

**Σχετικά με την  
ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ  
ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ  
ΠΕΡΙΒΑΛΛΟΝ**

**Θεωρία, αρχές & υποδείγματα για απαιτητικούς  
μαθητές**

**Γ' Λυκείου - Τεχνολογική Κατεύθυνση**

**Δημήτρης Τζήμας**



Copyright © 2015, Δημήτρης Ευ. Τζήμας

Βρείτε περισσότερες πληροφορίες για το βιβλίο:

<http://eclass.sch.gr/modules/document/document.php?course=EL845102>

& email: [dtzimas@sch.gr](mailto:dtzimas@sch.gr)

Ηλεκτρονική Σελιδοποίηση & Σχεδιασμός Εξωφύλλου:

Δ. Τζήμας (Εκπαιδευτικός Πληροφορικός)

Διάθεση βιβλίου, email: [dtzimas@sch.gr](mailto:dtzimas@sch.gr)

Δεύτερη έκδοση: Καστοριά Απρίλιος 2015

Παραχωρείται η άδεια προς αντιγραφή, διανομή και/ή τροποποίηση αυτού του εγγράφου, σύμφωνα με τους όρους της άδειας Creative Commons Attribution-NonCommercial 3.0 Unported License, η οποία είναι διαθέσιμη στη σελίδα <http://creativecommons.org/licenses/by-nc/3.0/>.

ISBN: 978-960-93-5269-7

Κωδικός στον Εύδοξο: 41964771



Στους γονείς μου  
Στην Όλγα, την Ευαγγελία, τον Ευθύμη



## ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω:

Την οικογένειά μου για τον χρόνο και την ενέργεια που τους στέρησα.

Τους μαθητές μου για την ειλικρινή, εγκάρδια και δημιουργική ανάδραση που έχουμε στη σχολική τάξη από το 2000.





## Περιεχόμενα

Ενότητα 1 Βασικές Έννοιες Αλγορίθμων - Δομή Ακολουθίας (κεφ. 2 και 7 σχολ. βιβλίου)	21
Ενότητα 2 Δομή Επιλογής (κεφ. 2 και 8 σχολ. βιβλίου)	41
Ενότητα 3 Δομή Επανάληψης (κεφ. 2 και 8 σχολ. βιβλίου)	59
Ενότητα 4 Δομές δεδομένων - Μονοδιάστατοι Πίνακες (κεφ. 3 και 9 σχολ. βιβλίου)	115
Ενότητα 5 Δισδιάστατοι Πίνακες (κεφ. 3 και 9 σχολ. βιβλίου)	175
Ενότητα 6 Εισαγωγή στον Προγραμματισμό (κεφ. 6 σχολ. βιβλίου)	231
Ενότητα 7 Υποπρογράμματα (κεφ. 10 σχολ. βιβλίου)	243
Παραρτήματα	289
Πηγές-βιβλιογραφία	307



## Εισαγωγή

Καλώς ορίσατε στο σχολικό βοήθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (ΑΕΠΠ). Με χαρακτηριστικά την απλότητά του παράλληλα με τις ισχυρές του δυνατότητες και μία ολιστική προσέγγιση, είναι προσαρμοσμένο στα θέματα των πανελληνίων που δίνουν την αίσθηση και το επίπεδο.

Όλα τα προβλήματα μπορούν να λυθούν με πολύ περισσότερους τρόπους από αυτούς που θεωρούμε ως τους μόνους δυνατούς, αρκεί να επανεξετάσουμε το ερώτημα και να σκεφτούμε ελεύθερα! Τις περισσότερες φορές όταν αντιμετωπίζουμε ένα πρόβλημα ψάχνουμε μια λύση που μας παγιδεύει στην αρχική του διατύπωση.

Δώστε προσοχή στη σωστή κατανομή χρόνου στα κεφάλαια και μην αρκείστε σε μαύρα κουτιά (παπαγαλία). **Εμπλακείτε με τον κώδικα**, προσπαθήστε, εργαστείτε ξανά και ξανά. Οι εκφωνήσεις των ασκήσεων θα έχουν διαφορετική απόχρωση για να προσαρμοστεί ο μαθητής σε όλες τις πιθανές διατυπώσεις των εξετάσεων. Η ΑΕΠΠ είναι κάτι καινούργιο για τον μαθητή, κάτι ολότελα διαφορετικό, χωρίς προαπαιτούμενα, στο οποίο όλοι αρχίζουν από την αρχή.

**Στόχοι** του βιβλίου είναι οι μαθητές να:

- ➔ καλλιεργήσουν την αυστηρότητα και σαφήνεια της έκφρασης και της διατύπωσης,
- ➔ αναπτύξουν ικανότητες μεθοδολογικού χαρακτήρα,
- ➔ αποκτήσουν δεξιότητες αλγοριθμικής προσέγγισης,
- ➔ κατανοούν πλήρως τα προβλήματα που τους τίθενται,
- ➔ μπορούν να ανιχνεύουν και να διακρίνουν τα μέρη ενός προβλήματος,
- ➔ μπορούν να προσδιορίζουν και να αναφέρουν με ακρίβεια και σαφήνεια τα δεδομένα και τα ζητούμενα ενός προβλήματος,
- ➔ γνωρίσουν και να επιλέγουν την κατάλληλη δομή-εντολή για την επίλυση ενός προβλήματος,
- ➔ μπορούν να παίρνουν θέση για την ορθότητα ενός αλγορίθμου,
- ➔ αναπτύξουν ικανότητες αναζήτησης εναλλακτικών λύσεων.

Και μην ξεχνάτε: Να αποκτήσετε προσωπικό, κομψό στυλ στον προγραμματισμό, αλλά όχι σε καλούπια & όλα τα μόρια είναι ισοδύναμα!

**Καλό προγραμματισμό!**

Δημήτρης Τζήμας  
Msc Εκπαιδευτικός Πληροφορικός







## Στόχοι Ενότητας

- ➔ Να κατανοήσετε το ρόλο των μεταβλητών (Η **πληροφορική μεταβλητή** διαφέρει από τη **μαθηματική** μεταβλητή) σε ένα πρόγραμμα ως δεσμευτικών χώρου για τιμές δεδομένων που μπορεί να αλλάζουν κατά την εκτέλεση ενός προγράμματος.
- ➔ Να αναγνωρίζετε την ύπαρξη διαφορετικών τύπων δεδομένων.
- ➔ Να μπορείτε να κάνετε απλούς υπολογισμούς χρησιμοποιώντας αριθμητικές παραστάσεις.
- ➔ Να εκπαιδεύσει το μαθητή στην αλγοριθμική σκέψη και στις προγραμματιστικές τεχνικές

## 1.1 Θεωρία-Οδηγίες

Αναλύεται η έννοια του αλγορίθμου και οι τρόποι αναπαράστασής του, οι βασικές έννοιες προγραμματισμού που απαιτούνται για την υλοποίηση ενός αλγορίθμου σε ένα προγραμματιστικό περιβάλλον (σταθερές, μεταβλητές, εντολές, πράξεις) και τέλος περιγράφεται η πρώτη αλγοριθμική δομή (**ακολουθία**).

Βήματα αντιμετώπισης ασκήσεων:

### 1. Αναλύστε το πρόβλημα

✓ να αναγνωρίσετε τα **ζητούμενα** του προβλήματος; Τι στοιχεία, ποιες ποσότητες θα βρείτε λύνοντας το πρόβλημα.

✓ να αναγνωρίσετε τα **δεδομένα** του προβλήματος; Τι στοιχεία, ποιες ποσότητες χρειάζεστε για να μπορέσετε να υπολογίσετε τα ζητούμενα.

### 2. Λύστε το πρόβλημα

✓ Περιγράψτε τις ενέργειες που κάνατε για να λύσετε το πρόβλημα.

✓ **Επαναδιατυπώστε** όλες τις εντολές του αλγόριθμου ώστε να αποθηκεύουν τα αποτελέσματά τους σε θέσεις μνήμης και να αξιοποιούν τα αποτελέσματα που βρίσκονται φυλαγμένα σε θέσεις μνήμης από προηγούμενες ενέργειες. Να δίνετε στις θέσεις μνήμης κατάλληλα ονόματα.

### 3. Σχεδιάστε τον αλγόριθμο

### 4. Εκτελέστε τον αλγόριθμο

Ο αλγόριθμός σας θα λειτουργεί χωρίς αλλαγές για πολλές διαφορετικές τιμές δεδομένων (στιγμιότυπα).

Με άλλα λόγια:

1. Διαβάζουμε καλά την εκφώνηση
2. Κάνουμε στο πρόχειρο ένα σκαρίφημα με τις αλγοριθμικές σκέψεις μας
3. Υπογραμμίζουμε τις λέξεις κλειδιά
4. Μεταφράζουμε σε κώδικα τις λέξεις κλειδιά
5. Κωδικοποίηση

Σημείωση: Η απάντησή σας σ' ένα υποερώτημα μπορεί να περιλαμβάνει αρκετά βήματα. Σχεδιάστε τα, αναλύστε τον τρόπο σκέψης σας και μετά προχωρήστε στην υλοποίησή τους. Ίσως ένα υποερώτημα να μην μπορείτε να το λύσετε, προχωρήστε στο επόμενο. Καλό είναι να γράψετε ένα σχετικό **σχόλιο** για να βοηθήσετε το βαθμολογητή.

**Αλγόριθμος** είναι μια **πεπερασμένη** σειρά ενεργειών, αυστηρά **καθορισμένων** και **εκτελέσιμων** σε πεπερασμένο χρόνο, που στοχεύουν στην **επίλυση** ενός προβλήματος. Η έννοια του αλγόριθμου **δεν** συνδέεται αποκλειστικά και μόνο με προβλήματα της Πληροφορικής.

Κάθε Αλγόριθμος απαραίτητα ικανοποιεί τα επόμενα **κριτήρια**.

- ✓ **Είσοδος** (input). Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών.
- ✓ **Έξοδος** (output). Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλον αλγόριθμο.
- ✓ **Καθοριστικότητα** (definiteness). Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. π.χ. μία εντολή διαίρεσης πρέπει να θεωρεί και την περίπτωση, όπου ο διαιρέτης της λαμβάνει μηδενική τιμή.
- ✓ **Περατότητα** (finiteness). Ο Αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα συγκεκριμένο αριθμό βημάτων, δεν αποτελεί αλγόριθμο, αλλά λέγεται απλά υπολογιστική διαδικασία .
- ✓ **Αποτελεσματικότητα** (effectiveness). Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

Στη βιβλιογραφία συναντώνται διάφοροι τρόποι **αναπαράστασης** ενός αλγορίθμου:

1. με **ελεύθερο κείμενο** (free text), που αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου. Έτσι εγκυμονεί τον κίνδυνο ότι μπορεί εύκολα να οδηγήσει σε μη εκτελέσιμη παρουσίαση, παραβιάζοντας το τελευταίο χαρακτηριστικό των αλγορίθμων, δηλαδή την αποτελεσματικότητα.
2. με **διαγραμματικές τεχνικές**, που συνιστούν ένα γραφικό τρόπο παρουσίασης του αλγορίθμου. Από τις διάφορες διαγραμματικές τεχνικές που έχουν επινοηθεί, η πιο παλιά και η πιο γνωστή ίσως, είναι το διάγραμμα ροής (flow chart).
3. με **φυσική γλώσσα** (natural language) κατά βήματα. Στην περίπτωση αυτή χρειάζεται προσοχή, γιατί μπορεί να παραβιασθεί το κριτήριο του καθορισμού.
4. με **κωδικοποίηση** (coding), δηλαδή με ένα πρόγραμμα που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.

Η **ακολουθιακή δομή** εντολών (σειριακών βημάτων) χρησιμοποιείται πρακτικά για την αντιμετώπιση απλών προβλημάτων, όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών

**Σταθερές** (constants). Με τον όρο αυτό αναφερόμαστε σε προκαθορισμένες τιμές που παραμένουν αμετάβλητες σε όλη τη διάρκεια της εκτέλεσης ενός αλγορίθμου. Οι σταθερές διακρίνονται σε:

- ✓ **αριθμητικές**, π.χ. 134, +7, -1,28
- ✓ **αλφαριθμητικές** π.χ. “Μανταρίνι”, “Hallo World!!”
- ✓ **λογικές** που είναι ακριβώς δύο, **Αληθής** και **Ψευδής**



**Μεταβλητές** (variables). Μια μεταβλητή είναι ένα γλωσσικό αντικείμενο, που χρησιμοποιείται για να παραστήσει ένα στοιχείο δεδομένου. Στη μεταβλητή εκχωρείται μια τιμή, η οποία μπορεί να αλλάζει κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Ανάλογα με το είδος της τιμής που μπορούν να λάβουν, οι μεταβλητές διακρίνονται σε αριθμητικές, αλφαριθμητικές και λογικές. Οι μεταβλητές πρέπει να **δηλώνονται στα προγράμματα** γιατί ο μεταγλωττιστής θέλει να γνωρίζει τον τύπο της πληροφορίας που θα αποθηκεύσει στη θέση μνήμης της RAM στην οποία ανατίθεται η μεταβλητή ώστε α) να δεσμεύσει τον κατάλληλο χώρο και β) να κάνει τον έλεγχο τύπων στις εκφράσεις που η μεταβλητή χρησιμοποιείται.

**Δήλωση** μεταβλητής: πρόταση που ενημερώνει το μεταγλωττιστή για το συμβολικό της όνομα και τον τύπο της. Καλό είναι να επιλέγετε **ονόματα μεταβλητών** σχετικά με το σκοπό που εξυπηρετεί η μεταβλητή και που θα είναι απίθανο να μπερδευτούν τυπογραφικά. Προτείνεται να χρησιμοποιείται σύντομα ονόματα για δείκτες βρόχων και μεγαλύτερα ονόματα για τις υπόλοιπες μεταβλητές.

**Τελεστές** (operators). Πρόκειται για τα γνωστά σύμβολα που χρησιμοποιούνται στις διάφορες πράξεις. Οι τελεστές διακρίνονται σε αριθμητικούς, λογικούς και συγκριτικούς. Οι μεταβλητές και οι σταθερές είναι τα βασικά αντικείμενα δεδομένων που χειρίζεται ένα πρόγραμμα. Οι τελεστές καθορίζουν τι θα γίνει μ' αυτές. Ο **τύπος** ενός αντικειμένου προσδιορίζει το σύνολο τιμών που μπορεί να έχει το αντικείμενο, καθώς και σε ποιες πράξεις μπορεί να συμμετέχει.

**Εκφράσεις** (expressions). Οι εκφράσεις διαμορφώνονται από τους τελεστές (operands), που είναι σταθερές και μεταβλητές και από τους τελεστές. Η διεργασία αποτίμησης μιας έκφρασης συνίσταται στην απόδοση τιμών στις μεταβλητές και στην εκτέλεση των πράξεων. Η τελική τιμή μιας έκφρασης εξαρτάται από την ιεραρχία των πράξεων και τη χρήση των παρενθέσεων. Μια έκφραση μπορεί να αποτελείται από μια μόνο μεταβλητή ή σταθερά μέχρι μια πολύπλοκη μαθηματική παράσταση

Ένα **πρόγραμμα** αποτελείται από κώδικα και δεδομένα.

**Κώδικας** είναι προτάσεις-εντολές που εκτελούνται στο χρόνο φόρτωσης και εκτέλεσης του προγράμματος.

Τα **δεδομένα** είναι μεταβλητές ή σταθερές.

**Δεσμευμένη λέξη**: λεκτική μονάδα της οποίας η σημασία καθορίζεται από τους κανόνες της γλώσσας και δεν μπορεί να αλλάξει.

Ακολουθούν και μερικοί χρήσιμοι ορισμοί:

- ✓ **αλφαριθμητικό**: Σύνολο χαρακτήρων που μπορεί να εμπεριέχει γράμματα, ψηφία και ειδικά σύμβολα, όπως π.χ. σημεία στίξης.
- ✓ **Εκχώρηση**: Μηχανισμός τιμοδότησης μιας μεταβλητής.
- ✓ **Εντολή**: Σε μια γλώσσα προγραμματισμού μια έκφραση που έχει νόημα και η οποία καθορίζει μια πράξη και προσδιορίζει τους τελεστές της, αν υπάρχουν.
- ✓ **φυσική γλώσσα** natural language: Γλώσσα οι κανόνες της οποίας βασίζονται στην τρέχουσα χρήση, χωρίς να είναι αυστηρά προδιαγεγραμμένο
- ✓ **σταθερά** constant: Γλωσσικό αντικείμενο που παίρνει μόνο μια ειδική τιμή.

Για να εκτελεστεί ένας αλγόριθμος στον υπολογιστή πρέπει να γραφτεί ένα πρόγραμμα

που να τον υλοποιεί, διαδικασία γνωστή και ως **κωδικοποίηση**.

Τα **σχόλια** περιγράφουν τη λειτουργία ενός προγράμματος ή γενικότερα ενός τμήματος κώδικα και γράφονται για να βοηθήσουν τους ανθρώπους και όχι τον υπολογιστή στην κατανόηση και συντήρηση ενός προγράμματος. Όταν ένα πρόγραμμα μεταφράζεται, τα σχόλια αγνοούνται. Τα σχόλια που περιγράφουν τη λειτουργία ενός προγράμματος, το ρόλο των μεταβλητών και τη λειτουργία πολύπλοκων τμημάτων κώδικα, αποτελούν παράδειγμα καλού προγραμματιστικού στυλ.

Άλλωστε ένα καλό **προγραμματιστικό στυλ** χτίζεται και από:

- Σχόλια
- Εσοχές
- Σχετικά ονόματα μεταβλητών
- Κατανοήσιμο κώδικα (more is not always better)
- Πριν την εντολή Διάβασε μία επεξηγηματική-προτροπική Γράψε.

Στον προγραμματισμό πρέπει πάντα να σκεφτόμαστε από πολλές οπτικές γωνίες:

- **Προγραμματιστής**
- **Η/Υ** (οθόνη, επεξεργαστής, μνήμη) (βλέπε πίνακας τιμών)
- **Τελικός χρήστης**

Χρησιμοποιώντας αποκλειστικά τη **δομή ακολουθίας**, μπορείτε να λύσετε μόνο προβλήματα στα οποία:

- ✓ η σειρά των βημάτων είναι καθορισμένη
- ✓ όλα τα βήματα εκτελούνται πάντοτε
- ✓ δεν υπάρχουν εξαιρέσεις!

Ακολουθούν μερικές **παρατηρήσεις**:

- ✓ Για τελεστές ίδιου βαθμού προτεραιότητας στην ίδια έκφραση, ισχύει η προτεραιότητα από τα αριστερά προς τα δεξιά.
- ✓ Η εντολή Διάβασε είναι **εκτελεστέα** εντολή, ενώ η εντολή Αλγόριθμος είναι **δηλωτική** εντολή.
- ✓ Οι πράξεις μέσα σε **παρενθέσεις** εκτελούνται κατά προτεραιότητα.
- ✓ Η **προτεραιότητα** των λογικών τελεστών θα θεωρούμε ότι έχει την ίδια προτεραιότητα και όταν γράφουμε σύνθετες συνθήκες, θα χρησιμοποιούμε πάντα παρενθέσεις.
- ✓ Δεν ορίζονται στη ΓΛΩΣΣΑ πράξεις μεταξύ τελεστών τύπου χαρακτήρα.
- ✓ Σε τελεστέους λογικού τύπου, ορίζονται μόνο οι λογικές πράξεις και από συγκρίσεις η ισότητα και η ανισότητα.
- ✓ Οι συγκριτικές πράξεις γίνονται μεταξύ τελεστών ίδιου τύπου.
- ✓ Συχνά κατά την ποσοστιαία αύξηση ή μείωση ξεχνάμε να πολλαπλασιάσουμε το ποσοστό με την αρχική τιμή. Γράφουμε για παράδειγμα  $x \leftarrow x + 10/100$ , ενώ εννοούμε  $x \leftarrow x + 10/100 * x$  (ή  $x \leftarrow 1.1 * x$ ).
- ✓ Η απόσταση μεταξύ δύο τιμών εκφράζεται από την απόλυτη τιμή της διαφοράς τους
- ✓ **Δεδομένα** και **Αποτελέσματα** μόνο σε αλγόριθμο (δηλωτικές εντολές).
- ✓ Η εντολή **Αντιμετάθεσε** δεν ορίζεται στη ΓΛΩΣΣΑ (μόνο σε αλγόριθμο).
- ✓ Το ΚΑΙ έχει την ίδια προτεραιότητα με το Ή. Επομένως καλό είναι να χρησιμοποιούνται παρενθέσεις στις λογικές εκφράσεις για να μην υπάρχει αμφιβολία για τη σειρά εκτέλεσής τους.

- ✓ Γενικά, θεωρούμε ότι κάθε ΓΡΑΨΕ τυπώνει σε μία γραμμή τη λίστα των ορισμάτων της και γενικά ότι το αποτέλεσμα που τυπώνει είναι ευκρινές δηλαδή όχι κολλημένο το ένα με το άλλο. Επίσης θεωρούμε ότι ως όρισμα στη ΓΡΑΨΕ μπορεί να μπει και μία έκφραση.
- ✓ Αν και προγραμματίζοντας στο χαρτί δεν αντιμετωπίζουμε πρόβλημα μεγέθους στους αριθμούς, στον υπολογιστή έχουμε πάντα περιορισμένο μέγεθος, όσο μεγάλο και αν είναι αυτό.
- ✓ Δεν υπάρχει διαφορετικός τύπος δεδομένων για ένα χαρακτήρα και για πολλούς χαρακτήρες.
- ✓ Υπάρχει διευκρινιστική οδηγία από το Παιδαγωγικό Ινστιτούτο (πλέον ΙΕΠ), η οποία αναφέρει ότι δεν πρέπει να χρησιμοποιούνται σε ασκήσεις με αρνητικούς αριθμούς ή πραγματικούς οι τελεστές **MOD, DIV**.
- ✓ Αποδεκτά ονόματα **αναγνωριστικών** (δηλαδή όνομα προγράμματος, σταθεράς ή μεταβλητής) είναι αυτά που ξεκινούν με γράμμα (είτε ελληνικό είτε αγγλικό) και στη συνέχεια περιλαμβάνουν γράμματα, αριθμούς ή την κάτω παύλα ( \_ ).
- ✓ Η **εντολή ανάθεσης** τιμής μπορεί να μεταφραστεί ως εξής: «Υπολόγισε την τιμή της έκφρασης δεξιά από το σύμβολο ανάθεσης τιμής και βάλε το αποτέλεσμα στη μεταβλητή που βρίσκεται αριστερά από την ανάθεση τιμής». Στο δεξί μέρος μπορεί να βρίσκεται οτιδήποτε έχει τιμή, ενώ στο αριστερό μία και μόνο μεταβλητή (ή στοιχείο πίνακα) ίδιου τύπου δεδομένων με την έκφραση.
- ✓ Η εντολή **ΔΙΑΒΑΣΕ** διαβάζει από το πληκτρολόγιο μία λίστα μεταβλητών, χωρισμένων από κόμματα. Δεν επιτρέπεται να διαβαστεί κάτι που δεν είναι μεταβλητή.
- ✓ Η συνάρτηση **A\_M** επιστρέφει το ακέραιο τμήμα ενός πραγματικού αριθμού, για παράδειγμα  $A\_M(6.98) = 6$ , ενώ  $A\_M(-9.77) = -9$ .
- ✓ Μια παράσταση σε παρενθέσεις: οι παρενθέσεις χρησιμοποιούνται για να υποδείξουν τη σειρά των πράξεων. Μια παράσταση σε παρενθέσεις είναι από μόνη της ένας όρος που αντιμετωπίζεται από τον μεταγλωττιστή ως μια μονάδα που πρέπει να υπολογιστεί πριν να συνεχιστεί ο υπολογισμός. Μια παράσταση (expression) αποτελείται από: **όρους** (terms) που παριστάνουν τιμές δεδομένων και **τελεστές** (operators) που υποδεικνύουν μια υπολογιστική πράξη.
- ✓  $x \leftarrow x * 10$  και  $x \leftarrow x \text{ div } 10$  μπορούμε να πούμε ότι είναι η ολίσθηση αριστερά και η ολίσθηση δεξιά αντίστοιχα στο δεκαδικό σύστημα αρίθμησης.
- ✓ Η **ιεραρχία** (προτεραιότητα) των τελεστών σε μια έκφραση είναι: αριθμητικοί, συγκριτικοί, λογικοί.
- ✓ Χρήσεις των τελεστών MOD/DIV: έλεγχος άρτιου, πολλαπλασία, απομόνωση ψηφίων.
- ✓ Για τη σύνταξη μιας λογικής έκφρασης ή **συνθήκης** χρησιμοποιούνται σταθερές, μεταβλητές, αριθμητικές παραστάσεις, συγκριτικοί και λογικοί τελεστές, καθώς και παρενθέσεις.

Τώρα που αρχίζεις να γράφεις προγράμματα, χρήσιμο θα ήταν να μάθεις να ακολουθείς γενικές αρχές έτσι ώστε, η συγγραφή, η κατανόηση και η τροποποίηση των προγραμμάτων σου να γίνεται εύκολα και γρήγορα. Τα προγράμματά σου πρέπει να είναι **απλά** και **κατανοητά**. Όχι μόνο για τους βαθμολογητές, αλλά και για σένα που θα επανέλθεις σε παλαιότερο σου πρόγραμμα για να το τροποποιήσεις ή να το ελεγκτείς.

Μερικά σημεία για να είναι κατανοήσιμος ο κώδικας:

- ✓ η χρήση **κενών γραμμών** διευκολύνει την ανάγνωση του προγράμματος και οριοθετεί τις ενότητες του.
- ✓ Η χρήση **εσοχών**.

- ✓ η χρησιμοποίηση **σταθερών** σε διευκολύνει σε πιθανές επόμενες αλλαγές και σε προστατεύει από αθέλητες τροποποιήσεις.
- ✓ Θα πρέπει να αποδίδεις **αρχικές τιμές** στις μεταβλητές που χρησιμοποιείς στο πρόγραμμα. Επιπλέον η απόδοση αρχικών τιμών βοηθάει στην καλύτερη κατανόηση του προγράμματος και στην ευκολότερη συντήρησή του.
- ✓ Να αποφεύγεις να χρησιμοποιείς μεγάλους υπολογισμούς. Η διάσπαση ενός υπολογισμού σε απλούστερους, διευκολύνει τους άλλους στην κατανόηση του προγράμματος και σένα στην αποφυγή λαθών.
- ✓ Είναι καλύτερο να χρησιμοποιείς **παρενθέσεις**, έστω και αν δεν είναι απαραίτητο, σε προφυλάσσει από πιθανά λάθη και αβλειψίες, ενώ ταυτόχρονα κάνει το πρόγραμμα πιο εύκολο στην κατανόηση του.

Στην πραγματικότητα όμως μόνο η **εξάσκηση** και η πείρα θα σου εξασφαλίσουν τη δυνατότητα να συντάσσεις εύκολα και γρήγορα σωστά προγράμματα.

Ιδιαίτερη έμφαση πρέπει να δίνεται στη **σχεδίαση** του αλγορίθμου. Οι μαθητές ενθαρρύνονται να επιλύουν το πρόβλημα στο πρόχειρο, να σχεδιάζουν τα βήματα του αλγορίθμου και μετά να το μετατρέπουν σε κώδικα. Αυτό πρέπει να το συνηθίσουν από την αρχή, ακόμη και για τα πιο απλά προβλήματα.

## 1.2 Ερωτήσεις Σωστού/Λάθους

1. Οι μεταβλητές αντιστοιχίζονται από το μεταγλωττιστή κάθε γλώσσας σε συγκεκριμένες θέσεις μνήμης του υπολογιστή.
2. Μπορούμε να αρχικοποιήσουμε μεταβλητές στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ.
3. Η τιμή της μεταβλητής είναι το περιεχόμενο της αντίστοιχης θέσης μνήμης και δεν μεταβάλλεται στη διάρκεια εκτέλεσης του προγράμματος.
4. Ο τύπος της μεταβλητής αλλάζει κατά την εκτέλεση του προγράμματος.
5. Η περατότητα ενός αλγορίθμου αναφέρεται στο γεγονός ότι καταλήγει στη λύση του προβλήματος μετά από πεπερασμένο αριθμό βημάτων.
6. Η εντολή Αλγόριθμος είναι δηλωτική εντολή ενώ η εντολή Διάβασε είναι εκτελεστέα εντολή.
7. Σε ένα διάγραμμα ροής, η έλλειψη δηλώνει την είσοδο ή την έξοδο στοιχείων.
8. Ένας αλγόριθμος μπορεί να μην έχει έξοδο.
9. Η σύγκριση 'ΑΛΗΘΗΣ' > 'ΑΛΗΘΕΣ' δίνει τιμή ΨΕΥΔΗΣ.
10. Ο ακέραιος τύπος δεν έχει συγκεκριμένο εύρος τιμών.
11. Οι τελεστές div, \* έχουν ίδια ιεραρχία.
12. Η δήλωση σταθερών είναι υποχρεωτική σε ένα πρόγραμμα.
13. Ένας αλγόριθμος είναι μία πεπερασμένη σειρά ενεργειών.
14. Οι ενέργειες που ορίζει ένας αλγόριθμος είναι αυστηρά καθορισμένες.
15. Η έννοια του αλγόριθμου συνδέεται αποκλειστικά με την Πληροφορική.
16. Ο αλγόριθμος τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης εντολών.
17. Ο πιο δομημένος τρόπος παρουσίασης αλγορίθμων είναι με ελεύθερο κείμενο.
18. Ένας αλγόριθμος στοχεύει στην επίλυση ενός προβλήματος.
19. Η διαίρεση με το 0 παραβιάζει το κριτήριο περατότητας ενός αλγορίθμου.
20. Η αποτελεσματικότητα είναι ένα από τα κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος.
21. Δεν υπάρχει αλγόριθμος για τη σχεδίαση αλγορίθμων.
22. Δεσμευμένες λέξεις καλούνται οι λέξεις που χρησιμοποιούνται από την ίδια τη ΓΛΩΣΣΑ για συγκεκριμένους λόγους και δεν μπορούν να χρησιμοποιηθούν ως ονόματα.
23. Ο τελεστής MOD χρησιμοποιείται για τον υπολογισμό του ηλίκου μίας διαίρεσης ακεραίων αριθμών.
24. Κάθε εντολή ενός αλγορίθμου πρέπει να καθορίζεται χωρίς αμφιβολία για τον τρόπο εκτέλεσής της.
25. Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μία έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες.
26. Ο πολλαπλασιασμός σε ένα πρόγραμμα γραμμένο σε ΓΛΩΣΣΑ, έχει μεγαλύτερη προτεραιότητα σε σχέση με την ύψωση σε δύναμη.
27. Η έξοδος ενός αλγορίθμου μπορεί να είναι είσοδος σε άλλον αλγόριθμο.
28. Μια αριθμητική έκφραση δεν μπορεί να περιέχει μεταβλητές, σταθερές, τελεστές και παρενθέσεις.
29. Μια μεταβλητή μπορεί να πάρει τιμή μόνο με την εντολή ΔΙΑΒΑΣΕ.
30. Το αποτέλεσμα της παρακάτω έκφρασης  $2^3 + 3 * (27 \bmod (25 \bmod 7))$  είναι 11.
31. Για την δήλωση μιας λογικής σταθεράς, μπορούμε να γράψουμε ΛΟΓΙΚΕΣ: flag = ΑΛΗΘΗΣ.
32. Σε μια μεταβλητή πραγματικού τύπου μπορούμε να εκχωρήσουμε και το αποτέλεσμα

μιας ακέραιας αριθμητικής παράστασης.

33. Ένας από τους τρόπους αναπαράστασης αλγορίθμων είναι και η κωδικοποίηση.

34. Ο τελεστής + είναι συγκριτικός.

35. Το γεωμετρικό σχήμα του πλάγιου παραλληλογράμμου αναπαριστά την είσοδο δεδομένων ή την έξοδο των αποτελεσμάτων σε ένα λογικό διάγραμμα.

36. Το διάγραμμα ροής είναι ένας τρόπος αναπαράστασης αλγορίθμου που μπορεί να παραβιάσει το κριτήριο της καθοριστικότητας.

37. Κατά την εκτέλεση ενός προγράμματος, η τιμή μιας μεταβλητής μπορεί να αλλάζει, ο τύπος της όμως μένει υποχρεωτικά αναλλοίωτος.

38. Μπορούμε να αρχικοποιήσουμε μεταβλητές στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ.

39. Η σύγκριση αλφαριθμητικών δεδομένων βασίζεται στη σύγκριση χαρακτήρα προς χαρακτήρα σε κάθε θέση μέχρις ότου βρεθεί κάποια διαφορά.

40. Οι τύποι μεταβλητών που δέχεται η ΓΛΩΣΣΑ είναι μόνο ΠΡΑΓΜΑΤΙΚΕΣ και ΑΚΕΡΑΙΕΣ.

41. Η σχέση (ΟΧΙ συνθήκη1 ΚΑΙ ΟΧΙ συνθήκη2) δίνει πάντα τα ίδια αποτελέσματα με τη σχέση ΟΧΙ (συνθήκη1 ΚΑΙ συνθήκη2).

42. Μια μεταβλητή που δεν έχει πάρει τιμή μέσω μιας εντολής εκχώρησης ή ΔΙΑΒΑΣΕ, έχει απροσδιόριστο περιεχόμενο.

43. Κάθε αλγόριθμος τελειώνει με τη δεσμευμένη λέξη Τέλος και το όνομα του αλγορίθμου.

44. Κάθε μεταβλητή παίρνει τιμή μόνο με την εντολή ΔΙΑΒΑΣΕ.

45. Ο αλγόριθμος είναι ένα μέσο παραγωγής δεδομένων.

46. Το όνομα μιας συμβολικής σταθεράς πρέπει να ξεκινάει με ψηφίο.

47. Ο αλγόριθμος είναι ένα μέσο παραγωγής πληροφοριών.

48. Το όνομα μιας συμβολικής σταθεράς πρέπει να ξεκινάει με γράμμα.

49. Κάθε μεταβλητή παίρνει τιμή μόνο με την εντολή εκχώρησης ακριβώς μία φορά.

50. Σε έναν αλγόριθμο στον οποίο υπάρχει μόνο η δομή ακολουθίας, κάθε εντολή εκτελείται.

51. Όταν μία μεταβλητή είναι αλφαριθμητικού τύπου, μπορούμε να της εκχωρήσουμε τον αριθμό τηλεφώνου κάποιου.

52. Η λέξη διάβαζε μπορεί να αποτελέσει το όνομα μιας μεταβλητής.

53. Αριστερά του τελεστή εκχώρησης ( $\leftarrow$ ), μπορεί να υπάρξει μόνο μεταβλητή.

54. Σε ένα διάγραμμα ροής, η εντολή Διάβασε συμβολίζεται με ένα ορθογώνιο παραλληλόγραμμο.

55. Κάθε αλγόριθμος πρέπει να πληρεί το κριτήριο της πληρότητας.

56. Η ακολουθιακή δομή εντολών χρησιμοποιείται για την αντιμετώπιση προβλημάτων, στα οποία είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών.

57. Για τον υπολογισμό μιας έκφρασης, όλες οι μεταβλητές που εμφανίζονται σ' αυτή πρέπει να έχουν οριστεί προηγουμένως, δηλαδή να έχουν κάποια τιμή.

58. Ενώ η τιμή μίας μεταβλητής μπορεί να αλλάζει κατά την εκτέλεση του προγράμματος, αυτό που μένει υποχρεωτικά αναλλοίωτο είναι ο τύπος της.

59. Σε μία εντολή εκχώρησης του αποτελέσματος μίας έκφρασης σε μία μεταβλητή, η μεταβλητή και η έκφραση πρέπει να είναι του ίδιου τύπου.

60. Η λογική πράξη ΚΑΙ μεταξύ δύο προτάσεων είναι ψευδής όταν οποιαδήποτε από τις δύο προτάσεις είναι ψευδής.

61. Η λογική έκφραση  $X \text{ Η } (\text{ΟΧΙ } X)$  είναι πάντα αληθής, για κάθε τιμή της λογικής μεταβλητής  $X$ .

62. Στην αριθμητική έκφραση  $A+B*Γ$  εκτελείται πρώτα η πρόσθεση και μετά ο πολλαπλασιασμός.

**63.** Κατά την εκτέλεση ενός προγράμματος μπορεί να αλλάζει η τιμή και ο τύπος μιας μεταβλητής.

**64.** Ο τελεστής ΚΑΙ αντιστοιχεί στη λογική πράξη της σύζευξης.

**Απάντηση**

1.Σ, 2.Λ, 3.Λ,4.Λ, 5.Σ, 6.Σ, 7.Λ, 8.Λ, 9.Λ, 10.Σ, 11.Σ, 12.Λ, 13.Σ, 14.Σ, 15.Λ, 16.Σ, 17.Λ, 18.Σ, 19.Λ, 20.Σ, 21.Σ, 22.Σ, 23.Λ, 24.Σ, 25.Σ, 26.Λ, 27.Σ, 28.Λ, 29.Λ, 30.Λ, 31.Λ, 32.Σ, 33.Σ, 34.Λ, 35.Σ, 36.Λ, 27.Σ, 38.Λ, 39.Σ, 40.Λ, 41.Λ, 42.Σ, 43.Σ, 44.Λ, 45.Λ, 46.Λ, 47.Σ, 48.Σ, 49.Λ, 50.Σ, 51.Σ, 52.Λ, 53.Σ, 54.Λ, 55.Λ, 56.Σ, 57.Σ, 58.Σ, 59.Σ, 60.Σ, 61.Σ, 62.Λ, 63.Λ, 64.Σ

## 1.3 Λυμένες Ασκήσεις

1. Να συμπληρώσετε τον κατάλληλο τύπο και το περιεχόμενο της μεταβλητής.

X ← 'ΑΛΗΘΗΣ'  
X ← 11.0 - 13.0  
X ← 7 > 4  
X ← ΨΕΥΔΗΣ  
X ← 4

### Απάντηση

Τύπος μεταβλητής X	Περιεχόμενο μεταβλητής X
ΧΑΡΑΚΤΗΡΑΣ	ΑΛΗΘΗΣ
ΠΡΑΓΜΑΤΙΚΗ	-2.0
ΛΟΓΙΚΗ	ΑΛΗΘΗΣ
ΛΟΓΙΚΗ	ΨΕΥΔΗΣ
ΑΚΕΡΑΙΑ	4

2. Να αναφέρετε τον τύπο της μεταβλητής αριστερά της εκχώρησης:

x ← a MOD b  
x ← z/g  
x ← "kastoria" = "chania"

### Απάντηση

ΑΚΕΡΑΙΑ  
ΠΡΑΓΜΑΤΙΚΗ  
ΛΟΓΙΚΗ

3. Δίνεται το παρακάτω τμήμα αλγορίθμου:

Z ← ΨΕΥΔΗΣ  
X ← ΑΛΗΘΗΣ  
Ψ ← ΨΕΥΔΗΣ  
A ← X ΚΑΙ (Ψ Η Ζ)  
B ← (ΟΧΙ A) ΚΑΙ (ΟΧΙ Z)

Να γράψετε τις τιμές των μεταβλητών A και B μετά την εκτέλεση του παραπάνω τμήματος αλγορίθμου.

### Απάντηση

A ΨΕΥΔΗΣ  
B ΑΛΗΘΗΣ

4. Να ξαναγράψετε την παρακάτω εντολή χωρίς τη χρήση λογικών τελεστών.

**Αν** ( A < B και C <> D ) **και** ( B > D ή B =D ) **τότε**

K ← 1

**Τέλος\_Αν**



## Απάντηση

Αν  $A < B$  τότε

Αν  $C < D$  τότε

Αν  $B >= D$  τότε

$K \leftarrow 1$

Τέλος\_Αν

Τέλος\_Αν

Τέλος\_Αν

5. Να γράψετε ποιες από τις παραπάνω έννοιες:

- είναι στοιχεία μιας γλώσσας προγραμματισμού;
- ανήκουν σε κατηγορίες προβλημάτων;

- Λογικός τύπος δεδομένων
- Επιλύσιμο
- Ακέραιος τύπος δεδομένων
- Περατότητα
- Μεταβλητή
- Ημιδομημένο
- Πραγματικός τύπος δεδομένων
- Σταθερά
- Αδόμητο
- Καθοριστικότητα
- Άλυτο
- Ανοικτό

## Απάντηση

1.α, 2.β, 3.α, 4.α, 5.α, 6.β, 7.α, 8.α, 9.β, 10.α, 11.β, 12.β

6. Ποια η λειτουργία του ακόλουθου κώδικα:

$\alpha \leftarrow \alpha + \beta$

$\beta \leftarrow \alpha - \beta$

$\alpha \leftarrow \alpha - \beta$

## Απάντηση

Αντιμεταθέτει τις τιμές των αριθμητικών μεταβλητών  $\alpha, \beta$ .

7. Επιλέξτε την κατάλληλη απάντηση:

- Ποιο από τα παρακάτω δεν είναι τύπος μεταβλητής στη ΓΛΩΣΣΑ προγραμματισμού;  
α. Ακέραιος β. Πραγματικός γ. Λογικός δ. Δενδρικός
- Ποιο από τα παρακάτω δεν μπορεί να είναι όνομα μεταβλητής;  
α. Είμαι\_καλά β. Τεστ123 γ. ΙΑΝΟΥάριος δ. Μεγάλο Δένδρο
- Ποια από τις παρακάτω εντολές χρησιμοποιείται για την εμφάνιση ενός μηνύματος στην οθόνη του υπολογιστή;  
α. ΓΡΑΨΕ β. ΤΥΠΩΣΕ γ. ΣΤΕΙΛΕ δ. ΕΙΔΟΠΟΙΗΣΕ

4. Στον προγραμματισμό η ανισότητα δηλώνεται με το σύμβολο;  
 α. <> β. != γ. ~= δ. Με όλα τα παραπάνω
5. Στην αριθμητική πράξη  $x^3/a*b$  η μεταβλητή b που πραγματικά θα υπολογιστεί;  
 α. Στον παρονομαστή β. Στον αριθμητή γ. Αναλόγως τον υπολογιστή δ. Αναλόγως τη γλώσσα προγραμματισμού
6. Ποιο από τα παρακάτω σύμβολα δηλώνει την εκχώρηση μιας τιμής σε μια μεταβλητή;  
 α. ~= β. := γ. ← δ. +=
7. Μια μεταβλητή του τύπου ΛΟΓΙΚΗ μπορεί να πάρει  
 α. μόνο τρεις τιμές (-1, 0, 1) β. μόνο δύο τιμές (ΑΛΗΘΗΣ, ΨΕΥΔΗΣ) γ. Ότι τιμή θέλουμε δ. Μόνο δυαδικές τιμές.
8. Με ποια εντολή διαβάζουμε από το πληκτρολόγιο την τιμή που μας δίνει ο χρήστης και την αποθηκεύουμε στη μεταβλητή χ.  
 α. χ ΔΙΑΒΑΣΕ β. ΔΩΣΕ χ γ. ΔΙΑΒΑΣΕ χ δ. ΕΙΣΟΔΟΣ χ
9. Στο πρόγραμμα μια μεταβλητή πριν την χρησιμοποιήσουμε πρέπει πρώτα να  
 α. μηδενιστεί β. τυπωθεί γ. αρχικοποιηθεί δ. δηλωθεί στο αντίστοιχο πεδίο δήλωσης μεταβλητών

### Απάντηση

1.δ, 2.δ, 3.α, 4.α, 5.β, 6.γ, 7.β, 8.γ, 9.δ

**8.** Εκτός από τις λογικές πράξεις ΚΑΙ, Η, ΟΧΙ, ορίζονται στην άλγεβρα Boole και άλλες όπως η ΧΟΡ η οποία δίνει αποτέλεσμα ΑΛΗΘΗΣ όταν ένας και μόνο ένας από τους τελεστές της έχει τιμή ΑΛΗΘΗΣ. Να γραφεί αλγόριθμος που υλοποιεί την ΧΟΡ.

### Λύση

<b>Αλγόριθμος</b> Αποκλειστική_Διάζευξη	! Έναρξη κώδικα
<b>Διάβασε</b> a,b	! Είσοδος δεδομένων
XOR←(a Η b) ΚΑΙ (a ΚΑΙ b)	! Λογική συνθήκη & ανάθεση
<b>Εμφάνισε</b> XOR	! Εντολή εξόδου
<b>Τέλος</b> Αποκλειστική_Διάζευξη	! Τερματισμός κώδικα

**9.** Η ΓΛΩΣΣΑ δεν διαθέτει τύπο για μιγαδικούς αριθμούς. Δημιουργήστε το δικό σας τύπο και χρησιμοποιήστε τον για να γράψετε συναρτήσεις για βασικές πράξεις με μιγαδικούς: πραγματικό μέρος, φανταστικό μέρος, μέτρο, συζυγής, πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση.

### Απάντηση

Ένας μιγαδικός αριθμός δεν είναι παρά ένα διατεταγμένο ζεύγος πραγματικών αριθμών, δηλαδή ένα σημείο στο πραγματικό επίπεδο. Το πιο κατάλληλο επομένως για τύπο μιγαδικών δεδομένων είναι δύο πραγματικές μεταβλητές.

**10.** Να διαβαστεί ένας αριθμός ημερών και να βρεθεί από πόσες εβδομάδες και από πόσες μέρες αποτελείται.

### Λύση

**Αλγόριθμος** Μετατροπή

**Εμφάνισε** “Δώσε ημέρες”

**Διάβασε** HM

weeks ← HM **div** 7 !ακέραιο ηλίκο

days ← HM **mod** 7 !ακέραιο υπόλοιπο

**Εμφάνισε** “Οι ημέρες αποτελούνται από”, Weeks, “εβδομάδες και”, days, “ημέρες”

**Τέλος** Μετατροπή

**11.** Η αξία ενός διαμερίσματος σε μια πολυκατοικία 7 ορόφων με ισόγειο υπολογίζεται ως εξής: στο ρετιρέ είναι 1400 € ανά τ.μ., ενώ η τιμή μειώνεται κατά 50 € ανά τ.μ. για κάθε όροφο. Ο φόρος μεταβίβασης είναι το 8% της αξίας του ενώ τα μεσιτικά έξοδα είναι 700 €. Να γραφεί αλγόριθμος που:

A) διαβάσει σε ποιο όροφο και πόσα τετραγωνικά μέτρα είναι ένα διαμέρισμα.

B) υπολογίζει την αξία του διαμερίσματος (αξία = σύνολο μέτρων \* τιμή μέτρου).

Γ) υπολογίζει το συνολικό κόστος κτίσης του διαμερίσματος (κόστος κτίσης = αξία + λοιπά έξοδα).

Δ) εμφανίζει το κόστος.

**Λύση**

**Αλγόριθμος** Αξία

**Διάβασε** TM, OP

Αξία ← (1400 - (7 - OP) \* 50) \* TM

Κόστος ← Αξία + 0,08 \* Αξία + 700

**Εμφάνισε** “Κόστος σε €”, Κόστος

**Τέλος** Αξία

**12.** Να μετατρέψετε σε εντολές εκχώρησης τις παρακάτω φράσεις:

α. Εκχώρησε στο I τον μέσο όρο των A, B, Γ.

β. Αύξησε την τιμή του M κατά 2.

γ. Διπλασίασε την τιμή του Λ.

δ. Μείωσε την τιμή του Χ κατά την τιμή του Ψ.

ε. Εκχώρησε στο A το υπόλοιπο της ακέραιας διαίρεσης του A με το B.

**Λύση**

α.  $I \leftarrow (A+B+Γ)/3$

β.  $M \leftarrow M+2$

γ.  $Λ \leftarrow Λ*2$

δ.  $Χ \leftarrow Χ-Ψ$

ε.  $A \leftarrow A \text{ MOD } B$

**13.** Αλγόριθμος ο οποίος διαβάζει μια τιμή που παριστάνει το χρόνο σε δευτερόλεπτα και την μετατρέπει σε τρεις νέες τιμές που παριστάνουν τον ίδιο χρόνο σε ώρες, λεπτά και δευτερόλεπτα.

**Λύση**

**Αλγόριθμος** Ασκ\_1

**Διάβασε** time

Hour ← time div 3600

! Ακέραια διαίρεση

Min ← (time mod 3600) div 60

Sec ← (time mod 3600) mod 60

**Εμφάνισε** Hour, "h :", Min, "m :", Sec, "s"

**Τέλος** Ασκ\_1

**14.** Ένας έμπορος αγόρασε σε τιμή χονδρικής εμπορεύματα αξίας  $K \in$  και τα πούλησε  $A \in$ . Αν το ποσό  $A$  προκύπτει από την καθαρή αξία  $K$  και το ποσοστό κέρδους  $X\%$  του εμπόρου επί της καθαρής αξίας  $K$ , να γίνει αλγόριθμος που να υπολογίζει και να εμφανίζει: α) το κέρδος του εμπόρου β) το εισπραχθέν ποσό  $A$ .

**Λύση**

**Αλγόριθμος** Εμπόριο

Διάβασε  $K$

$A \leftarrow K + K * (X/100)$

Εμφάνισε  $A - K, A$

**Τέλος** Εμπόριο

**15.** Να γραφεί αλγόριθμος που διαβάζει τις ώρες εργασίας, την ωριαία αντιμισθία και το ποσοστό των κρατήσεων ενός εργάτη και στη συνέχεια υπολογίζει κι εμφανίζει τις ακαθάριστες αποδοχές του, το ποσό των κρατήσεων που έχει και τις καθαρές του αποδοχές. Ο υπολογισμός των ποσών θα γίνεται ως εξής:

Καθ. Αποδοχές = ώρες \* ωριαία αντιμισθία

κρατήσεις = καθαρές αποδοχές \* ποσοστό

ακαθάριστες αποδοχές = καθαρές αποδοχές + κρατήσεις

**Λύση**

**Αλγόριθμος** Εργασία

Διάβασε ωρ, αντ, ποσ καθ  $\leftarrow$  ωρ \* αντ

κρατ  $\leftarrow$  καθ \* ποσ

ακαθ  $\leftarrow$  καθ + κρατ

Εμφάνισε ακαθ, κρατ, καθ

**Τέλος** Εργασία

**16.** Να γραφεί αλγόριθμος που διαβάζει τα στοιχεία ενός εργαζόμενου και στη συνέχεια υπολογίζει και εμφανίζει το χρηματικό ποσό που πρέπει να του δώσει η επιχείρηση ως δώρο στο τέλος του έτους. Τα στοιχεία του εργαζόμενου δίνονται με την παρακάτω σειρά: ημέρες εργασίας έτους, ημερήσια αμοιβή, ποσοστό δώρου επί τις %.

**Λύση**

**Αλγόριθμος** Εργασία

Διάβασε ημ, αμοιβη, ποσ

ετ  $\leftarrow$  ημ \* αμοιβη

δωρο  $\leftarrow$  ετ \* ποσ

Εμφάνισε δωρο

**Τέλος** Εργασία

## 1.4 Ασκήσεις προς επίλυση

1. Κατασκευάστε τον πίνακα αλήθειας της λογικής παράστασης:  $X$  και  $Y$  η  $Z$ .

2. Μία οικογένεια πήγε για μία εβδομάδα εκδρομή στην Τήνο, αγοράζοντας το πρόγραμμα Fly & Drive που προσφέρεται για μία εβδομάδα σε οικονομικές τιμές από τουριστικό πρακτορείο. Η συμμετοχή στην εκδρομή κοστίζει 200 € για κάθε άτομο και επιτρέπει ενοικίαση αυτοκινήτου προς 25 € την ημέρα, για όσες ημέρες επιλέξει ο πελάτης. Η οικογένεια χρησιμοποίησε αυτοκίνητο όλες τις ημέρες των διακοπών. Σχεδιάστε τον αλγόριθμο που (1) θα ζητάει τιμές για τα μεταβλητά δεδομένα του προβλήματος, (2) θα υπολογίζει το κόστος του ταξιδιού και (3) θα το εμφανίζει στο χρήστη.

3. Γράψτε ένα πρόγραμμα το οποίο διαβάζει τα μήκη των δύο κάθετων πλευρών ενός ορθογωνίου τριγώνου κι εκτυπώνει το εμβαδόν του τριγώνου.

4. Μια ναυτιλιακή εταιρεία μεταφέρει οικιακές συσκευές τυποποιημένου μεγέθους και χρησιμοποιεί ειδικά μεταφορικά κιβώτια τα οποία χωρούν 1, 5, 20 ή 50 οικιακές συσκευές. Γράψτε ένα πρόγραμμα το οποίο να δέχεται ως δεδομένο τον αριθμό των οικιακών συσκευών που πρόκειται να μεταφερθούν και υπολογίζει κι εμφανίζει τον απαιτούμενο αριθμό μεταφορικών κιβωτίων από κάθε μέγεθος, έτσι ώστε η μεταφορά να πραγματοποιηθεί με τον πλέον οικονομικό τρόπο (τον ελάχιστο δυνατό αριθμό μεταφορικών κιβωτίων χωρίς αχρησιμοποίητο χώρο).

5. Οι γωνίες μετρούνται σε μοίρες (ένας κύκλος έχει  $360^\circ$ ), οι μοίρες διαιρούνται σε λεπτά (1 μοίρα έχει 60 λεπτά) και τα λεπτά σε δευτερόλεπτα (1 λεπτό έχει 60 δευτερόλεπτα). Γράψτε ένα πρόγραμμα το οποίο διαβάζει τα μεγέθη δυο γωνιών (σε μοίρες, λεπτά και δευτερόλεπτα) και υπολογίζει κι εμφανίζει το άθροισμα τους με αντίστοιχη μορφή.

6. Να γραφεί πρόγραμμα που θα υπολογίζει τα ρέστα που πρέπει να δώσει ένα αυτόματο μηχάνημα έκδοσης εισιτηρίων. Τα εισιτήρια κοστίζουν 0.85€. Ο αλγόριθμος θα δέχεται το ποσό που πληρώνει ο πελάτης και θα επιστρέφει τον αριθμό των κερμάτων με αξία 2€, 1€, 0.50€, 0.20€, 0.10€ Ο αλγόριθμος πρέπει να λειτουργεί με τέτοιο τρόπο ώστε τα ρέστα να δίνονται με το μικρότερο δυνατό αριθμό νομισμάτων.

7. Να γραφεί πρόγραμμα που διαβάζει τα στοιχεία ενός εργαζόμενου και στη συνέχεια υπολογίζει και εμφανίζει το χρηματικό ποσό που πρέπει να του δώσει η επιχείρηση ως δώρο στο τέλος του έτους. Τα στοιχεία του εργαζόμενου δίνονται με την παρακάτω σειρά: ημέρες εργασίας έτους, ημερήσια αμοιβή, ποσοστό δώρου επί τις %.

8. Να γραφεί πρόγραμμα που διαβάζει τις συνολικές εισπράξεις τριμήνου μιας επιχείρησης ανά κατηγορία Φ.Π.Α. και στη συνέχεια υπολογίζει κι εμφανίζει το συνολικό ποσό του Φ.Π.Α. που πρέπει να καταβάλλει στο τέλος στην εφορία. Τα ποσοστά Φ.Π.Α. είναι οι σταθερές (0.06, 0.09, 0.23) και τα ποσά τριμήνου πρέπει να εισάγονται με την αντίστοιχη σειρά.

**9.** Να γραφεί πρόγραμμα που διαβάζει τον αριθμό επιστολών που στέλνει μια εταιρεία σε πελάτες εσωτερικού και εξωτερικού που έχει και στη συνέχεια υπολογίζει κι εμφανίζει το συνολικό κόστος. Τα ταχυδρομικά έξοδα κάθε επιστολής για το εσωτερικό και το εξωτερικό είναι αντίστοιχα 2 και 3,5 ευρώ.

**10.** Ένα κατάστημα πουλά τα εμπορεύματα του με έκπτωση  $E\%$ . Δεδομένων της αρχικής αξίας του εμπορεύματος  $K$  και του ποσοστού έκπτωσης  $E\%$ , να γραφεί πρόγραμμα που θα υπολογίζει και θα εμφανίζει το ποσό που θα πληρώσει ένας πελάτης για την αγορά εμπορευμάτων αξίας  $K$  και το ποσό της έκπτωσης που του αναλογεί.

**11.** Να γραφεί πρόγραμμα που θα διαβάζει την τιμή τεμαχίου ενός  $H/Y$ , το ποσοστό  $\Phi.P.A.$  και το πλήθος των  $H/Y$  προς παραγγελία. Στη συνέχεια να υπολογίζει και να εμφανίζει το συνολικό κόστος.

**12.** Να γίνει πρόγραμμα που θα δέχεται μία τιμή  $x$  και θα υπολογίζει την τιμή της παράστασης  $4\sinh(x) + 9\sinh(x + 2)$ , όπου το  $\sinh(x)$  είναι το υπερβολικό ημίτονο του  $x$  και ορίζεται  $\sinh(x) = (e^x - e^{-x})/2$ .

**13.** Να γίνει αλγόριθμος, και το σχετικό διάγραμμα ροής που θα διαβάζει το μισθό ενός υπαλλήλου και θα υπολογίζει από πόσα χαρτονομίσματα των 500, 200, 100, 50, 20, 10 και 5 ευρώ θα πρέπει να πληρωθεί. Ο αριθμός των χαρτονομισμάτων θα πρέπει να είναι ο λιγότερος δυνατός.

**14.** Ένας 6ψήφιος κωδικός θεωρείται έγκυρος αν ισχύουν τα ακόλουθα:  
1) Το άθροισμα του 1ου και του 2ου ψηφίου είναι ίσο με το 3ο ψηφίο  
2) το υπόλοιπο της διαίρεσης του 3ου με το 4ο ψηφίο είναι ίσο με το 5ο ψηφίο μείον 2  
3) και η διαφορά του 6ου με το 2ο ψηφίο είναι ίση με 3.  
Να γίνει αλγόριθμος που θα διαβάζει έναν εξαψήφιο αριθμό και θα ελέγχει αν ο κωδικός είναι έγκυρος ή όχι.

**15.** Ένας αυτόματος πωλητής αναψυκτικών λειτουργεί δεχόμενος μόνο κέρματα των 50, 20 και 10 λεπτών του ευρώ. Επίσης μπορεί να δώσει ρέστα του ίδιου ακριβώς τύπου (50, 20 και 10 λεπτών). Να γίνει αλγόριθμος που θα προσομοιώνει την λειτουργία του αυτόματου πωλητή: Αρχικά θα διαβάζει το αριθμό των αναψυκτικών που κάποιος επιθυμεί να αγοράσει. Στην συνέχεια θα διαβάζει τρεις τιμές, που αντιστοιχούν στον αριθμό των κερμάτων που το μηχάνημα μπορεί να δεχτεί. Στο τέλος, θα εμφανίζει τα ρέστα που πρέπει να επιστρέψει το μηχάνημα, αναλυτικά σε αριθμό 50λεπτων, 20λεπτων και 10λεπτων. Να σημειωθεί πως κάθε αναψυκτικό κοστίζει 60 λεπτά.









### Στόχοι ενότητας:

- ➔ Να μάθετε να χειρίζεστε λογικά δεδομένα και να εκτιμήσετε τη σπουδαιότητά τους.
- ➔ Να κατανοήσετε τη συμπεριφορά των σχεσιακών και λογικών τελεστών.
- ➔ Να μάθετε τις λεπτομέρειες των εντολών **AN** και τις περιπτώσεις στις οποίες χρησιμοποιούμε την κάθε μια.

## 2.1 Θεωρία-Οδηγίες

Χρησιμοποιώντας αποκλειστικά τη **δομή ακολουθίας**, μπορείτε να λύσετε μόνο προβλήματα στα οποία:

- ✓ όλα τα βήματα εκτελούνται πάντοτε
- ✓ δεν υπάρχουν απρόοπτα-εξαιρέσεις
- ✓ η σειρά των βημάτων είναι καθορισμένη

Για να δώσετε λύση σε πιο σύνθετα προβλήματα πρέπει να μπορείτε να δημιουργείτε στον αλγόριθμο λογικά μονοπάτια, εξετάζοντας απλά λογικά ερωτήματα. Σε αυτή την ενότητα θα ασχοληθείτε με μία νέα δομή, τη **δομή επιλογής**.

Οι ασκήσεις αυτής της ενότητας θα σας βοηθήσουν να κατανοήσετε πώς να σχεδιάζετε έναν αλγόριθμο, όταν αυτός περιλαμβάνει κάποιο βήμα που **δεν θα εκτελείται πάντοτε**. Έχετε πλέον τη δυνατότητα να δημιουργείτε **λογική** στους αλγορίθμους σας και να προσδιορίζετε εναλλακτικά μονοπάτια, ορίζοντας φίλτρα-κριτήρια επιλογής.

Η εντολή απλής επιλογής **AN** συνθήκη **TOTE** ορίζεται διαφορετικά στη Ψευδογλώσσα και διαφορετικά στη ΓΛΩΣΣΑ. Στη ΓΛΩΣΣΑ απαιτείται ένα **ΤΕΛΟΣ\_AN** για κάθε AN, ενώ στη Ψευδογλώσσα αν στο τότε έχεις μόνο μια εντολή, επιτρέπεται να γραφεί στην ίδια ευθεία. Δηλαδή: **AN** συνθήκη **TOTE** εντολή

Η διαδικασία της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές και ακολουθεί η απόφαση εκτέλεσης κάποιας ενέργειας, ανάλογα με την τιμή της συνθήκης.

Και λίγοι **ορισμοί**:

**Εμφωλευμένα AN** ονομάζονται δύο ή περισσότερες εντολές της μορφής **AN ...ΑΛΛΙΩΣ** που περιέχονται η μία μέσα στην άλλη.

**Συνθήκη**: Μια έκφραση σε πρόγραμμα που μπορεί να εκτιμηθεί είτε ως αληθής είτε ως ψευδής, όταν εκτελείται το πρόγραμμα.

**Μεταβλητή ελέγχου**: Μεταβλητή της οποίας η τιμή ελέγχει τον αριθμό εκτελέσεων ενός βρόχου.

**Εντολές ελέγχου**: εντολές που επηρεάζουν τον τρόπο με τον οποίο εκτελούνται άλλες εντολές. Οι εντολές ελέγχου χωρίζονται σε δύο βασικές κατηγορίες:

- ✓ Εντολές **υπό συνθήκη**: χρησιμοποιούνται για την επιλογή δύο ή περισσότερων ανεξάρτητων διαδρομών σε ένα πρόγραμμα, ανάλογα με το αποτέλεσμα κάποιου ελέγχου συνθήκης.

✓ Εντολές **επανάληψης (Βλέπε Ενότητα 4)**: χρησιμοποιούνται όταν χρειάζεται να επαναλαμβάνεται στο πρόγραμμα μια λειτουργία ένα καθορισμένο πλήθος φορών (**ΓΙΑ**) ή όσο ισχύει μια συγκεκριμένη συνθήκη (**ΟΣΟ/ΜΕΧΡΙΣ\_ΟΤΟΥ**).

Οι μεταβλητές **λογικού τύπου** χρησιμοποιούνται συχνά στα προγράμματα ως **σημαίες** (flags), για να σηματοδοτήσουν δηλαδή αν έχετε ολοκληρώσει ή όχι κάποια φάση της λειτουργίας.

Η εντολή **ΑΛΛΙΩΣ\_ΑΝ** είναι διαφορετική από την **ΑΛΛΙΩΣ ΑΝ**.

Πριν χρησιμοποιήσεις **εμφωλευμένα ΑΝ**, σκέψου μήπως το ίδιο πρόγραμμα μπορεί να υλοποιηθεί απλούστερα με σύνθετες λογικές εκφράσεις ή την **πολλαπλή ΑΝ**.

Μην ξεχνάτε: η χρήση **εσοχών** καθιστά πολύ πιο ευκολονόητη τη δομή του προγράμματος, αυξάνοντας τη σαφήνειά του.

## 2.2 Ερωτήσεις Σωστού-Λάθους

Επιλέξτε Σωστό ή Λάθος για καθένα από τα παρακάτω:

1. Η λογική πράξη "ή" μεταξύ δύο προτάσεων είναι ψευδής, όταν οποιαδήποτε από τις δύο προτάσεις είναι ψευδής.
2. Στη δομή επιλογής μπορεί μία ή περισσότερες εντολές να μην εκτελεστούν.
3. Η δομή επιλογής επιτρέπει τη διενέργεια ελέγχων κατά τη διάρκεια εκτέλεσης των αλγορίθμων.
4. Αν η M είναι μια λογική μεταβλητή, η εντολή εκχώρησης  $M \leftarrow \text{'Αληθής'}$  είναι επιτρεπτή.
5. Αν έχουμε την εντολή  $A \leftarrow (X < 5)$ , τότε η μεταβλητή A είναι λογικού τύπου.
6. Το διάγραμμα ροής ενός αλγορίθμου που έχει μία εντολή  $AN \dots ALLIΩΣ$  θα περιέχει δύο ρόμβους.
7. Η χρήση εμφωλευμένων  $AN$  είναι καλή προγραμματιστική τακτική.
8. Στην εντολή  $a \leftarrow \text{"Φλεβάρης"} > \text{"Μάρτης"}$  το a παίρνει την τιμή  $ALLHΘΣ$ .
9. Η λογική συνθήκη "α" < "β" είναι αληθής.
10. Η λογική παράσταση  $x \text{ ΚΑΙ } OXI(x)$  είναι πάντα ψευδής ανεξάρτητα από την τιμή του x.

### Απάντηση

1.Λ, 2.Σ, 3.Σ, 4.Λ, 5.Σ, 6.Λ, 7.Λ, 8.Σ, 9.Σ, 10.Σ

Δίνονται οι παρακάτω εντολές από ένα τμήμα προγράμματος:

...

**ΔΙΑΒΑΣΕ** x,y

$z \leftarrow x > y$

...

Να επιλέξετε Σωστό ή Λάθος για τους παρακάτω ισχυρισμούς:

1. Η x είναι λογική μεταβλητή.
2. Τα z , x είναι πάντα μεταβλητές διαφορετικού τύπου.
3. Το x μπορεί να έχει τιμή μεγαλύτερη του y.

### Απάντηση

1.Λ, 2. Σ, 3.Σ

## 2.3 Λυμένες Ασκήσεις

1. Ο ακόλουθος αλγόριθμος μετατρέπει τα πόδια σε μέτρα ή τα μέτρα σε πόδια, ανάλογα με την αίτηση που κάνει ο χρήστης.

### Λύση

**Αλγόριθμος** Μετατροπή

**Εμφάνισε** “Δώσε τιμή”

**Διάβασε**  $x$

**Εμφάνισε** “1. Πόδια σε μέτρα, 2. Μέτρα σε πόδια”

**Εμφάνισε** “Δώσε επιλογή σου”

**Διάβασε** choice

**Αν** choice=1 **τότε Εμφάνισε**  $(x/3.28)$

! Απλή επιλογή

**Αν** choice=2 **τότε Εμφάνισε**  $(x*3.28)$

**Τέλος** Μετατροπή

2. Απλοποιήστε το παρακάτω σύνολο εντολών.

**Διάβασε**  $a$

**Αν**  $(a > 0) \vee (a < 0)$  **τότε**

$\beta \leftarrow \text{ΑΛΗΘΗΣ ΚΑΙ } (a < 100)$

! Σύνθετη επιλογή

**αλλιώς**

$\beta \leftarrow (a = 0)$

**Τέλος\_Αν**

**Εμφάνισε**  $\beta$

### Λύση

**Διάβασε**  $a$

**Αν**  $a < > 0$  **τότε**

$\beta \leftarrow a < 100$

**αλλιώς**

$\beta \leftarrow \text{Αληθής}$

**Τέλος\_Αν**

**Εμφάνισε**  $\beta$

3. Αντικαταστήστε την παρακάτω δομή σύνθετης επιλογής με μία μόνο εντολή εκχώρησης, που να επιφέρει το ίδιο ακριβώς αλγοριθμικό αποτέλεσμα.

**Αν**  $X*Y+Z > 100$  **τότε**

$\rho \leftarrow \text{Ψευδής}$

**αλλιώς**

$\rho \leftarrow \text{Αληθής}$

**Τέλος\_Αν**

### Λύση

$\rho \leftarrow \text{ΟΧΙ}(X*Y+Z > 100)$

4. Ποιό ή ποιά αλγοριθμικά κριτήρια παραβιάζονται στον παρακάτω αλγόριθμο.

**Αλγόριθμος ΑΣΚ**

**Διάβασε** λ

μ ← κ+λ

**Αν** μ > 100 **τότε**

μ ← μ-10

**Τέλος\_Αν**

**Τέλος ΑΣΚ**

**Λύση**

α) **Είσοδος**, γιατί το κ δεν έχει τιμή και παρ' όλα αυτά χρησιμοποιείται στην εντολή μ ← κ+λ.

β) **Έξοδος**, γιατί ο Αλγόριθμος δεν εξάγει κανένα αποτέλεσμα.

5. Δίνεται το παρακάτω τμήμα αλγορίθμου.

**Διάβασε** κ

**Αν** (κ > 0) ή (κ < 0) **τότε**

λ ← όχι (Ψευδής ή κ > 20)

Αλλιώς

λ ← (κ=0)

**Τέλος\_Αν**

**Εκτύπωσε** λ

Στο παραπάνω τμήμα υπάρχουν περιττοί έλεγχοι. Να γράψετε ξανά το παραπάνω τμήμα χωρίς τους περιττούς ελέγχους.

**Λύση**

**Διάβασε** κ

**Αν** κ <> 0 **τότε**

λ ← όχι(κ > 20)

Αλλιώς

λ ← Αληθής

**Τέλος\_Αν**

**Εκτύπωσε** λ

6. Ξαναγράψτε το παρακάτω σύνολο εντολών εξαφανίζοντας τους περιττούς ελέγχους.

**Αν** α < 0 **τότε**

**Εμφάνισε** α

**αλλιώς\_Αν** (α >= 0) **ΚΑΙ** (α < 199) **τότε**

! Πολλαπλή επιλογή

**Εμφάνισε** -α

**αλλιώς\_Αν** (α >= 199) **ΚΑΙ** (α < 234) **τότε**

**Εμφάνισε** α+1

**αλλιώς**

**Αν** α >= 234 **τότε**

**Εμφάνισε** 1-α

**Τέλος\_Αν**

**Τέλος\_Αν**

## Λύση

Αν  $a < 0$  τότε  
Εμφάνισε  $a$   
αλλιώς\_Αν  $a < 199$  τότε  
Εμφάνισε  $-a$   
αλλιώς\_Αν  $a < 234$  τότε  
Εμφάνισε  $a+1$   
αλλιώς  
Εμφάνισε  $1-a$   
Τέλος\_Αν

7. Πως μετατρέπεται το παρακάτω τμήμα αλγορίθμου σε απλή δομή επιλογής;

Αν  $k > 0$  τότε

$f \leftarrow 130+z$

Αλλιώς

$f \leftarrow 100$

Τέλος\_Αν

## Λύση

$f \leftarrow 100$

Αν  $k > 0$  τότε

$f \leftarrow f+30+z$

Τέλος\_Αν

8. Να γραφεί η συνθήκη χωρίς λογικό τελεστή:

Αν  $a > b$  η  $a = b$  τότε

## Απάντηση

Αν  $a \geq b$  τότε

9. Να γραφεί με μία εντολή Αν:

Αν  $z > 0$  τότε

Εμφάνισε 'Hallo world'

Αλλιώς

Αν  $g > 0$  τότε

Εμφάνισε 'Hallo world'

Τέλος\_Αν

Τέλος\_Αν

## Λύση

Αν  $z > 0$  Η  $g > 0$  τότε

Εμφάνισε 'Hallo world'

Τέλος\_Αν

**10.** Δίνεται το παρακάτω τμήμα αλγορίθμου:

**Αν**  $X > 1$  **τότε**

$K \leftarrow$  **Αληθής**

**Αλλιώς**

$K \leftarrow$  **Ψευδής**

**Τέλος\_Αν**

Να γράψετε συμπληρωμένη την παρακάτω εντολή εκχώρησης, ώστε να έχει το ίδιο αποτέλεσμα με το παραπάνω τμήμα αλγορίθμου.

$K \leftarrow$ .....

**Λύση**

$K \leftarrow X > 1$

**11.** Με το νέο σύστημα πληρωμής των διοδίων, οι οδηγοί των τροχοφόρων έχουν τη δυνατότητα να πληρώνουν το αντίτιμο των διοδίων με ειδική μαγνητική κάρτα. Υποθέστε ότι υπάρχει μηχανήμα το οποίο διαθέτει είσοδο για την κάρτα και φωτοκύτταρο. Το μηχανήμα διαβάζει από την κάρτα το υπόλοιπο των χρημάτων και το αποθηκεύει σε μία μεταβλητή  $Y$  και με το φωτοκύτταρο, αναγνωρίζει τον τύπο του τροχοφόρου και το αποθηκεύει σε μία μεταβλητή  $T$ . Υπάρχουν τρεις τύποι τροχοφόρων: δίκυκλα ( $\Delta$ ), επιβατικά ( $E$ ) και φορτηγά ( $\Phi$ ), με αντίτιμο διοδίων 1, 2 και 3 ευρώ αντίστοιχα. Να αναπτύξετε αλγόριθμο, ο οποίος:

α. ελέγχει τον τύπο του τροχοφόρου και εκχωρεί στη μεταβλητή  $A$  το αντίτιμο των διοδίων, ανάλογα με τον τύπο του τροχοφόρου

β. ελέγχει την πληρωμή των διοδίων με τον παρακάτω τρόπο. Αν το υπόλοιπο της κάρτας επαρκεί για την πληρωμή του αντιτίμου των διοδίων, αφαιρεί το ποσό αυτό από την κάρτα. Αν η κάρτα δεν έχει υπόλοιπο, το μηχανήμα ειδοποιεί με μήνυμα για το ποσό που πρέπει να πληρωθεί. Αν το υπόλοιπο δεν επαρκεί, μηδενίζεται η κάρτα και δίνεται με μήνυμα το ποσό που απομένει να πληρωθεί. (Θέμα Πανελλαδικών).

**Λύση**

**Αλγόριθμος** Θέμα\_3

**Δεδομένα** //  $Y, T$  //

**Αν**  $T = \text{"}\Delta\text{"}$  **τότε**

$A \leftarrow 1$

**αλλιώς\_Αν**  $T = \text{"}E\text{"}$  **τότε**

$A \leftarrow 2$

**αλλιώς**

$A \leftarrow 3$

**Τέλος\_Αν**

**Αν**  $A \geq Y$  **τότε**

$Y \leftarrow Y - A$

**αλλιώς\_Αν**  $Y = 0$  **τότε**

**Εκτύπωσε** "Πρέπει να πληρώσετε",  $A$

**αλλιώς**

**Εκτύπωσε** "Πρέπει να πληρώσετε",  $A - Y$

$Y \leftarrow 0$

**Τέλος\_Αν**

**Τέλος** Θέμα\_3



**12.** Να γίνει πρόγραμμα το οποίο να ζητάει μια χρονολογία και να εμφανίζει την ημερομηνία του Πάσχα αυτής της χρονιάς με σωστή αναφορά του μήνα, για παράδειγμα: "Ημερομηνία του Πάσχα: 19 Απριλίου" ή "Ημερομηνία του Πάσχα: 3 Μαΐου".

### Λύση

Η ημερομηνία του Πάσχα κάποιας χρονολογίας υπολογίζεται ως εξής:

Ετ = η χρονολογία (δεδομένο)

Π = η ημερομηνία του Πάσχα (αποτέλεσμα) . Θέτουμε:

Ετ:19 = α

Ετ:4 = β

Ετ:7 = γ

$(19\alpha+16):30 = \delta$

$(2\beta+4\gamma+6\delta):7 = \zeta$

όπου το σύμβολο ":" σημαίνει ΑΚΕΡΑΙΟ ΥΠΟΛΟΙΠΟ ΔΙΑΙΡΕΣΗΣ.

Η ημερομηνία του Πάσχα θα δίνεται από τον τύπο:  $\Pi = \delta + \zeta + 3$  Απριλίου. Αν  $\delta + \zeta + 3 > 30$  τότε σημαίνει ότι το Πάσχα πέφτει μέσα στον Μάιο. Δηλαδή αν  $\delta + \zeta + 3 = 35$ , η ημερομηνία είναι 5 Μαΐου. (ΣΗΜ.: Το Πάσχα πέφτει μόνο Απρίλιο & Μάιο).

**ΠΡΟΓΡΑΜΜΑ** Πάσχα

**ΜΕΤΑΒΛΗΤΕΣ**

*! Τμήμα δήλωσης μεταβλητών*

**ΑΚΕΡΑΙΕΣ:** α,β,γ,δ,ζ,Ετ,Π

**ΧΑΡΑΚΤΗΡΕΣ:** μήνας

**ΑΡΧΗ**

**ΓΡΑΨΕ** 'Δώσε τη χρονολογία:'

**ΔΙΑΒΑΣΕ** Ε

$\alpha \leftarrow \text{Ετ MOD } 19$

$\beta \leftarrow \text{Ετ MOD } 4$

$\gamma \leftarrow \text{Ετ MOD } 7$

$\delta \leftarrow (19 * \alpha + 16) \text{ MOD } 30$

$\zeta \leftarrow (2 * \beta + 4 * \gamma + 6 * \delta) \text{ MOD } 7$

$\Pi \leftarrow \delta + \zeta + 3$

**ΑΝ**  $\Pi \leq 30$  **ΤΟΤΕ**

μήνας  $\leftarrow$  ' Απριλίου'

**ΑΛΛΙΩΣ**

μήνας  $\leftarrow$  ' Μαΐου'

$\Pi \leftarrow \Pi - 30$

**ΤΕΛΟΣ\_ΑΝ**

**ΓΡΑΨΕ** 'Ημερομηνία του Πάσχα: ', Π, μήνας

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**13.** Δίνονται τα παρακάτω τμήματα αλγορίθμου σε φυσική γλώσσα. Να γραφεί το αντίστοιχο τμήμα κώδικα σε ΓΛΩΣΣΑ

1. Αν η βαθμολογία (ΒΑΘ) είναι μεγαλύτερη από το Μέσο Όρο (ΜΟ), τότε να τυπώνει «Πολύ Καλά», αν είναι ίση ή μικρότερη του Μέσου Όρου μέχρι και τρεις μονάδες να τυπώνει «Καλά», σε κάθε άλλη περίπτωση να τυπώνει «Μέτρια».

2. Αν το τμήμα (ΤΜΗΜΑ) είναι το Β2 και η βαθμολογία (ΒΑΘ) είναι μεγαλύτερη από 17, τότε να τυπώνει το επώνυμο (ΕΠΩΝ).

3. Αν η απάντηση (ΑΠ) δεν είναι Ν ή ν ή Ο ή ο, τότε να τυπώνει «Λάθος απάντηση».
4. Αν ο αριθμός (Χ) είναι αρνητικός ή το ημίτονό του είναι μηδέν, τότε να τυπώνει «Λάθος δεδομένο», αλλιώς να υπολογίζει και να τυπώνει την τιμή της Χ. (Θέμα Πανελλαδικών)

#### Λύση

**ΑΝ ΒΑΘ>ΜΟ ΤΟΤΕ ! 1**

ΓΡΑΨΕ “Πολύ καλά”

**ΑΛΛΙΩΣ\_ΑΝ ΒΑΘ<=ΜΟ-3 ΤΟΤΕ**

ΓΡΑΨΕ “ΚΑΛΑ”

**ΑΛΛΙΩΣ**

ΓΡΑΨΕ “ΜΕΤΡΙΑ”

**ΤΕΛΟΣ\_ΑΝ**

**ΑΝ ΤΜΗΜΑ=“Β2” ΚΑΙ ΒΑΘ>17 ΤΟΤΕ ! 2**

ΓΡΑΨΕ ΕΠΩΝ

**ΤΕΛΟΣ\_ΑΝ**

**ΑΝ ΑΠ<>“Ν” ΚΑΙ ΑΠ<>“ν” ΑΠ<>“Ο” ΑΠ<>“ο” ΤΟΤΕ ! 3**

ΓΡΑΨΕ “Λάθος απάντηση”

**ΤΕΛΟΣ\_ΑΝ**

**ΑΝ Χ<0 Η ΗΜ(Χ)=0 ΤΟΤΕ ! 4**

ΓΡΑΨΕ “Λάθος δεδομένο”

**ΑΛΛΙΩΣ**

ΓΡΑΨΕ Χ

**ΤΕΛΟΣ\_ΑΝ**

**14.** Ένας πωλητής λαμβάνει bonus 5% για πωλήσεις μέχρι τις 100.000 € και διπλασιάζεται (δηλ. γίνεται 10%) μόνον για τις πωλήσεις που υπερβαίνουν τις 100.000 €. Να γράψετε έναν αλγόριθμο για να υπολογίσετε το συνολικό ποσό προμήθειας, αν δίνεται ως δεδομένο το ποσό των πωλήσεων.

#### Λύση

**Αλγόριθμος Πωλήσεις**

Διάβασε ποσό

Αν ποσό < 100000 τότε

προμ <- (5/100)\*ποσό

αλλιώς

προμ <- (5/100)\*100000 + (10/100)\*(ποσό-100000)

τέλος\_αν

Εμφάνισε προμ

**Τέλος Πωλήσεις**

**15.** Σε κάποιο σχολικό αγώνα, για το άθλημα «Άλμα εις μήκος» καταγράφεται για κάθε αθλητή η καλύτερη έγκυρη επίδοσή του. Τιμής ένεκεν, πρώτος αγωνίζεται ο περσινός πρωταθλητής. Η Επιτροπή του αγώνα διαχειρίζεται τα στοιχεία των αθλητών που αγωνίστηκαν.

Να γράψετε αλγόριθμο ο οποίος:

1. Να ζητάει το ρεκόρ αγώνων και να το δέχεται, εφόσον είναι θετικό και μικρότερο των 10 μέτρων.
2. Να ζητάει τον συνολικό αριθμό των αγωνιζομένων και για κάθε αθλητή το όνομα και την επίδοσή του σε μέτρα με τη σειρά που αγωνίστηκε.
3. Να εμφανίζει το όνομα του αθλητή με τη χειρότερη επίδοση.
4. Να εμφανίζει τα ονόματα των αθλητών που κατέρριψαν το ρεκόρ αγώνων. Αν δεν υπάρχουν τέτοιοι αθλητές, να εμφανίζει το πλήθος των αθλητών που πλησίασαν το ρεκόρ αγώνων σε απόσταση όχι μεγαλύτερη των 50 εκατοστών.
5. Να βρίσκει και να εμφανίζει τη θέση που κατέλαβε στην τελική κατάταξη ο περσινός πρωταθλητής.

Σημείωση: Να θεωρήσετε ότι κάθε αθλητής έχει έγκυρη επίδοση και ότι όλες οι επιδόσεις των αθλητών που καταγράφονται είναι διαφορετικές μεταξύ τους. (Θέμα Πανελλαδικών)

## Λύση

### Αλγόριθμος Θέμα3\_2010

Αρχή\_επανάληψης

Διάβασε ρεκόρ

Μέχρις\_ότου ρεκόρ > 0 και ρεκόρ < 10

Διάβασε αρ\_αθλητών

Διάβασε όνομα, μήκος\_1ου

min <- μήκος\_1ου

min\_όνομα <- όνομα

τελική\_θέση\_πρωταθλητή <- 1

βρέθηκε <- ψευδής

Αν μήκος\_1ου > ρεκόρ τότε

Εμφάνισε όνομα

βρέθηκε <- αληθής

Αλλιώς

    Αν μήκος\_1ου >= ρεκόρ - 0.5 τότε

        κοντά\_στο\_ρεκόρ <- κοντά\_στο\_ρεκόρ + 1

    Τέλος\_αν

Τέλος\_αν

Για i από 2 μέχρι αρ\_αθλητών

Διάβασε όνομα, μήκος\_αθλητή

    Αν μήκος\_αθλητή < min τότε

        min <- μήκος\_αθλητή

        min\_όνομα

<-

!!broken!!

Τέλος\_αν

    Αν μήκος\_αθλητή < μήκος\_1ου τότε

        τελική\_θέση\_πρωταθλητή <- τελική\_θέση\_πρωταθλητή + 1

    Τέλος\_αν

    Αν μήκος\_αθλητή > ρεκόρ τότε

        Εμφάνισε όνομα

        βρέθηκε <- αληθής

    Αλλιώς

        Αν μήκος\_αθλητή >= ρεκόρ - 0.5 τότε

            κοντά\_στο\_ρεκόρ <- κοντά\_στο\_ρεκόρ + 1

        Τέλος\_αν

Τέλος\_αν  
Τέλος\_επανάληψης  
Αν βρέθηκε = ψευδής τότε  
Εμφάνισε κοντά\_στο\_ρεκόρ  
Τέλος\_αν  
Εμφάνισε min\_όνομα  
Εμφάνισε τελική\_θέση\_πρωταθλητή  
**Τέλος** Θέμα3\_2010

## 2.4 Ασκήσεις προς επίλυση

1. Να γραφεί ένα πρόγραμμα που θα βρίσκει ποιος από τους εξής αριθμούς είναι μεγαλύτερος:  $x^n$  ή  $n^x$ .

2. Γράψτε εκφράσεις Boolean οι οποίες να ικανοποιούν τις παρακάτω συνθήκες:

a) το  $b$  είναι αυστηρά ανάμεσα στο 2 και το 5

b) το  $a$  είναι μικρότερο του 6 ή μεγαλύτερο του 10

c)  $p=q=r$

d) το  $x$  είναι μικρότερο του 3 ή το  $y$  είναι μικρότερο του 3, αλλά όχι και τα δυο μαζί.

3. Θεωρώντας δεδομένες τις λογικές μεταβλητές  $a$ ,  $b$ ,  $c$ , γράψτε λογικές εκφράσεις οι οποίες να ικανοποιούν τις παρακάτω συνθήκες:

a) να είναι αληθής όταν και μόνον όταν  $a$ ,  $b$  είναι αληθείς (ΑΛΗΘΗΣ) και  $c$  είναι ψευδής (ΨΕΥΔΗΣ)

b) να είναι αληθής όταν και μόνον όταν  $a$  είναι αληθής και τουλάχιστον μία από τις  $b$ ,  $c$  είναι αληθής

c) να είναι αληθής όταν και μόνον όταν ακριβώς μία από τις  $a$ ,  $b$  είναι αληθής.

4. Να γραφεί αλγόριθμος ο οποίος δέχεται δύο λογικές εισόδους και βρίσκει- εμφανίζει τη λογική τους σύζευξη χωρίς χρήση λογικού τελεστή.

5. Γράψτε ξεχωριστά προγράμματα που να επιλύουν και να εμφανίζουν τα αποτελέσματα για καθένα από τα παρακάτω προβλήματα:

α. Αν η μεταβλητή  $Code$  έχει τιμή 'T', τότε η  $Timh$  να αυξηθεί με την πρόσθεση  $Rate$  ποσοστού της  $Timh$  στην  $Timh$  και να εμφανίσετε το αποτέλεσμα, αλλιώς να εμφανίσετε το μήνυμα ERROR.

β. Αν  $A$  και  $B$  είναι μεταβλητές και  $0 < A < 5$ , τότε το  $B$  να γίνει ίσο με  $1/A$ , αλλιώς το  $B$  να γίνει ίσο με  $A$ . Να εμφανίσετε την τιμή του  $B$ .

γ. Η μεταβλητή  $Disekto$  να πάρει την τιμή αληθής, αν η ακέραια μεταβλητή  $Year$  είναι αριθμός δίσεκτου έτους (δηλαδή πολλαπλασίου του 4, αλλά όχι του 100. Δίσεκτα είναι και τα πολλαπλάσια του 400 εκτός των πολλαπλασίων του 2000), αλλιώς να πάρει την τιμή ψευδής. Να εμφανίσετε την τιμή της  $Disekto$ .

6. Για την παραγγελία μιας ποσότητας ενός προϊόντος σε τεμάχια δίνονται ο κωδικός του προϊόντος (4 ψηφίος ακέραιος) και ο αριθμός τεμαχίων. Το ποσοστό έκπτωσης εξαρτάται από τον αριθμό τεμαχίων ως εξής:

Αριθμός τεμαχίων	Έκπτωση %
1-30	10
31-70	20
71-...	35

Η τιμή πώλησης του προϊόντος βρίσκεται αν προσθέσουμε στα 2 τελευταία ψηφία του κωδικού τα 2 πρώτα (π.χ. 1234 46). Να υπολογιστούν και να εμφανιστούν η τιμή πώλησης

προϊόντος, η έκπτωση και η τελική τιμή της παραγγελίας.

**7.** Ο υπολογισμός των ταχυδρομικών τελών για αποστολή δέματος πραγματοποιείται με τον εξής τρόπο:

Πάγιο τέλος 2 ευρώ (επιβάρυνση όλων των μικροδεμάτων ανεξαρτήτως λοιπών τελών).  
Επιπλέον χρέωση:

Για δέματα από 1 ως 20 γραμμάρια 0,2 ευρώ ανά γραμμάριο. Για δέματα από 20 ως 100 γραμμάρια χρέωση 0,3 ευρώ ανά γραμμάριο για κάθε γραμμάριο επιπλέον των 20. Για δέματα άνω 100 γραμμαρίων χρέωση 0,4 ευρώ ανά γραμμάριο για κάθε γραμμάριο επιπλέον των 100.

Τα δέματα εξωτερικού επιβαρύνονται με προσαύξηση 16% αν αποστέλλονται προς χώρα της Ευρωπαϊκής Ένωσης ή με 19% προς τις λοιπές χώρες. Η επιβάρυνση για κάθε δέμα υπολογίζεται: (πάγιο + χρέωση) \* ποσοστό προσαύξησης %.

Να γραφεί ένα πρόγραμμα για τον υπολογισμό και την εμφάνιση των ταχυδρομικών τελών με δεδομένο το βάρος ενός δέματος και τη χώρα αποστολής.

0: αποστολή προς χώρα της ΕΕ, 1: αποστολή προς χώρα εκτός της ΕΕ.

Τα δεδομένα που πρέπει να εμφανίζει το πρόγραμμα είναι το βάρος του δέματος, η επιπλέον χρέωση, η επιβάρυνση και τα ταχυδρομικά τέλη της αποστολής.

**8.** Ο παρακάτω αλγόριθμος προτάθηκε για να ελέγχει και να εκτυπώνει, αν ένας μη αρνητικός ακέραιος αριθμός είναι μονοψήφιος, διψήφιος ή τριψήφιος. Στην περίπτωση που δοθεί αριθμός αρνητικός ή με περισσότερα από 3 ψηφία ο αλγόριθμος πρέπει να εμφανίζει το μήνυμα «Λάθος Δεδομένα».

**Αλγόριθμος Ψηφία**

**Διάβασε**  $x$

**Αν**  $x \geq 0$  **και**  $x < 10$  **τότε**

**Εμφάνισε** 'Μονοψήφιος'

**αλλιώς\_Αν**  $x < 100$  **τότε**

**Εμφάνισε** 'Διψήφιος'

**αλλιώς\_Αν**  $x < 1000$  **τότε**

**Εμφάνισε** 'Τριψήφιος'

**αλλιώς**

**Εμφάνισε** 'Λάθος Δεδομένα'

**Τέλος\_Αν**

**Τέλος Ψηφία**

Ο παραπάνω Αλγόριθμος έχει λάθος. Δώστε ένα παράδειγμα εισόδου που θα καταδείξει το λάθος που υπάρχει στον αλγόριθμο. Στη συνέχεια να γράψετε τον αλγόριθμο κάνοντας τις απαραίτητες διορθώσεις, έτσι ώστε να λειτουργεί σωστά. (Θέμα Πανελλαδικών)

**9.** Να μετατραπούν οι παρακάτω προτάσεις σε σύνθετες εκφράσεις (συνθήκες) στη ΓΛΩΣΣΑ:

1. Ο  $x$  είναι μεγαλύτερος του  $-1$  και μικρότερος ή ίσος του  $10$ .

2. Ο  $x$  είναι ίσος με  $1$  ή με  $5$  ή με  $-40$ .

3. Ο  $x$  είναι μεγαλύτερος του  $50$  αλλά όχι ίσος με  $100$ .

4. Ο ακέραιος  $x$  είναι θετικός αριθμός πολλαπλάσιο του  $3$ .

5. Ο ακέραιος  $x$  διαιρείται ακριβώς με το  $4$  αλλά όχι με το  $100$ . (Θέμα Πανελλαδικών)

**10.** Να αναπτύξετε έναν αλγόριθμο, ώστε:

α) να διαβάσει έναν πραγματικό αριθμό μεγαλύτερο του μηδενός και μικρότερο του  $1000$

και να κάνει έλεγχο ορθής καταχώρησης του αριθμού,  
β) να ελέγχει αν είναι ακέραιος και να εμφανίζει τη λέξη «ΑΚΕΡΑΙΟΣ» αλλιώς να εμφανίζει τη λέξη «ΠΡΑΓΜΑΤΙΚΟΣ»,  
γ) να ελέγχει, στην περίπτωση που ο αριθμός είναι ακέραιος, αν είναι άρτιος ή περιττός και να εμφανίζει τη λέξη «ΑΡΤΙΟΣ» ή «ΠΕΡΙΤΤΟΣ» αντίστοιχα. (Θέμα Πανελλαδικών)

**11.** Σε κάποια εξεταστική δοκιμασία ένα γραπτό αξιολογείται από δύο βαθμολογητές στη βαθμολογική κλίμακα [0, 100]. Αν η διαφορά μεταξύ των βαθμολογιών του α' και του β' βαθμολογητή είναι μικρότερη ή ίση των 20 μονάδων της παραπάνω κλίμακας, ο τελικός βαθμός είναι ο μέσος όρος των δύο βαθμολογιών. Αν η διαφορά μεταξύ των βαθμολογιών είναι μεγαλύτερη από 20 μονάδες, το γραπτό δίνεται για αναβαθμολόγηση σε τρίτο βαθμολογητή. Ο τελικός βαθμός του γραπτού προκύπτει τότε από τον μέσο όρο των τριών βαθμολογιών. Να αναπτύξετε αλγόριθμο ο οποίος, αφού ελέγξει την εγκυρότητα των βαθμών στην βαθμολογική κλίμακα [0,100], να υλοποιεί την παραπάνω διαδικασία εξαγωγής τελικού βαθμού και να εμφανίζει τον τελικό βαθμό του γραπτού στην εικοσαβάθμια κλίμακα. (Θέμα Πανελλαδικών)

**12.** Γράψτε ένα πρόγραμμα που θα εμφανίζει στην οθόνη το παρακάτω μενού επιλογών και θα δίνει τη δυνατότητα στο χρήστη να επιλέγει μία από τις περιπτώσεις. Μετά την επιλογή θα εμφανίζεται στην οθόνη ο τίτλος της επιλογής και σε διαφορετική σειρά το μήνυμα "PRESS ENTER TO CONTINUE...".

PROCESSING STUDENTS' DATA

1. INSERT
2. EDIT
3. DELETE
4. SHOW
5. EXIT

CHOICE:

**13.** Μία εταιρία κινητής τηλεφωνίας προσφέρει ένα πρόγραμμα χρήσης, με φθηνές κλήσεις ειδικά για φοιτητές. Στο συγκεκριμένο πρόγραμμα η εταιρία δεν προσφέρει τη δυνατότητα αποστολής γραπτών μηνυμάτων. Το κόστος των κλήσεων εξαρτάται αποκλειστικά από τη διάρκεια ομιλίας και δεν υπάρχει ελάχιστος χρόνος χρέωσης. Η εταιρία προσφέρει αυτό το πακέτο χωρίς πάγιο αλλά με ελάχιστη συνολική διάρκεια κλήσεων τα 60 δευτερόλεπτα. Αυτό σημαίνει ότι εάν ο μαθητής κάνει κλήσεις με συνολική διάρκεια μικρότερη από 60 δευτερόλεπτα, τότε και μόνο τότε το κόστος των κλήσεων επιβαρύνεται με πάγιο 50 λεπτών του €. Διαφορετικά δεν υπάρχει πάγιο. Να γραφεί αλγόριθμος ο οποίος υλοποιεί τα παραπάνω.

**14.** Το ύψος που θα έχει ένας άνθρωπος στην ηλικία των 18 ετών εξαρτάται από το ύψος των γονέων με απόκλιση  $\pm 10$  εκατοστά και μπορεί να υπολογιστεί με τους εξής μαθηματικούς υπολογισμούς:

Για τα αγόρια είναι το άθροισμα του ύψους του πατέρα (σε εκ.) συν το ύψος της μητέρας (σε εκ.) συν 17 εκ. και όλο διαιρούμενο δια του 2 δηλ.  $(Υ_{\text{πατ}} + Υ_{\text{μητ}} + 17) / 2$ .

Για τα κορίτσια είναι το άθροισμα του ύψους του πατέρα συν το ύψος της μητέρας μείον 12 εκ. και όλο διαιρούμενο δια του 2. Θα πρέπει να γράψετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

Α) Να διαβάζει το φύλο ενός ανηλίκου, το ύψος του πατέρα και το ύψος της μητέρας (Θεωρήστε ότι για το φύλο ο χρήστης δίνει τιμές: Α για τα αγόρια και Κ για τα κορίτσια)

Β) Να εμφανίζει το ύψος ανάλογα με το φύλο.

**15.** Υποθέστε ότι πρέπει να φτιάξετε ένα σύστημα πυρασφάλειας που να ελέγχει τη θερμοκρασία, την υγρασία και την περιεκτικότητα σε καπνό ενός χώρου. Αν η θερμοκρασία είναι κάτω από 30 C, η υγρασία πάνω από 25% και ο καπνός κάτω από 50 mg/m<sup>3</sup>, τότε οι συνθήκες είναι φυσιολογικές.

Για το σκοπό αυτό θα πρέπει να γράψετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο να διαβάζει τα παραπάνω τρία στοιχεία και να εμφανίζει το μήνυμα «Φυσιολογικές Συνθήκες» αν όλα είναι εντάξει, διαφορετικά το μήνυμα «Συναγερμός».

**16.** Για να έχει κάποιος δικαίωμα συμμετοχής στις εξετάσεις για τη Σχολή των Ευελπίδων θα πρέπει η ηλικία του να είναι από 17 έως και 22 ετών. Αν η ηλικία του υποψηφίου είναι έξω από τα όρια αυτά τότε δεν πληροί τις προϋποθέσεις. Για το σκοπό αυτό θα πρέπει να γράψετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο να εισάγει την ηλικία του υποψηφίου και να εμφανίζει μήνυμα «Δεν πληροί τις προϋποθέσεις» αν η ηλικία του είναι έξω από τα παραπάνω όρια.

**17.** Η φυσιολογική θερμοκρασία ενός παιδιού παρουσιάζει διακυμάνσεις ανάλογα με την ηλικία του, τη δραστηριότητα του και την ώρα της ημέρας. Τα βρέφη έχουν συνήθως υψηλότερη θερμοκρασία απ' ό,τι τα μεγαλύτερα παιδιά, ενώ η θερμοκρασία είναι γενικώς υψηλότερη τις απογευματινές ώρες και χαμηλότερη τις πρώτες πρωινές. Όταν θερμομετρήσουμε από το στόμα τις πρώτες πρωινές ώρες ένα παιδί ηλικίας κάτω των τριών ετών, τότε αν η θερμοκρασία του είναι από 37,4 C και πάνω, θεωρούμε ότι έχει πυρετό, διαφορετικά έχει φυσιολογική θερμοκρασία. Όταν η ηλικία του παιδιού είναι από τρία έτη και πάνω τότε το όριο αυτό αλλάζει και γίνεται 37,1 C. Για το παραπάνω πρόβλημα να αναπτύξετε αλγόριθμο, ο οποίος να εμφανίζει το μήνυμα "Βρέφος Εμπύρετο" ή "Βρέφος Απύρετο" ή "Παιδί Εμπύρετο" ή "Παιδί Απύρετο", ανάλογα με την περίπτωση. Θεωρήστε ότι ο χρήστης του προγράμματος θα εισαγάγει τιμές για τα δεδομένα που θα είναι πραγματικές και δεν θα χρειάζονται έλεγχο από το δημιουργό του προγράμματος.

**18.** Κάθε χρόνο το υπουργείο παιδείας προκηρύσσει πρωτάθλημα δρόμου για τα σχολεία της χώρας. Κάθε σχολείο κάνει προκριματικούς αγώνες, με σκοπό να στείλει τους μαθητές που τερμάτισαν πρώτοι. Επειδή η αντοχή κάθε μαθητή επηρεάζεται από το φύλο και την ηλικία, η απόσταση της διαδρομής είναι διαφορετική σε κάθε περίπτωση. Η απόσταση διαδρομής για τα κορίτσια που φοιτούν σε γυμνάσια είναι 3.000μ. και για τα αγόρια 4.000μ, ενώ για τα κορίτσια που φοιτούν σε λύκεια είναι 4.000μ και για τα αγόρια 5.000μ. Για το παραπάνω πρόβλημα να αναπτύξετε αλγόριθμο, ο οποίος:

1. Να εισάγει τον τύπο του σχολείου που φοιτά ο μαθητής (Γ για γυμνάσιο και Λ Για λύκειο)
2. Να εισάγει το φύλο του μαθητή (Κ για κορίτσι και Α για αγόρι)
3. Να εμφανίζει την απόσταση διαδρομής.

**19.** Μια εταιρεία κινητής τηλεφωνίας ακολουθεί ανά μήνα την πολιτική που φαίνεται στο παρακάτω πίνακα:

Χρόνος τηλεφωνημάτων (sec)	Χρονοχρέωση (ευρώ / sec)
1-600	0.01



601-900	0,001
901 και άνω	0,0001
	Διευκρίνιση: Πάγιο 9 ευρώ.

Η χρονοχρέωση θεωρείται κλιμακωτή. Δηλαδή τα πρώτα 600 δευτερόλεπτα χρεώνονται με 0,01 ευρώ/δευτερόλεπτο, τα επόμενα 300 δευτερόλεπτα με 0,001 ευρώ /δευτερόλεπτο και τα υπόλοιπα άνω των 900 με 0,0001 ευρώ/δευτ.

Να αναπτύξετε αλγόριθμο ο οποίος:

- α) να διαβάξει τη χρονική διάρκεια των τηλεφωνημάτων ενός συνδρομητή σε διάστημα ενός μήνα.
- β) να υπολογίζει τη μηνιαία χρέωση του συνδρομητή
- γ) να εμφανίζει τη λέξη "ΧΡΕΩΣΗ" και τη μηνιαία χρέωση του συνδρομητή.

**20.** Ο υπολογισμός των ταχυδρομικών τελών για αποστολή επιστολής πραγματοποιείται με τον εξής τρόπο:

Για γράμματα από 1 ως 20 γραμμάρια 2,5 ευρώ.

Για γράμματα από 20 ως 200 γραμμάρια 3 ευρώ.

Για γράμματα άνω των 200 γραμμαρίων χρεώνεται 0,05 ευρώ για κάθε γραμμάριο (π.χ. ένα γράμμα 250 γραμμαρίων χρεώνεται  $0,05 \times 250 = 12,5$  ευρώ).

Να γραφεί ένα πρόγραμμα για τον υπολογισμό και την εμφάνιση των ταχυδρομικών τελών με δεδομένο το βάρος ενός γράμματος.

**21.** Οι εκπτώσεις ενός καταστήματος προσφέρονται ανάλογα με το ποσό των αγορών που έχουν πραγματοποιηθεί. Έτσι, αν το ποσό των αγορών είναι μεγαλύτερο από 500€, τότε το ποσό της έκπτωσης είναι 80€ αυξημένο κατά 2% του συνολικού ποσού αγορών που έχει πραγματοποιήσει ο πελάτης. Διαφορετικά, το ποσό της έκπτωσης είναι 60€. Ωστόσο, αν το συνολικό ποσό αγορών είναι μικρότερο από 150€, η έκπτωση είναι 2% του συνολικού ποσού αγορών. Να αναπτυχθεί αλγόριθμος που θα διαβάξει το συνολικό ποσό αγορών ενός πελάτη και θα εμφανίζει το ποσό της έκπτωσης καθώς και το ποσό που καλείται τελικά να πληρώσει για τις αγορές του.

**22.** Η διαφορά ώρας ανάμεσα στην Ελλάδα και την Ινδία είναι 3 ώρες και 30 λεπτά. Αυτό σημαίνει πως όταν στην Ελλάδα η ώρα είναι 17.00 στην Ινδία είναι 20.30. Να γίνει αλγόριθμος που θα διαβάξει σε δύο μεταβλητές (μία για την ώρα και μία για τα λεπτά) την ώρα της Ελλάδας, σε 24ώρη μορφή και θα εμφανίζει την ώρα της Ινδίας, π.χ. Ώρα Ελλάδας: 23.45, Ώρα Ινδίας: 3.15.

**23.** Η Secure Airlines εκτελεί πτήσεις Κέρκυρα-Ρώμη. Το κόστος για ενήλικους είναι 100 ευρώ, για παιδιά έως τριών χρονών 25 ευρώ, ενώ για παιδιά έως 18 χρονών 35 ευρώ. Κάθε επιβάτης μπορεί να έχει αποσκευές μέχρι 25 κιλά. Κάθε επιπλέον κιλό χρεώνεται 1,5 ευρώ. Να γραφεί αλγόριθμος οποίος αφού διαβάσει τα κατάλληλα δεδομένα θα υπολογίσει-εμφανίσει το κόστος μιας οικογένειας για να μετακινηθεί από την Κέρκυρα στη Ρώμη.



# Ενότητα 3 Δομή Επανάληψης (κεφ. 2 και 8 σχολ. βιβλίου)





### 3.1 Θεωρία-Οδηγίες

Η διαδικασία της επανάληψης είναι ιδιαίτερα συχνή, αφού πλήθος προβλημάτων μπορούν να επιλυθούν με κατάλληλες επαναληπτικές διαδικασίες. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις, όπου μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων, που έχουν κάτι κοινό.

Η δομή **επανάληψης** συναντάται κυρίως σε προβλήματα όπου έχουμε να επεξεργαστούμε μια σειρά από παρόμοιες τιμές δεδομένων αρκετές φορές. Τα προβλήματα αυτά άλλοτε, έχουν άγνωστο πλήθος φορών της παραπάνω επεξεργασίας και άλλοτε γνωστό. Πρέπει να είμαστε σε θέση μέσα από τις προδιαγραφές του προβλήματος, να καταλάβουμε σε ποια περίπτωση είμαστε.

**Βρόχος loop:** Σύνολο εντολών που μπορεί να εκτελεστεί επανειλημμένα, όσο ισχύει μια ορισμένη συνθήκη. Το τμήμα του αλγόριθμου που επαναλαμβάνεται.

**Επανάληψη:** Η διαδικασία επαναληπτικής εκτέλεσης ενός συνόλου εντολών μέχρι την ικανοποίηση κάποιας συνθήκης.

**Λογική των εντολών επανάληψης (βρόχων):** εφαρμόζεται σε περιπτώσεις, οι οποίες έχουν κάτι κοινό, όπου μία ακολουθία εντολών πρέπει να εκτελεστεί περισσότερες από μία φορά. Οι εκτελέσεις ελέγχονται πάντοτε από κάποια λογική έκφραση η οποία καθορίζει την έξοδο από το **βρόχο**.

Η τιμή **φρουρός** χρησιμοποιείται για τη διακοπή εκτέλεσης ενός βρόχου.

Η εντολή **Για** είναι η πιο απλή εντολή επανάληψης. Χρησιμοποιεί ένα μετρητή για να μετράει πόσες επαναλήψεις γίνονται. Επιλέγουμε να τη χρησιμοποιήσουμε όταν γνωρίζουμε από πριν πόσες επαναλήψεις θέλουμε. Ο αριθμός των επαναλήψεων που πραγματοποιείται ισούται με Ακέραιο Μέρος του (τελική τιμή - αρχική τιμή)/βήμα+1.

Η εντολή **ΓΙΑ** χρησιμοποιείται μόνο για προκαθορισμένο αριθμό επαναλήψεων. Αν λουπόν ξέρεις τον αριθμό των επαναλήψεων ή μπορείς να τον υπολογίσεις, τότε να χρησιμοποιείς την εντολή **ΓΙΑ**. Ποτέ μη χρησιμοποιείς εντολές που αλλάζουν την αρχική τιμή, την τελική τιμή, το βήμα ή τη μεταβλητή που ελέγχει την επανάληψη μέσα σε ένα βρόχο **ΓΙΑ**. Διαφυγή (έξοδος από τον χρήστη) βρόχου δε γίνεται στη **ΓΙΑ**.

Η εντολή **Όσο** είναι η πιο **ισχυρή** εντολή επανάληψης. Χρησιμοποιείται συνήθως όταν δεν ξέρουμε τον αριθμό των επαναλήψεων, αλλά οι επαναλήψεις εξαρτώνται από κάποια συνθήκη. Την προτιμούμε από την **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** όταν είναι πιθανό να μη γίνει καμία επανάληψη, γιατί στην **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** γίνεται πάντα τουλάχιστον μία.

Αν θέλετε να βάλετε μετρητή στην **ΟΣΟ**, θυμηθείτε τα τρία βήματα του μετρητή:

- ✓ Αρχικοποίηση πριν την **ΟΣΟ** (π.χ.  $i \leftarrow 1$ )
- ✓ Μεταβολή πριν το **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** (π.χ.  $i \leftarrow i+1$ )
- ✓ Συνθήκη τέλους στην **ΟΣΟ** (π.χ. **ΟΣΟ**  $i \leq 20$ )

Η εντολή **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** χρησιμοποιείται όταν δεν υπάρχει συγκεκριμένος αριθμός επαναλήψεων, αλλά οι επαναλήψεις εξαρτώνται από κάποια συνθήκη. Την προτιμούμε από την **ΟΣΟ** όταν θέλουμε να γίνει τουλάχιστον μία επανάληψη. (Έλεγχος εισαγωγής ακεραίου κλπ).

**Αρχή\_ επανάληψης**

## Διάβασε χ

Μέχρις\_ότου  $\chi = A\_M(\chi)$

Οι μεταβλητές που ελέγχουν την επανάληψη του βρόχου **ΟΣΟ** και **ΜΕΧΡΙΣ\_ΟΤΟΥ** πρέπει υποχρεωτικά να αλλάζουν τιμή μέσα στο σώμα του βρόχου, αλλιώς ή δεν εκτελείται ποτέ ή συνηθέστερα δεν σταματάει η εκτέλεση του (**ατέρμων** βρόχος).

Οι επαναλήψεις που υλοποιούνται με την εντολή **ΟΣΟ**, μπορεί να μην εκτελεστούν ούτε μία φορά, αφού ο έλεγχος γίνεται στην είσοδο του βρόχου, αντίθετα οι επαναλήψεις **ΜΕΧΡΙΣ\_ΟΤΟΥ** θα πραγματοποιηθούν τουλάχιστον μία φορά.

ΑΝ γνωστός ο αριθμός των επαναλήψεων ΤΟΤΕ

ενδείκνυται η ΓΙΑ

ΑΛΛΙΩΣ\_ΑΝ η επανάληψη θα εκτελεστεί υποχρεωτικά μια φορά ΤΟΤΕ

ενδείκνυται η ΜΕΧΡΙΣ\_ΟΤΟΥ

ΑΛΛΙΩΣ

ενδείκνυται η ΟΣΟ

ΤΕΛΟΣ\_ΑΝ

**Φώλιασμα:** Η ένθεση βρόχου ή τμήματος κώδικα μέσα σε άλλους βρόχους ή μπλοκ κώδικα.

Στη χρήση των **εμφωλευμένων** βρόχων ισχύουν συγκεκριμένοι κανόνες που πρέπει να ακολουθούνται αυστηρά για την σωστή λειτουργία των προγραμμάτων. Συγκεκριμένα:

- ✓ Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό. Ο βρόχος που ξεκινάει τελευταίος, πρέπει να ολοκληρώνεται πρώτος.
- ✓ Η είσοδος σε κάθε βρόχο υποχρεωτικά γίνεται από την αρχή του.
- ✓ Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

## 3.2 Ερωτήσεις Σωστού-Λάθους

1. Ο ρόμβος έχει 2 εξόδους.
2. Η Μέχρι\_ότου προτιμάται στον έλεγχο εγκυρότητας.
3. Κάθε εντολή ΟΣΟ μπορεί να μετατραπεί σε μια ισοδύναμη εντολή ΓΙΑ.
4. Όταν το βήμα σε μια εντολή ΓΙΑ μειώνεται κατά ένα τότε αυτό μπορεί να παραληφθεί.
5. Αν η συνθήκη της ΟΣΟ είναι αληθής τότε ο βρόχος θα εκτελεστεί οπωσδήποτε πάνω από μια φορά.
6. Η είσοδος σε κάθε βρόχο γίνεται υποχρεωτικά από την αρχή του.
7. Η ολίσθηση ενός αριθμού προς τα αριστερά στο δυαδικό σύστημα, ισούται με τον υποδιπλασιασμό του.
8. Το βήμα σε μια εντολή επανάληψης Για πρέπει να είναι πάντα ακέραιος αριθμός.
9. Ο βρόχος Για  $x$  από 5 μέχρι 6 με\_βήμα -1 εκτελείται μία μόνο φορά.
10. Η μορφή επανάληψης Για μπορεί να χρησιμοποιηθεί και όταν δεν γνωρίζουμε στην αρχή τον αριθμό των επαναλήψεων.
11. Ο βρόχος Για  $i$  από 0 μέχρι 3 με\_βήμα 3 εκτελείται τρεις φορές.
12. Με χρήση της εντολής Όσο...επανάλαβε επιτυγχάνεται η επανάληψη μίας διαδικασίας με βάση κάποια συνθήκη.
13. Οι εντολές της επαναληπτικής Όσο θα εκτελεστούν τουλάχιστον μια φορά.
14. Όταν το πλήθος των επαναλήψεων είναι γνωστό, δεν μπορεί να χρησιμοποιηθεί η εντολή επανάληψης Όσο.
15. Η εντολή επανάληψης ΜΕΧΡΙΣ\_ΟΤΟΥ εκτελείται υποχρεωτικά τουλάχιστον μία φορά.
16. Σε ένα διάγραμμα ροής, ο ρόμβος δηλώνει την αρχή και το Τέλος του αλγόριθμου.
17. Ο βρόχος Για  $K$  από 5 μέχρι 5 δεν εκτελείται καμία φορά.
18. Ο βρόχος Για  $K$  από 5 μέχρι 1 εκτελείται 5 φορές.
19. Στην επαναληπτική δομή Για ... από ... μέχρι ...με\_βήμα οι τιμές από, μέχρι και με\_βήμα δεν είναι απαραίτητο να είναι ακέραιες.
20. Η εντολή επανάληψης ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ ...ΜΕ\_ΒΗΜΑ μπορεί να χρησιμοποιηθεί, όταν έχουμε άγνωστο αριθμό επαναλήψεων.
21. Για  $k$  από - 4 μέχρι - 3 εκτελείται ακριβώς δύο φορές.
22. Η εντολή ΓΙΑ μπορεί να χρησιμοποιηθεί Για έλεγχο εγκυρότητας τιμής.
23. Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.
24. Η ΓΛΩΣΣΑ υποστηρίζει τρεις εντολές επανάληψης, την εντολή ΟΣΟ, την εντολή ΜΕΧΡΙΣ\_ΟΤΟΥ και την εντολή ΓΙΑ.
25. Όταν ένας βρόχος είναι εμφωλευμένος σε άλλο, ο βρόχος που ξεκινάει τελευταίος πρέπει να ολοκληρώνεται πρώτος.
26. Όταν το πλήθος των επαναλήψεων είναι γνωστό, δεν μπορεί να χρησιμοποιηθεί η εντολή επανάληψης Όσο ...Επανάλαβε.
27. Μία εντολή ΓΙΑ μπορεί να μετατραπεί σε ΟΣΟ ακόμα κι Αν το βήμα της είναι αρνητικό.
28. Η επανάληψη που δεν τελειώνει ποτέ παραβιάζει το κριτήριο της περατότητας.
29. Όταν μια ομάδα εντολών δεν ολοκληρώνεται μετά από ένα πεπερασμένο αριθμό βημάτων, αποκαλείται υπολογιστική διαδικασία.
30. Η εντολή ΓΙΑ  $i$  ΑΠΟ 0 ΜΕΧΡΙ 20 θα εκτελεστεί 20 φορές.
31. Μια δομή επιλογής είναι δυνατό να περιέχει μια δομή επανάληψης.

### Απαντήσεις

1.Σ, 2.Σ, 3.Λ, 4.Λ, 5.Λ, 6.Σ, 7.Λ, 8.Λ, 9.Λ, 10.Λ, 11.Λ, 12.Σ, 13.Λ, 14.Λ, 15.Σ, 16.Λ, 17.Λ, 18.Λ, 19.Σ, 20.Λ, 21.Σ, 22.Λ, 23.Σ, 24.Σ, 25.Σ, 26.Λ, 27.Σ, 28.Σ, 29.Σ, 30.Λ, 31.Σ

32. Δίνεται το παρακάτω τμήμα αλγορίθμου με αριθμημένες τις εντολές του:

(1)  $\Sigma \leftarrow 0$

(2)  $K \leftarrow 0$

(3) **Αρχή\_Επανάληψης**

(4) **Διάβασε** X

(5)  $\Sigma \leftarrow \Sigma + X$

(6) **Αν**  $X > 0$  **τότε**

(7)  $K \leftarrow K + 1$

(8) **Τέλος\_Αν**

(9) **Μέχρις\_ότου**  $\Sigma > 1000$

(10) **Εμφάνισε** X

Να γράψετε τη λέξη ΣΩΣΤΟ, αν είναι σωστή η πρόταση, ή τη λέξη ΛΑΘΟΣ, αν είναι λανθασμένη.

1. Η εντολή (4) θα εκτελεστεί τουλάχιστον μία φορά.
2. Η εντολή (1) θα εκτελεστεί ακριβώς μία φορά.
3. Στη μεταβλητή K καταχωρείται το πλήθος των θετικών αριθμών που δόθηκαν.
4. Η εντολή (7) εκτελείται πάντα λιγότερες φορές από την εντολή (4).
5. Η τιμή που θα εμφανίσει η εντολή (10) μπορεί να είναι αρνητικός αριθμός.

**Απάντηση**

1.Σ, 2.Σ, 3.Σ, 4.Λ, 5.Λ

33. Δίνεται το παρακάτω τμήμα αλγορίθμου:

**Για** i **από** -3 **μέχρι** A **με\_βήμα** B

**Εμφάνισε** i

**Τέλος\_επανάληψης**

Να χαρακτηρίσετε καθεμία από τις προτάσεις που ακολουθούν και αναφέρονται στο παραπάνω τμήμα αλγορίθμου, αν αυτή είναι Σωστή, ή Λανθασμένη.

1. Αν το A είναι 0 και το B είναι 1 δεν ικανοποιείται το κριτήριο της περατότητας.
2. Αν το A είναι μεγαλύτερο του 0 και το B είναι μικρότερο του -4, ο βρόχος δεν εκτελείται καμία φορά.
3. Αν το A είναι 2 και το B είναι 2, ο βρόχος εκτελείται ακριβώς 3 φορές.
4. Αν το A και το B είναι θετικοί αριθμοί, ο βρόχος μπορεί να μετατραπεί με τη χρήση της εντολής Όσο...επανάλαβε.

**Απάντηση**

1.Λ, 2.Σ, 3.Σ, 4.Σ



### 3.3 Λυμένες Ασκήσεις

1. Να γραφεί αλγόριθμος που υλοποιεί τον πολλαπλασιασμό αλά ρωσικά.

**Λύση**

**Αλγόριθμος** Πολλαπλασιασμός\_αλά\_ρωσικά

**Δεδομένα** // M1,M2 ακέραιοι //

P ← 0

**Όσο** M2 > 0 **επανάλαβε**

**Αν** M2 mod 2 = 1 **τότε** P ← P+M1

    M1 ← M1\*2

    M2 ← M2 div 2

**Τέλος\_επανάληψης**

**Αποτελέσματα** // P, το γινόμενο των ακεραίων M1,M2 //

**Τέλος** Πολλαπλασιασμός\_αλά\_ρωσικά

2. Να γραφεί αλγόριθμος που υλοποιεί τον υπολογισμό του παραγοντικού.

**Λύση**

**Αλγόριθμος** Παραγοντικό

**Δεδομένα** // n //

product ← 1

**Για** i **από** 2 **μέχρι** n                   ! Εντολή επανάληψης **Για**

    product ← product \* i

**Τέλος\_επανάληψης**

**Αποτελέσματα** // product //

**Τέλος** Παραγοντικό

3. Να γραφεί αλγόριθμος που υλοποιεί τον υπολογισμό του μέγιστου κοινού διαιρέτη.

**Λύση**

Ουσιαστικά λαμβάνει το μικρότερο από τους δύο ακεραίους, τον z, και εξετάζει με τη σειρά όλους τους ακεραίους ξεκινώντας από τον z και μειώνοντας συνεχώς κατά μία μονάδα μέχρι και οι δύο αριθμοί, x και y, να διαιρούνται από τη νέα τιμή του z.

**Αλγόριθμος** Μέγιστος\_Κοινός\_Διαιρέτης

**Δεδομένα** // x,y //

**Αν** x < y **τότε**

    z ← x

    αλλιώς

    z ← y

**Τέλος\_Αν**

**Όσο** (x mod z ≠ 0) ή (y mod z ≠ 0) **επανάλαβε**

        z ← z-1

**Τέλος\_επανάληψης**

**Αποτελέσματα** // z //

**Τέλος** Μέγιστος\_Κοινός\_Διαιρέτης

## 2η Λύση

Αλγόριθμος Ευκλείδης

Δεδομένα // x,y //

$z \leftarrow y$

Όσο  $z \neq 0$  επανάλαβε

$z \leftarrow x \bmod y$

$x \leftarrow y$

$y \leftarrow z$

Τέλος\_επανάληψης

Αποτελέσματα // x //

Τέλος Ευκλείδης

4. Να γραφεί αλγόριθμος που υλοποιεί τον υπολογισμό της ακολουθίας Fibonacci.

## Λύση

Οι πρώτοι όροι της ακολουθίας Fibonacci πρώτης τάξης είναι:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 κλπ.

Ο επόμενος αλγόριθμος υπολογίζει τον αριθμό Fibonacci πρώτης τάξης  $F_n$ , με επαναληπτική μέθοδο. Υποτίθεται ότι κατά την κλήση του αλγορίθμου μία μη αρνητική τιμή περνά ως όρισμα στη μεταβλητή  $n$ .

Αλγόριθμος Fibonacci

Δεδομένα // n //

Αν  $n \leq 1$  τότε  $Fib \leftarrow n$

$f0 \leftarrow 0$

$f1 \leftarrow 1$

Για  $i$  από 2 μέχρι  $n$

$fib \leftarrow f0 + f1$

$f0 \leftarrow f1$

$f1 \leftarrow fib$

Τέλος\_επανάληψης

Αποτελέσματα // Fib //

Τέλος Fibonacci

5. Να γραφεί αλγόριθμος που υλοποιεί τον υπολογισμό της δύναμης χωρίς χρήση του τελεστή  $^$ .

## Λύση

Αλγόριθμος Δύναμη

Δεδομένα // a, b //

power  $\leftarrow$  1

Για  $i$  από 1 μέχρι  $b$

power  $\leftarrow$  power \* a

Τέλος\_επανάληψης

Αποτελέσματα // power //

Τέλος Δύναμη

6. Ποια είναι η τιμή φρουρός σε κάθε ένα από τα παρακάτω τμήματα κώδικα;

Διάβασε  $y$   $!(1)$

Όσο  $Y > 3$  επανάλαβε

Εμφάνισε  $y$

Διάβασε  $y$

Τέλος\_επανάληψης

Απάντηση

Η τιμή φρουρός είναι το 3 και οποιαδήποτε μικρότερη.

Για  $I$  από 2 μέχρι 8 με\_βήμα 2  $!(2)$

Εμφάνισε  $I$

Τέλος\_επανάληψης

Απάντηση

Η τιμή φρουρός είναι το 10.

7. Να εκτελέσετε το παρακάτω τμήμα αλγορίθμου, για  $K = 24$  και  $L = 40$ . Να γράψετε στο τετράδιό σας τις τιμές των μεταβλητών  $X$ ,  $Y$  καθώς αυτές τυπώνονται με την εντολή Εμφάνισε  $X$ ,  $Y$  (τόσο μέσα στη δομή επανάληψης όσο και στο τέλος του αλγορίθμου). (Θέμα Πανελλαδικών)

$X \leftarrow K$

$Y \leftarrow L$

Αν  $X < Y$  τότε

$TEMP \leftarrow X$

$X \leftarrow Y$

$Y \leftarrow TEMP$

Τέλος\_Αν

Όσο  $Y <> 0$  επανάλαβε

$TEMP \leftarrow Y$

$Y \leftarrow X \text{ MOD } Y$

$X \leftarrow TEMP$

Εμφάνισε  $X$ ,  $Y$

Τέλος\_επανάληψης

$Y \leftarrow (K * L) \text{ DIV } X$

Εμφάνισε  $X$ ,  $Y$

Λύση

Αρχικές τιμές:

$K = 24$ ,  $L = 40$ ,  $X = 24$ ,  $Y = 40$

	X	Y
1η Επανάληψη	24	16
2η Επανάληψη	16	8
3η Επανάληψη	8	0

Τελική	8	120
--------	---	-----

8. Δίνεται το παρακάτω τμήμα αλγορίθμου γραμμένο σε «ψευδογλώσσα»:

$K \leftarrow 2$

**Αρχή\_επανάληψης**

$K \leftarrow K+4$

**Μέχρις\_ότου**  $K > 20$

$\Lambda \leftarrow T\_P(K-23)$

Το οποίο δεν ικανοποιεί ένα αλγοριθμικό κριτήριο. Εντοπίστε και αιτιολογήστε.

**Λύση**

Δεν ικανοποιεί το κριτήριο της καθοριστικότητας, σύμφωνα με το οποίο κάθε εντολή ενός αλγορίθμου θα πρέπει να είναι σαφώς ορισμένη σχετικά με τον τρόπο της εκτέλεσης της. Συγκεκριμένα, όταν ολοκληρώνεται η δομή επανάληψης η μεταβλητή  $K$  έχει τιμή ίση με 22. Επομένως, εφόσον η διαφορά  $K-23$  είναι αρνητική, είναι αδύνατη η εκτέλεση της εντολής εκχώρησης  $\Lambda \leftarrow T\_P(K-23)$ , διότι δεν ορίζεται η τετραγωνική ρίζα.

9. Απλοποιήστε το παρακάτω σύνολο εντολών (θεωρήστε  $k$  ακέραιος)

**Αρχή\_επανάληψης**

Διάβασε  $k$

**Μέχρις\_ότου** ( $k \geq 0$ ) **ΚΑΙ** ( $k < 100$ ) **ΚΑΙ** ( $k < > 0$ )

**Αν** ( $k > 98$ ) **Ή** ( $k > 99$ ) **τότε**

Εμφάνισε 'α'

**αλλιώς\_Αν**  $k > 59$  **τότε**

Εμφάνισε 'β'

**αλλιώς\_Αν**  $k > 0$  **τότε**

Εμφάνισε 'γ'

**αλλιώς**

Εμφάνισε 'δ'

**Τέλος\_Αν**

**Λύση**

**Αρχή\_επανάληψης**

Διάβασε  $k$

**Μέχρις\_ότου** ( $k > 0$ ) **ΚΑΙ** ( $k < 100$ )

**Αν**  $k = 99$  **τότε**

Εμφάνισε 'α'

**αλλιώς\_Αν**  $k > 59$  **τότε**

Εμφάνισε 'β'

**αλλιώς**

Εμφάνισε 'γ'

**Τέλος\_Αν**

10. Να γραφεί αλγόριθμος ο οποίος να διαβάζει 25 ακέραιους αριθμούς μεγαλύτερους του μηδενός και να ελέγχει την αξιόπιστη καταχώρησή τους. Στην περίπτωση όπου

κάποιος από αυτούς είναι πολλαπλάσιος του 8 και μεγαλύτερος του 30, θα πρέπει να εκχωρείται σε μια μεταβλητή Δ το ηλίκο της διαίρεσης του αριθμού με το 12 και στη συνέχεια να εμφανίζεται το μήνυμα 'Η τιμή της Δ είναι:', \_\_ όπου στο κενό να εμφανίζεται η τιμή της Δ.

Στην περίπτωση όπου κάποιος από αυτούς είναι μεγαλύτερος του 60, θα πρέπει να εκχωρείται σε μια μεταβλητή Ρ η τετραγωνική ρίζα του αριθμού και στη συνέχεια να εμφανίζεται το μήνυμα: 'Η τετραγωνική ρίζα του αριθμού:', \_\_\_\_, 'είναι:', \_\_\_\_, όπου στο πρώτο κενό να εμφανίζεται ο αριθμός και στο δεύτερο κενό η τετραγωνική ρίζα του αριθμού.

Σε οποιαδήποτε άλλη περίπτωση, θα πρέπει να εκχωρείται σε μια μεταβλητή Υ η διπλάσια τιμή του τετραγώνου του αριθμού. Αν αυτή είναι μεγαλύτερη του 125 να εμφανίζεται το μήνυμα 'Το διπλάσιο τετράγωνο του αριθμού είναι:', \_\_\_\_, όπου στο κενό να εμφανίζεται η τιμή Υ, διαφορετικά να εκχωρείται στην μεταβλητή Υ η τιμή 9.

### **Λύση**

**ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** I, X, Δ, Y

**ΠΡΑΓΜΑΤΙΚΕΣ:** P

**ΑΡΧΗ**

**Για I ΑΠΟ 1 ΜΕΧΡΙ 25**

**ΓΡΑΨΕ** 'Δώστε τον', I, 'ο ακέραιο αριθμό μεγαλύτερο του μηδενός:'

**ΔΙΑΒΑΣΕ X**

**Όσο X <= 0 ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ** 'Δώσατε λάθος τιμή. Προσπαθήστε ξανά'

**ΓΡΑΨΕ** 'Δώστε τον', I, 'ο ακέραιο αριθμό μεγαλύτερο του μηδενός:'

**ΔΙΑΒΑΣΕ X**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Αν (X MOD 8 = 0) ΚΑΙ (X > 30) ΤΟΤΕ**

$\Delta \leftarrow X \text{ DIV } 12$

**ΓΡΑΨΕ** 'Η τιμή της Δ είναι:', Δ

**ΑΛΛΙΩΣ\_ΑΝ X>60 ΤΟΤΕ**

$P \leftarrow \sqrt{X}$

**ΓΡΑΨΕ** 'Η τετραγωνική ρίζα του αριθμού:', X, 'είναι:', P

**ΑΛΛΙΩΣ**

$Y \leftarrow 2 * (X * X)$

**ΑΝ Y > 125 ΤΟΤΕ**

**ΓΡΑΨΕ** 'Το διπλάσιο τετράγωνο του αριθμού είναι:', Y

**ΑΛΛΙΩΣ**

$Y \leftarrow 9$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**11.** Να γραφεί πρόγραμμα το οποίο διαβάζει τυχαίο ακέραιο και υπολογίζει, εμφανίζει το πλήθος των ψηφίων του και τη συχνότητα εμφάνισης του κάθε ψηφίου.

### **Λύση**

**ΠΡΟΓΡΑΜΜΑ** Ασκ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** N, I, Ψ, ΜΕΤΡ, ΠΛ[10]

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** N

**ΓΙΑ** I **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

ΠΛ[I] ← 0

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΜΕΤΡ ← 0

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

Ψ ← N MOD 10

N ← N DIV 10

ΜΕΤΡ ← ΜΕΤΡ + 1

ΠΛ[Ψ + 1] ← ΠΛ[Ψ + 1] + 1

**ΜΕΧΡΙΣ\_ΟΤΟΥ** N = 0

**ΓΡΑΨΕ** ΜΕΤΡ

**ΓΙΑ** I **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

**ΓΡΑΨΕ** ΠΛ[I]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**12.** Στις εξετάσεις του ΑΣΕΠ οι υποψήφιοι εξετάζονται σε τρεις θεματικές ενότητες. Ο βαθμός κάθε θεματικής ενότητας είναι από 1 έως 100. Η συνολική βαθμολογία κάθε υποψηφίου προκύπτει από τον μέσο όρο των βαθμών του στις τρεις θεματικές ενότητες. Ο υποψήφιος θεωρείται ως επιτυχών, αν η συνολική βαθμολογία του είναι τουλάχιστον 55 και ο βαθμός του σε κάθε θεματική ενότητα είναι τουλάχιστον 50. Να γράψετε αλγόριθμο ο οποίος για κάθε υποψήφιο:

1. Να διαβάζει το όνομά του και τους βαθμούς του σε καθεμία από τις τρεις θεματικές ενότητες. (Δεν απαιτείται έλεγχος εγκυρότητας δεδομένων).
2. Να εμφανίζει τον μεγαλύτερο από τους βαθμούς που πήρε στις τρεις θεματικές ενότητες.
3. Να εμφανίζει το όνομα και τη συνολική βαθμολογία του στην περίπτωση που είναι επιτυχών.
4. Ο αλγόριθμος να τερματίζει όταν δοθεί ως όνομα η λέξη "ΤΕΛΟΣ".
5. Στο τέλος να εμφανίζει το όνομα του επιτυχόντα με τη μικρότερη συνολική βαθμολογία. (Θέμα Πανελλαδικών)

**Λύση**

**Αλγόριθμος** ΘέμαΓ

min ← 101

**Διάβασε** Όνομα

**Όσο** (Όνομα <> "ΤΕΛΟΣ") **επανάλαβε**

**Διάβασε** α, β, γ

max ← α

**Αν** β > max **τότε** max ← β

!Γ5 ερώτημα

!Γ1 ερώτημα

!Γ4 ερώτημα

!Γ1 ερώτημα

```

Αν  $\gamma > \max$  τότε  $\max \leftarrow \gamma$ 
Εμφάνισε  $\max$  !Γ2 ερώτημα
 $MO \leftarrow (\alpha + \beta + \gamma)/3$  !Γ3 ερώτημα
Αν  $(MO \geq 55)$  και  $(\alpha \geq 50)$  και  $(\beta \geq 50)$  και  $(\gamma \geq 50)$  τότε
    Εμφάνισε Όνομα,  $MO$  !Γ3 ερώτημα
    Αν  $MO < \min$  τότε !Γ5 ερώτημα
         $\min \leftarrow MO$ 
         $\min \text{Όνομα} \leftarrow \text{Όνομα}$ 
    Τέλος_Αν
Τέλος_Αν
Διάβασε Όνομα !Γ1 ερώτημα
Τέλος_επανάληψης
Εμφάνισε  $\min \text{Όνομα}$  !Γ5 ερώτημα
Τέλος ΘέμαΓ

```

**13.** Ένας καθηγητής πληροφορικής, που διδάσκει το μάθημα ΑΕΠΠ, αποφάσισε να αναθέσει μια εργασία στους μαθητές του. Τους ζήτησε να κατασκευάσουν ένα πρόγραμμα το οποίο δέχεται άγνωστο πλήθος ακεραίων διψήφιων αριθμών (να γίνεται έλεγχος έγκυρης εισαγωγής μόνο διψήφιων ακεραίων). Στη συνέχεια να υπολογίζουν και να εμφανίζουν το γινόμενο των ψηφίων τους μόνο στην περίπτωση που το δεύτερο ψηφίο ήταν μεγαλύτερο ή και ίσο από το πρώτο. Για να τους κάνει τη ζωή ακόμα πιο δύσκολη, τους υποχρέωσε ο αλγόριθμος τους να υπολογίζει αυτό το γινόμενο με την μέθοδο του «Πολλαπλασιασμού αλά Ρωσικά». Επειδή δεν τους καθόρισε το πλήθος των ακεραίων που θα επεξεργαστούν, τους επισήμανε ότι μετά από την επεξεργασία κάθε ακεραίου ο αλγόριθμος τους θα έθετε την εξής ερώτηση «Θέλετε να συνεχίσετε; ΝΑΙ ή ΟΧΙ;». Με την εισαγωγή της απάντησης «ΟΧΙ» ο αλγόριθμος σταματά να δέχεται άλλους ακεραίους και εμφανίζει με τη χρήση κατάλληλα διαμορφωμένων μηνυμάτων, το πλήθος των διψήφιων ακεραίων που δόθηκαν, το πλήθος των διψήφιων ακεραίων που χρειάστηκε να υπολογιστεί το γινόμενο τους, καθώς και ποιο ήταν το μεγαλύτερο γινόμενο από αυτά που χρειάστηκε να υπολογιστούν.

### Λύση

**ΠΡΟΓΡΑΜΜΑ ΕΡΓΑΣΙΑ**

ΜΕΤΑΒΛΗΤΕΣ

**ΑΚΕΡΑΙΕΣ:** ΑΡ, Ψ1, Ψ2, ΓΙΝ, ΠΛ1, ΠΛ2, ΜΑΧ

**ΧΑΡΑΚΤΗΡΕΣ:** ΑΡ

ΑΡΧΗ

ΠΛ1←0

ΠΛ2←0

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** ' Δώστε έναν διψήφιο ακέραιο'

**ΔΙΑΒΑΣΕ** ΑΡ

**ΑΝ** (ΑΡ<10) 'Η (ΑΡ >99) **ΤΟΤΕ**

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

**ΓΡΑΨΕ** 'Μη αποδεκτή τιμή. Δώστε πάλι διψήφιο ακέραιο'

**ΔΙΑΒΑΣΕ** ΑΡ

**ΜΕΧΡΙΣ\_ΟΤΟΥ** (ΑΡ>=10) ΚΑΙ (ΑΡ<=99)

**ΤΕΛΟΣ\_ΑΝ**

```

Ψ1←AP DIV 10
Ψ2←AP MOD 10
ΑΝ Ψ2 >= Ψ1 ΤΟΤΕ
ΓΙΝ←0
    ΟΣΟ Ψ2 >0 ΕΠΑΝΑΛΑΒΕ
    ΑΝ Ψ2 MOD 2 <> 0 ΤΟΤΕ
    ΓΙΝ←ΓΙΝ + Ψ1
    ΤΕΛΟΣ_ΑΝ
    Ψ1←2 * Ψ1
    Ψ2← Ψ2 DIV 2
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ' Το γινόμενο των ψηφίων του', AP, ' είναι:', ΓΙΝ
ΠΛ2←ΠΛ2 +1
    ΑΝ ΠΛ2 = 1 ΤΟΤΕ
    ΜΑΧ←ΓΙΝ
    ΑΛΛΙΩΣ
        ΑΝ ΓΙΝ > ΜΑΧ ΤΟΤΕ
        ΜΑΧ← ΓΙΝ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΠΛ1← ΠΛ1 + 1
ΓΡΑΨΕ ' Θέλεις να δώσεις και άλλον διψήφιο ακέραιο; ΝΑΙ ή ΟΧΙ;'
ΔΙΑΒΑΣΕ ΑΠ
ΜΕΧΡΙΣ_ΟΤΟΥ ΑΠ = 'ΟΧΙ'
ΓΡΑΨΕ ' Δόθηκαν:', ΠΛ1, ' διψήφιοι ακέραιοι'
ΑΝ ΠΛ2 <> 0 ΤΟΤΕ
ΓΡΑΨΕ ' Χρειάστηκε να υπολογιστεί το γινόμενο σε:', ΠΛ2,' διψήφιους ακεραίους'
ΓΡΑΨΕ ' Το μεγαλύτερο γινόμενο που υπολογίστηκε ήταν:', ΜΑΧ
ΑΛΛΙΩΣ
ΓΡΑΨΕ ' Δεν δόθηκαν διψήφιοι που χρειαζόταν να υπολογιστεί το γινόμενο τους'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

14. Ποιο ή ποια αλγοριθμικά κριτήρια παραβιάζονται από το παρακάτω σύνολο εντολών;

```

χ ← 1
Όσο χ < 19 επανάλαβε
    Εμφάνισε χ
    χ ← χ+2
Τέλος επανάληψης
κ ← T_P(χ-22)
Εμφάνισε κ

```

**Λύση**

Παραβιάζεται το κριτήριο της καθοριστικότητας. Το χ παίρνει διαδοχικά τις τιμές 1, 3, 5, ..., 19, 21. Έτσι ο υπολογισμός του κ είναι αδύνατος, αφού  $\chi - 22 = 21 - 22 = -1 < 0$  και η τετραγωνική ρίζα αυτής της παράστασης δεν μπορεί να υπολογιστεί.



**15.** Δημόσιος οργανισμός διαθέτει ένα συγκεκριμένο ποσό για την επιδότηση επενδυτικών έργων. Η επιδότηση γίνεται κατόπιν αξιολόγησης και αφορά δύο συγκεκριμένες κατηγορίες έργων με βάση τον προϋπολογισμό τους. Οι κατηγορίες και τα αντίστοιχα ποσοστά επιδότησης επί του προϋπολογισμού είναι:

Κατηγορία έργου	Προϋπολογισμός έργου σε ευρώ	Ποσοστό επιδότησης
Μικρή	200.000 – 299.999	60%
Μεγάλη	300.000 – 399.999	70%

Μετά από κάθε εκταμίευση μειώνεται το ποσό που διαθέτει ο οργανισμός. Να αναπτύξετε αλγόριθμο ο οποίος:

1. Να διαβάσει το ποσό που διαθέτει ο οργανισμός για το πρόγραμμα επενδύσεων συνολικά, ελέγχοντας ότι το ποσό είναι μεγαλύτερο από 5.000.000 Ευρώ.
2. Να διαβάσει το όνομα κάθε έργου. Η σειρά ανάγνωσης είναι η σειρά υποβολής των έργων. Η επαναληπτική διαδικασία να τερματίζεται, όταν αντί για όνομα έργου δοθεί η λέξη «ΤΕΛΟΣ», ή όταν το διαθέσιμο ποσό έχει μειωθεί τόσο, ώστε να μην είναι δυνατή η επιδότηση ούτε ενός έργου μικρής κατηγορίας. Για κάθε έργο, αφού διαβάσει το όνομά του, να διαβάσει και τον προϋπολογισμό του (Δεν απαιτείται έλεγχος εγκυρότητας του προϋπολογισμού).
3. Για κάθε έργο να ελέγχει αν το διαθέσιμο ποσό καλύπτει την επιδότηση, και μόνον τότε να γίνεται η εκταμίευση του ποσού. Στη συνέχεια, να εμφανίζει το όνομα του έργου και το ποσό της επιδότησης που δόθηκε.
4. Να εμφανίζει το πλήθος των έργων που επιδοτήθηκαν από κάθε κατηγορία καθώς και τη συνολική επιδότηση που δόθηκε σε κάθε κατηγορία.
5. Μετά το τέλος της επαναληπτικής διαδικασίας να εμφανίζει το ποσό που δεν έχει διατεθεί, μόνο αν είναι μεγαλύτερο του μηδενός. (Θέμα Πανελλαδικών)

### **Λύση**

#### **Αλγόριθμος Οργανισμός**

*! Ερώτημα Γ1 - Έλεγχος Εγκυρότητας*

#### **Αρχή\_επανάληψης**

**Διάβασε** Ποσό

**Μέχρις\_ότου** Ποσό > 5000000

*! Ερώτημα Γ2*

i1 ← 0

i2 ← 0

sum1 ← 0

sum2 ← 0

**Διάβασε** Όνομα

**Όσο** ((Όνομα ≠ "ΤΕΛΟΣ") και (Ποσό ≥ 200000\*60/100)) **επανάλαβε**

**Διάβασε** Προϋπολογισμός *! Δεν απαιτείται έλεγχος εγκυρότητας*

*! Ερώτημα Γ3*

**Αν** Προϋπολογισμός ≤ 299999 **τότε**

Επιδότηση ← Προϋπολογισμός\*60/100

**Αν** Ποσό ≥ Επιδότηση **τότε**

i1 ← i1 + 1

sum1 ← sum1 + Επιδότηση

Ποσό ← Ποσό - Επιδότηση  
**Εμφάνισε** Όνομα, Επιδότηση  
**Τέλος\_Αν**  
**αλλιώς**  
 Επιδότηση ← Προϋπολογισμός\*70/100  
**Αν** Ποσό ≥ Επιδότηση **τότε**  
 $i2 \leftarrow i2 + 1$   
 sum2 ← sum2 + Επιδότηση  
 Ποσό ← Ποσό - Επιδότηση  
**Εμφάνισε** Όνομα, Επιδότηση  
**Τέλος\_Αν**  
**Τέλος\_Αν**  
**Αν** Ποσό ≥ 200000\*60/100 **τότε**  
**Διάβασε** Όνομα  
**Τέλος\_Αν**  
**Τέλος\_επανάληψης**  
 ! Ερώτημα Γ4  
**Εμφάνισε** i1, i2, sum1, sum2  
 ! Ερώτημα Γ5  
**Αν** Ποσό > 0 **τότε**  
**Εμφάνισε** Ποσό  
**Τέλος\_Αν**  
**Τέλος** Οργανισμός

**16.** Σε ένα εργαστήριο αξιολόγησης υλικού προσωπικών υπολογιστών πραγματοποιούνται δοκιμές σε ένα μοντέλο εκτυπωτή. Στις δοκιμές χρησιμοποιούνται 70 έγγραφα. Η ονομαστική ταχύτητα εκτύπωσης του συγκεκριμένου μοντέλου είναι ίση με 11 σελίδες/λεπτό. Όμως στην περίπτωση που ο αριθμός των σελίδων εκτύπωσης του εγγράφου υπερβαίνει τις 90, τότε αυτή μειώνεται κατά 3 σελίδες/λεπτό. Για παράδειγμα, αν οι σελίδες εκτύπωσης ενός εγγράφου είναι 90, τότε οι 70 πρώτες θα εκτυπωθούν με ταχύτητα 11 σελίδων/λεπτό και οι υπόλοιπες 20 σελίδες θα εκτυπωθούν με ταχύτητα 8 σελίδων/λεπτό. Στις ανωτέρω δοκιμές, κάθε έγγραφο δεν πρέπει να περιέχει περισσότερες από 2000 σελίδες αλλά ούτε και λιγότερες από 20.

Να αναπτυχθεί αλγόριθμος ο οποίος να:

Διαβάζει το πλήθος των σελίδων κάθε εγγράφου καθώς και το πλήθος των σελίδων εκτύπωσης κάθε εγγράφου και στη συνέχεια να ελέγχει την αξιόπιστη καταχώρηση για κάθε ένα από τα πλήθη.

Υπολογίζει και εμφανίζει τον χρόνο εκτύπωσης κάθε εγγράφου.

Βρίσκει και εμφανίζει τον μέσο χρόνο εκτύπωσης των εγγράφων.

Βρίσκει και εμφανίζει πόσα έγγραφα είχαν περισσότερες από 110 σελίδες.

Βρίσκει και εμφανίζει σε πόσα έγγραφα εκτυπώθηκαν περισσότερες από 400 σελίδες.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ** ΘΕΜΑ 3

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** I, ΣΕΓ, ΣΕΕ, Π1, Π2

**ΠΡΑΓΜΑΤΙΚΕΣ:** ΧΕ, ΣΧ, ΜΧ

**ΑΡΧΗ**

Π1 ← 0

! Αρχικοποιήσεις

Π2 ← 0

ΣΧ ← 0

Για Ι ΑΠΟ 1 ΜΕΧΡΙ 55

**ΓΡΑΨΕ** 'Δώστε τις σελίδες του', Ι, 'ου εγγράφου'

**ΓΡΑΨΕ** 'Όχι κάτω των 15, ούτε άνω των 1000:'

**ΔΙΑΒΑΣΕ** ΣΕΓ

**Όσο** (ΣΕΓ < 15) **Η'** (ΣΕΓ > 1000) **ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ** 'Δώσατε λάθος τιμή. Προσπαθήστε ξανά'

**ΓΡΑΨΕ** 'Δώστε τις σελίδες του', Ι, 'ου εγγράφου'

**ΓΡΑΨΕ** 'Όχι κάτω των 15, ούτε άνω των 1000:'

**ΔΙΑΒΑΣΕ** ΣΕΓ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Δώστε τις σελίδες εκτύπωσης του', Ι, 'ου εγγράφου'

**ΓΡΑΨΕ** 'Όχι κάτω των', ΣΕΓ, 'ούτε άνω των 1000:'

**ΔΙΑΒΑΣΕ** ΣΕΕ

**Όσο** (ΣΕΕ > ΣΕΓ) **Η'** (ΣΕΕ > 1000) **ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ** 'Δώσατε λάθος τιμή. Προσπαθήστε ξανά'

**ΓΡΑΨΕ** 'Δώστε τις σελίδες εκτύπωσης του', Ι, 'ου εγγράφου'

**ΓΡΑΨΕ** 'Όχι κάτω των', ΣΕΓ, 'ούτε άνω των 1000:'

**ΔΙΑΒΑΣΕ** ΣΕΕ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Αν** ΣΕΕ < = 80 **ΤΟΤΕ**

ΧΕ ← ΣΕΕ / 12

**ΑΛΛΙΩΣ**

ΧΕ ← 80/12 + (ΣΕΕ - 80) / 10

**ΤΕΛΟΣ\_ΑΝ**

**ΓΡΑΨΕ** 'Ο χρόνος εκτύπωσης του', Ι, 'ου εγγράφου είναι:', ΧΕ, 'λεπτά'

ΣΧ ← ΣΧ + ΧΕ !Υπολογισμός συνολικού χρόνου εκτύπωσης

**Αν** ΣΕΓ > 105 **ΤΟΤΕ**

Π1 ← Π1 + 1

**ΤΕΛΟΣ\_ΑΝ**

**Αν** ΣΕΕ > 500 **ΤΟΤΕ**

Π2 ← Π2 + 1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΜΧ ← ΣΧ / 55

**ΓΡΑΨΕ** 'Ο μέσος χρόνος εκτύπωσης είναι:', ΜΧ, 'λεπτά'

**ΓΡΑΨΕ** 'Βρέθηκαν', Π1, 'έγγραφα με περισσότερες από 105 σελίδες'

**ΓΡΑΨΕ** 'Βρέθηκαν', Π2, 'έγγραφα άνω των 500 σελίδων εκτύπωσης'

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**17.** Να σχεδιαστεί αλγόριθμος ο οποίος να διαβάζει έναν ακέραιο αριθμό και να υπολογίζει και εμφανίζει τον κατοπτρικό του. Θεωρείστε ότι ο αριθμός μπορεί να έχει απεριόριστο αριθμό ψηφίων. Παράδειγμα ο κατοπτρικός του 1279 είναι ο 9721.

**Λύση**

**Αλγόριθμος** Κατοπτρικός

**Διάβασε**  $\chi$   
 νέος  $\leftarrow 0$   
 digit  $\leftarrow 0$   
**Όσο**  $\chi > 0$  επανάλαβε  
 digit  $\leftarrow \chi \bmod 10$   
 νέος  $\leftarrow$  νέος  $\cdot 10 +$  digit  
 $\chi \leftarrow \chi \operatorname{div} 10$   
**Τέλος\_Επανάληψης**  
 Γράψε νέος  
**Τέλος** Κατοπτρικός

**18.** Σε ένα βιβλιοπωλείο υπάρχουν 8 διαφορετικές συσκευασίες με στυλό μιας συγκεκριμένης μάρκας. Γράψτε αλγόριθμο που διαβάζει την τιμή πώλησης και το πλήθος των τεμαχίων κάθε συσκευασίας και υπολογίζει τον αύξοντα αριθμό (1-8) της πιο συμφέρουσας.

**Λύση**

**Αλγόριθμος** ΑΣΚ  
**Διάβασε** Τιμή  
**Διάβασε** Τεμάχια  
 MIN  $\leftarrow$  Τιμή/Τεμάχια  
 ΑΑΜΙΝ  $\leftarrow 1$   
**Για**  $\kappa$  **από** 2 **μέχρι** 8  
     **Διάβασε** Τιμή  
     **Διάβασε** Τεμάχια  
     Τεμ  $\leftarrow$  Τιμή/Τεμάχια  
     **Αν** Τεμ < MIN **τότε**  
         MIN  $\leftarrow$  Τεμ  
         ΑΑΜΙΝ  $\leftarrow \kappa$   
     **Τέλος\_Αν**  
**Τέλος\_επανάληψης**  
**Αποτελέσματα** // ΑΑΜΙΝ //  
**Τέλος** ΑΣΚ

**19.** Θέλετε να γεμίσετε το ντεπόζιτο του ΙΧ σας, συνολικού όγκου 60 λίτρων, με βενζίνη. Κάνετε λοιπόν μια έρευνα αγοράς προκειμένου να εντοπίσετε το πρατήριο καυσίμων που πουλάει το πετρέλαιο στην πιο συμφέρουσα για εσάς τιμή. Γράψτε πρόγραμμα το οποίο:

- διαβάζει την τιμή πώλησης ανά λίτρο βενζίνης σε διάφορα πρατήρια καυσίμων (η είσοδος δεδομένων ολοκληρώνεται μόλις δοθεί μια μη θετική τιμή),
- υπολογίζει και εμφανίζει: τον αύξοντα αριθμό του πρατηρίου με την πιο συμφέρουσα τιμή πώλησης και το συνολικό κόστος αγοράς

**Λύση**

**ΠΡΟΓΡΑΜΜΑ** ΠΕΤΡΕΛΑΙΟ  
**ΜΕΤΑΒΛΗΤΕΣ**  
     **ΑΚΕΡΑΙΕΣ:** Πληθ, ΑΑΜΙΝ  
     **ΠΡΑΓΜΑΤΙΚΕΣ:** Τιμή, MIN, Κόστος

## ΑΡΧΗ

MIN ← 1000000

Πληθ ← 0

## ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

**ΔΙΑΒΑΣΕ** Τιμή

**Αν** Τιμή > 0 **ΤΟΤΕ**

Πληθ ← Πληθ+1

**Αν** Τιμή < MIN **ΤΟΤΕ**

MIN ← Τιμή

AAMIN ← Πληθ

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_Αν**

**ΜΕΧΡΙΣ\_ΟΤΟΥ** Τιμή <= 0

**Αν** Πληθ > 0 **ΤΟΤΕ**

Κόστος ← 2000\*MIN

**ΓΡΑΨΕ** AAMIN, Κόστος

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'Δε δόθηκαν δεδομένα'

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**20.** Ένας μαθητής ζήτησε από τους γονείς του να του αγοράσουν ένα smartphone αξίας 650 €. Οι γονείς του δήλωσαν ότι μπορούν να του διαθέσουν σταδιακά το ποσό, δίνοντας του κάθε εβδομάδα ποσό διπλάσιο από την προηγούμενη, αρχίζοντας την πρώτη εβδομάδα με ποσό 35 €. Να γραφεί αλγόριθμος που:

α) υπολογίζει μετά από πόσες εβδομάδες ο μαθητής θα μπορέσει να αγοράσει το smartphone.

β) υπολογίζει και ελέγχει υπόλοιπο χρημάτων.

### Λύση

**Αλγόριθμος** Ασκ\_1

ποσό←0

χαρτζηλίκι←35

week←0

**Όσο** (ποσό < 650) **επανάλαβε**

ποσό←ποσό + χαρτζηλίκι

χαρτζηλίκι←χαρτζηλίκι \* 2

week←week + 1

Τέλος\_επανάληψης

**Εμφάνισε** "Αγορά σε ", week, "εβδομάδες"

**Αν** ποσό > cost **τότε**

**Εμφάνισε** "Περίσσευμα: ", ποσό-650

**Τέλος\_Αν**

**Τέλος** Ασκ\_1

**21.** Σύμφωνα με την παράδοση, η δημιουργία του σκακιού οφείλεται στην προτροπή ενός βασιλιά να ανταμείψει όποιον από τους συμβούλους του θα έφτιαχνε ένα πρωτότυπο

παιχνίδι, διαφορετικό κάθε φορά στην εξέλιξή του. Ένας λοιπόν από τους συμβούλους του σκέφτηκε να δημιουργήσει το σκάκι. Ο βασιλιάς ενθουσιάστηκε και ρώτησε για την αμοιβή του συμβούλου. Αυτός ζήτησε να πάρει το ρύζι που θα μαζευόταν ως εξής: Στο 1ο τετράγωνο του σκακιού να τοποθετηθεί 1 κόκκος ρυζιού, στο 2ο τετράγωνο 2 κόκκοι ρυζιού, στο 3ο τετράγωνο 4 κόκκοι, στο 4ο τετράγωνο 8 κόκκοι κλπ. Το άθροισμα των κόκκων θα ήταν η αμοιβή του. Να γραφεί αλγόριθμος που θα υπολογίζει πόσοι τόνοι ρυζιού θα ήταν η ποσότητα αυτή, να θεωρηθεί ότι 1 Kg ρυζιού έχει 10000 κόκκους.

### Λύση

**Αλγόριθμος** Ασκ

κόκκοι ← 1

P ← 1

**Για** I από 2 μέχρι 64

P ← P \* 2

κόκκοι ← κόκκοι + P

**Τέλος\_επανάληψης**

τόνοι ← (κόκκοι/10000)/1000

**Εμφάνισε** “Σύνολο τόνων ρυζιού”, τόνοι

**Τέλος** Ασκ

**22.** Αλγόριθμος που διαβάσει ένα ακέραιο αριθμό και διασφαλίζει ότι είναι θετικός. Αν ο αριθμός είναι περιττός τότε εμφανίζει όλους τους θετικούς περιττούς αριθμούς που προηγούνται από αυτόν, ενώ αν είναι άρτιος εμφανίζει όλους τους θετικούς άρτιους αριθμούς που προηγούνται από αυτόν.

### Λύση

**Αλγόριθμος** Ασκ\_1

**Αρχή\_επανάληψης** ! Διασφάλιση τιμής

**Διάβασε** num

**Μέχρις\_ότου** (num > 0)

**Αν** (num mod 2 = 1) τότε

**Για** i από 1 μέχρι num-2 **με\_βήμα** 2

**Εμφάνισε** i

**Τέλος\_επανάληψης**

**αλλιώς**

**Για** i από 2 μέχρι num-2 **με\_βήμα** 2

**Εμφάνισε** i

**Τέλος\_επανάληψης**

**Τέλος\_Αν**

**Τέλος** Ασκ\_1

**23.** Να γραφεί αλγόριθμος ο οποίος εμφανίζει όλους τους τριψήφιους θετικούς αριθμούς που το άθροισμα των ψηφίων τους ισούται με 9 π.χ. οι αριθμοί 900, 513.

### Λύση

**Αλγόριθμος** Ασκ\_3

**Για** i από 100 μέχρι 999

```
A1 ← i div 100
A2 ← i mod 100 div 10
A3 ← i mod 100 mod 10 !αλλιώς A3 ← i mod 10
S ← A1 + A2 + A3
Αν S = 6 τότε Εμφάνισε i
```

**Τέλος\_επανάληψης**

**Τέλος** Ασκ\_3

**24.** Να γραφεί αλγόριθμος ο οποίος διαβάζει τυχαίους αριθμούς μέχρι να δοθεί τιμή φρουρός 0. Υπολογίζει τη μέγιστη συχνότητα ίσων συνεχόμενων αριθμών και ποιος ήταν αυτός.

**Λύση**

**Αλγόριθμος** Συχνότητες

**Διάβασε** x

max\_συχν ← -1

συχν ← 1

**Όσο** αριθμ <> 0 **επανάλαβε**

προηγ ← x

διαβασε x

**Αν** προηγ=x **τότε**

συχν ← συχν+1

**Αν** συχν > max\_συχν **τότε**

max\_συχν ← συχν

γ ← αριθμ

**Τέλος\_Αν**

Αλλιώς

συχν ← 1

**Τέλος\_Αν**

**Τέλος\_Επανάληψης**

**Εμφάνισε** γ, max\_συχν

**Τέλος** Συχνότητες

**25.** Σε ένα πρωτάθλημα ποδοσφαίρου συμμετέχουν 16 ομάδες. Γράψτε πρόγραμμα το οποίο:

α) διαβάζει το όνομα κάθε ομάδας και τα αποτελέσματα (Νίκη ή Ισοπαλία ή Ήττα) που έφερε σε κάποιους αγώνες του πρωταθλήματος (η είσοδος ολοκληρώνεται μόλις δοθεί ως τιμή εισόδου η λέξη 'ΤΕΛΟΣ')

β) υπολογίζει και εμφανίζει: το όνομα της ομάδας με τη μεγαλύτερη βαθμολογία, το όνομα της ομάδας με τις λιγότερες ισοπαλίες. Παρατηρήσεις: Καμιά ομάδα δεν είχε περισσότερες από 10 ισοπαλίες στους αγώνες της και κάθε νίκη αποτιμάται με 3 βαθμούς, η ισοπαλία με 1, ενώ η ήττα δε δίνει κανένα βαθμό.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ** ΠΟΔΟΣΦΑΙΡΟ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** κ, Ισοπαλίες, MIN, MAX, Βαθμ

**ΧΑΡΑΚΤΗΡΕΣ:** Απ, ΟΝ, ΟΝΜΑΧ, ΟΝΜΙΝ

**ΑΡΧΗ**

ΜΑΧ ← -100

ΜΙΝ ← 11

**Για κ ΑΠΟ 1 ΜΕΧΡΙ 16**

**ΔΙΑΒΑΣΕ ΟΝ**

Βαθμ ← 0

Ισοπαλίες ← 0

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ Απ**

**Αν Απ < > 'ΤΕΛΟΣ' ΤΟΤΕ**

**Αν Απ = 'Νίκη' ΤΟΤΕ**

Βαθμ ← Βαθμ+2

**ΑΛΛΙΩΣ**

**Αν Απ = 'Ισοπαλία' ΤΟΤΕ**

Βαθμ ← Βαθμ+1

Ισοπαλίες ← Ισοπαλίες+1

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_Αν**

**ΜΕΧΡΙΣ\_ΟΤΟΥ Απ = 'ΤΕΛΟΣ'**

**Αν Βαθμ > ΜΑΧ ΤΟΤΕ**

ΜΑΧ ← Βαθμ

ΟΝΜΑΧ ← ΟΝ

**ΤΕΛΟΣ\_Αν**

**Αν Ισοπαλίες < ΜΙΝ ΤΟΤΕ**

ΜΙΝ ← Ισοπαλίες

ΟΝΜΙΝ ← ΟΝ

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ ΟΝΜΑΧ, ΟΝΜΙΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**26.** Γράψτε αλγόριθμο που διαβάζει τις ημερομηνίες γέννησης των 350 μαθητών ενός γυμνασίου και υπολογίζει πόσοι μαθητές γεννήθηκαν: χειμωνιάτικα, την τελευταία μέρα του χρόνου, πρωτομαγιά, μετά το 1999.

**Λύση**

**Αλγόριθμος** Γυμν

Χειμ ← 0

Τελ ← 0

Πρ ← 0

Μετά ← 0

**Για κ από 1 μέχρι 350**

**Διάβασε** Ημέρα, Μήνας, Χρόνος

**Αν** (Μήνας = 12) Ή (Μήνας = 1) Ή (Μήνας = 2) **τότε**

Χειμ ← Χειμ+1



**Τέλος\_Αν**  
**Αν** (Ημέρα = 31) ΚΑΙ (Μήνας = 12) **τότε**  
Τελ ← Τελ+1

**Τέλος\_Αν**  
**Αν** (Ημέρα = 1) ΚΑΙ (Μήνας = 5) **τότε**  
Πρ ← Πρ+1

**Τέλος\_Αν**  
**Αν** Χρόνος > 1999 **τότε**  
Μετά ← Μετά+1

**Τέλος\_Αν**  
**Τέλος\_επανάληψης**  
**Αποτελέσματα** // Χειμ, Τελ, Πρ, Μετά //  
**Τέλος** Γυμν

**27.** Γράψτε αλγόριθμο που διαβάζει

α) το πλήθος των εντολών 18 προγραμμάτων,

β) υπολογίζει και εμφανίζει: το μέσο όρο εντολών ανά πρόγραμμα, πόσα προγράμματα είχαν περισσότερες από 200 εντολές και το ποσοστό (%) των προγραμμάτων που είχαν το πολύ 120 εντολές.

**Λύση**

**Αλγόριθμος** Προγ

σ ← 0

φ1 ← 0

φ2 ← 0

**Για** λ **από** 1 **μέχρι** 18

**Διάβασε** Εντ

σ ← σ+Εντ

**Αν** Εντ > 200 **τότε**

φ1 ← φ1+1

**Τέλος\_αν**

**Αν** Εντ <= 120 **τότε**

φ2 ← φ2+1

**Τέλος\_αν**

**Τέλος\_επανάληψης**

ΜΟ ← σ/18

ΠΟΣ ← 100\*φ2/18

**Εμφάνισε** ΜΟ, φ1, ΠΟΣ

**Τέλος** Προγ

**28.** Σε ένα ραδιοφωνικό σταθμό γίνεται ένας διαγωνισμός για να μοιραστούν δώρα συνολικής αξίας 2000 €. Κάθε ακροατής που διαγωνίζεται απαντά τηλεφωνικά σε 4 ερωτήσεις που η καθεμία μπορεί να πάρει βαθμό από 0 μέχρι και 12. Ο διαγωνιζόμενος κερδίζει αν έχει συνολικό βαθμό και από τις 4 ερωτήσεις τουλάχιστον 40. Η αξία του δώρου που θα πάρει είναι ίση με το τριτλάσιο της συνολικής του βαθμολογίας σε €. Να γραφεί αλγόριθμος ο οποίος:

Α) διαβάσει το όνομα κάθε διαγωνιζόμενου και την βαθμολογία του σε καθεμία από τις 4

ερωτήσεις

Β) ελέγχει για κάθε διαγωνιζόμενο αν δικαιούται δώρο μέχρι να μοιραστεί όλο το αρχικό ποσό του σταθμού

Γ) εμφανίζει πόσοι διαγωνίστηκαν, πόσοι κέρδισαν (μαζί με τα ονόματά τους) και πόσοι κέρδισαν δώρα αξίας 120 €.

### Λύση

#### Αλγόριθμος Ασκ\_1

poso  $\leftarrow$  2000

diag  $\leftarrow$  0

win  $\leftarrow$  0

win120  $\leftarrow$  0

**Όσο** (poso > 0) **επανάλαβε**

**Διάβασε** name

$\Sigma \leftarrow 0$

**Για** i **από** 1 **μέχρι** 4

**Διάβασε** answer

$\Sigma \leftarrow \Sigma + \text{answer}$

**Τέλος\_επανάληψης**

**Αν** ( $\Sigma \geq 40$ ) **τότε**

        doro  $\leftarrow 2 * \Sigma$

**Εμφάνισε** "Ο παίκτης", name, "κέρδισε", doro

        win  $\leftarrow \text{win} + 1$

**Αν** doro = 120 **τότε**

            win120  $\leftarrow \text{win120} + 1$

**Τέλος\_Αν**

**Αν** (poso  $\geq$  doro) **τότε**

            poso  $\leftarrow \text{poso} - \text{doro}$

**αλλιώς**

        poso  $\leftarrow 0$  !Αν το δώρο είναι περισσότερο από το ποσό

**Τέλος\_Αν** !που απομένει, ο σταθμός δίνει ότι έχει και μηδενίζεται

**Τέλος\_Αν**

    diag  $\leftarrow \text{diag} + 1$

**Τέλος\_Επανάληψης**

**Εμφάνισε** "Διαγωνίστηκαν", diag, "ακροατές"

**Εμφάνισε** "Κέρδισαν", win, "ακροατές"

**Εμφάνισε** "Δώρα αξίας 120 € κέρδισαν", win120

**Τέλος** Ασκ\_1

**29.** Ένα μικρό πλοίο μπορεί να δεχτεί συνολικό φορτίο 9500 kgr. Κατά τη φόρτωσή του πρέπει το συνολικό φορτίο που τοποθετείται στην αριστερή πλευρά να είναι περίπου ίσο με το συνολικό φορτίο που τοποθετείται στη δεξιά πλευρά. Κάθε κιβώτιο ζυγίζεται πριν γίνει η φόρτωσή του. Αν το συνολικό φορτίο της αριστερής πλευράς είναι μικρότερο από αυτό της δεξιάς τότε το κιβώτιο τοποθετείται στην αριστερή πλευρά, διαφορετικά τοποθετείται στην δεξιά πλευρά. Να γραφεί αλγόριθμος ο οποίος:

Α) διαβάζει το βάρος κάθε κιβωτίου

Β) εμφανίζει το πλήθος των κιβωτίων κάθε πλευράς

Γ) εμφανίζει το συνολικό βάρος κάθε πλευράς

## Λύση

### Αλγόριθμος Ασκ\_2

```
left ← 0
right ← 0
left_kiv ← 0
right_kiv ← 0
sum ← 0
```

### Διάβασε varos

Όσο (sum + varos < 9500) επανάλαβε

Αν (left < right) τότε

```
left ← left + varos
left_kiv ← left_kiv + 1
```

αλλιώς

```
right ← right + varos
right_kiv ← right_kiv + 1
```

Τέλος\_Αν

```
sum ← sum + varos
```

Διάβασε varos

Τέλος\_επανάληψης

Εμφάνισε "Κιβώτια δεξιάς πλευράς", right\_kiv, "με βάρος", right

Εμφάνισε "Κιβώτια αριστερής πλευράς", left\_kiv, "με βάρος", left

Τέλος Ασκ\_2

30. Ο πληθυσμός μιας πόλης A είναι 30.000 και αυξάνεται ετησίως κατά 3%. Μια πόλη B έχει πληθυσμό 45.000 και αυξάνεται ετησίως 1%. Να γίνει αλγόριθμος που θα βρίσκει σε πόσα χρόνια από τώρα ο πληθυσμός της πόλης A θα ξεπεράσει τον πληθυσμό της πόλης B.

## Λύση

### Αλγόριθμος Πληθυσμοί

```
πληθ_A ← 30000
πληθ_B ← 45000
χρόνια ← 0
```

Όσο πληθ\_A <= πληθ\_B επανάλαβε

! Για κάθε χρόνο που περνά, αύξησε τον πληθυσμό της πόλης A κατά 3% και της B 2%

```
πληθ_A ← πληθ_A + πληθ_A * 3 / 100
```

```
πληθ_B ← πληθ_B + πληθ_B * 1 / 100
```

! επίσης μέτρησε τα χρόνια που έχουν έως τώρα περάσει

```
χρόνια ← χρόνια + 1
```

Τέλος\_επανάληψης

Εμφάνισε "Μετά από ", χρόνια, " χρόνια, ο πληθυσμός της πόλης A θα ξεπεράσει της B"

Τέλος Πληθυσμοί

31. Να γίνει Αλγόριθμος που να εμφανίζει όλους τους αριθμούς από το 1 έως το 100 που

έχουν ακριβώς 5 διαιρέτες.

### Λύση

#### Αλγόριθμος Διαιρέτες

! Ψάξε όλους τους υποψήφιους αριθμούς από το 1 μέχρι το 100...

Για  $x$  από 1 μέχρι 100

! Για κάθε αριθμό  $x$  βρες το πλήθος των διαιρετών  
πλήθος\_διαιρετών  $\leftarrow 0$

Για  $y$  από 1 μέχρι  $x$

Αν  $x \bmod y = 0$  τότε

πλήθος\_διαιρετών  $\leftarrow$  πλήθος\_διαιρετών + 1

Τέλος\_Αν

Τέλος\_επανάληψης

! Αν το πλήθος διαιρετών του αριθμού  $x$  είναι 5, εμφάνισε τον αριθμό  $x$

Αν πλήθος\_διαιρετών = 5 τότε

Εμφάνισε  $x$

Τέλος\_Αν

Τέλος\_επανάληψης

Τέλος Διαιρέτες

**32.** Για να υπολογίσουμε το εμβαδόν  $E$  ενός τριγώνου, όταν δίνονται τα μήκη των πλευρών του, εφαρμόζουμε τον μαθηματικό τύπο του Ήρωνα:

$E = t * (t - a) * (t - b) * (t - c)$ , σε τετραγωνική ρίζα, όπου  $t = (a+b+c)/2$  η ημιπερίμετρος του τριγώνου και  $a, b, c$  τα μήκη των δοθέντων πλευρών του τριγώνου.

Γράψτε ένα πρόγραμμα το οποίο να υπολογίζει το εμβαδόν  $E$  ενός τριγώνου καθώς και την περίμετρο του όταν δίνονται τα μήκη των πλευρών του. Το πρόγραμμα θα ελέγχει αν τα δοθέντα μήκη των πλευρών είναι θετικοί αριθμοί.

### Λύση

Αλγόριθμος ΠΔΠ12

! gymnasio 12ος ΠΔΠ

Αρχή\_επανάληψης

Διάβασε  $a$

Μέχρις\_ότου  $a > 0$

Αρχή\_επανάληψης

Διάβασε  $b$

Μέχρις\_ότου  $b > 0$

Αρχή\_επανάληψης

Διάβασε  $c$

Μέχρις\_ότου  $c > 0$

Αν  $a < b + c$  και  $b < a + c$  και  $c < a + b$  τότε ! Σύνθετη συνθήκη

$t \leftarrow (a + b + c)/2$

$EM \leftarrow T\_P(t*(t - a)*(t - b)*(t - c))$

$PER \leftarrow t*2$

Εμφάνισε  $EM, PER$

αλλιώς

**Εμφάνισε "ΔΕΝ ΟΡΙΖΕΤΑΙ ΤΡΙΓΩΝΟ"**

**Τέλος\_Αν**

**Τέλος ΠΔΠ12**

**33.** Δημιουργήστε έναν αλγόριθμο ο οποίος θα εκτυπώνει τους αριθμούς από το 1 έως το 100.

**Λύση**

Αλγόριθμος πχ1

Για κ από 1 μέχρι 100

Εκτύπωσε κ

τέλος\_επανάληψης

Τέλος πχ1

**34.** Γράψτε ένα πρόγραμμα το οποίο θα εκτυπώνει τους αριθμούς μεταξύ 17 και 100 οι οποίοι μπορούν να διαιρούνται τέλεια με το 17.

**Λύση**

**Αλγόριθμος πχ1**

**Για κ από 17 μέχρι 100 με\_βήμα 17**

Εκτύπωσε κ

**τέλος\_επανάληψης**

**Τέλος πχ1**

**35.** Δίνεται το παρακάτω τμήμα αλγορίθμου, με δεδομένο έναν μονοδιάστατο πίνακα A[20]. Ο πίνακας περιέχει άρτιους και περιττούς θετικούς ακераίους, σε τυχαίες θέσεις. Το τμήμα αλγορίθμου δημιουργεί ένα νέο πίνακα B[20], στον οποίο υπάρχουν πρώτα οι άρτιοι και μετά ακολουθούν οι περιττοί. Να συμπληρωθούν τα κενά.

K ← 0

**Για i από ..... μέχρι .....**

**Αν A[i] mod 2 = 0 τότε**

K ← .....

B[.....] ← A[i]

**Τέλος\_Αν**

Τέλος\_επανάληψης

**Για i από ..... μέχρι .....**

**Αν A[i] mod 2 = ..... τότε**

K ← .....

B[.....] ← A[.....]

**Τέλος\_Αν**

Τέλος\_επανάληψης

**Λύση**

1, 20, K+1, K, 1, 20, 1, K+1, K, i

**36.** Στο παρακάτω τμήμα εντολών να συμπληρωθούν τα κενά με μία σταθερά ή μεταβλητή.

Για i από \_\_\_\_ μέχρι \_\_\_\_ με βήμα \_\_\_\_

Για j από \_\_\_\_ μέχρι \_\_\_\_ με βήμα \_\_\_\_

Εμφάνισε j

Τέλος\_επανάληψης

Τέλος\_επανάληψης

έτσι ώστε, αν εκτελεστεί να εμφανιστούν οι αριθμοί: 10, 14, 18, 22, 13, 17, 21, 16, 20.

**Λύση**

10, 16, 3, i, 22, 4

**37.** Να γραφεί ένα πρόγραμμα για την κατασκευή κι εμφάνιση ενός πίνακα προπαίδειας από το 1 έως το 10.

**Λύση**

**Αλγόριθμος** προπαίδεια

Για κ από 1 μέχρι 10

    Για λ από 1 μέχρι 10

        Εμφάνισε κ,"\*",λ,"=",κ\*λ

    τέλος\_επανάληψης

τέλος\_επανάληψης

Τέλος προπαίδεια

**38.** Να γραφεί αλγόριθμος που θα βρίσκει σε ποιον όρο το άθροισμα  $1+2+3+\dots$  γίνεται μεγαλύτερο του 2000.

**Λύση**

**Αλγόριθμος** πχ3

πλ <- 0

αθρ <- 0

χ <- 1

**Αρχή\_επανάληψης**

αθρ <- αθρ + χ

χ <- χ+1

μέχρις\_ότου αθρ>1000

Εμφάνισε χ

Τέλος πχ3

**39.** Να γραφεί ένας αλγόριθμος που θα δέχεται επαναληπτικά αριθμούς από το πληκτρολόγιο μέχρι να εισαχθούν 100 αριθμοί ή το άθροισμά τους ξεπεράσει ένα γνωστό και δεδομένο όριο M. Ο αλγόριθμος θα εμφανίζει στο τέλος το πλήθος των αριθμών που έχουν εισαχθεί καθώς και το άθροισμά τους.

**Λύση**

### Αλγόριθμος πχ2

Δεδομένα //M

πλ <- 0

αθρ <- 0

### Αρχή\_επανάληψης

Διάβασε χ

πλ <- πλ+1

αθρ <- αθρ + χ

**μέχρις\_ότου** πλ=100 Η αθρ>M

Εμφάνισε πλ, αθρ

**Τέλος** πχ2

**40.** Ένας θετικός ακέραιος λέγεται ατελής, τέλειος ή πλούσιος αν το άθροισμα των διαιρετών του εκτός από τον ίδιο τον αριθμό είναι αντίστοιχα μικρότερο, ίσο ή μεγαλύτερο από τον αριθμό αυτό, π.χ. το 6 είναι τέλειος αριθμός διότι  $6=1+2+3$  και οι αριθμοί 1,2,3 είναι οι διαιρέτες του. Να γραφεί αλγόριθμος για την εμφάνιση της αντίστοιχης λέξης για μια περιοχή 50 συνεχόμενων ακεραίων αριθμών. Τα όρια θα εισάγονται από το πληκτρολόγιο.

### Λύση

**ΔΙΑΒΑΣΕ** N1

**ΑΝ** N1>0 **ΤΟΤΕ**

N2=N1+50

**ΤΕΛΟΣ\_ΑΝ**

**ΓΙΑ** A **ΑΠΟ** N1 **ΜΕΧΡΙ** N2 **ΜΕ\_ΒΗΜΑ** 1

SUM =0

**ΓΙΑ** K **ΑΠΟ** 1 **ΜΕΧΡΙ** A **DIV** 2

**ΑΝ** (A MOD K)==0 **ΤΟΤΕ** SUM=SUM+K

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ** (A==SUM) **ΤΟΤΕ** **ΕΜΦΑΝΙΣΕ** "ΤΕΛΕΙΟΣ"

**ΑΝ** (A<SUM) **ΤΟΤΕ** **ΕΜΦΑΝΙΣΕ** "ΑΤΕΛΗΣ"

**ΑΝ** (A>SUM) **ΤΟΤΕ** **ΕΜΦΑΝΙΣΕ** "ΠΛΟΥΣΙΟΣ"

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**41.** Η άσκηση περιλαμβάνει 4 δραστηριότητες:

Στην πρώτη να μετατρέψετε τη μορφή ΟΣΟ σε μορφή ΜΕΧΡΙΣ\_ΟΤΟΥ

Στη δεύτερη να μετατραπεί η ΓΙΑ σε μορφή ΟΣΟ

Στην τρίτη τη μορφή ΟΣΟ σε μορφή ΓΙΑ

Στην τέταρτη δραστηριότητα, καλείσθε να επιλέξετε την κατάλληλη μορφή για ένα δεδομένο πρόβλημα. Κάθε δραστηριότητα (πλην της τέταρτης) θα περιλαμβάνει τον αρχικό κώδικα σε ΨΕΥΔΟΓΛΩΣΣΑ. Θα ακολουθεί λεκτική περιγραφή του αλγορίθμου, δημιουργία διαγράμματος ροής, ενδεχομένως πίνακας τιμών μεταβλητών και τέλος μετατροπή σε ισοδύναμη μορφή.

### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Ο Αλγόριθμος σε ψευδογλώσσα:

**Αλγόριθμος** Δ1

! Σύστημα ελέγχου ωρών στάθμευσης

! Διαβάζει τις ώρες στάθμευσης κάθε οχήματος και τερματίζει όταν δοθεί ως &είσοδος η τιμή 0

S←0

**Εμφάνισε** 'Δώσε ώρες στάθμευσης:'

**Διάβασε** ώρες

**Όσο** ώρες ≠ 0 **επανάλαβε**

**Αν** ώρες ≤2 **τότε**

        XP←ώρες\*1,5

**αλλιώς**

        XP←2\*1,5 + (ώρες-2)\*1

**Τέλος\_Αν**

S←S+XP

**Εμφάνισε** 'Χρυστάς', XP, 'Ευρώ'

**Διάβασε** ώρες

**Τέλος\_επανάληψης**

**Εμφάνισε** S

**Τέλος** Δ1

**Λεκτική περιγραφή:** Θα πρέπει να είστε σε θέση να αρχικοποιείτε τις μεταβλητές και να περιγράφετε επακριβώς τη συνθήκη τερματισμού.

**Διάγραμμα ροής:** Μπορεί να υλοποιηθεί μέσω του οδηγού διαγραμμάτων ροής (ΑΛΓΟΡΙΘΜΙΚΗ) ή στην περίπτωση που δεν επαρκεί ο χρόνος, στο χαρτί.

**Μετατροπή:** Να μετατραπεί στην ισοδύναμη μορφή ΜΕΧΡΙΣ\_ΟΤΟΥ θεωρώντας ως δεδομένη την καταλληλότητα της μορφής αυτής. Για τον συγκεκριμένο αλγόριθμο επισημαίνονται τα ακόλουθα:

- ✓ Οι αρχικοποιήσεις και το σώμα του βρόχου παραμένουν ανέπαφα
- ✓ Η συνθήκη τερματισμού αντιστρέφεται

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Ο Αλγόριθμος σε ψευδογλώσσα:

**Αλγόριθμος** Δ2

! Υπολογισμός αριθμητικής προόδου

S←0

**Για** i από 1 μέχρι 100 **με βήμα** 2

S← S + 1 / i

**Τέλος\_επανάληψης**

**Εμφάνισε** S

**Τέλος** Δ2

**Να τονιστεί ότι τα διαγράμματα ροής της ΟΣΟ και της ΓΙΑ είναι ταυτόσημα.**

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Ο Αλγόριθμος σε ψευδογλώσσα:

**Αλγόριθμος** Δ3

! Υπολογισμός μέσου όρου μαθήματος

i←1

**ΟΣΟ** i≤100 **επανάλαβε**



**Εμφάνισε** 'Δώσε προφορικό και γραπτό βαθμό:'

**Διάβασε** πρ,γρ

**Εμφάνισε** (πρ+γρ)/2

i←i+1

**Τέλος\_επανάληψης**

**Τέλος Δ3**

#### **ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4**

Διατύπωση προβλήματος: Ως επέκταση της Δραστηριότητας 3, ο αλγόριθμος τερματίζει όταν δοθεί οποιοσδήποτε αρνητικός βαθμός ή το πολύ 100 μαθητές.

**42.** Ένα πρατήριο υγρών καυσίμων διαθέτει έναν τύπο καυσίμου που αποθηκεύεται σε δεξαμενή χωρητικότητας 10.000 λίτρων. Να αναπτύξετε αλγόριθμο ο οποίος: 1. να διαβάσει την ποσότητα (σε λίτρα) του καυσίμου που υπάρχει αρχικά στη δεξαμενή μέχρι να δοθεί έγκυρη τιμή. Για κάθε όχημα που προσέρχεται στο πρατήριο: 2. να διαβάσει τον τύπο του οχήματος ("B" για βυτιοφόρο όχημα που προμηθεύει το πρατήριο με καύσιμο και "E" για επιβατηγό όχημα που προμηθεύεται καύσιμο από το πρατήριο). 3. Αν το όχημα είναι βυτιοφόρο τότε να γεμίζει τη δεξαμενή μέχρι την πλήρωσή της. Αν το όχημα είναι επιβατηγό τότε να διαβάσει την ποσότητα καυσίμου την οποία θέλει να προμηθευτεί (μονάδες 2) και, αν υπάρχει επάρκεια καυσίμου στη δεξαμενή, τότε το επιβατηγό όχημα να εφοδιάζεται με τη ζητούμενη ποσότητα καυσίμου, διαφορετικά το όχημα να μην εξυπηρετείται 4. Η επαναληπτική διαδικασία να τερματίζεται, όταν αδειάσει η δεξαμενή του πρατηρίου ή όταν δεν εξυπηρετηθούν τρία διαδοχικά επιβατηγά οχήματα. 5. Στο τέλος ο Αλγόριθμος να εμφανίζει: α. τη μέση ποσότητα καυσίμου ανά επιβατηγό όχημα που εξυπηρετήθηκε β. τη συνολική ποσότητα καυσίμου με την οποία τα βυτιοφόρα ανεφοδιάσαν τη δεξαμενή. Θεωρήστε ότι στο πρατήριο προσέρχεται ένα τουλάχιστον επιβατηγό όχημα για το οποίο η ποσότητα καυσίμου στη δεξαμενή επαρκεί.

#### **Λύση**

**Αλγόριθμος** πρατήριο

Αρχή\_επανάληψης

**Διάβασε** ποσ

**Μέχρις\_ότου** ποσ >= 0 **ΚΑΙ** ποσ <= 10000

πλήθος ← 0, ν ← 0, sum1 ← 0, sum2 ← 0

**Όσο** ποσ > 0 **και** πλήθος < 3 **επανάλαβε**

**Διάβασε** τύπο

**Αν** τύπο = 'B' **τότε**

sum2 ← sum2 + (10000 - ποσ)

ποσ ← 10000

εξυπηρέτηση ← αληθής

αλλιώς

**Εμφάνισε** "Δώστε λίτρα"

**Διάβασε** λιτ

**Αν** λιτ <= ποσ **τότε**

ποσ ← ποσ - λιτ

sum1 ← sum1 + λιτ

ν ← ν + 1

εξυπηρέτηση ← αληθής

```

αλλιώς
εξυπηρέτηση ← ψευδής
Τέλος_Αν
Τέλος_Αν
Αν εξυπηρέτηση = αληθής τότε
Πλήθος ← 0
αλλιώς
πλήθος ← πλήθος + 1
Τέλος_Αν
Τέλος_επανάληψης
Αν  $v > 0$  τότε ! Ελεγχος καθοριστικότητας
ΜΟ ←  $\text{sum1} / v$ 
Εμφάνισε "Ο μέσος όρος ποσότητας καυσίμου είναι ", ΜΟ
Τέλος_Αν
Εμφάνισε "Η συνολική ποσότητα που εφοδιάστηκε η δεξαμενή είναι", sum2
Τέλος πρατήριο

```

**45.** Ένας συλλέκτης γραμματοσήμων επισκέπτεται στο διαδίκτυο το αγαπημένο του ηλεκτρονικό κατάστημα προκειμένου να αγοράσει γραμματόσημα. Προτίθεται να ξοδέψει μέχρι 1500 ευρώ. Να αναπτύξετε αλγόριθμο ο οποίος:

α. Για κάθε γραμματόσημο, να διαβάζει την τιμή και την προέλευσή του (ελληνικό/ξένο) και να επιτρέπει την αγορά του, εφόσον η τιμή του δεν υπερβαίνει το διαθέσιμο υπόλοιπο χρημάτων. Διαφορετικά να τερματίζει τυπώνοντας το μήνυμα «ΤΕΛΟΣ ΑΓΟΡΩΝ». ΣΗΜΕΙΩΣΗ: Δεν απαιτείται έλεγχος εγκυρότητας για τα δεδομένα εισόδου.

β. Να τυπώνει:

1. Το συνολικό ποσό που ξόδεψε ο συλλέκτης.
2. Το πλήθος των ελληνικών και το πλήθος των ξένων γραμματοσήμων που αγόρασε.
3. Το ποσό που περίσσεψε, εφόσον υπάρχει, διαφορετικά το μήνυμα «ΞΑΝΤΛΗΘΗΚΕ ΟΛΟ ΤΟ ΠΟΣΟ». (Θέμα Πανελλαδικών)

### Λύση

**ΑΛΓΟΡΙΘΜΟΣ** Γραμματόσημα

πλήθοςE ← 0

πλήθοςΞ ← 0

σύνολο ← 0

υπέρβαση ← Ψευδές

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ τιμή

ΔΙΑΒΑΣΕ προέλευση

ΑΝ σύνολο + τιμή ≤ 1500 ΤΟΤΕ

σύνολο ← σύνολο + τιμή

ΑΝ προέλευση="ελληνικό" ΤΟΤΕ

πλήθοςE ← πλήθοςE + 1

ΑΛΛΙΩΣ

πλήθοςΞ ← πλήθοςΞ + 1

ΤΕΛΟΣ\_ΑΝ

ΑΝ σύνολο=1500 ΤΟΤΕ

υπέρβαση ← ΑΛΗΘΕΣ

ΤΕΛΟΣ\_ΑΝ  
ΑΛΛΙΩΣ  
υπέρβαση ← ΑΛΗΘΕΣ  
ΤΕΛΟΣ\_ΑΝ  
**ΜΕΧΡΙΣ\_ΟΤΟΥ** υπέρβαση  
**ΓΡΑΨΕ** "ΤΕΛΟΣ ΑΓΟΡΩΝ"  
**ΓΡΑΨΕ** σύνολο  
**ΓΡΑΨΕ** πλήθοςE, πλήθοςΞ  
**ΑΝ** σύνολο < 1500 **ΤΟΤΕ**  
υπόλοιπο ← 1500-σύνολο  
**ΓΡΑΨΕ** υπόλοιπο  
ΑΛΛΙΩΣ  
**ΓΡΑΨΕ** "ΕΞΑΝΤΛΗΘΗΚΕ ΟΛΟ ΤΟ ΠΟΣΟ"  
ΤΕΛΟΣ\_ΑΝ  
**ΤΕΛΟΣ** Γραμματόσημα

### 3.4 Ασκήσεις προς επίλυση

1. Να αιτιολογήσετε ποιο αλγοριθμικό κριτήριο παραβιάζεται στους βρόχους:

$x \leftarrow 1$

**Όσο**  $x < 0$  **επανάλαβε**

**Διάβασε**  $y$

$z \leftarrow x + y$

$x \leftarrow x + 2$

Τέλος\_επανάληψης

**Εκτύπωσε**  $z$

Αρχή\_επανάληψης

**Διάβασε**  $on$

**Μέχρις\_ότου**  $on < >$  "Τέλος"

$sum \leftarrow 0$

**Όσο**  $on < >$  "Τέλος" **επανάλαβε**

**Διάβασε**  $x$

$sum \leftarrow sum + x$

Τέλος\_επανάληψης

**Εμφάνισε**  $sum$

2. Να μετατραπούν σε ισοδύναμη εντολή Αρχή\_επανάληψης ... Μέχρις\_ότου τα παρακάτω τμήματα αλγορίθμου.

**Διάβασε**  $x$

**Όσο**  $x > 0$  και  $x < 10$  **επανάλαβε**

**Διάβασε**  $x$

$x \leftarrow x + y$

Τέλος\_επανάληψης

.

**Διάβασε**  $a$

**Όσο**  $a > 0$  **επανάλαβε**

$a \leftarrow a - 3$

**Εμφάνισε**  $a$

Τέλος\_επανάληψης

3. Να μετατραπεί ισοδύναμα σε Όσο, Μέχρις\_ότου το παρακάτω τμήμα αλγορίθμου.

$Sum \leftarrow 0$

**Για**  $i$  **από** 100 **μέχρι** 0 **με\_βήμα** -2

$sum \leftarrow sum + 1$

Τέλος\_επανάληψης

**Εκτύπωσε**  $sum$

4. Ο σημερινός αριθμός αυτοκινήτων που κυκλοφορούν σε μια πόλη είναι 80000. Αν ο αριθμός αυτός αυξάνεται με ρυθμό 7%, να γραφεί αλγόριθμος που να υπολογίζει σε πόσα χρόνια ο αριθμός των αυτοκινήτων θα ξεπεράσει τις 160000. Να εμφανίζει πόσα θα

είναι τότε τα αυτοκίνητα; Γενικεύστε το πρόγραμμα για Arith\_Aytko αυτοκίνητα, που αυξάνονται με ετήσιο ρυθμό Rhythmos και ξεπερνούν ένα δοσμένο όριο Orio.

**5.** Το στατιστικό τμήμα της τραπεζής της Ελλάδος διαπίστωσε μια ετήσια μείωση του τουριστικού συναλλάγματος της τάξεως του 4,5%. Αν το φετινό τουριστικό συναλλαγμα ανέρχονταν στο ύψος των 20500000 ευρώ, να υπολογιστεί σε πόσα χρόνια -με τον ίδιο ρυθμό μείωσης- θα πέσει κάτω από 9500000 ευρώ. Πόσο θα είναι τότε το συναλλαγμα; να εμφανίζουν τα έτη και το ύψος του συναλλάγματος όταν θα ξεπεράσει το δοσμένο όριο.

**6.** Σε ένα ταξιδιωτικό πρακτορείο γίνονται κάθε μέρα ορισμένες "κινήσεις" που αναφέρονται σε κρατήσεις θέσεων, για ένα ταξίδι συνολικού κόστους 15000 ευρώ. Το εισιτήριο για ένα άτομο στοιχίζει 900 ευρώ και προβλέπεται μια έκπτωση 6% κατά άτομο για ομάδες άνω των 15 ατόμων. Για κάθε κίνηση δίνονται: ο κωδικός κίνησης (1=κράτηση θέσεως, 2=τερματισμός καταχωρήσεων των κινήσεων) και ο αριθμός ατόμων. Να γραφεί πρόγραμμα που θα διαβάζει τον κωδικό κίνησης και ανάλογα με την τιμή του τον αριθμό των ατόμων και στη συνέχεια θα υπολογίζει και θα εμφανίζει το ποσό εισπράξεως από κάθε κράτηση και τέλος το καθαρό κέρδος ή τη ζημία του πρακτορείου για το ταξίδι αυτό.

**7.** Δίνονται N πεντάδες αριθμών. Σε κάθε πεντάδα αριθμών ο πρώτος είναι ο κωδικός, ο δεύτερος είναι το φύλο, 0 για άνδρα ή 1 για γυναίκα., ο τρίτος είναι το βάρος, ο τέταρτος το ύψος και ο πέμπτος η ηλικία ενός ατόμου. Να γραφεί πρόγραμμα που θα διαβάζει το N και στη συνέχεια θα διαβάζει τα στοιχεία της κάθε πεντάδας και θα βρίσκει τον ψηλότερο άνδρα και θα εμφανίζει τα παρακάτω στοιχεία του: κωδικός, φύλο, βάρος, ύψος, ηλικία.

**8.** Γράψτε ένα πρόγραμμα για το διάβασμα μιας σειράς δεδομένων ακεραίου τύπου (επίδοση αθλητών), για τον υπολογισμό και την εμφάνιση του πλήθους τους, του συνολικού τους αθροίσματος και του μέσου όρου τους. Το πρόγραμμα θα συνεχίσει να διαβάζει τις επιδόσεις μέχρι να δοθεί ως επίδοση ο αριθμός -1.

**9.** Να γίνει αλγόριθμος ο οποίος θα διαβάζει 100 αριθμούς και θα ελέγχει αν αποτελούν διαδοχικούς όρους αριθμητικής προόδου.

**10.** Χρησιμοποιώντας την πιο κάτω σειρά βημάτων, να βρείτε τον 10ο αριθμό της πιο κάτω σειράς: 1, 2, 4, 7, 11...

Βήματα:

1. Ο 1ος αριθμός είναι το 1

2. Ο 2ος αριθμός είναι το 2

3. Κάθε επόμενος αριθμός είναι το άθροισμα των δύο προηγούμενων. Αν ο αριθμός που προκύπτει διαιρείται ακριβώς με το 3 τότε προσθέτουμε ακόμα μια μονάδα στον αριθμό.

**11.** Σε μια κινηματογραφική αίθουσα υπάρχουν συνολικά 500 θέσεις. Για να κάνουν κρατήσεις θέσεων οι θεατές επικοινωνούν με την έκδοση εισιτηρίων και ρωτάνε αν υπάρχουν κενές θέσεις. Να δοθεί αλγόριθμος ο οποίος να διαβάζει τον αριθμό των θέσεων που θέλει κάθε φορά ένας θεατής και αν υπάρχουν διαθέσιμες θέσεις τότε να εμφανίζεται ο αριθμός των θέσεων που μένουν κενές μετά την κράτηση, ο συνολικός αριθμός κρατήσεων και η τελευταία κράτηση. Σε περίπτωση που ο αριθμός των θέσεων

που θέλει κάποιος θεατής είναι μεγαλύτερος από εκείνες που είναι ακόμα κενές, να εμφανίζεται μήνυμα που να δίνει το μέγιστο αριθμό θέσεων που μπορεί να κάνει κράτηση.

**12.** Ένας αριθμός λέγεται παλίνδρομος αν μπορεί να διαβαστεί το ίδιο από την αρχή και το τέλος του (π.χ. 2772, 6666). Να γραφεί ένα πρόγραμμα που θα υλοποιεί τα εξής:

Θα διαβάξει από το πληκτρολόγιο έναν ακέραιο θετικό τετραψήφιο αριθμό.

Θα ελέγχει αν ο αριθμός αυτός ανήκει στο σωστό διάστημα τιμών.

Αν ο αριθμός δεν ανήκει στο παραπάνω διάστημα θα εμφανίζεται ένα κατάλληλο μήνυμα.

Αν ο αριθμός ικανοποιεί τους περιορισμούς θα εμφανίζονται κατάλληλο μήνυμα αν ο αριθμός είναι παλίνδρομος ή όχι.

**13.** Να γράψετε πρόγραμμα για την εύρεση της τετραγωνικής ρίζας ενός αριθμού, σύμφωνα με τα παρακάτω βήματα:

1. Θα διαβάξει από το πληκτρολόγιο τον αριθμό  $x$  για τον οποία αναζητούμε την τετραγωνική ρίζα. Αν ο αριθμός είναι αρνητικός το πρόγραμμα θα σταματά.

2. Θα χρησιμοποιεί ως πρώτη τιμή για την τετραγωνική ρίζα το μισό του αριθμού  $x$ .

3. Θα χρησιμοποιεί μια δομή επανάληψης για να βρει τη ρίζα με διαδοχικές προσεγγίσεις με μια δοσμένη ακρίβεια  $\epsilon=10^{-6}$ . Η νέα τιμή σε κάθε επανάληψη θα υπολογίζεται από τον τύπο: νέα τιμή =  $0.5 * (x/\text{προηγούμενη τιμή} + \text{προηγούμενη τιμή})$ . Θα εμφανίζει σε κάθε βήμα τον αύξοντα αριθμό επανάληψης, την νέα τιμή, την αμέσως προηγούμενη τιμή και τη διαφορά αυτών των δύο τιμών.

**14.** Να γραφεί πρόγραμμα που θα βρίσκει και θα εμφανίζει σε ποιο έτος μια κατάθεση ενός ποσού  $A$  με ετήσιο επιτόκιο  $c\%$  διπλασιάζεται, με την υπόθεση ότι οι τόκοι προστίθενται κάθε έτος στο αρχικό κεφάλαιο. Το πρόγραμμα θα εμφανίζει Για κάθε έτος το αρχικό ποσό, τους τόκους και το τελικό ποσό (κεφάλαιο + τόκοι).

**15.** Να εμφανιστούν όλοι οι τριψήφιοι θετικοί ακέραιοι που το άθροισμα των τετραγώνων των ψηφίων τους ισούται με τον ίδιο τον αριθμό.

**16.** Αλγόριθμος (έξυπνος) υπολογισμού ακεραίων λύσεων της τριτοβάθμιας εξίσωσης στο  $(-100,200]$ .

**17.** Σε διαγωνισμό της Μαθηματικής Εταιρίας έλαβαν μέρος αρκετοί υποψήφιοι και βαθμολογήθηκαν σε ακέραια κλίμακα από το 1 ως το 100 (να γίνει έλεγχος). Στον επόμενο γύρο προκρίνονται όσοι υποψήφιοι έχουν βαθμολογία πάνω από 80 βαθμούς. Να γραφεί αλγόριθμος ο οποίος:

α) να διαβάξει τη βαθμολογία και τον κωδικό κάθε υποψηφίου, θα σταματάει αν δοθεί ως κωδικός το «000».

β) να υπολογίζει και να εμφανίζει πόσοι από τους υποψηφίους απορρίφθηκαν από τον επόμενο γύρο.

γ) να υπολογίζει και να εμφανίζει το μέσο όρο βαθμολογίας των υποψηφίων που προκρίθηκαν στον επόμενο γύρο.

δ) να υπολογίζει και να εμφανίζει το ποσοστό αυτών που προκρίθηκαν στον επόμενο γύρο.

**18.** Η συχνότητα δειγματοληψίας ενός μονοδιάστατου σήματος ήχου είναι  $f_s=1000$  και

άρα η περίοδος δειγματοληψίας  $T_s=1/f_s$ . Δειγματοληπτούμε το σήμα  $g_1=\cos(\delta \cdot \pi \cdot n)$  στο χρονικό διάστημα  $n=0:T_s:1$ . Να μοντελοποιηθεί-επιλυθεί αλγοριθμικά το πρόβλημα.

**19.** Στην ψηφιακή επεξεργασία ήχου και εικόνας χρησιμοποιούνται δύο βασικές συναρτήσεις:

$u(n)$  η οποία ονομάζεται συνάρτηση μοναδιαίου βήματος, η οποία για μη αρνητικές τιμές του ακεραίου  $n$  δίνει έξοδο 1, ενώ για αρνητικές δίνει έξοδο 0.

$\delta(n)$  η οποία ονομάζεται κρουστική συνάρτηση, η οποία δίνει έξοδο 1 για κάθε ακέραιο  $n$ , με εξαίρεση όταν το  $n$  ισούται με 0 τότε δίνει έξοδο 1.

Να γραφεί αλγόριθμος ο οποίος τις υλοποιεί.

**20.** Ένας εργάτης δουλεύει 5 μέρες την εβδομάδα. Γράψτε αλγόριθμο που:

διαβάζει τις ώρες που δούλεψε ο εργάτης κάθε μέρα και για χρονικό διάστημα των 20 εβδομάδων και υπολογίζει/εμφανίζει:

1. το μικρότερο ημερήσιο πλήθος ωρών εργασίας του εργάτη, τον αύξοντα αριθμό (1-20) της εβδομάδας και τον αύξοντα αριθμό της ημέρας (1-6) στις οποίες καταγράφηκε το παραπάνω γεγονός.

2. το μέσο όρο των ημερήσιων ωρών εργασίας του ανά εβδομάδα.

3. πόσες φορές δούλεψε ημερησίως περισσότερες από 9 ώρες.

**21.** Έστω ότι στα πλαίσια μιας ανθρωπιστικής αποστολής στην Αφρική, παρέχεται ιατρική και φαρμακευτική βοήθεια στους κατοίκους της από κάποια εθελοντική οργάνωση, ώστε να αντιμετωπιστεί η ηπατίτιδα. Η βοήθεια προβλέπει την αποστολή 10.000 εμβολίων τον πρώτο χρόνο (2015). Για τα επόμενα χρόνια, έως και το έτος 2022, η αποστολή εμβολίων θα γίνεται ανά διετία και η ποσότητα θα αυξάνεται κατά 2.000 εμβόλια σε κάθε αποστολή. Να γραφεί αλγόριθμος ο οποίος εμφανίζει το πλήθος των εμβολίων που παρέχονται όλα τα παραπάνω χρόνια.

**22.** Οι υπεύθυνοι από ένα κέντρο υποδοχής μεταναστών θέλουν να αναπτύξουν έναν αλγόριθμο που να έχει ως έξοδο την ηλικία του μεγαλύτερου άνδρα και της μεγαλύτερης γυναίκας που φιλοξενούν στις εγκαταστάσεις τους.

Για το σκοπό αυτό δημιουργήστε έναν αλγόριθμο ο οποίος:

1. Να εισάγει τον αριθμό των ανθρώπων που φιλοξενούνται στις εγκαταστάσεις του.

2. Για κάθε φιλοξενούμενο να εισάγει το φύλο του (Α για άντρας και Γ για γυναίκα) και την ηλικία του.

3. Να εμφανίζει το εύρος της ηλικίας των γυναικών (εύρος θεωρούμε τη διαφορά της μεγαλύτερης και μικρότερης ηλικίας). Θεωρήστε ότι ο χρήστης του προγράμματος θα δίνει τιμές για την ηλικία και το φύλο που θα είναι ρεαλιστικές και δε θα χρειάζονται έλεγχο από το δημιουργό του προγράμματος.

**23.** Να υλοποιηθεί αλγόριθμος για το εξής σενάριο. Έχουμε σκληρό δίσκο 500 GB από τα οποία τα 170 είναι ήδη δεσμευμένα από λογισμικά. Διαβάζεται το μέγεθος και τύπος-κατάληξη διαδοχικών αρχείων για download. Πόσα και με τι μέσο όρο μεγέθους. Τι ποσοστό είναι αρχεία doc.

**24.** Αρχίστε με έναν οποιοδήποτε ακέραιο θετικό αριθμό, έστω τον 25. Βρείτε το άθροισμα των κύβων των ψηφίων του 25: 133. Επαναλάβετε τη διαδικασία με τον 133 για να πάρετε τον 55. Συνεχίστε αθροίζοντας τους κύβους των ψηφίων των αριθμών για να δημιουργείτε νέους αριθμούς έως ότου βρείτε έναν αριθμό που να έχει ήδη

παρουσιαστεί στη λίστα των αριθμών που δημιουργούνται. Για παράδειγμα, αρχίζοντας με τον 25 η λίστα των αριθμών που δημιουργούνται είναι: 25 133 55 250 133. Η λίστα επαναλαμβάνει τον αριθμό 133 στη θέση 5.

Ζητείται να γράψετε ένα πρόγραμμα που να διαβάζει από την είσοδο έναν ακέραιο θετικό αριθμό μεταξύ 0 και 999999 και να δημιουργεί τη λίστα των αριθμών όπως περιγράφηκε παραπάνω. Εμφανίζει τη θέση στην οποία βρέθηκε ο πρώτος αριθμός που επαναλαμβάνεται για δεύτερη φορά μέσα στη λίστα.

**25.** Σε ένα θέατρο υπάρχουν 700 θέσεις. Να γραφεί πρόγραμμα το οποίο διαβάζει τις θέσεις που θέλει να κάνει κράτηση ένας πελάτης-θεατής ελέγχοντας αν υπάρχουν επαρκείς διαθέσιμες. Αν επαρκούν εμφανίζει κατάλληλο μήνυμα. Τερματίζεται η διαδικασία όταν μηδενιστούν οι θέσεις ή απαιτηθούν περισσότερες από τις κενές.

**26.** Τέλειος αριθμός ονομάζεται ο ακέραιος εκείνος του οποίου οι παράγοντες έχουν ως άθροισμα τον ίδιο τον αριθμό. Στους παράγοντες αυτούς συμπεριλαμβάνεται το 1 αλλά όχι ο ίδιος ο αριθμός. Για παράδειγμα τέλειοι αριθμοί είναι:

το 6 αφού  $6 = 1 \times 6$  ή  $2 \times 3$  και  $1 + 2 + 3 = 6$ .

το 28 αφού  $28 = 1 \times 28$ ,  $2 \times 14$  ή  $4 \times 7$  και  $1 + 2 + 14 + 4 + 7 = 28$ .

Να γράψετε ένα πρόγραμμα που θα λαμβάνει ως είσοδο ακέραιους αριθμούς και βρίσκει αν αυτοί είναι τέλειοι. Ο αριθμός 1 δεν θεωρείται τέλειος αριθμός.

**27.** Δίνεται ένας θετικός ακέραιος διψήφιος αριθμός XY με δύο ψηφία X και Y. Αντιστρέψτε τα ψηφία και δημιουργήστε τον αριθμό YX. Στη συνέχεια βρείτε τη θετική διαφορά μεταξύ XY και YX. Επαναλάβετε τη διαδικασία με το θετικό διψήφιο αριθμό που προέκυψε ως διαφορά και μέχρις ότου ως νέα διαφορά προκύψει μονοψήφιος αριθμός. Η λίστα των αριθμών που δημιουργούνται καλείται "λίστα προς το 9" γιατί καταλήγει πάντα στον αριθμό 9.

Γράψτε πρόγραμμα που

α) να ζητά διψήφιο θετικό ακέραιο αριθμό και να δημιουργεί τη λίστα του προς το 9. Σε περίπτωση που ο αριθμός έχει τα δύο ψηφία όμοια ( $A=B$ ) να ζητά νέο αριθμό.

β) να εμφανίζει τους αριθμούς της λίστας καθώς και το μήκος της όπως στο παράδειγμα. Λίστα του αριθμού 19 προς το 9: 19, 72, 45, 9. Μήκος λίστας: 4

γ) Να επαναλαμβάνει τα βήματα α και β μέχρι να δοθεί ο αριθμός 99.

**28.** Δίνεται η δομή επανάληψης

Για i από κ μέχρι λ με βήμα β

Εντολές

Τέλος\_Επανάληψης

Αν  $\beta > 0$ , να γράψετε ισοδύναμες εντολές, χρησιμοποιώντας αντί της εντολής Για την εντολή Αρχή\_επανάληψης . . . μέχρις ότου.

Ποιες αλλαγές θα κάνετε στις εντολές που δώσατε παραπάνω, αν  $\beta < 0$ ;

**29.** Να γράψετε πρόγραμμα για κάθε μία από τις παρακάτω περιπτώσεις:

a. Ο σημερινός αριθμός αυτοκινήτων που κυκλοφορούν σε μια πόλη είναι 80000. Αν ο αριθμός αυτός αυξάνεται με ρυθμό 7%, να γραφεί πρόγραμμα που να υπολογίζει σε πόσα χρόνια ο αριθμός των αυτοκινήτων θα ξεπεράσει τις 160000. Πόσα θα είναι τότε τα αυτοκίνητα;

b. Γενικεύστε το πρόγραμμα για Αριθμο\_Αυτοκ αυτοκίνητα, που αυξάνονται με ετήσιο ρυθμό Ρυθμος και ξεπερνούν ένα δοσμένο όριο Οριο.



Τα προγράμματα a και b να εμφανίζουν τα έτη και τον αριθμό των αυτοκινήτων όταν θα ξεπεράσει το δοσμένο όριο. Για τον υπολογισμό του ετήσιου αριθμού αυτοκινήτων να χρησιμοποιηθεί ο τύπος: Αριθμός αυτοκινήτων = (Αριθμός αυτοκινήτων  $\cdot$  (1+Ρυθμός αύξησης)).

**30.** Μια βιομηχανία κρασιού αποστέλλει κρασιά (στο εξωτερικό και στο εσωτερικό). Κάθε αποστολή κρασιών περιλαμβάνει έναν αριθμό κιβωτίων που προορίζονται για το εσωτερικό κι έναν αριθμό κιβωτίων που προορίζονται για το εξωτερικό. Κάθε κιβώτιο που αποστέλλεται έχει ταχυδρομικά τέλη 5 ευρώ για το εσωτερικό και 7 ευρώ για το εξωτερικό. Στη διάρκεια μιας μέρας μπορούν να πραγματοποιηθούν πολλές αποστολές. Να γράψετε ένα πρόγραμμα το οποίο για κάθε αποστολή (στην οποία περιλαμβάνεται ένας γνωστός αριθμός κιβωτίων για το εσωτερικό και το εξωτερικό) να υπολογίζει το απαιτούμενο ποσό για ταχυδρομικά τέλη εσωτερικού και εξωτερικού καθώς και το άθροισμά τους. Για κάθε αποστολή είναι γνωστός ο αριθμός κιβωτίων εσωτερικού και ο αριθμός κιβωτίων εξωτερικού. Το πρόγραμμα στο τέλος της ημέρας υπολογίζει και τα συνολικά ποσά εσωτερικού κι εξωτερικού και το γενικό σύνολο για όλες τις αποστολές. Η εισαγωγή στοιχείων να πραγματοποιείται έως ότου τερματισθούν οι αποστολές (π.χ. εισάγοντας ένα πλασματικό πλήθος κιβωτίων εσωτερικού: -9999).

**31.** Σε ένα ταξιδιωτικό πρακτορείο γίνονται κάθε μέρα ορισμένες "κινήσεις" που αναφέρονται σε κρατήσεις θέσεων, για ένα ταξίδι συνολικού κόστους 30000 ευρώ. Το εισιτήριο για ένα άτομο στοιχίζει 500 ευρώ και προβλέπεται μια έκπτωση 8% κατά άτομο για ομάδες άνω των 15 ατόμων. Για κάθε κίνηση δίνονται:

ο κωδικός κίνησης (1=κράτηση θέσεως, 2=τερματισμός καταχωρήσεων των κινήσεων και ο αριθμός ατόμων (άνω των 10 θεωρούνται ομάδα).

Να γραφεί πρόγραμμα που θα διαβάζει τον κωδικό κίνησης και ανάλογα με την τιμή του τον αριθμό των ατόμων και στη συνέχεια θα υπολογίζει και θα εμφανίζει το ποσό εισπράξεως από κάθε κράτηση και τέλος το καθαρό κέρδος ή τη ζημία του πρακτορείου για το ταξίδι αυτό.

**32.** Γράψτε ένα πρόγραμμα για την εμφάνιση του παρακάτω σχήματος:

```
*
***
****
*****
*****
*****
*****
****
***
*
```

**33.** Ένα ασανσέρ έχει όρια 300 κιλά και πλήθος ατόμων 4. Να γραφεί αλγόριθμος ο οποίος εμφανίζει πόσα άτομα θα επιβιβαστούν και με τι συνολικό βάρος αφού διαβάσει το βάρος κάθε ατόμου που εισέρχεται.

**34.** Να γράψετε πρόγραμμα το οποίο διαβάζει ακέραιο και ελέγχει αν το άθροισμα των ψηφίων του ισούται με το γινόμενο των ψηφίων του, εμφανίζοντας κατάλληλο μήνυμα.

**35.** AN  $\chi > 0$  Ή ( $\psi = 6$  και  $\delta < 10$ ) ΤΟΤΕ

**ΕΜΦΑΝΙΣΕ** “άσπρο”

**ΑΛΛΙΩΣ**

**ΕΜΦΑΝΙΣΕ** “μαύρο”

**ΤΕΛΟΣ\_ΑΝ**

Γράψτε την δομή επιλογής σε μία ισοδύναμη μορφή, χρησιμοποιώντας όσες δομές επιλογής κρίνετε απαραίτητο, αλλά χωρίς να χρησιμοποιήσετε κανένα λογικό τελεστή.

**36.** Δίνονται τα παρακάτω τμήματα αλγορίθμου σε φυσική γλώσσα.

- i. Αν η τιμή της μεταβλητής  $\alpha$  είναι άρτιος αριθμός, να μειώνεται κατά 3 μονάδες.
- ii. Όσο η τιμή της μεταβλητής  $\alpha$  είναι μεγαλύτερη από την τιμή 2, να μειώνεται κατά μία μονάδα.
- iii. Αν η μεταβλητή ΓΡΑΜΜΑ έχει τιμή «α» ή «ε» ή «η» ή «ι» ή «ο» ή «υ» ή «ω», να εκχωρείται στη μεταβλητή ΑΠΟΤΕΛΕΣΜΑ η τιμή «Φωνήεν» και σε κάθε άλλη περίπτωση να εκχωρείται η τιμή «Σύμφωνο».

Να γράψετε το αντίστοιχο τμήμα αλγορίθμου σε ψευδογλώσσα.

**37.** Σε ένα πειραματικό σχολείο ο τελικός βαθμός κάθε μαθητή είναι ίσος με το μέσο βαθμό του αν όλοι οι βαθμοί του είναι μεγαλύτεροι του 7. Διαφορετικά, αν έστω κι ένας βαθμός του είναι μικρότερος ή ίσος του 7, τότε ο τελικός βαθμός του είναι ο μέσος βαθμός του μειωμένος κατά 2,5 μονάδες. Να γίνει πρόγραμμα το οποίο:

A) Για κάθε μαθητή

i) Διαβάζει το όνομά του και τους 6 βαθμούς του

iii) Υπολογίζει το τελικό βαθμό του σύμφωνα με την παραπάνω διαδικασία

iv) Τυπώνει το όνομα του μαθητή και τον τελικό του βαθμό

Το πρόγραμμα τερματίζει όταν για όνομα μαθητή δοθεί η λέξη 'όχι'.

B) Υπολογίζει και τυπώνει το ποσοστό των μαθητών με τελικό βαθμό μεγαλύτερο του 16.

**38.** Να δοθεί πρόγραμμα που να υπολογίζει τον αριθμό των careta careta στον πλανήτη σε δέκα χρόνια. Δίνεται ότι ο σημερινός πληθυσμός τους είναι 1000 και αυξάνονται ετησίως κατά 1%.

**39.** Μια μυρμηγκοφωλιά έχει 1000 σπόρους κριθάρι. Ένα μαύρο μυρμηγκί βάζει X σπόρους κριθάρι ημερησίως, ενώ ένα κόκκινο βγάζει Y σπόρους αντίστοιχα. Να γραφεί αλγόριθμος ο οποίος αφού διαβάσει τα X και Y (ελέγχοντας  $X > Y$ ) να υπολογίζει-εμφανίζει ανά ημέρα το πλήθος των κόκκων στη φωλιά. Τέλος, εμφανίζει σε πόσες μέρες η φωλιά θα έχει τουλάχιστον 2000 σπόρους κριθάρι.

**40.** Γράψτε τμήματα αλγορίθμων που θα βρίσκουν το μέσο όρο N ακεραίων:

α) Για  $N = 30$

β) Για άγνωστο N, όπου η εισαγωγή αριθμών θα τερματίζεται όταν δοθεί το -1, το οποίο δεν θα λαμβάνεται υπόψη στους υπολογισμούς.

γ) Όταν το N δίνεται από το χρήστη, αλλά κάθε αριθμός που δίνεται να είναι μικρότερος του προηγούμενου.

**41.** Ένας καταναλωτής έχει ένα δάνειο με αρχικό ποσό A. Το πιστωτικό του όριο είναι 7000 €. Κάθε μήνα το ποσό που χρωστά τοκίζεται κατά 4%, ενώ αν έχει ξεπεράσει το πιστωτικό όριο, το επιπλέον ποσό τοκίζεται με 9%. Η ελάχιστη δόση κάθε μήνα είναι το 1/50 του συνολικού ποσού που χρωστάει. Γράψτε αλγόριθμο που:

α) Θα διαβάζει το αρχικό ποσό A.

β) Κάθε μήνα θα του εμφανίζει το ποσό που χρωστάει και θα δέχεται το ποσό που πληρώνει ο καταναλωτής. Αν υπάρχει υπέρβαση πιστωτικού ορίου, θα του εμφανίζει σχετικό μήνυμα. Η επανάληψη θα ολοκληρώνεται όταν εισαχθούν 24 δόσεις ή αν το ποσό εξοφληθεί. Στο τέλος να εμφανίζεται το τελικό υπόλοιπο, αν υπάρχει.

**42.** Αναπτύξτε αλγόριθμο που να διαβάζει έναν ακέραιο αριθμό και να υπολογίζει ποιο ή ποια ψηφία του εμφανίζονται πιο πολλές φορές.

**43.** Το παρακάτω πρόγραμμα διαβάζει τη θερμοκρασία και τη σχετική υγρασία της ημέρας και εμφανίζει ένα χαρακτηρισμό για τον καιρό σύμφωνα με τον ακόλουθο πίνακα. Να συμπληρωθούν τα κενά.

Θερμοκρασία	Σχετική Υγρασία	Χαρακτηρισμός
$\leq 10^\circ \text{C}$	$\leq 60$	Κρύος και ξηρός
$\leq 10^\circ \text{C}$	$> 60$	Κρύος και υγρός
$> 10^\circ \text{C}$	$\leq 60$	Ζεστός και ξηρός
$> 10^\circ \text{C}$	$> 60$	Ζεστός και υγρός

#### ΠΡΟΓΡΑΜΜΑ ΕΜΥ

#### ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: .....

#### ΑΡΧΗ

ΓΡΑΨΕ ' Δώσε την τιμή της θερμοκρασίας'

ΔΙΑΒΑΣΕ υγρ .....

ΑΝ ..... ΚΑΙ .....ΤΟΤΕ

ΓΡΑΨΕ ' Ο καιρός είναι κρύος και ξηρός'

ΑΛΛΙΩΣ\_ΑΝ.....ΚΑΙ υγρ>60 ΤΟΤΕ

.....

ΑΛΛΙΩΣ\_ΑΝ .....

.....

ΑΛΛΙΩΣ\_ΑΝ .....

.....

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

**44.** Σε ένα στάδιο υπάρχουν 33 σειρές καθισμάτων. Στην πρώτη σειρά υπάρχουν 400 καθίσματα και σε κάθε επόμενη σειρά τα καθίσματα αυξάνονται κατά 10. Να γράψετε πρόγραμμα στη γλώσσα με το όνομα askisi45, το οποίο να υπολογίζει και να τυπώνει το συνολικό αριθμό καθισμάτων που υπάρχουν στο στάδιο.

**45.** Σε ένα διαγωνισμό υπάρχουν συνολικά 100 διαθέσιμες συμμετοχές σε τρεις κατηγορίες: K1 για άντρες, K2 για γυναίκες και K3 για παιδιά κάτω από 17 χρονών. Οι υποψήφιοι για συμμετοχή δηλώνουν την ηλικία (σε ακέραιο αριθμό), το όνομα και το φύλο τους (M άντρες, F γυναίκες). Να σχεδιάσετε λογικό διάγραμμα και ακολούθως να το μετατρέψετε στο αντίστοιχο πρόγραμμα στη γλώσσα με το όνομα askisi47, το οποίο να:

(α) δέχεται τα δεδομένα για τον κάθε υποψήφιο,

(β) τερματίζει την είσοδο δεδομένων όταν συμπληρωθεί ο μέγιστος αριθμός συμμετοχών ή όταν εισαχθεί ηλικία ίση με μηδέν,  
(γ) βρίσκει την κατηγορία για τον κάθε συμμετέχοντα και υπολογίζει και τυπώνει το πλήθος των συμμετεχόντων σε κάθε κατηγορία,  
(δ) τυπώνει το όνομα και την ηλικία του νεαρότερου συμμετέχοντα στην κατηγορία K3. Σε περίπτωση που δεν υπάρχουν άτομα στην κατηγορία αυτή να τυπώνει το κατάλληλο μήνυμα. Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων.

**46.** Να υλοποιηθεί αλγόριθμος για το ακόλουθο σενάριο. Κάποια χώρα θέλει να εφαρμόσει σύστημα καταγραφής τροχαίων παραβάσεων με κάμερες στα φώτα τροχαίας. Μέρος του συστήματος περιλαμβάνει τα εξής:

Μετά από κάθε φωτογράφιση παράβασης, καταχωρούνται στο αρχείο καταγγελιών τα στοιχεία της παράβασης (ο αριθμός κυκλοφορίας του μηχανοκινήτου, ο τόπος, η ημερομηνία και ώρα της παράβασης, η φωτογραφία, το προβλεπόμενο πρόστιμο και οι βαθμοί ποινής) καθώς και ο αριθμός καταγγελίας, ο οποίος δημιουργείται αυτόματα από το σύστημα. Ακολουθώντας, με βάση τον αριθμό κυκλοφορίας, γίνεται αναζήτηση των προσωπικών στοιχείων του ιδιοκτήτη του μηχανοκινήτου (ονοματεπώνυμο, αρ. ταυτότητας, διεύθυνση) από το αρχείο μηχανοκινήτων. Στην συνέχεια τυπώνεται έντυπο στο οποίο καταγράφονται ο αριθμός της καταγγελίας, η ημερομηνία, η ώρα, τα στοιχεία του ιδιοκτήτη, το είδος της παράβασης, το πρόστιμο και οι βαθμοί ποινής. Το έντυπο αυτό αποστέλλεται στον ιδιοκτήτη.

**47.** Στα πλαίσια της περιβαλλοντικής πολιτικής του Υπουργείου Παιδείας, οργανώθηκε εκστρατεία συλλογής ανακυκλώσιμων ειδών σε αριθμό σχολείων της χώρας. Στα πλαίσια της εκστρατείας, εθελοντές μαθητές από το κάθε σχολείο μάζεψαν ποσότητες από χαρτί και αλουμίνιο. Να σχεδιάσετε λογικό διάγραμμα και να γράψετε το αντίστοιχο πρόγραμμα στη γλώσσα με το όνομα askisi2, το οποίο να:

(α) ζητά από το χρήστη το όνομα του κάθε σχολείου και το βάρος (σε κιλά) κάθε υλικού ξεχωριστά. Να θεωρήσετε ότι καταχωρούνται στοιχεία για τουλάχιστον ένα σχολείο. Η επανάληψη να τερματίζεται όταν ο χρήστης απαντήσει με «ΟΧΙ» στην ερώτηση «Υπάρχουν άλλα σχολεία; (ΝΑΙ/ΟΧΙ)».

(β) υπολογίζει και τυπώνει το συνολικό βάρος του αλουμινίου που μαζεύτηκε από όλα τα σχολεία.

(γ) βρίσκει και τυπώνει το όνομα του σχολείου που μάζεψε το περισσότερο χαρτί (σε περίπτωση ύπαρξης δύο ή περισσότερων τέτοιων σχολείων να τυπώνεται μόνο ένα από αυτά).

(δ) υπολογίζει και τυπώνει το μέσο όρο βάρους του αλουμινίου που μαζεύτηκε από όλα τα σχολεία. Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων.

**48.** Να γράψετε πρόγραμμα με το όνομα askisi8, το οποίο να διαβάσει τυχαίους ακέραιους αριθμούς που δίνονται από το χρήστη. Η επανάληψη να τερματίζει όταν δοθεί αρνητικός αριθμός ή το μηδέν. Για καθένα από τους αριθμούς το πρόγραμμα να ελέγχει αν είναι διψήφιος. Σε περίπτωση που είναι διψήφιος να εκτυπώνει σε διαφορετικές γραμμές το ψηφίο των δεκάδων και το ψηφίο των μονάδων, με τα κατάλληλα μηνύματα. Στο τέλος, το πρόγραμμα να τυπώνει το άθροισμα των ψηφίων των μονάδων όλων των διψήφιων αριθμών με τα κατάλληλα μηνύματα.

**49.** Ένας καταστηματάρχης αποφάσισε να πωλεί τα προϊόντα του καταστήματός του αποκόπτοντας το δεκαδικό μέρος της αρχικής τιμής τους. Για παράδειγμα, αν η αρχική τιμή του προϊόντος είναι 10.74€, τότε η τελική τιμή του προϊόντος υπολογίζεται σε 10€. Να σχεδιάσετε λογικό διάγραμμα και να γράψετε το αντίστοιχο πρόγραμμα στη γλώσσα με το όνομα askisi50, το οποίο:

(α) Να διαβάζει από το πληκτρολόγιο το όνομα και την αρχική τιμή κάθε προϊόντος. Για τερματισμό θα δίνεται αρνητική τιμή προϊόντος.

(β) Να υπολογίζει και να τυπώνει την τελική τιμή του κάθε προϊόντος.

(γ) Να υπολογίζει και να τυπώνει το πλήθος των προϊόντων που είχαν ζημιά καθώς και τη συνολική ζημιά των προϊόντων. Η ζημιά για κάθε προϊόν προκύπτει αφαιρώντας την τελική τιμή από την αρχική τιμή του προϊόντος.

(δ) Να βρίσκει και να τυπώνει το όνομα του προϊόντος με τη μεγαλύτερη ζημιά. Αν δεν υπάρχει προϊόν με ζημιά, τότε να τυπώνει το μήνυμα «ΔΕΝ ΥΠΑΡΧΕΙ ΠΡΟΪΟΝ ΜΕ ΖΗΜΙΑ». Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων.

**50.** Να γράψετε πρόγραμμα στη γλώσσα με όνομα askisi51, το οποίο να δέχεται δύο θετικούς ακέραιους αριθμούς A και B. Θεωρείστε ότι  $A < B$ . Αν σε κάθε επανάληψη ο A διπλασιάζεται και ο B μειώνεται κατά 5, τότε το πρόγραμμα να υπολογίζει και να τυπώνει τον ελάχιστο αριθμό των επαναλήψεων που χρειάζονται έτσι ώστε η τιμή του B να γίνει μικρότερη της τιμής του A.

**51.** Να γραφεί πρόγραμμα στη γλώσσα το οποίο δέχεται έναν τριψήφιο αριθμό. Αν το ψηφίο των εκατοντάδων είναι 1, τότε να παρουσιάζεται το μήνυμα «Ο μαθητής δίνει εξετάσεις για πρόσβαση», αν είναι 2 τότε να παρουσιάζεται το μήνυμα «Ο μαθητής δίνει εξετάσεις για απόλυση» και αν είναι 3 τότε να παρουσιάζεται το μήνυμα «Ο μαθητής παραπέμπεται το Σεπτέμβριο». Αν το ψηφίο είναι εκτός ορίων, τότε να παρουσιάζεται το μήνυμα «Εκτός Ορίου».

**52.** Να γράψετε πρόγραμμα στη γλώσσα με το όνομα askisi8, το οποίο να διαβάζει ένα πενταψήφιο αριθμό. Ο αριθμός αυτός δηλώνει μια ημερομηνία (μέρα, μήνας) και κατά πόσο είναι πρωί ή βράδυ. Συγκεκριμένα, στον πενταψήφιο αριθμό xxyz, τα ψηφία xx αντιπροσωπεύουν τη μέρα, τα ψηφία yy το μήνα και το ψηφίο z καθορίζει αν είναι πρωί (έχει τιμή μηδέν) ή βράδυ (έχει τιμή ένα). Με βάση τον αριθμό που δόθηκε, το πρόγραμμα να τυπώνει τη μέρα, το όνομα του μήνα, και αν είναι πρωί ή βράδυ. Σημειώστε ότι η μέρα που θα τυπώνεται να είναι πάντα διψήφιος αριθμός. Για παράδειγμα, η είσοδος του πενταψήφιου αριθμού 23030 να έχει ως αποτέλεσμα την εκτύπωση «23 Μαρτίου πρωί» και του αριθμού 02041 να έχει ως αποτέλεσμα την εκτύπωση «02 Απριλίου βράδυ». Να θεωρήσετε ότι ο χρήστης δίνει έναν πενταψήφιο αριθμό όπως απαιτεί η άσκηση και άρα δεν χρειάζεται έλεγχος.

**53.** Μια αυτοκινητοβιομηχανία αποφάσισε να αποσύρει, λόγω κάποιου σοβαρού κατασκευαστικού προβλήματος, 2 μοντέλα αυτοκινήτων M1 και M2. Τα μοντέλα αυτά διατέθηκαν στην αγορά τα 4 τελευταία χρόνια (2007 - 2010) και οι πωλήσεις τους ανήλθαν συνολικά στις 10000 αυτοκίνητα. Για κάθε αυτοκίνητο που αποσύρεται (δηλαδή επιστρέφεται στην αυτοκινητοβιομηχανία από τον ιδιοκτήτη), η αυτοκινητοβιομηχανία θα αποζημιώνει τον ιδιοκτήτη ως εξής:

Ετος πώλησης	Αποζημίωση
--------------	------------

2007	50%
2008	70%
2009	90%
2010	100%

**54.** Να σχεδιάσετε λογικό διάγραμμα και ακολούθως να το μετατρέψετε στο αντίστοιχο πρόγραμμα με το όνομα askisi54, το οποίο:

α) για το κάθε αυτοκίνητο που αποσύρεται, να δέχεται το μοντέλο (M1 ή M2), το έτος που πωλήθηκε (2010-2013) και την τιμή στην οποία πωλήθηκε. Να γίνεται έλεγχος μόνο για το έτος πώλησης του αυτοκινήτου και σε περίπτωση που αυτό δεν είναι στα αποδεκτά πλαίσια, να παρουσιάζεται στην οθόνη το μήνυμα «Λάθος Έτος» και να ζητείται ξανά το έτος. Να θεωρήσετε ότι το μοντέλο και η τιμή πώλησης του αυτοκινήτου δίνονται ορθά και δε χρειάζεται έλεγχος.

β) να τερματίζει την είσοδο δεδομένων όταν στο ερώτημα του προγράμματος «Συνεχίζεται η επιστροφή αυτοκινήτων (ΝΑΙ ή ΟΧΙ)» δοθεί η απάντηση «ΟΧΙ» ή αν επιστραφούν και τα 10000 αυτοκίνητα.

γ) να υπολογίζει και τυπώνει τα πιο κάτω:

i. Το συνολικό αριθμό αυτοκινήτων από κάθε μοντέλο που αποσύρθηκε.

ii. Το συνολικό ποσό που θα πληρώσει η εταιρεία για αποζημιώσεις.

iii. Τον αριθμό αυτοκινήτων μοντέλου M1 που αποσύρθηκαν το έτος 2011, καθώς και το ποσό της αποζημίωσης που θα πληρώσει η εταιρεία για αυτά τα αυτοκίνητα. Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων.

**55.** Να γράψετε πρόγραμμα με το όνομα askisi7, το οποίο να διαβάσει μέχρι πενήντα τυχαίους ακέραιους αριθμούς. Η εισαγωγή των αριθμών να τερματίζεται μόλις δοθεί αριθμός που έχει ξαναδοθεί προηγουμένως. Το πρόγραμμα να τυπώνει το άθροισμα των αριθμών που έχουν καταχωρηθεί χωρίς να λαμβάνει υπόψη τον τελευταίο αριθμό (π.χ. αν καταχωρούνται οι αριθμοί 2, 3, 6, 5, 4, 7, 6 θα εμφανίζεται άθροισμα 27).

**56.** Ένας πελάτης για να χρησιμοποιήσει μια Αυτόματη Ταμειακή Μηχανή (ATM) μιας τράπεζας πρέπει να πληκτρολογήσει τον προσωπικό κωδικό αριθμό PIN (ακέραιο αριθμό). Ο πελάτης έχει τη δυνατότητα να δώσει τον κωδικό 3 φορές μόνο. Σε περίπτωση που δοθεί σωστός κωδικός τότε παρουσιάζεται το μήνυμα «Επιτυχής Πρόσβαση». Διαφορετικά για κάθε φορά που ο κωδικός είναι λάθος, παρουσιάζεται το μήνυμα «Λάθος κωδικός» και ξαναζητείται.

**57.** Στα πλαίσια της πολιτικής εξοικονόμησης νερού, το Υπουργείο Γεωργίας χρειάζεται στοιχεία από κάθε Δήμο της Κύπρου. Να γράψετε πρόγραμμα στη γλώσσα προγραμματισμού ΓΛΩΣΣΑ με όνομα askisi57, για να εκτελούνται τα πιο κάτω:

(α) να ζητείται από το χρήστη το όνομα του Δήμου, η επαρχία στην οποία ανήκει (με κωδικό Λευκωσία - 1, Λεμεσός - 2, Λάρνακα - 3, Λεμεσός - 4, Αμμόχωστος - 5 και Πάφος - 6) και τους τόνους νερού που καταναλώθηκαν για κάθε Δήμο για ένα έτος. Σε περίπτωση καταχώρησης κωδικού επαρχίας εκτός των πιο πάνω ορίων, να επιβάλλεται επανάληψη της καταχώρησης με μήνυμα «Επαρχία 1 - 6 μόνο».

(Να θεωρήσετε ότι καταχωρούνται στοιχεία για τουλάχιστον ένα Δήμο). Η επανάληψη να τερματίζεται όταν ο χρήστης απαντήσει με «ΟΧΙ» στην ερώτηση «Υπάρχουν άλλα

στοιχεία; (ΝΑΙ/ΟΧΙ)».

(β) να υπολογίζει και να εμφανίζει τη συνολική κατανάλωση νερού όλων των Δήμων.

(γ) να εντοπίζει και να εμφανίζει τον Δήμο της επαρχίας Λευκωσίας με τη μεγαλύτερη κατανάλωση.

(δ) να βρίσκει και να εμφανίζει την επαρχία με τη μεγαλύτερη κατανάλωση μεταξύ Λευκωσίας και Λεμεσού. (Θεωρήστε ότι οι δύο επαρχίες δεν έχουν την ίδια κατανάλωση).

**58.** Δίνεται η παρακάτω ακολουθία εντολών:

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

A ← 10

**Για i ΑΠΟ 1 ΜΕΧΡΙ 3**

        A ← A - 10

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΜΕΧΡΙΣ\_ΟΤΟΥ** A = 0

Να αναφέρετε ποιο κριτήριο αλγορίθμου δεν ικανοποιείται και να δικαιολογήσετε την απάντησή σας.

**59.** Μια εταιρεία δημοσκοπήσεων θέτει σ' ένα δείγμα 2000 πολιτών ένα ερώτημα. Για την επεξεργασία των δεδομένων να αναπτύξετε αλγόριθμο που:

1. να διαβάσει το φύλο του πολίτη (Α=Ανδρας, Γ=Γυναίκα) και να ελέγχει την ορθή εισαγωγή

2. να διαβάσει την απάντηση στο ερώτημα, η οποία μπορεί να είναι «ΝΑΙ», «ΟΧΙ», «ΔΕΝ ΞΕΡΩ» και να ελέγχει την ορθή εισαγωγή

3. να υπολογίζει και να εμφανίζει το πλήθος των ατόμων που απάντησαν «ΝΑΙ»

4. στο σύνολο των ατόμων που απάντησαν «ΝΑΙ», να υπολογίζει και να εμφανίζει το ποσοστό των ανδρών και το ποσοστό των γυναικών.

**60.** Το παρακάτω τμήμα προγράμματος να μετατραπεί σε ισοδύναμο, χρησιμοποιώντας αποκλειστικά τη δομή επανάληψης Όσο... ΕΠΑΝΑΛΑΒΕ.

S ← 0

**ΓΙΑ** K **από** 1 **μέχρι** 5

**ΓΙΑ** L **από** 1 **μέχρι** 7

        S ← S + 1

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** S

**61.** Δίνεται το παρακάτω τμήμα αλγορίθμου:

α ← 5

β ← 3

**Για** X **από** 2 **μέχρι** 7 **με\_βήμα** 4

**Όσο** α < = 10 **επανάλαβε**

        β ← β + α

        α ← α + 4

**Τέλος\_επανάληψης**

**Εμφάνισε** α, β

    α ← 4

**Τέλος\_επανάληψης**

**Εμφάνισε** α

Να γράψετε στο τετράδιό σας τις τιμές που εμφανίζονται κατά την εκτέλεση του παραπάνω τμήματος αλγορίθμου.

**62.** Δίνεται το παρακάτω τμήμα αλγορίθμου:

$X \leftarrow 0$

$A \leftarrow 10$

$B \leftarrow 14$

**Όσο**  $B > 0$  **ΕΠΑΝΑΛΑΒΕ**

**Αν**  $B \text{ MOD } 2 = 0$  **ΤΟΤΕ**

$X \leftarrow X + A$

**ΤΕΛΟΣ\_ΑΝ**

$A \leftarrow A * 2$

$B \leftarrow B \text{ DIV } 2$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**α.** Να γράψετε στο τετράδιό σας τις τιμές των μεταβλητών  $X$ ,  $A$  και  $B$  στο τέλος κάθε επανάληψης κατά την εκτέλεση του παραπάνω τμήματος αλγορίθμου.

**β.** Να σχεδιάσετε το αντίστοιχο διάγραμμα ροής.

**63.** Να ξαναγράψετε στο τετράδιό σας καθένα από τα παρακάτω τμήματα αλγορίθμου, χρησιμοποιώντας μόνο μία δομή επανάληψης Για ... από .... μέχρι και χωρίς τη χρήση δομής επιλογής:

$i \leftarrow 1$  (1)

$j \leftarrow 1$

**Αρχή\_επανάληψης**

**Εμφάνισε**  $A[i,j]$

$i \leftarrow i + 1$

$j \leftarrow j + 1$

**Μέχρις\_ότου**  $j > 10$

**Για**  $i$  **από** 1 **μέχρι** 100 (2)

**Για**  $j$  **από** 1 **μέχρι** 100

**Αν**  $i = 50$  **τότε**

! 2 επίπεδα εμφώλευσης

**Εμφάνισε**  $A[i,j]$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**64.** Δίνεται το παρακάτω τμήμα αλγορίθμου:

**Για**  $X$  **από**  $A$  **μέχρι**  $M$  **με\_βήμα**  $B$

**Εμφάνισε**  $X$

**Τέλος\_επανάληψης**

Να γράψετε στο τετράδιό σας για καθένα από τις παρακάτω περιπτώσεις τις τιμές των  $A$ ,  $M$ ,  $B$ , έτσι ώστε το αντίστοιχο τμήμα αλγορίθμου να εμφανίζει όλους:

1. τους ακέραιους από 1 μέχρι και 100
2. τους ακέραιους από 10 μέχρι και 200 σε φθίνουσα σειρά
3. τους ακέραιους από -1 μέχρι και -200 σε αύξουσα σειρά
4. τους άρτιους ακέραιους από 100 μέχρι και 200
5. τους θετικούς ακέραιους που είναι μικρότεροι του 8128.



**65.** Εστω το τμήμα του αλγορίθμου με μεταβλητές A, B, C, D, X και Y.

$D \leftarrow 2$

**Για X από 2 μέχρι 5 με\_βήμα 2**

$A \leftarrow 10 * X$

$B \leftarrow 5 * X + 10$

$C \leftarrow A + B - (5 * X)$

$D \leftarrow 3 * D - 5$

$Y \leftarrow A + B - C + D$

**Τέλος\_επανάληψης**

Να βρείτε τις τιμές των μεταβλητών A, B, C, D, X και Y σε όλες τις επαναλήψεις.

**66.** Δίνεται η δομή επανάληψης.

**Για i από τιμή1 μέχρι τιμή2 με βήμα β**

Εντολές

Τέλος\_επανάληψης

Να μετατρέψετε την παραπάνω δομή σε ισοδύναμη δομή επανάληψης Όσο ... επανάλαβε.

**67.** Δίνεται το παρακάτω τμήμα αλγορίθμου.

$X \leftarrow 1$

**Όσο  $X < 5$  επανάλαβε**

$A \leftarrow X + 2$

$B \leftarrow 3 * A - 4$

$C \leftarrow B - A + 4$

**Αν  $A > B$  τότε**

**Αν  $A > C$  τότε**

$MAX \leftarrow A$

αλλιώς

$MAX \leftarrow C$

Τέλος\_Αν

**αλλιώς**

**Αν  $B > C$  τότε**

$MAX \leftarrow B$

αλλιώς

$MAX \leftarrow C$

Τέλος\_Αν

Τέλος\_Όσο

**Εμφάνισε X, A, B, C, MAX**

$X \leftarrow X + 2$

**Τέλος επανάληψης**

Ποιες είναι οι τιμές των μεταβλητών X, A, B, C, MAX που θα εμφανιστούν κατά την εκτέλεση του παραπάνω τμήματος αλγορίθμου;

**68.** Να γράψετε στο τετράδιό σας τις τιμές των μεταβλητών N, M και B, όπως αυτές τυπώνονται σε κάθε επανάληψη, και την τιμή της μεταβλητής X που τυπώνεται μετά το τέλος της επανάληψης, κατά την εκτέλεση του παρακάτω αλγορίθμου.

**Αλγόριθμος Αριθμοί**

$A \leftarrow 1$

$B \leftarrow 1$

$N \leftarrow 0$

$M \leftarrow 2$

**Όσο**  $B < 6$  **επανάλαβε**

$X \leftarrow A + B$

**Αν**  $X \text{ MOD } 2 = 0$  **τότε**

$N \leftarrow N + 1$

αλλιώς

$M \leftarrow M + 1$

**Τέλος\_Αν**

$A \leftarrow B$

$B \leftarrow X$

**Εμφάνισε**  $N, M, B$

**Τέλος\_επανάληψης**

**Εμφάνισε**  $X$

**Τέλος** Αριθμοί

**69.** Δίνεται το παρακάτω τμήμα αλγορίθμου:

$S \leftarrow 0$

**Για**  $I$  **από** 2 **μέχρι** 100 **με\_βήμα** 2

$S \leftarrow S + I$

**Τέλος\_επανάληψης**

**Εμφάνισε**  $S$

1. Να μετατραπεί σε ισοδύναμο με χρήση της δομής Όσο ... επανάλαβε.

2. Να μετατραπεί σε ισοδύναμο με χρήση της δομής Αρχή\_επανάληψης... Μέχρις\_ότου.

**70.** Να γραφεί αλγόριθμος ο οποίος δέχεται τους συντελεστές αγώνων ποδοσφαίρου για ένα ιντερνετικό στοίχημα και υπολογίζει τον τελικό συντελεστή (γινόμενο των επιμέρους). Ο παίκτης μπορεί να ποντάρει το πολύ 8 αγώνες. Τιμή φρουρός το μηδέν. Το στοίχημα είναι έγκυρο αν παιχτούν τουλάχιστον 3 αγώνες.

**71.** Δίνεται το παρακάτω τμήμα αλγορίθμου:

$\Delta \leftarrow$  **Αληθής**

**Για**  $a$  **από** 1 **μέχρι**  $N$

$\Delta \leftarrow$  **ΟΧΙ**  $\Delta$

**Τέλος\_επανάληψης**

**Εμφάνισε**  $\Delta$

Να το εκτελέσετε για καθεμία από τις παρακάτω περιπτώσεις:

1)  $N=0$  2)  $N=1$  3)  $N=4$  4)  $N=2011$  5)  $N=8128$

και να γράψετε τη λογική τιμή που θα εμφανιστεί μετά την εκτέλεση της αντίστοιχης περίπτωσης.

**72.** Δίνεται το παρακάτω τμήμα αλγορίθμου:

$K \leftarrow 1$

**ΟΣΟ**  $K \leq 200$  **ΕΠΑΝΑΛΑΒΕ**

**ΕΜΦΑΝΙΣΕ**  $K$

$K \leftarrow K + 2$

**ΤΕΛΟΣ\_ ΕΠΑΝΑΛΗΨΗΣ**

Να γράψετε στο τετράδιό σας

α. τις σταθερές,

- β. τους αριθμητικούς τελεστές,
- γ. τους συγκριτικούς τελεστές,
- δ. τις λογικές εκφράσεις.

**73.** Σε ΚΤΕΟ της χώρας το 2010 προσέρχονται οχήματα για έλεγχο. Τα οχήματα είναι τριών κατηγοριών ΦΟΡΤΗΓΟ, ΕΠΙΒΑΤΗΓΟ, ΔΙΚΥΚΛΟ και πληρώνουν 60€, 40€ και 20€ αντίστοιχα. Ένα όχημα χαρακτηρίζεται ως προς την προσέλευσή του “ΕΜΠΡΟΘΕΣΜΟ” ή “ΕΚΠΡΟΘΕΣΜΟ”. Τα οχήματα που προσέρχονται εκπρόθεσμα επιβαρύνονται με πρόστιμο 15,80€. Να αναπτύξετε πρόγραμμα σε “ΓΛΩΣΣΑ” το οποίο:

1. Περιλαμβάνει τμήμα δηλώσεων μεταβλητών.
2. Για κάθε όχημα το οποίο προσέρχεται στο ΚΤΕΟ για έλεγχο
  - α. διαβάζει την κατηγορία του, το έτος της πρώτης κυκλοφορίας και τον τύπο προσέλευσης χωρίς κανένα έλεγχο εγκυρότητας.
  - β. υπολογίζει και εμφανίζει, με βάση την κατηγορία του και την εμπρόθεσμη ή εκπρόθεσμη προσέλευσή του, το ποσό πληρωμής.
 Η διαδικασία εισαγωγής δεδομένων τερματίζει όταν δοθεί η τιμή “Τ” σαν κατηγορία οχήματος.
3. Εμφανίζει το πλήθος των φορτηγών που προσήλθαν στο ΚΤΕΟ.
4. Εμφανίζει την κατηγορία του παλαιότερου οχήματος.
5. Εμφανίζει το συνολικό ποσό προστίμου.

**74.** Τι θα εμφανίσει το παρακάτω σύνολο εντολών.

$\mu \leftarrow 9$

$\lambda \leftarrow 13$

**Αρχή\_επανάληψης**

$\alpha \leftarrow \lambda - \mu$

**Αν  $\alpha > 0$  τότε**

**Εμφάνισε  $\mu+1$ ,  $\lambda-2$**

**Τέλος\_Αν**

$\mu \leftarrow \mu - 3$

$\lambda \leftarrow \lambda - 5$

**Εμφάνισε  $\mu$ ,  $\lambda$**

**Μέχρις\_ότου  $\mu < 2$**

**75.** Το τετράγωνο ενός ακεραίου αριθμού  $N$  μπορεί να υπολογιστεί προσθέτοντας όλους τους ακεραίους από το 1 έως το  $N$  και επιστρέφοντας πάλι πίσω στο 1, π.χ.  $4^2 = 1 + 2 + 3 + 4 + 3 + 2 + 1 = 16$ . Να γραφεί ο κατάλληλος αλγόριθμος που θα υπολογίζει και θα εμφανίζει το τετράγωνο οποιουδήποτε ακεραίου  $N$  χρησιμοποιώντας τη μέθοδο αυτή. Ο αλγόριθμος θα δέχεται ως είσοδο τον αριθμό  $N$ . Κατά την εισαγωγή θα ελέγχεται η συνθήκη  $N > 0$ .

**76.** Δεδομένα που αφορούν  $N$  εργαζόμενους ( $N =$  γνωστό) εισάγονται με την εξής σειρά: κωδικός (τριψήφιος ακεραίος), τμήμα (1 ή 2), μισθός. Να γραφεί αλγόριθμος που θα βρίσκει και θα εμφανίζει: το μεγαλύτερο μισθό σε κάθε τμήμα, τον κωδικό εργαζομένου που λαμβάνει το μεγαλύτερο μισθό (αν υπάρχουν περισσότεροι από ένας εργαζόμενοι θα εμφανίζεται ο πρώτος) και το μέσο μισθό κάθε τμήματος.

**77.** Δεδομένα εισόδου είναι ζεύγη πραγματικών θετικών αριθμών που αντιστοιχούν στο επιθυμητό και το πραγματικό μήκος μεταλλικών ράβδων που παράγει μια μονάδα

παραγωγής. Η μέγιστη αποδεκτή απόκλιση μεταξύ επιθυμητού και πραγματικού μήκους είναι 5%. Η εισαγωγή δεδομένων θα σταματά όταν το ποσοστό των ζευγών με απόκλιση μεγαλύτερη από τη μέγιστη αποδεκτή απόκλιση ξεπεράσει το 20% του συνόλου όλων των ζευγών που έχουν εισαχθεί μέχρι εκείνη τη στιγμή. Ο αλγόριθμος θα εμφανίζει στο τέλος το πλήθος των ζευγών που έχουν εισαχθεί.

**78.** Ο σημερινός αριθμός αυτοκινήτων που κυκλοφορούν σε μια πόλη είναι  $A$ . Αν ο αριθμός αυτός αυξάνεται με ετήσιο ρυθμό  $c\%$ , να γραφεί αλγόριθμος που να υπολογίζει σε πόσα έτη ο αριθμός των αυτοκινήτων θα ξεπεράσει μια δεδομένη γνωστή τιμή  $B$  (να υποθέσετε ότι θα ισχύει  $B > A$ ). Ο αλγόριθμος θα εμφανίζει στο τέλος τον αριθμό των ετών καθώς και τον τελικό αριθμό των αυτοκινήτων.

**79.** Να γραφεί αλγόριθμος που θα υπολογίζει και θα εμφανίζει τον πρώτο ακέραιο και θετικό αριθμό, το τετράγωνο του οποίου διαφέρει από το τετράγωνο του επομένου του τουλάχιστον κατά 50.

**80.** Να γραφεί αλγόριθμος που θα εισάγει ένα άγνωστο πλήθος μετρήσεων (οι μετρήσεις αντιστοιχούν σε πραγματικούς αριθμούς), θα υπολογίζει και θα εκτυπώνει:

- Τη μέγιστη τιμή και τη θέση της στο πλήθος
- Την ελάχιστη τιμή και τη θέση της στο πλήθος
- Τη μέση τιμή
- Ο τελευταίος αριθμός θα είναι ο αριθμός  $-999.9$  και δεν αποτελεί μέτρηση (ο τελευταίος αριθμός καθορίζει και το τέλος εισαγωγής των δεδομένων).

**81.** Για την εύρεση μιας ρίζας ενός πολυωνύμου χρησιμοποιείται ο παρακάτω αλγόριθμος (μέθοδος Newton):

- 1) εισάγονται ως δεδομένα η τάξη του πολυωνύμου και οι συντελεστές κάθε όρου
- 2) εισάγονται ως δεδομένα δύο τυχαίες τιμές (με ονόματα  $posX$ ,  $negX$ ) εντός των οποίων ενδέχεται να υπάρχει μια ρίζα  $x$
- 3) ο αλγόριθμος ελέγχει αν υπάρχει ρίζα μεταξύ αυτών των δύο τιμών (οι τιμές του πολυωνύμου για κάθε μια από τις δύο αυτές τιμές  $posX$ ,  $negX$  πρέπει να έχουν αντίθετο πρόσημο). Αν αυτό δεν ισχύει ο αλγόριθμος θα σταματά και θα εμφανίζει το κατάλληλο μήνυμα.
- 4) Ο αλγόριθμος θα χρησιμοποιεί ως προσεγγιστική τιμή της ρίζας  $x$  τη μέση τιμή των δύο αυτών τιμών  $posX$ ,  $negX$  επαναληπτικά.
- 5) Αν για τη νέα τιμή  $x$  η τιμή του πολυωνύμου είναι θετική θα αντικαθιστά το  $posX$  με το  $x$ , ενώ αν είναι αρνητική θα αντικαθιστά το  $negX$  με το  $x$ .
- 6) ο αλγόριθμος θα επαναλαμβάνει τα βήματα 4 και 5 μέχρις ότου δύο διαδοχικές τιμές του  $x$  θα διαφέρουν τουλάχιστον κατά μια σταθερά  $\epsilon=10^{-4}$ . Ο αλγόριθμος σε κάθε βήμα πρέπει να εμφανίζει όλες τις υπολογιζόμενες τιμές.

**82.** Δίνονται  $N$  σειρές ακέραιων και θετικών αριθμών ( $N$ = γνωστό). Σε κάθε σειρά ο πρώτος αριθμός δείχνει το πλήθος αυτών που υπάρχουν στη συνέχεια. Να γραφεί αλγόριθμος που θα βρίσκει και θα εμφανίζει το πλήθος των άρτιων και των περιττών αριθμών που υπάρχουν σε κάθε σειρά. Κατά την εισαγωγή θα ελέγχεται ότι όλοι οι αριθμοί είναι θετικού.

Παράδειγμα για  $N=3$  :

5, 2,45,77,4,33 άρτιοι = 2 περιττοί=3

4,17,27,44,55 άρτιοι = 1 περιττοί=3

2, 3,9 άρτιοι = 0 περιττοί=2

**83.** Δύο θετικοί ακέραιοι αριθμοί είναι «φίλιοι» αν ο καθένας ισούται με το άθροισμα όσων διαιρούν τον άλλον (λαμβάνονται υπόψη μόνον οι γνήσιοι διαιρέτες). Οι πιο διάσημοι «φίλιοι» αριθμοί είναι οι αριθμοί 220 και 284 (αποδίδονται στον Πυθαγόρα).

Διαιρέτες του 220: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110 (άθροισμα=284).

Διαιρέτες του 284 : 1, 2, 4, 71, 142 (άθροισμα=220).

Να γραφεί αλγόριθμος που θα διαβάζει δύο ακεραίους και θα εμφανίζει αν είναι φίλιοι ή όχι.

**84.** Να σχεδιάσετε τον πίνακα τιμών, το διάγραμμα ροής και τις τιμές που θα εμφανιστούν στην οθόνη μετά την εκτέλεση του παρακάτω αλγορίθμου:

**Αλγόριθμος ΠίνακαςΤιμών3**

$x \leftarrow 2$

**Για k από 1 μέχρι 9 με βήμα 3**

$x \leftarrow x + k$

$z \leftarrow 7$

**Όσο  $z < x$  επανάλαβε**

$z \leftarrow z + 2$

**Εμφάνισε z**

**Τέλος\_επανάληψης**

**Εμφάνισε x**

Τέλος\_επανάληψης

**Τέλος ΠίνακαςΤιμών3**

**85.** Κάθε χρόνο έρχονται στην Ελλάδα τουρίστες από τη Γερμανία και τη Ρωσία. Να γραφεί πρόγραμμα το οποίο:

α. διαβάζει για κάθε επισκέπτη την ηλικία, τη χώρα προέλευσης (Ρ ή Γ) και το φύλο (Α, Γ).

β. Να υπολογίζει-εμφανίζει: 1. το σύνολο των γυναικών που επισκέφθηκαν την Ελλάδα 2. τη χώρα καταγωγής του νεαρότερου τουρίστα 3. το μέσο όρο ηλικίας των ανδρών επισκεπτών από τη Γερμανία. (Να τερματίζει όταν δοθεί ηλικία αρνητική).

**86.** Να αναπτυχθεί πρόγραμμα που θα διαβάζει άγνωστο πλήθος θετικών αριθμών και θα τερματίζει όταν εισαχθεί αρνητικός αριθμός ή μηδέν. Να εκτυπώνεται:

- Ο μεγαλύτερος αριθμός που διαβάστηκε
- Ο μικρότερος αριθμός που διαβάστηκε
- Το πλήθος των αριθμών που διαβάστηκαν
- Το πλήθος των άρτιων αριθμών που διαβάστηκαν
- Το πλήθος των περιττών αριθμών που διαβάστηκαν
- Ο μέσος όρος των στοιχείων που διαβάστηκαν
- Ο μέσος όρος των άρτιων αριθμών που διαβάστηκαν
- Ο μέσος όρος των περιττών αριθμών που διαβάστηκαν

**87.** Ένας κωδικός χρήστη αποτελείται το πολύ από 10 χαρακτήρες και το ελάχιστο από 6. Ο κωδικός αυτός μπορεί να περιέχει οποιοδήποτε χαρακτήρα αλλά τουλάχιστον ένα σύμβολο ('\$', '#', '@', '!', '%', '\*') και δύο αριθμούς (0-9). Επιπλέον απαγορεύεται η εισαγωγή του κενού. Να γίνει αλγόριθμος που θα ελέγχει την εγκυρότητα ενός κωδικού.

**88.** Στα μαθηματικά πρώτος αριθμός είναι ένας φυσικός αριθμός μεγαλύτερος της μονάδας με την ιδιότητα οι μόνοι φυσικοί διαιρέτες του να είναι η μονάδα και ο εαυτός του. Το μηδέν και το ένα δεν είναι πρώτοι αριθμοί.

Για παράδειγμα η ακολουθία των 10 πρώτων αριθμών είναι η εξής:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...

Να αναπτύξετε αλγόριθμο που να εμφανίζει τους πρώτους αριθμούς μέχρι το 1000.

**89.** Ο μηνιαίος μισθός κάθε υπαλλήλου μιας εταιρείας προκύπτει από τον βασικό μισθό, το bonus και το επίδομα ως εξής:

Βασικός μισθός: 100 ευρώ.

Bonus: Αν οι μηνιαίες πωλήσεις είναι κάτω από 4000 ευρώ τότε παίρνει bonus 100 ευρώ, αλλιώς 7% επί των μηνιαίων πωλήσεών του. Το επίδομα συνδέεται με τον κωδικό του κάθε υπαλλήλου. Επίδομα 200 ευρώ δίνεται μόνο σε αυτούς που έχουν κωδικό τουλάχιστον 355.

Να γραφεί αλγόριθμος ο οποίος για τους 35 υπαλλήλους της εταιρείας:

α. Διαβάζει τον κωδικό και τις πωλήσεις ανά μήνα κάθε υπαλλήλου.

β. Υπολογίζει και τυπώνει το μέσο όρο των μηνιαίων πωλήσεων όλων των υπαλλήλων με κωδικό μεγαλύτερο του 300.

γ. Υπολογίζει και τυπώνει το σύνολο των εξόδων της εταιρείας για ένα μήνα σε πληρωμές των υπαλλήλων της.

δ. Εμφανίζει τον κωδικό του υπαλλήλου με τη μεγαλύτερη προμήθεια.

**90.** Για ορισμένους ακέραιους και θετικούς αριθμούς ισχύει η εξής ιδιότητα: Το άθροισμα που προκύπτει από την επαναλαμβανόμενη πρόσθεση είτε των ψηφίων του αριθμού είτε από την πρόσθεση του νέου αριθμού σε αυτόν που δημιουργήθηκε αμέσως πριν, ισούται με τον αριθμό που εξετάζεται, π.χ. για τον αριθμό 75:  $7+5=12$ ,  $5+12=17$ ,  $12+17=29$ ,  $17+29=46$ ,  $29+46=75$ .

Να γραφεί αλγόριθμος που θα βρίσκει και θα εμφανίζει όλους τους αριθμούς που ικανοποιούν αυτή την ιδιότητα και είναι μικρότεροι του 1000. Στο τέλος να εμφανίζεται και το πλήθος αυτών των αριθμών.

**91.** Η καταλανική τριάδα είναι το τρίγωνο αριθμών

1

1 1

1 2 2

1 3 5 5

1 4 9 14 14

1 5 14 28 42 42

1 6 20 48 90 132 132

Κάθε στοιχείο είναι το άθροισμα του στοιχείου που βρίσκεται ακριβώς πάνω από αυτό και αυτού που βρίσκεται αριστερά του, π.χ.  $9=5+4$ . Να γραφεί αλγόριθμος που θα εμφανίζει το σχετικό τρίγωνο καταλανικών τριάδων για  $n=10$ .

**92.** Για τον υπολογισμό του  $n!$  όταν η τιμή του  $n$  είναι μεγάλη χρησιμοποιείται ο προσεγγιστικός τύπος του Stirling:  $n! \approx e^{-n} n^n \sqrt{2\pi n}$ .

Να γραφεί αλγόριθμος για τον υπολογισμό της τιμής  $n!$  σύμφωνα με τον παραπάνω τύπο. Στον ίδιο αλγόριθμο να υπολογιστεί το  $n!$  σύμφωνα με τον τύπο  $n! = 1 * 2 * 3 * \dots * n$  και να βρεθούν οι διαφορές % που έχουν οι δύο μέθοδοι.



### 3.5 Ασκήσεις προς επίλυση με Διαγράμματα Ροής

**Οδηγία:** Αυτό που συνήθως δυσκολεύει σε μια μετατροπή ενός διαγράμματος ροής σε αλγόριθμο είναι οι **συνθήκες (ρόμβοι)**. Άλλες φορές οι συνθήκες εννοούν μια εντολή επιλογής ενώ άλλες φορές μια εντολή επανάληψης. Μία καλή μέθοδος για να ελέγξουμε αν ένα διάγραμμα περιέχει δομή επανάληψης είναι να παρατηρήσουμε τα βέλη του διαγράμματος. **Εάν τα βέλη γυρνάνε προς τα πίσω**, (ή ανεβαίνουν προς τα επάνω) τότε το διάγραμμα περιέχει **δομή επανάληψης**. Σε διαφορετική περίπτωση δεν υπάρχει δομή επανάληψης.

Να μετατραπούν οι ακόλουθοι αλγόριθμοι σε ισοδύναμα διαγράμματα ροής.

#### 1. Αλγόριθμος ΑσκήσηΔΡ1

$\pi \leftarrow 0$

Για  $i$  από 1 μέχρι 14 με\_βήμα 3

$y \leftarrow 0$

Όσο  $y < 8$  επανάλαβε

    Διάβασε  $x$

    Αν  $x < 0$  τότε

$\pi \leftarrow \pi + 1$

$y \leftarrow y + 1$

    Τέλος\_Αν

$y \leftarrow y + 1$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Εμφάνισε  $\pi, y$

Τέλος ΑσκήσηΔΡ1

#### 2. Αλγόριθμος Μετατροπή

$X \leftarrow 0$

Για  $K$  από 1 μέχρι 10

    Διάβασε  $\Lambda$

    Αν  $\Lambda > 0$  τότε

$X \leftarrow X + \Lambda$

    Αλλιώς

$X \leftarrow X - \Lambda$

    Τέλος\_Αν

Τέλος\_Επανάληψης

Εμφάνισε  $X$

Τέλος Μετατροπή

#### 3. Αλγόριθμος ΑσκήσηΔΡ3

$\Sigma \leftarrow 0$

Αρχή\_επανάληψης

    Διάβασε  $x$

    Όσο  $x > 0$  επανάλαβε

        Εμφάνισε  $x$

$x \leftarrow x - 3$

$\Sigma \leftarrow \Sigma + x$



Τέλος\_επανάληψης  
Μέχρις\_ότου  $\Sigma > 200$   
Τέλος ΑσκησηΔΡ3

4.  $\mu \leftarrow -31$   
Αρχή\_επανάληψης  
Για  $\lambda$  από  $-2$  μέχρι  $5$  με\_βήμα  $4$   
 $\kappa \leftarrow 19$   
Όσο  $\kappa \leq 42$  επανάλαβε  
Εμφάνισε  $\kappa + \lambda + \mu$   
 $\kappa \leftarrow \kappa + 13$   
Τέλος\_επανάληψης  
Τέλος\_επανάληψης  
Εμφάνισε  $\mu + \lambda$   
 $\mu \leftarrow \mu - 14$   
Μέχρις\_ότου  $\mu < 4$   
Εμφάνισε  $\mu$

5. Αλγόριθμος Διοφαντική  
Για  $x$  από  $0$  μέχρι  $100$   
Για  $y$  από  $0$  μέχρι  $100$   
Για  $z$  από  $0$  μέχρι  $100$   
Αν  $3 * x + 2 * y - 7 * z = 5$  τότε Εκτύπωσε  $x, y, z$   
Τέλος\_επανάληψης  
Τέλος\_επανάληψης  
Τέλος\_επανάληψης  
Τέλος Διοφαντική



# Ενότητα 4 Δομές Δεδομένων-Μονοδιάστατοι Πίνακες (κεφ. 3 & 9 σχολ. Βιβλίου)





### Στόχοι:

- ➔ Να εκτιμήσετε τη σημασία που έχουν για τον προγραμματισμό οι πίνακες δεδομένων.
- ➔ Να μπορείτε να δηλώνετε και να χειρίζεστε απλούς πίνακες.
- ➔ Να κατανοήσετε τον τρόπο με τον οποίο αποθηκεύονται στη μνήμη οι τιμές δεδομένων.
- ➔ Να αναγνωρίσετε σε τι διαφέρει η μεταβίβαση πινάκων ως παραμέτρων συναρτήσεων από τη μεταβίβαση απλών μεταβλητών.
- ➔ Να μάθετε πώς να κάνετε στατική ανάθεση αρχικών τιμών σε πίνακες.
- ➔ Να κατανοήσετε τη δομή των πολυδιάστατων πινάκων.

## 4.1 Θεωρία-Οδηγίες

Οι **δομές δεδομένων** είναι ο μηχανισμός της γλώσσας που μας δίνει τη δυνατότητα να ομαδοποιήσουμε δεδομένα που σχετίζονται μεταξύ τους ώστε να τα επεξεργαστούμε καλύτερα.

Οι δομές δεδομένων διακρίνονται σε δύο μεγάλες κατηγορίες: τις **στατικές** (static) και τις **δυναμικές** (dynamic). Οι δυναμικές δομές δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης, αλλά στηρίζονται στην τεχνική της λεγόμενης δυναμικής παραχώρησης μνήμης (dynamic memory allocation). Με άλλα λόγια, οι δομές αυτές δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων τους μεγαλώνει και μικραίνει καθώς στη δομή εισάγονται νέα δεδομένα ή διαγράφονται κάποια δεδομένα αντίστοιχα. Όλες οι σύγχρονες γλώσσες προγραμματισμού προσφέρουν τη δυνατότητα δυναμικής παραχώρησης μνήμης.

Συχνά δεν μας αρκούν οι μεταβλητές! Πώς θα αποθηκεύαμε;

- ✓ Ένα σύνολο μαθητών;
- ✓ Μία λίστα από πόλεις;
- ✓ Μαθητές μαζί με τους βαθμούς τους;

Άρα βοηθούν οι πίνακες στην επίλυση προβλημάτων που δεν θα μπορούσαμε να λύσουμε μόνο με μεταβλητές.

Οι **πίνακες** αποτελούν ένα βολικό τρόπο διαχείρισης πολλών δεδομένων **ίδιου τύπου**. Η χρήση τους είναι αναγκαία όταν τα δεδομένα ενός προγράμματος πρέπει να διατηρηθούν στη μνήμη του μέχρι το τέλος της εκτέλεσης του. **Πίνακας** είναι ένα σύνολο αντικειμένων ίδιου τύπου, τα οποία αναφέρονται με ένα κοινό όνομα. Κάθε ένα από τα αντικείμενα λέγεται **στοιχείο** του πίνακα. Η αναφορά σε ατομικά στοιχεία του πίνακα γίνεται με το όνομα του πίνακα ακολουθούμενο από ένα δείκτη.

Με τη χρήση του πίνακα, όλα τα δεδομένα καταχωρούνται κάτω από το ίδιο όνομα μεταβλητής και έχει δύο χαρακτηριστικά:

1. Ο πίνακας είναι **διατεταγμένος**: τα μεμονωμένα συστατικά στοιχεία ενός πίνακα μπορούμε να τα ξεχωρίσουμε ως πρώτο, δεύτερο, κ.ο.κ.
2. Ο πίνακας είναι **ομοιογενής**: όλες οι τιμές που αποθηκεύονται σε έναν πίνακα είναι του ίδιου τύπου.

Οι πίνακες θεωρούμε ότι είναι στατικές δομές και άρα πρέπει να ορίζονται στην αρχή

κάθε προγράμματος.

Το όνομα του πίνακα μπορεί να είναι οποιοδήποτε δεκτό όνομα της **ΓΛΩΣΣΑΣ** και ο δείκτης είναι μία ακέραια έκφραση, σταθερή ή μεταβλητή που περικλείεται μέσα στα σύμβολα [ και ]. Κάθε πίνακας πρέπει υποχρεωτικά να περιέχει δεδομένα του ίδιου τύπου, δηλαδή ακέραια, πραγματικά, λογικά, ή αλφαριθμητικά. Ο τύπος του πίνακα δηλώνεται μαζί με τις άλλες μεταβλητές του προγράμματος στο τμήμα δήλωσης μεταβλητών. Εκτός από τον τύπο του πίνακα πρέπει να δηλώνεται και ο αριθμός των στοιχείων που περιέχει ή καλύτερα ο μεγαλύτερος αριθμός στοιχείων που μπορεί να έχει ο συγκεκριμένος πίνακας και αυτό για να δεσμευτούν οι αντίστοιχες συνεχόμενες θέσεις μνήμης.

Η χρήση πινάκων είναι ένας **βολικός τρόπος** για τη διαχείριση πολλών δεδομένων ίδιου τύπου, αλλά συχνά η χρήση τους είναι **περιττή** και επιζήμια στην ανάπτυξη του προγράμματος.

- ✓ **Οι πίνακες απαιτούν μνήμη.** Κάθε πίνακας δεσμεύει από την αρχή του προγράμματος πολλές θέσεις μνήμης. Σε ένα μεγάλο και σύνθετο πρόγραμμα η άσκοπη χρήση μεγάλων πινάκων μπορεί να οδηγήσει ακόμη και σε αδυναμία εκτέλεσης του προγράμματος.
- ✓ **Οι πίνακες περιορίζουν τις δυνατότητες του προγράμματος.** Αυτό γιατί οι πίνακες είναι στατικές δομές και το μέγεθος τους πρέπει να δηλώνεται στην αρχή του προγράμματος, ενώ παραμένει υποχρεωτικά σταθερό κατά την εκτέλεση του προγράμματος.

Κάθε πίνακας έχει δύο βασικές ιδιότητες: τον **τύπο στοιχείου**, δηλαδή τον τύπο των τιμών οι οποίες είναι δυνατό να αποθηκευτούν στα στοιχεία του πίνακα και το **μέγεθος πίνακα**, δηλαδή το πλήθος των στοιχείων που περιέχει. Κάθε φορά που δημιουργείτε έναν πίνακα στο πρόγραμμά σας, θα πρέπει να καθορίζετε τόσο τον τύπο των στοιχείων όσο και το μέγεθος του πίνακα:

Όταν χρησιμοποιείτε πίνακες στα προγράμματά σας πρέπει να εξασφαλίζετε ότι οι τιμές των αριθμοδεικτών που χρησιμοποιείτε για να επιλέγετε στοιχεία από τους πίνακες θα παραμένουν **εντός των ορίων** των πινάκων.

Με τους πίνακες γράφεις και διαβάζεις σε όποια θέση θέλεις άμεσα:

π.χ.  $A[5] = 14$

Άρα εύκολη σκέψη και χρήση.

Υπάρχουν δύο τρόποι για να προσπελάσουμε τα στοιχεία ενός πίνακα έστω  $A$ .

**Άμεσα**, π.χ.  $A[i]$ .

**Έμμεσα** π.χ.  $A[B[i]]$  όπου ο δείκτης είναι κελί άλλου πίνακα (υποχρεωτικά ακεραίου με θετικές τιμές).

**Οδηγία 1:** Η ανάγνωση, η επεξεργασία και η εμφάνιση των στοιχείων ενός πίνακα υλοποιείται από βρόχους και συνήθως με την εντολή **ΓΙΑ**. Στους δισδιάστατους πίνακες συνήθως έχουμε εμφωλευμένες επαναλήψεις. Η αναφορά στα στοιχεία των πινάκων γίνεται με τη χρήση **ακέραιων θετικών δεικτών** (τόσοι όσες και οι διαστάσεις). Όταν θέλουμε να προσπελάσουμε τον δισδιάστατο πίνακα κατά γραμμές, βολεύει να βάλουμε πρώτα την εντολή επανάληψης που θα επεξεργαστεί τις γραμμές (αντίστοιχα για τις

στήλες).

**Οδηγία 2:** Δύο ή περισσότεροι πίνακες λέγονται **παράλληλοι** αν τα περιεχόμενα που βρίσκονται στις ίδιες θέσεις τους, αντιστοιχούν σε χαρακτηριστικά της ίδιας οντότητας. Άρα στους παράλληλους πίνακες να προσέχουμε να μεταβάλλουμε τους δείκτες τους ταυτόχρονα και με τον ίδιο τρόπο. Φυσικά δεν είναι υποχρεωτικό να είναι του ίδιου τύπου.

Η χρήση των πινάκων είναι ένας βολικός τρόπος για την αποθήκευση μεγάλου αριθμού δεδομένων ίδιου τύπου. Συνήθως οι νέοι προγραμματιστές χρησιμοποιούν πίνακες ακόμη και όταν η χρήση τους δεν είναι απαραίτητη! Εξέτασε αν πραγματικά χρειάζεται πίνακας για την επίλυση του προβλήματος. Αν δεν είναι απαραίτητος μην τον χρησιμοποιείς. Να έχεις πάντα στο νου σου ότι οι πίνακες ξοδεύουν μεγάλα ποσά μνήμης.

Για να αποφύγεις τα πλέον κοινά λάθη στη χρήση των πινάκων να προσέχεις πάντα:

- ✓ Να δίνεις αρχικές τιμές αν χρειάζεται στους πίνακες.
- ✓ Μην ξεπερνάς τα **όρια** του πίνακα σου. Το πιο συνηθισμένο λάθος στη χρήση των πινάκων είναι η προσπάθεια ανάγνωσης ή εκχώρησης τιμής έξω από τα όρια του πίνακα.
- ✓ Η επεξεργασία γίνεται στα στοιχεία του πίνακα. Άρα σε όλες τις εντολές πρέπει να εμφανίζονται τα στοιχεία του πίνακα και όχι το όνομα του ίδιου του πίνακα.
- ✓ Όλα τα στοιχεία του πίνακα έχουν τον ίδιο τύπο, για παράδειγμα όλα είναι ακέραια ή όλα είναι χαρακτήρες όπως ορίστηκαν στο τμήμα δηλώσεων.

Συνήθεις ασκήσεις με πίνακες.

- ➔ Υπολογισμός αθροισμάτων στοιχείων του πίνακα.
- ➔ Εύρεση του μέγιστου ή του ελάχιστου στοιχείου.
- ➔ Ταξινόμηση των στοιχείων του πίνακα.
- ➔ Αναζήτηση ενός στοιχείου του πίνακα.
- ➔ Συγχώνευση δύο πινάκων.

**Λειτουργίες Δ.Δ.**

**Δομή Δεδομένων** είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών. Κάθε μορφή δομής δεδομένων αποτελείται από ένα σύνολο **κόμβων** (nodes).

Οι βασικές **λειτουργίες** (ή αλλιώς πράξεις) επί των δομών δεδομένων είναι οι ακόλουθες:

- ✓ **Προσπέλαση** (access), πρόσβαση σε έναν κόμβο με σκοπό να εξετασθεί ή να τροποποιηθεί το περιεχόμενό του.
- ✓ **Εισαγωγή** (insertion), δηλαδή η προσθήκη νέων κόμβων σε μία υπάρχουσα δομή.
- ✓ **Διαγραφή** (deletion), που αποτελεί το αντίστροφο της εισαγωγής, δηλαδή ένας κόμβος αφαιρείται από μία δομή.
- ✓ **Αναζήτηση** (searching), κατά την οποία προσπελάζονται οι κόμβοι μιας δομής, προκειμένου να εντοπιστούν ένας ή περισσότεροι που έχουν μια δεδομένη ιδιότητα.
- ✓ **Ταξινόμηση** (sorting), όπου οι κόμβοι μιας δομής διατάσσονται κατά αύξουσα ή φθίνουσα σειρά.
- ✓ **Αντιγραφή** (copying), κατά την οποία όλοι οι κόμβοι ή μερικοί από τους κόμβους μίας δομής αντιγράφονται σε μία άλλη δομή.
- ✓ **Συγχώνευση** (merging), κατά την οποία δύο ή περισσότερες δομές συνενώνονται σε

μία ενιαία δομή.

✓ **Διαχωρισμός** (separation), που αποτελεί την αντίστροφη πράξη της συγχώνευσης.

Δεν μπορούν να εφαρμοστούν η εισαγωγή και η διαγραφή σε πίνακα. Συγκεκριμένα, στη στατική δομή δεδομένων το ακριβές μέγεθος της απαιτούμενης μνήμης καθορίζεται κατά τη στιγμή του προγραμματισμού τους. Αυτό συνεπάγεται ότι αυτό το μέγεθος καθορίζεται κατά τη στιγμή μετάφρασης τους και όχι κατά τη στιγμή εκτέλεσης (on the fly) του προγράμματος. Οι πράξεις της εισαγωγής και της διαγραφής αυξάνουν και μειώνουν αντίστοιχα το μέγεθος μίας δομής δεδομένων, επομένως μπορούν να εφαρμοστούν μόνο σε δυναμικές δομές δεδομένων. Στην πράξη σπάνια χρησιμοποιούνται και οι οκτώ λειτουργίες για κάποια δομή.

**Δείκτης:** Στον προγραμματισμό, ένας ακέραιος που αναγνωρίζει τη θέση ενός στοιχείου δεδομένων σε μια ακολουθία στοιχείων δεδομένων.

Εξίσωση **Wirth**: Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα.

**Στοιβα stack:** Δομή δεδομένων με ένα άκρο στην οποία το τελευταίο στοιχείο που εισάγεται είναι και το πρώτο που μπορεί να εξαχθεί. Δύο είναι οι κύριες λειτουργίες σε μία **στοίβα**:

Η **ώθηση** (push) στοιχείου στην κορυφή της στοίβας, και η **απώθηση** (pop) στοιχείου από τη στοίβα.

Η διαδικασία της ώθησης πρέπει οπωσδήποτε να ελέγχει, αν η στοίβα είναι γεμάτη, οπότε λέγεται ότι συμβαίνει **υπερχείλιση** (overflow) της στοίβας. Αντίστοιχα, η διαδικασία απώθησης ελέγχει, αν υπάρχει ένα τουλάχιστον στοιχείο στη στοίβα, δηλαδή ελέγχει αν γίνεται **υποχείλιση** (underflow) της στοίβας.

Μια στοίβα μπορεί να υλοποιηθεί πολύ εύκολα με τη βοήθεια ενός μονοδιάστατου πίνακα, Μια βοηθητική μεταβλητή (με όνομα συνήθως **top**) χρησιμοποιείται για να δείχνει το στοιχείο που τοποθετήθηκε τελευταίο στην κορυφή της στοίβας. Για την εισαγωγή ενός νέου στοιχείου στη στοίβα (ώθηση) αρκεί να αυξηθεί η μεταβλητή **top** κατά ένα και στη θέση αυτή να εισέλθει το στοιχείο. Αντίθετα για την εξαγωγή ενός στοιχείου από τη στοίβα (απώθηση) εξέρχεται πρώτα το στοιχείο που δείχνει η μεταβλητή **top** και στη συνέχεια η **top** μειώνεται κατά ένα για να δείχνει τη νέα κορυφή.

**Ουρά** queue: Δομή δεδομένων με δύο άκρα, στην οποία το πρώτο στοιχείο που εισάγεται είναι και το πρώτο που μπορεί να εξαχθεί. Δύο είναι οι κύριες λειτουργίες που εκτελούνται σε μία **ουρά**:

Η **εισαγωγή** (enqueue) στοιχείου στο πίσω άκρο της ουράς, και η εξαγωγή (**dequeue**) στοιχείου από το εμπρός άκρο της ουράς. απαιτούνται **δύο δείκτες**: ο εμπρός (front) και ο πίσω (rear) δείκτης, που μας δίνουν τη θέση του στοιχείου που σε πρώτη ευκαιρία θα εξαχθεί και τη θέση του στοιχείου που μόλις εισήλθε. π.χ. σε μια αποθήκη φαρμάκων η εξαγωγή-πώληση αυτών γίνεται με τη μέθοδο **FIFO**.

Τα στοιχεία ενός **αρχείου** ονομάζονται **εγγραφές**.

Κάθε εγγραφή αποτελείται από ένα τουλάχιστον **πεδίο (πρωτεύων κλειδί)** που ταυτοποιεί την εγγραφή και από επιπλέον πεδία που καθορίζουν χαρακτηριστικά της.



## 4.2 Ερωτήσεις Σωστού-Λάθους

1. Οι δείκτες ενός πίνακα μπορούν να είναι χαρακτήρες.
2. Η μέθοδος της ταξινόμησης ευθείας ανταλλαγής (φυσάλιδα) βασίζεται στην αρχή της σύγκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων, μέχρις ότου διαταχθούν όλα τα στοιχεία.
3. Για την εισαγωγή ενός νέου στοιχείου στην ουρά ο δείκτης rear αυξάνεται κατά 1 και ο δείκτης front μειώνεται κατά 1.
4. Στη δομή FIFO απαιτείται η ύπαρξη δύο δεικτών
5. Ένας 3-διάστατος πίνακας είναι δυναμική δομή δεδομένων.
6. Το όνομα ενός πίνακα καθορίζει μια ομάδα διαδοχικών θέσεων στη μνήμη.
7. Ο πίνακας δεν δηλώνεται ως σταθερά.
8. Η παραβίαση ορίων πίνακα αναιρεί την καθοριστικότητα.
9. Το πρόβλημα ταξινόμησης πίνακα με τον πιο γρήγορο τρόπο είναι πρόβλημα βελτιστοποίησης.
10. Σκοπός της ταξινόμησης των στοιχείων ενός πίνακα είναι και να διευκολυνθεί στη συνέχεια η αναζήτηση.
11. Στην πράξη, είναι σύνηθες να χρησιμοποιούνται και οι οκτώ λειτουργίες για κάποια δομή.
12. Οι πίνακες επεκτείνουν τις δυνατότητες του προγράμματος.
13. Η δυαδική αναζήτηση εφαρμόζεται σε ταξινομημένο πίνακα.
14. Η σειριακή αναζήτηση θα πρέπει να αποφεύγεται όταν ο πίνακας είναι μεγάλος.
15. Οι πίνακες είναι δυναμικές δομές δεδομένων.
16. Ένας πίνακας δεν έχει το ίδιο μέγεθος σε όλη τη διάρκεια εκτέλεσης του αλγόριθμου.
17. Τα δεδομένα που αποθηκεύονται σε έναν πίνακα δεν μπορούν να είναι διαφορετικού τύπου.
18. Για να προσπελάσουμε έναν πίνακα δεν μπορούμε να χρησιμοποιήσουμε τη δομή επανάληψης Όσο ... Επανάλαβε.
19. Η σειριακή αναζήτηση είναι η πιο αποδοτική μέθοδος αναζήτησης.
20. Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα.
21. Η ουρά και η στοίβα μπορούν να υλοποιηθούν με δομή πίνακα.
22. Η εξαγωγή (dequeue) στοιχείου γίνεται από το εμπρός άκρο της ουράς.
23. Η απώθηση (pop) στοιχείου γίνεται από το πίσω άκρο της στοίβας.
24. Κατά τη διαδικασία της ώθησης πρέπει να ελέγχεται αν η στοίβα είναι γεμάτη.
25. Η ώθηση (push) στοιχείου είναι μία από τις λειτουργίες της ουράς.
26. Ο αλγόριθμος της σειριακής αναζήτησης χρησιμοποιείται αποκλειστικά σε ταξινομημένους πίνακες.
27. Ένας πίνακας έχει σταθερό περιεχόμενο αλλά μεταβλητό μέγεθος.
28. Οι δυναμικές δομές δεδομένων αποθηκεύονται πάντα σε συνεχόμενες θέσεις μνήμης.
29. Η μέθοδος επεξεργασίας «πρώτο μέσα πρώτο έξω» (FIFO) εφαρμόζεται στη δομή δεδομένων ΟΥΡΑ.
30. Η ταξινόμηση είναι μια από τις βασικές λειτουργίες των δομών δεδομένων.
31. Η μέθοδος της σειριακής αναζήτησης δικαιολογείται στην περίπτωση που ο πίνακας είναι μη ταξινομημένος και μικρού μεγέθους.
32. Σε μια στατική δομή το ακριβές μέγεθος της απαιτούμενης κύριας μνήμης καθορίζεται κατά την εκτέλεση του προγράμματος.
33. Όταν γίνεται σειριακή αναζήτηση κάποιου στοιχείου σε ένα μη ταξινομημένο πίνακα

και το στοιχείο δεν υπάρχει στον πίνακα, τότε υποχρεωτικά σαρώνονται όλα τα στοιχεία του πίνακα.

34. Ένα από τα πλεονεκτήματα των πινάκων είναι ότι χρειάζονται λίγο χώρο στη μνήμη.

35. Οι πίνακες είναι μία δομή δεδομένων που στηρίζεται στην τεχνική δυναμικής παραχώρησης μνήμης.

36. Η χρήση πινάκων αυξάνει την απαιτούμενη μνήμη για την εκτέλεση του προγράμματος.

37. Οι δομές δεδομένων διακρίνονται σε τρεις μεγάλες κατηγορίες: τις στατικές, τις δυναμικές και τις ημιδομημένες.

38. Ο πίνακας που χρησιμοποιεί ένα μόνο δείκτη για την αναφορά των στοιχείων του ονομάζεται μονοδιάστατος.

39. Η ταξινόμηση φυσαλίδας είναι ο πιο απλός και ταυτόχρονα ο πιο γρήγορος αλγόριθμος ταξινόμησης.

40. Οι λειτουργίες ώθηση και απώθηση είναι οι κύριες λειτουργίες σε μια ουρά.

41. Η προσπέλαση είναι μια από τις βασικές πράξεις των δομών δεδομένων.

42. Με τη λειτουργία της συγχώνευσης, δύο ή περισσότερες δομές δεδομένων συνενώνονται σε μία ενιαία δομή.

43. Για να προσδιορίσουμε την θέση σε ένα διδιάστατο πίνακα χρειαζόμαστε μία μεταβλητή.

44. Όλες οι δομές δεδομένων μπορούν να υλοποιήσουν όλες τις βασικές πράξεις.

45. Σε κάθε πίνακα η δυαδική αναζήτηση είναι ο καταλληλότερος τρόπος αναζήτησης

46. Ο μικρότερος μονοδιάστατος πίνακας έχει 2 στοιχεία.

47. Ο μικρότερος δισδιάστατος πίνακας έχει 4 στοιχεία.

48. Η εγγραφή δεν είναι απαραίτητο να έχει πεδία ίδιου τύπου.

49. Η εισαγωγή και η διαγραφή είναι δύο λειτουργίες που δε γίνονται στους πίνακες.

50. Το πλήθος των εγγραφών ενός αρχείου μπορεί να αυξομειώνεται ανάλογα με τις ανάγκες που παρουσιάζονται.

51. Σε μια ουρά στην οποία ο δείκτης FRONT έχει την τιμή 5 και ο δείκτης REAR έχει την τιμή 7, μπορεί να γίνει εξαγωγή το πολύ 3 φορές αν στο μεταξύ δε γίνει καμία εισαγωγή.

52. Ένας πίνακας καταλαμβάνει μέρος της κύριας μνήμης του υπολογιστή.

53. Οι δισδιάστατοι πίνακες δεν μπορούν να δοθούν ως παράμετροι σε μια διαδικασία.

54. Οι κόμβοι μιας στατικής δομής καταλαμβάνουν συνεχόμενες θέσεις στη μνήμη του υπολογιστή

55. Γνωρίζοντας ότι η μέγιστη τιμή ενός πίνακα  $A_{N \times M}$  βρίσκεται σε μία μοναδική θέση του πίνακα, ο εντοπισμός της θέσης αυτής μπορεί να υλοποιηθεί μέσω συνάρτησης.

56. Σε μία ουρά θα πρέπει πάντα να ισχύει  $rear \geq front$ , διαφορετικά θα σημαίνει ότι συνέβη υποχείλιση.

57. Η αναφορά  $A[3, 2, 4, 5]$  αφορά στοιχείο τετραδιάστατου πίνακα.

58. Όλες οι πράξεις των δομών δεδομένων μπορούν να εφαρμοστούν και σε πίνακες.

59. Τα δεδομένα στους πίνακες χάνονται όταν τελειώσει το πρόγραμμα.

60. Η φυσαλίδα είναι η μόνη μέθοδος ταξινόμησης ενός μονοδιάστατου πίνακα.

61. Ένας λογικός πίνακας δεν μπορεί να ταξινομηθεί με φυσαλίδα

62. Αναζήτηση μιας τιμής μπορεί να γίνει μόνο αν ο πίνακας είναι ταξινομημένος

### **Απαντήσεις**

1.Λ, 2.Σ, 3.Λ, 4.Σ, 5.Λ, 6.Σ, 7.Σ, 8.Σ, 9.Σ, 10.Σ, 11.Λ, 12.Λ, 13.Σ, 14.Σ, 15.Λ, 16.Λ, 17.Σ, 18.Λ, 19.Λ, 20.Σ, 21.Σ, 22.Σ, 23.Λ, 24.Σ, 25.Λ, 26.Λ, 27.Λ, 28.Λ, 29.Σ, 30.Σ, 31.Σ, 32.Λ, 33.Σ, 34.Λ, 35.Λ, 36.Σ, 37.Λ, 38.Σ, 39.Λ, 40.Λ, 41.Σ, 42.Σ, 43.Λ, 44.Λ, 45.Λ, 46.Λ, 47.Σ, 48.Σ, 49.Σ, 50.Σ, 51.Σ, 52.Σ, 53.Λ, 54.Σ, 55.Σ, 56.Σ,

57.Σ, 58.Λ, 59.Σ, 60.Λ, 61.Σ, 62.Λ

## 4.3 Λυμένες Ασκήσεις

1. Δίνεται το παρακάτω τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ:

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** k, z, ΠΙΝ[11]

**ΧΑΡΑΚΤΗΡΕΣ:** Χαρ

Για κάθε μια από τις παρακάτω λειτουργίες και λαμβάνοντας υπόψη το παραπάνω τμήμα δηλώσεων, να δώσετε την εντολή που την υλοποιεί:

Εκχώρηση του αριθμού 7 στη μεταβλητή k

Εμφάνιση της τιμής του 3<sup>ου</sup> στοιχείου του πίνακα ΠΙΝ

Εμφάνιση της θέσης του στοιχείου ΠΙΝ[k]

Εμφάνιση του ονόματος της μεταβλητής k

Μείωση του τελευταίου στοιχείου του πίνακα ΠΙΝ κατά 7

**Λύση**

k<-7

ΓΡΑΨΕ ΠΙΝ[3]

ΓΡΑΨΕ k

ΠΙΝ[11] <- ΠΙΝ[11] - 7

2. Να διαβαστούν 20 τιμές σ' έναν πίνακα ακεραίων και να βρεθεί η μεγαλύτερη και η δεύτερη μεγαλύτερη τιμή

**Λύση**

Για i από 1 μέχρι 20

Εμφάνισε "Δώσε τον", i, "ο αριθμό του πίνακα"

Διάβασε a[i]

Τέλος\_επανάληψης

max←a[1];

! εδώ βρίσκουμε τη μεγαλύτερη τιμή του πίνακα

Για i από 1 μέχρι 20

Αν (a[i]>max) τότε

max=a[i];

Τέλος\_Αν

Τέλος\_επανάληψης

max2←a[1];

! βρίσκουμε τη 2η μεγαλύτερη τιμή του πίνακα

Για i από 1 μέχρι 20

Αν (a[i]>max2 ΚΑΙ a[i]<>max) τότε

ma2x=a[i];

Τέλος\_Αν

Τέλος\_επανάληψης

Εμφάνισε "Η μεγαλύτερη τιμή του πίνακα είναι: ", max);

Εμφάνισε "Η δεύτερη μεγαλύτερη τιμή του πίνακα είναι: ", max2);

3. Ένας μαθητής αποταμιεύει καθημερινά ένα ποσό από το ημερήσιο χαρτζιλίκι του. Την

πρώτη ημέρα το ποσό που αποταμίευσε ήταν Μ ευρώ, ενώ για κάθε άλλη ημέρα αποταμίευσε ποσό κατά Ν ευρώ μεγαλύτερο από την προηγούμενη. Αν η αποταμίευση του μαθητή πραγματοποιήθηκε σε Π ημέρες, γράψτε αλγόριθμο που, με δεδομένα τα Μ, Ν και Π, υπολογίζει και εμφανίζει τα ημερήσια ποσά αποταμίευσης κατά φθίνουσα σειρά.

#### Λύση

**Αλγόριθμος** Μαθητής

**Δεδομένα** // Μ, Ν, Π //

Ποσό[1] ← Χ

**Για** κ από 2 μέχρι Π

    Ποσό[κ] ← Ποσό[κ-1]+Ν

**Τέλος\_επανάληψης**

**Για** κ από 2 μέχρι Π

! Ταξινόμηση Φυσαλίδας

**Για** λ από Π μέχρι κ με\_βήμα -1

**Αν** Ποσό[λ-1] < Ποσό[λ] **τότε**

            Βοηθ ← Ποσό[λ-1]

            Ποσό[λ-1] ← Ποσό[λ]

            Ποσό[λ] ← Βοηθ

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Για** κ από 1 μέχρι Π

**Εμφάνισε** Ποσό[κ]

**Τέλος\_επανάληψης**

**Τέλος** Μαθητής

4. Γράψτε αλγόριθμο που:

α) διαβάσει τα τελευταία δύο ψηφία των αριθμών δελτίου ταυτότητας κάποιων πολιτών (η είσοδος δεδομένων ολοκληρώνεται μόλις δοθεί μια αρνητική τιμή ή μηδέν),

β) υπολογίζει τη συχνότητα εμφάνισης των τιμών 30, 31, ..., 78, 79, 80, ανάμεσα στα εισερχόμενα δεδομένα.

#### Λύση

**Αλγόριθμος** ΑΣΚ

! Πίνακας συχνοτήτων

**Για** κ από 1 μέχρι 51

    Φ[κ] ← 0

**Τέλος\_επανάληψης**

**Αρχή\_επανάληψης**

**Διάβασε** Ψ

**Αν** Ψ > 0 **τότε**

**Αν** (Ψ >= 30) **ΚΑΙ** (Ψ <= 80) **τότε**

            Θ ← Ψ-29

            Φ[Θ] ← Φ[Θ]+1

**Τέλος\_Αν**

**Τέλος\_Αν**

**Μέχρις\_ότου** Ψ <= 0

**Αποτελέσματα** // Φ //

**Τέλος** ΑΣΚ

5. Εύρεση των κορυφών μόλυνσης της ατμοσφαιρικής ρύπανσης από ένα σύνολο 20 ημερών - κορυφές μόλυνσης θεωρούνται οι μέρες εκείνες που η μόλυνσή τους είναι μεγαλύτερη από τη μόλυνση της προηγούμενης και της επόμενης μέρας - η μόλυνση έχει τιμή από 0 έως 100.

#### Λύση

Για  $i$  από 1 μέχρι 20

Αρχή\_επανάληψης

Εμφάνισε "Δώσε την μόλυνση"

Διάβασε ΜΟΛ[ $i$ ]

Μέχρις\_ότου ΜΟΛ[ $i$ ] >=0 και ΜΟΛ[ $i$ ] <=100

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι 20

Αν (ΜΟΛ[ $i$ ] > ΜΟΛ[ $i+1$ ] ΚΑΙ ΜΟΛ[ $i$ ] > ΜΟΛ[ $i-1$ ]) τότε

Εμφάνισε "Υπάρχει κορυφή μόλυνσης την μέρα",  $i$ );

Τέλος\_Αν

Τέλος\_επανάληψης

6. Σε μια μπουτίκ διατίθενται παντελόνια και φούστες από τρεις διαφορετικές μάρκες: ΜαρκΑ, ΜαρκΒ και ΜαρκΓ. Το χρώμα του κάθε ρούχου είναι κόκκινο, πράσινο, μπλε και μαύρο. Γράψτε πρόγραμμα το οποίο εμφανίζει όλα τα είδη ρούχων που μπορεί να αγοράσει ένας πελάτης.

#### Λύση

ΠΡΟΓΡΑΜΜΑ ΡΟΥΧΑ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: κ, λ, μ

ΧΑΡΑΚΤΗΡΕΣ: ΚΑΤ[2], ΧΡ[4], ΜΑΡ[3]

ΑΡΧΗ

ΚΑΤ[1] ← 'παντελόνια' ! Αρχικοποίηση πίνακα από τον προγραμματιστή

ΚΑΤ[2] ← 'φούστες'

ΧΡ[1] ← 'κόκκινο'

ΧΡ[2] ← 'πράσινο'

ΧΡ[3] ← 'μπλε'

ΧΡ[4] ← 'μαύρο'

ΜΑΡ[1] ← 'ΜαρκΑ'

ΜΑΡ[2] ← 'ΜαρκΒ'

ΜΑΡ[3] ← 'ΜαρκΓ'

Για κ ΑΠΟ 1 ΜΕΧΡΙ 3

    Για λ ΑΠΟ 1 ΜΕΧΡΙ 2

        Για μ ΑΠΟ 1 ΜΕΧΡΙ 4

            ΓΡΑΨΕ ΜΑΡ[κ], ΚΑΤ[λ], ΧΡ[μ]

            ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

        ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

    ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

**7.** Θέλετε να προμηθευτείτε μια εξάδα εμφιαλωμένων νερών (ίδιας χωρητικότητας το καθένα) και διαπιστώνετε ότι το σούπερ μάρκετ της γειτονιάς σας εμπορεύεται 13 διαφορετικές μάρκες. Γράψτε πρόγραμμα το οποίο:

- α) διαβάζει τις μάρκες των νερών και τις καταχωρεί στον αλφαριθμητικό πίνακα  $M[13]$ ,
- β) διαβάζει τις τιμές πώλησης κάθε εξάδας και τις καταχωρεί στον πραγματικό πίνακα  $T[13]$  (θεωρείστε ότι οι τιμές αυτές είναι διαφορετικές μεταξύ τους),
- γ) υπολογίζει και εμφανίζει: τη μάρκα της εξάδας που είναι πιο συμφέρουσα για αγορά και το μέσο όρο τιμής πώλησης κάθε μπουκαλιού.

### **Λύση**

**ΠΡΟΓΡΑΜΜΑ ΝΕΡΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** κ, ΠΙΟ

**ΧΑΡΑΚΤΗΡΕΣ:**  $M[13]$

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $T[13]$ , ΜΙΚ, Σ, ΜΟ

**ΑΡΧΗ**

**Για κ ΑΠΟ 1 ΜΕΧΡΙ 13**

**ΔΙΑΒΑΣΕ**  $M[k]$

**ΔΙΑΒΑΣΕ**  $T[k]$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

$ΜΙΚ \leftarrow T[1]$

$ΠΙΟ \leftarrow 1$

**Για κ ΑΠΟ 2 ΜΕΧΡΙ 13**

**Αν**  $T[k] < ΜΙΚ$  **ΤΟΤΕ**

$ΜΙΚ \leftarrow T[k]$

$ΠΙΟ \leftarrow κ$

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ**  $M[ΠΙΟ]$

$Σ \leftarrow 0$

**Για κ ΑΠΟ 1 ΜΕΧΡΙ 13**

$Σ \leftarrow Σ + T[k] / 6$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

$ΜΟ \leftarrow Σ / 13$

**ΓΡΑΨΕ** ΜΟ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**8.** Ας υποθέσουμε ότι μας δίνουν έναν πίνακα, π.χ. 10 θέσεων και μας ζητάνε να βρούμε αν ο πίνακας αυτός είναι ταξινομημένος ή όχι, π.χ. σε αύξουσα διάταξη. Για να προκύψει το συμπέρασμα, θα πρέπει να ελέγξουμε τα διαδοχικά στοιχεία του πίνακα. Δηλαδή, το 1ο με το 2ο, το 2ο με το 3ο, το 3ο με το 4ο κ.ο.κ. Για να είναι ο πίνακας ταξινομημένος σε αύξουσα διάταξη θα πρέπει κάθε προηγούμενο στοιχείο να είναι μικρότερο από κάθε επόμενο. Δηλαδή, το  $i$  στοιχείο του πίνακα να είναι μικρότερο από το  $i+1$ . Προφανώς, αν βρούμε έστω κι ένα ζεύγος διαδοχικών στοιχείων για το οποίο να μην ισχύει το παραπάνω, τότε ο πίνακας δεν είναι ταξινομημένος, οπότε δεν υπάρχει λόγος να συνεχίσουμε. Ο αριθμός των συγκρίσεων που θα κάνουμε θα είναι το πολύ όσο το

μέγεθος του πίνακα μείον ένα (π.χ για τον πίνακα 10 θέσεων που προαναφέραμε θα κάνουμε το πολύ 10 συγκρίσεις).

### Λύση

Θα εφαρμόσουμε σειριακή αναζήτηση για να βρούμε κάποιο στοιχείο  $i$  που να είναι μεγαλύτερο από το  $i+1$ . Αν βρούμε τέτοιο τότε προφανώς ο πίνακας δεν είναι ταξινομημένος.

**Αλγόριθμος** Είναι\_ταξινομημένος

**Δεδομένα** //A//

βρέθηκε  $\leftarrow$  ΨΕΥΔΗΣ

$i \leftarrow 1$

**Όσο**  $i \leq 9$  **ΚΑΙ** βρέθηκε = ΨΕΥΔΗΣ **επανάλαβε** !το  $i$  είναι μέχρι 9 γιατί ψάχνω μέχρι το  $i+1=10$

**Αν**  $A[i] > A[i+1]$  **τότε**

βρέθηκε  $\leftarrow$  ΑΛΗΘΗΣ

**αλλιώς**

$i \leftarrow i + 1$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Αν** βρέθηκε = ΑΛΗΘΗΣ **τότε**

**Εμφάνισε** "Ο πίνακας δεν είναι ταξινομημένος σε αύξουσα σειρά"

**αλλιώς**

**Εμφάνισε** "Ο πίνακας είναι ταξινομημένος σε αύξουσα σειρά"

**Τέλος\_Αν**

**Τέλος** Είναι\_ταξινομημένος

**9.** Όταν το πρωτάθλημα ποδοσφαίρου τελειώσει, βραβεύεται ο ποδοσφαιριστής που έχει σημειώσει τα περισσότερα τέρματα. Ωστόσο, αν υπάρχουν παραπάνω από ένας ποδοσφαιριστές με τον ίδιο αριθμό τερμάτων, νικητής βγαίνει αυτός που έχει τα λιγότερα λεπτά συμμετοχής. Να γίνει αλγόριθμος, που με δεδομένα τα ονόματα, τον αριθμό των τερμάτων και τα λεπτά συμμετοχής (στους πίνακες ΟΝ, ΓΚΟΛ και ΛΣ αντίστοιχα) των 400 ποδοσφαιριστών που συμμετέχουν στο πρωτάθλημα, να εντοπίζει και να εμφανίζει το όνομα του πρώτου σκόρερ.

### Λύση

**Αλγόριθμος** ποδόσφαιρο

**Δεδομένα** //ΟΝ, ΓΚΟΛ, ΛΣ//

! Έστω, ότι καλύτερος είναι ο πρώτος ...

$\max \leftarrow 1$

**Για**  $i$  από 2 μέχρι 400

! Αν ο  $i$ -στός παίκτης, έχει περισσότερα γκολ από τον  $\max$  ή Αν έχει τα ίδια

! αλλά λιγότερο λεπτά συμμετοχής ...

**Αν**  $\Gamma\text{ΚΟΛ}[i] > \Gamma\text{ΚΟΛ}[\max]$  **Η** ( $\Gamma\text{ΚΟΛ}[i] = \Gamma\text{ΚΟΛ}[\max]$  **ΚΑΙ**  $\Lambda\text{Σ}[i] < \Lambda\text{Σ}[\max]$ ) **τότε**

$\max \leftarrow i$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Εμφάνισε** ΟΝ[ $\max$ ]

**Τέλος** ποδόσφαιρο



**10.** Να γίνει αλγόριθμος ο οποίος θα εμφανίζει στο χρήστη 4 ερωτήσεις. Ο χρήστης θα πρέπει να απαντήσει σε όλες τις ερωτήσεις. Για κάθε ερώτηση θα έχει τρεις ευκαιρίες. Δηλαδή για κάθε ερώτηση θα μπορεί να δώσει μέχρι και τρεις απαντήσεις. Ο αλγόριθμος θα εμφανίζει την επόμενη ερώτηση όταν ο χρήστης απαντά σωστά ή όταν ξεπεράσει τις τρεις προσπάθειες. Για κάθε ερώτηση που ο χρήστης απαντά με την πρώτη ευκαιρία κερδίζει 6 πόντους. Με τη δεύτερη ευκαιρία κερδίζει 4 και με την τρίτη ευκαιρία 2 πόντους. Να γίνει αλγόριθμος που θα εμφανίζει τις ακόλουθες ερωτήσεις και ανάλογα με τις απαντήσεις του παίκτη, στο τέλος να εμφανίζει το σκορ του.

Ερώτηση 1: Ποιο είναι το αποτέλεσμα της πράξης  $12 + 24$ ;

Ερώτηση 2: Ποιο είναι το αποτέλεσμα της πράξης  $122 - 67$ ;

Ερώτηση 3: Ποιο είναι το αποτέλεσμα της πράξης  $6 * 12$ ;

Ερώτηση 4: Ποιο είναι το αποτέλεσμα της πράξης  $81 / 3$ ;

### **Λύση**

#### **Αλγόριθμος Quiz**

*! Οργάνωσε τα δεδομένα μέσα σε πίνακες.*

*! Αυτό θα μας βοηθήσει κατά την επανάληψη.*

Ερωτήσεις[1] ← "Ποιο είναι το αποτέλεσμα της πράξης  $12 + 24$ ;"

Ερωτήσεις[2] ← "Ποιο είναι το αποτέλεσμα της πράξης  $122 - 67$ ;"

Ερωτήσεις[3] ← "Ποιο είναι το αποτέλεσμα της πράξης  $6 * 12$ ;"

Ερωτήσεις[4] ← "Ποιο είναι το αποτέλεσμα της πράξης  $81 / 3$ ;"

Απαντήσεις[1] ← 36

Απαντήσεις[2] ← 55

Απαντήσεις[3] ← 72

Απαντήσεις[4] ← 27

Σ\_πόντοι ← 0

**Για i από 1 μέχρι 4**

**Εμφάνισε** Ερωτήσεις[i]

**Διάβασε** απάντηση\_χρήστη

*! Αυτή είναι η πρώτη προσπάθεια.*

προσπάθεια ← 1

*! Αν η απάντηση δεν είναι σωστή, ζητά νέα απάντηση, αρκεί οι*

*! προσπάθειες που έχεις κάνει να μην είναι τρεις.*

*! Αν είναι τρεις σταμάτα την επανάληψη*

**Όσο** απάντηση\_χρήστη <> Απαντήσεις[i] **ΚΑΙ** προσπάθεια < 3 **επανάλαβε**

προσπάθεια ← προσπάθεια + 1

**Εμφάνισε** "Κάνατε λάθος, προσπαθήστε ξανά:"

**Διάβασε** απάντηση\_χρήστη

**Τέλος επανάληψης**

*! Εφόσον η απάντηση του χρήστη είναι σωστή, υπολόγισε τους πόντους που &του αναλογούν*

**Αν** απάντηση\_χρήστη = Απαντήσεις[i] **τότε**

*! Ανάλογα με τον αριθμό των προσπαθειών, οι πόντοι μειώνονται κατά δύο.*

πόντοι\_ερώτησης ← 6 - (προσπάθεια - 1) \* 2

Σ\_πόντοι ← Σ\_πόντοι + πόντοι\_ερώτησης

**Τέλος\_Αν**

## Τέλος επανάληψης

Εμφάνισε "Οι συνολικοί πόντοι που συγκέντρωσες είναι: ",  $\Sigma$  πόντοι

Τέλος Quiz

**11.** Ένας πίνακας λέγεται **αραιός** αν μεγάλο ποσοστό των στοιχείων του (έστω 80%) έχουν μηδενική τιμή. Το πρόβλημα είναι ότι δαπανούν μεγάλο χώρο μνήμης για την αποθήκευση μηδενικών. Ένας οικονομικός τρόπος διαχείρισής τους είναι ο εξής: Αντί να αποθηκεύουμε τα στοιχεία σε έναν δισδιάστατο πίνακα, δημιουργούμε έναν μονοδιάστατο όπου τοποθετούμε μόνο τα μη μηδενικά στοιχεία καθώς και την αντίστοιχη γραμμή και στήλη όπου βρίσκονται. Έτσι για κάθε μη μηδενικό στοιχείο χρησιμοποιούμε μια τριάδα στοιχείων {γραμμή, στήλη, τιμή} που αποθηκεύεται στο μονοδιάστατο πίνακα. Να γραφεί αλγόριθμος οποίος να δέχεται έναν αραιό δισδιάστατο πίνακα A με N γραμμές και M στήλες και θα υλοποιεί την παραπάνω επεξεργασία παράγοντας τον αντίστοιχο μονοδιάστατο πίνακα. Στη συνέχεια υπολογίζει και εκτυπώνει το ποσοστό των μηδενικών στο δισδιάστατο πίνακα.

### Λύση

**Αλγόριθμος** Αραιός\_Πίνακας

Δεδομένα // A, N, M //

K ← 0

Π ← 0

Για i από 1 μέχρι N

    Για j από 1 μέχρι M

        Αν  $A[i, j] \neq 0$  τότε

$B[K+1] \leftarrow i$

$B[K+2] \leftarrow j$

$B[K+3] \leftarrow B[i, j]$

$K \leftarrow K + 3$

            αλλιώς

$\Pi \leftarrow \Pi + 1$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

Ποσοστό  $\leftarrow \Pi / (N * M) * 100$

Εκτύπωσε Ποσοστό, " % "

Τέλος Αραιός\_Πίνακας

**12.** Να γραφεί αλγόριθμος ο οποίος να βελτιώνει τον αλγόριθμο ταξινόμησης **φυσάλιδας** ενός πίνακα κατά φθίνουσα σειρά, ελέγχοντας αν έχουν γίνει αντιμεταθέσεις στοιχείων οπότε συνεχίζει τις συγκρίσεις ή αντιθέτως αν δεν έχει γίνει καμία αντιμετάθεση σε αυτήν την περίπτωση σταματά. Έτσι πραγματοποιεί λιγότερες συγκρίσεις (**βελτιστοποίηση**).

### Λύση

**Αλγόριθμος** Εξυπνος\_Αλγόριθμος\_Φυσάλιδας

δεδομένα //A, N//

i ← 2

ταξινομημένος ← ψευδής

**Όσο** ταξινομημένος= ψευδής και  $i \leq N$  επανάλαβε

ταξινομημένος ← αληθής

**Για**  $j$  από  $N$  μέχρι  $i$  με βήμα  $-1$

**Αν**  $A[j-1] < A[j]$  τότε

**Αντιμετάθεσε**  $A[j-1], A[j]$  ! Επιτρέπεται σε αλγορίθμους ταξινομημένους ← ψευδής

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

$i \leftarrow i + 1$

**Τέλος\_επανάληψης**

**Τέλος** Εξυπνος\_Αλγόριθμος\_Φουσαλίδας

**13.** Σε κάποιο τυχερό παιχνίδι κληρώνονται 15 αριθμοί από το 1 μέχρι και το 80. Κάποιος παίκτης πρέπει να σημειώσει 8 αριθμούς, με σκοπό να πετύχει όσο το δυνατόν περισσότερους από τους 15 που κληρώθηκαν. Ο παίκτης κερδίζει σε κάθε μια από τις ακόλουθες περιπτώσεις:

- 1η κατηγορία κέρδους: Αν σημειώσει 8 επιτυχίες (πετύχει και τους 8 αριθμούς).
- 2η κατηγορία κέρδους: Αν σημειώσει 7 επιτυχίες.
- 3η κατηγορία κέρδους: Αν σημειώσει 6 επιτυχίες.

Να αναπτυχθεί αλγόριθμος, ο οποίος να:

1. διαβάζει τους 15 αριθμούς που κληρώθηκαν και θα τους αποθηκεύει σε πίνακα  $A[15]$ .
2. διαβάζει τους 8 αριθμούς που έπαιξε κάποιος παίκτης και θα τους αποθηκεύει σε πίνακα  $B[8]$ .
3. υπολογίζει τις επιτυχίες του παίκτη δηλαδή όσους αριθμούς έτυχε.
4. υπολογίζει και θα εμφανίζει την κατηγορία κέρδους του δελτίου του παίκτη. Σε περίπτωση που δεν ανήκει σε καμία κατηγορία, να εμφανίζει το μήνυμα "Δυστυχώς δεν κερδίζετε".

**Λύση**

**Αλγόριθμος** Παιχνίδι

! Ερώτημα 1

**Για**  $I$  από 1 μέχρι 15

**Εμφάνισε** "Δώσε το",  $I$ , "αριθμό που κληρώθηκε"

διάβασε  $A[I]$

**Τέλος\_επανάληψης**

! Ερώτημα 2

**Για**  $I$  από 1 μέχρι 8

**Εμφάνισε** "Δώσε το",  $I$ , "αριθμό που έπαιξε ο παίκτης"

**Διάβασε**  $B[I]$

**Τέλος\_επανάληψης**

! Ερώτημα 3

!Υπολογισμός επιτυχιών του δελτίου του παίκτη. Ο αλγόριθμος θα αναζητάει κάθε στοιχείο του πίνακα  $B$  στον πίνακα  $A$  με σειριακή αναζήτηση. Για κάθε στοιχείο του πίνακα  $B$  που υπάρχει στον πίνακα  $A$ , να αυξάνει την τιμή ενός μετρητή επιτυχιών κατά 1.

$N \leftarrow 0$

**Για**  $K$  από 1 μέχρι 8

αριθμός  $\leftarrow B[K]$

$I \leftarrow 1$

βρέθηκε ←ψευδής

Όσο βρέθηκε=ψευδής και  $I \leq 15$  επανάλαβε

Αν  $A[I]=$ αριθμός τότε

βρέθηκε ←αληθής

αλλιώς

$I \leftarrow I+1$

Τέλος\_Αν

Τέλος\_επανάληψης

Αν βρέθηκε = αληθής τότε

$N \leftarrow N+1$

Τέλος\_Αν

Τέλος\_επανάληψης

Αν  $N=8$  τότε

Εμφάνισε "1η κατηγορία κέρδους"

αλλιώς\_Αν  $N=7$  τότε

Εμφάνισε "2η κατηγορία κέρδους"

αλλιώς\_Αν  $N=6$  τότε

Εμφάνισε "3η κατηγορία κέρδους"

αλλιώς

Εμφάνισε "Δυστυχώς δεν κερδίζετε"

Τέλος\_Αν

Τέλος Παιχνίδι

**14.** Δυο φίλοι παίζουν ένα παιχνίδι. Ο πρώτος εισάγει έναν αριθμό από το 1 έως το 100. Ο δεύτερος έχει τρεις προσπάθειες να τον βρει. Αν τον βρει, κερδίζει. Το παιχνίδι να επαναλαμβάνεται μέχρι να δοθεί το ΟΧΙ ως απάντηση στην ερώτηση: «θέλετε να ξαναπαίξετε;». Στο παιχνίδι να εμφανίζεται:

1. Κατάλληλο μήνυμα αν ο παίχτης κέρδισε.
2. Πόσο κοντά έφθασε, στην περίπτωση που δεν κέρδισε.
3. Το συνολικό αριθμό νικών στις παρτίδες που έπαιξαν.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ** παιχνίδι

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** I, AP, EL, PR, Δ, ΝΙΚΕΣ

**ΛΟΓΙΚΕΣ:** Φ

**ΧΑΡΑΚΤΗΡΕΣ:**ΑΠ

**ΑΡΧΗ**

ΝΙΚΕΣ ← 0

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

ΕΜΦΑΝΙΣΕ "ΠΡΩΤΟΣ ΠΑΙΧΤΗΣ: ΔΩΣΕ ΑΡΙΘΜΟ ΑΠΟ 1 ΕΩΣ 100"

**ΔΙΑΒΑΣΕ** AP

**ΜΕΧΡΙΣ\_ΟΤΟΥ** AP  $\geq 1$  ΚΑΙ AP  $\leq 100$

Φ ← ΨΕΥΔΗΣ

EL ← 100

I ← 0

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

```

ΓΡΑΨΕ 'ΠΡΩΤΟΣ ΠΑΙΧΤΗΣ:ΔΩΣΕ ΕΝΑΝ ΑΡΙΘΜΟ ΑΠΟ ΤΟ 1 ΕΩΣ ΤΟ 100'
ΔΙΑΒΑΣΕ ΠΡ
Δ ← A_T(ΠΡ - ΑΡ)
ΑΝ Δ = 0 ΤΟΤΕ
Φ ← ΑΛΗΘΗΣ
ΑΛΛΙΩΣ
Ι ← Ι + 1
ΤΕΛΟΣ_ΑΝ
ΑΝ Δ < ΕΛ ΤΟΤΕ
ΕΛ ← Δ
ΤΕΛΟΣ_ΑΝ

```

```

ΜΕΧΡΙΣ_ΟΤΟΥ Ι = 3 Η Φ = ΑΛΗΘΗΣ
ΑΝ Φ = ΑΛΗΘΗΣ ΤΟΤΕ
ΓΡΑΨΕ 'ΚΕΡΔΙΣΕΣ, Ο ΑΡΙΘΜΟΣ ΕΙΝΑΙ ΤΟ:', ΑΡ
ΝΙΚΕΣ ← ΝΙΚΕΣ + 1
ΑΛΛΙΩΣ
ΓΡΑΨΕ 'ΕΧΑΣΕΣ, ΕΦΘΑΣΕΣ', ΕΛ, 'ΜΟΝΑΔΕΣ ΚΟΝΤΑ'
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ 'ΘΕΛΕΤΕ ΝΑ ΞΑΝΑΠΑΙΞΕΤΕ;'
ΔΙΑΒΑΣΕ ΑΠ
ΜΕΧΡΙΣ_ΟΤΟΥ ΑΠ= 'ΟΧΙ'
ΓΡΑΨΕ ' Ο ΠΑΙΧΤΗΣ ΝΙΚΗΣΕ', ΝΙΚΕΣ, 'ΦΟΡΕΣ'
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

**15.** Υπολογισμός νομισμάτων: κάθε στιγμή θα γίνεται επιλογή του νομίσματος που αντιστοιχεί στο μεγαλύτερο δυνατό ποσό. Τα βήματα επιλογής νομισμάτων περιγράφονται στον επόμενο αλγόριθμο, που παρουσιάζει μία απλοποιημένη έκδοση του προβλήματος.

### **Λύση**

**Αλγόριθμος** Νομίσματα

**Δεδομένα** // C, n, poso //

find ← poso

coins ← 0

choice ← n

**Όσο** (choice>0) **και** (find > 0) **επανάλαβε**

**Αν** C[choice]<= find **τότε**

coins ← coins + 1

find ← find - C[choice]

αλλιώς

choice ← choice - 1

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Αποτελέσματα** // coins //

**Τέλος** Νομίσματα

Στον αλγόριθμο αυτό ο πίνακας C διαθέτει όλες τις τιμές των n χρησιμοποιούμενων νομισμάτων από το μικρότερο προς το μεγαλύτερο. Ο αλγόριθμος υπολογίζει τον ελάχιστο αριθμό νομισμάτων για ένα συγκεκριμένο ποσό (μεταβλητή coins).

**16.** Αναπτύξτε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει έναν ακέραιο αριθμό και να υπολογίζει ποιο ή ποια ψηφία του εμφανίζονται πιο πολλές φορές. Παρατήρηση: θεωρήστε ότι ο αριθμός είναι ακέραιος και δε χρειάζεται να κάνετε έλεγχο. Επεξήγηση: Για να απαντήσουμε ποιο ψηφίο εμφανίζεται πιο πολλές φορές, πρέπει να βρούμε πόσες φορές εμφανίζεται το κάθε ένα. Αντί λοιπόν να δημιουργήσουμε 10 διαφορετικά πλήθη, που θα αυξάνονται σε μια πολύ μεγάλη δομή επιλογής, δημιουργούμε έναν πίνακα 10 θέσεων με το κάθε πλήθος. Το πρόβλημα είναι ότι τα πλήθη του μηδέν δεν μπορούν να αποθηκευτούν στη μηδενική θέση του πίνακα γιατί δεν υπάρχει. Έτσι τα αποθηκεύουμε μία θέση δεξιά στον πίνακα. Στη συνέχεια βρίσκουμε το μεγαλύτερο πλήθος και εμφανίζουμε τις θέσεις στις οποίες υπάρχει, με μία αλλαγή: Αν το max βρεθεί στην 5η θέση σημαίνει ότι το ψηφίο 4 έχει εμφανιστεί πιο πολλές φορές με βάση αυτό που είπαμε παραπάνω.

### Λύση

**ΠΡΟΓΡΑΜΜΑ** ΑΝΑΛΥΣΗ\_ΑΡΙΘΜΟΥ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** ΠΛ[10],Χ,ΜΑΧ,Ι,Τ\_Ψ

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** Χ

**ΓΙΑ** Ι **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

ΠΛ[Ι] ← 0

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΟΣΟ** Χ<>0 **ΕΠΑΝΑΛΑΒΕ**

Τ\_Ψ ← Χ MOD 10

ΠΛ[Τ\_Ψ+1] ← ΠΛ[Τ\_Ψ+1]+1

Χ ← Χ DIV 10

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΜΑΧ ← ΠΛ[1]

**ΓΙΑ** Ι **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

**ΑΝ** ΠΛ[Ι]>ΜΑΧ **ΤΟΤΕ**

ΜΑΧ ← ΠΛ[Ι]

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ** Ι **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

**ΑΝ** ΠΛ[Ι]=ΜΑΧ **ΤΟΤΕ**

**ΓΡΑΨΕ** 'ΤΟ ΨΗΦΙΟ', Ι-1, 'ΕΜΦΑΝΙΖΕΤΑΙ', ΠΛ[Ι], 'ΦΟΡΕΣ'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**17.** Έστω ένας ταξινομημένος πίνακας 100 θέσεων. Αν επιθυμούσαμε αντίστροφη ταξινόμηση από την υπάρχουσα θα ήταν καλή λύση ο αλγόριθμος ταξινόμησης ευθείας ανταλλαγής-φυσαλίδας; Πόσες συγκρίσεις και αντιμεταθέσεις κάνει; Μπορείτε να προτείνετε κάποιον καλύτερο;

### Λύση

Για κάθε λειτουργία δημιουργείται και ένας αλγόριθμος. Αρκετές φορές υπάρχουν διαφορετικοί αλγόριθμοι που υλοποιούν μία συγκεκριμένη λειτουργία. Όταν συμβαίνει αυτό, επιλέγουμε τον αλγόριθμο που είναι πιο αποδοτικός (π.χ. πιο γρήγορος) για τα συγκεκριμένα δεδομένα. Ο αλγόριθμος ταξινόμησης ευθείας ανταλλαγής-φυσάλιδας θα έκανε επαναλήψεις  $99(i=2)+98(i=3)+97(i=4)+\dots+3(i=97)+2(i=98)+1(i=100)$ . Παρατηρούμε ότι ο πίνακάς μας έχει μια ιδιομορφία. Τα στοιχεία του είναι ήδη ταξινομημένα (αντίστροφα από ότι θέλουμε). Οπότε δεν χρειάζεται να συγκρίνουμε κάθε φορά ένα στοιχείο με το γειτονικό του, αλλά μπορούμε να το αντιμετωπίσουμε κατευθείαν με το σωστό στοιχείο. Δηλαδή το  $1^{\circ}$  με το  $100^{\circ}$ , το  $2^{\circ}$  με το  $99^{\circ}$ , το  $3^{\circ}$  με το  $98^{\circ}$  κ.ο.κ. κάνοντας έτσι μόνο 50 επαναλήψεις:

**Για  $i$  από 1 μέχρι 50**

temp  $\leftarrow$  Π[i]

Π[i]  $\leftarrow$  Π[101- i]

Π[101- i]  $\leftarrow$  temp

**Τέλος \_επανάληψης**

**18.** Σε μία βιβλιοθήκη ο υπεύθυνος θέλει να καταχωρήσει σε δύο πίνακες ΟΝΟΜΑ και ΑΝΤΙΤΥΠΑ όλα τα διαφορετικά βιβλία που υπάρχουν και τον αριθμό τον αντιτύπων τους αντίστοιχα. Στο τέλος θα παραγγείλει όσα βιβλία υπάρχουν μόνο μία φορά. Η εισαγωγή θα γίνεται με τον εξής τρόπο: θα πληκτρολογεί τον τίτλο κάθε βιβλίου και για να τερματίσει θα πληκτρολογήσει τη λέξη ΤΕΛΟΣ. Να γραφεί αλγόριθμος ο οποίος θα:

A. διαβάσει το όνομα ενός βιβλίου, αν δεν έχει ήδη καταχωρηθεί στη βάση δεδομένων να δημιουργήσει μία νέα εγγραφή, διαφορετικά θα ενημερώνει τον αριθμό αντιτύπων του.

B. υπολογίζει και θα εμφανίζει πόσα βιβλία υπάρχουν μόνο μια φορά. (Θέμα Πανελλαδικών)

**Λύση**

**ΑΛΓΟΡΙΘΜΟΣ ΒΙΒΛΙΟΠΩΛΕΙΟ**

ΠΛ  $\leftarrow$  0

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

ΠΛ  $\leftarrow$  ΠΛ+1

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ** ΟΝΟΜΑ[ΠΛ]

ΒΡΕΘΗΚΕ  $\leftarrow$  ΨΕΥΔΗΣ

K  $\leftarrow$  1

**ΟΣΟ** K  $\leq$  ΠΛ - 1 **ΚΑΙ** ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ **ΕΠΑΝΑΛΑΒΕ**

**ΑΝ** ΟΝΟΜΑ[ΠΛ] = ΟΝΟΜΑ[K] **ΤΟΤΕ**

ΒΡΕΘΗΚΕ  $\leftarrow$  **ΑΛΗΘΗΣ**

**ΑΛΛΙΩΣ**

K  $\leftarrow$  K + 1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ** ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ **ΤΟΤΕ**

ΑΝΤΙΤΥΠΑ[ΠΛ]  $\leftarrow$  1

**ΑΛΛΙΩΣ**

ΑΝΤΙΤΥΠΑ[K]  $\leftarrow$  ΑΝΤΙΤΥΠΑ[K] + 1

**ΤΕΛΟΣ\_ΑΝ**

```

ΜΕΧΡΙΣ_ΟΤΟΥ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ
ΜΕΧΡΙΣ_ΟΤΟΥ ΟΝΟΜΑ[ΠΛ] = "ΤΕΛΟΣ"
ΠΛ ← ΠΛ-1 !ΕΠΕΙΔΗ ΤΟ "ΤΕΛΟΣ" ΔΕΝ ΕΙΝΑΙ ΒΙΒΛΙΟ
ΜΟΝΑΔΙΚΑ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ ΠΛ
    ΑΝ ΑΝΤΙΤΥΠΑ[Ι]=1 ΤΟΤΕ
        ΜΟΝΑΔΙΚΑ ← ΜΟΝΑΔΙΚΑ + 1
        ΕΜΦΑΝΙΣΕ "ΤΟ ΒΙΒΛΙΟ", ΟΝΟΜΑ[Ι], "ΕΧΕΙ ΜΟΝΟ ΕΝΑ ΑΝΤΙΤΥΠΟ"
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΕΜΦΑΝΙΣΕ "ΤΑ ΒΙΒΛΙΑ ΜΕ ΕΝΑ ΑΝΤΙΤΥΠΟ ΜΟΝΟ ΕΙΝΑΙ", ΜΟΝΑΔΙΚΑ
ΤΕΛΟΣ ΒΙΒΛΙΟΠΩΛΕΙΟ

```

**19.** Να γραφεί αλγόριθμος σε "ΓΛΩΣΣΑ" ο οποίος:

Α. Να διαβάξει το πλήθος των στοιχείων ενός μονοδιάστατου πίνακα με όνομα Α. Το πλήθος δε θα πρέπει να υπολείπεται του 2 αλλά ούτε και να υπερβαίνει το 100.

Β. Να δέχεται από το πληκτρολόγιο όλα τα στοιχεία του πίνακα τα οποία πρέπει να είναι θετικοί ακέραιοι αριθμοί.

Γ. Να δημιουργεί ένα μονοδιάστατο πίνακα Β κάθε στοιχείο του οποίου πρέπει να είναι το γινόμενο των γειτονικών στοιχείων του Α. Για παράδειγμα, το στοιχείο Β[1] πρέπει να είναι το Α[1] \* Α[2], το Β[2] είναι το Α[2] \* Α[3], κ.ο.κ. Για τον υπολογισμό του γινομένου να εφαρμοστεί η μέθοδος του πολλαπλασιασμού αλά ρωσικά.

Δ. Να εμφανίζει (με κατάλληλα διαμορφωμένο μήνυμα) κάθε ζευγάρι στοιχείων του πίνακα Α καθώς και το αντίστοιχο γινόμενο του.

Ε. Να υπολογίζει το μεγαλύτερο γινόμενο και να εμφανίζει (με κατάλληλα διαμορφωμένο μήνυμα) ποιο ζευγάρι από τα στοιχεία του Α το είχε καθώς ποιοι ήταν οι δείκτες τους στον πίνακα Α.

**Λύση**

```

ΠΡΟΓΡΑΜΜΑ Θέμα1
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: Α[100]
ΑΚΕΡΑΙΕΣ: Β[99], Γ[99,2], Ι, ΜΑΧ, ΘΕΣΗ, Ν, ΓΙΝ, Κ, Μ1, Μ2, Λ
ΑΡΧΗ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Δώστε το πλήθος των στοιχείων του πίνακα Α. Να είναι >= 2 και <= 100'
ΔΙΑΒΑΣΕ Ν
ΜΕΧΡΙΣ_ΟΤΟΥ (Ν >= 2) ΚΑΙ (Ν <= 100)
Για Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΡΑΨΕ 'Δώστε στοιχεία πίνακα Α'
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΔΙΑΒΑΣΕ Α[Ι]
        ΜΕΧΡΙΣ_ΟΤΟΥ (Α[Ι] > 0) ΚΑΙ (Α_Μ(Α[Ι]) = Α[Ι])
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΙΝ ← 0
    Κ ← 0
Για Ι ΑΠΟ 1 ΜΕΧΡΙ Ν -1
    Μ1 ← Α[Ι]

```



```

M2 ← A[I+1]
Όσο M2 > 0 ΕΠΑΝΑΛΑΒΕ
    Αν M2 MOD 2 = 1 ΤΟΤΕ
        ΓΙΝ ← ΓΙΝ + M1
    ΤΕΛΟΣ_ΑΝ
M1 ← 2 * M1
M2 ← M2 DIV 2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Κ ← Κ + 1
B[K] ← ΓΙΝ
Γ[K, 1] ← Ι
Γ[K, 2] ← Ι + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Λ ← 0
Για Ι ΑΠΟ 1 ΜΕΧΡΙ Ν - 1
    Λ ← Λ + 1
    ΓΡΑΨΕ 'Στοιχεία A[' , A[I], ']', 'A[I+1', A[I+1], ']' Γινόμενο:', B[Λ]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΑΧ ← B[1]
    ΘΕΣΗ ← 1
    Για Ι ΑΠΟ 2 ΜΕΧΡΙ Κ
        Αν B[I] > ΜΑΧ ΤΟΤΕ
            ΜΑΧ ← B[I]
            ΘΕΣΗ ← Ι
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ 'Στοιχεία με το μεγαλύτερο γινόμενο: A[' , Γ[ΘΕΣΗ, 1], ']' A[' , Γ[ΘΕΣΗ, 2],']
    ΓΡΑΨΕ 'Με δείκτες:', Γ[ΘΕΣΗ, 1], Γ[ΘΕΣΗ, 2]
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

**20.** Γράψτε πρόγραμμα το οποίο:

α) διαβάζει τα ύψη (σε εκατοστά) των μαθητών μιας τάξης (η είσοδος δεδομένων ολοκληρώνεται μόλις δοθεί η απάντηση 'ΟΧΙ' στην υποβαλλόμενη προς το χρήστη ερώτηση "Υπάρχει άλλος μαθητής;").

β) υπολογίζει και εμφανίζει πόσοι από αυτούς έχουν ύψος 160, 161, ..., 175 εκατοστά.

**Λύση**

```

ΠΡΟΓΡΑΜΜΑ ΥΨΗ
ΜΕΤΑΒΛΗΤΕΣ
ΧΑΡΑΚΤΗΡΕΣ: ΑΠ
ΑΚΕΡΑΙΕΣ: Φ[16], Θ, κ, Υ
ΑΡΧΗ
Για κ ΑΠΟ 1 ΜΕΧΡΙ 16
    Φ[κ] ← 0
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ Υ
    Αν (Υ >= 160) ΚΑΙ (Υ <= 175) ΤΟΤΕ

```

```

Θ ← Υ-159
[Θ] ← Φ[Θ]+1
ΤΕΛΟΣ_Αν
ΓΡΑΨΕ 'Υπάρχει άλλος μαθητής;'
ΔΙΑΒΑΣΕ ΑΠ
ΜΕΧΡΙΣ_ΟΤΟΥ ΑΠ = 'ΟΧΙ'
Για κ ΑΠΟ 1 ΜΕΧΡΙ 16
ΓΡΑΨΕ Φ[κ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

**21.** Να γράψετε πρόγραμμα το οποίο θα καταχωρεί σε έναν πίνακα 50 ακέριες τιμές από το 10 έως το 20 και θα βρίσκει την τιμή που εμφανίζεται με μεγαλύτερη συχνότητα. Σε περίπτωση που υπάρχουν δύο ή περισσότεροι αριθμοί με ίδια συχνότητα, τους εμφανίζει όλους.

### Λύση

```

ΠΡΟΓΡΑΜΜΑ Άσκηση
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: ι, j, Μ, ΠΙΝ[50], ΣΥΧΝ[11], ΑΡ[11], max
ΑΡΧΗ
! Γεμίζω έναν πίνακα ΑΡ[11] με αριθμούς από το 10 έως το 20
ΓΙΑ ι ΑΠΟ 10 ΜΕΧΡΙ 20
ΑΡ[ι-9] ← ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
! Γεμίζω τον πίνακα ΠΙΝ[50] με τυχαίους αριθμούς από το 10 έως το 20
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 50
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ ΠΙΝ[ι]
ΜΕΧΡΙΣ_ΟΤΟΥ ΠΙΝ[ι]>=10 ΚΑΙ ΠΙΝ[ι]<=20
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
! Γεμίζω τον πίνακα ΣΥΧΝ[11] με τον αριθμό εμφάνισης κάθε αριθμού στον πίνακα
ΠΙΝ[50]
ΓΙΑ j ΑΠΟ 10 ΜΕΧΡΙ 20
Μ ← 0
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 50
ΑΝ ΠΙΝ[ι]=j ΤΟΤΕ
Μ ← Μ+1
ΤΕΛΟΣ_Αν
ΣΥΧΝ[j-9] ← Μ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
! Βρίσκω το μέγιστο αριθμό εμφάνισης (max) στον πίνακα ΣΥΧΝ[11]
max ← ΣΥΧΝ[1]
ΓΙΑ ι ΑΠΟ 2 ΜΕΧΡΙ 11
ΑΝ max<ΣΥΧΝ[ι] ΤΟΤΕ
max ← ΣΥΧΝ[ι]
ΤΕΛΟΣ_Αν

```

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**! Εμφανίζω τον αριθμό ή τους αριθμούς με την μεγαλύτερη συχνότητα εμφάνισης**

**ΓΡΑΨΕ** 'Βρέθηκε (Αν) στον πίνακα με μεγαλύτερη συχνότητα (' ,max,'φορές' ), ο αριθμός ή οι αριθμοί:'

**ΓΙΑ**  $i$  **ΑΠΟ** 1 **ΜΕΧΡΙ** 11

**ΑΝ** max=ΣΥΧΝ[ $i$ ] **ΤΟΤΕ**

**ΓΡΑΨΕ** AP[ $i$ ]

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**22.** Δίδεται ταξινομημένος μονοδιάστατος πίνακας ακεραίων αριθμών 1000 θέσεων. Ζητείται να κατασκευαστεί πρόγραμμα, το οποίο να βρίσκει τη συχνότητα εμφάνισης κάθε αριθμού του πίνακα. Τα αποτελέσματα να εμφανίζονται στην οθόνη.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ** Συχνότητα1

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $i$ , S, Προηγ\_A, A[1000]

**ΑΡΧΗ**

$i \leftarrow 1$

$S \leftarrow 0$

Προηγ\_A  $\leftarrow$  A[ $i$ ]

**Όσο**  $i < 1001$  **ΕΠΑΝΑΛΑΒΕ**

**Αν** Προηγ\_A  $\neq$  A[ $i$ ] **ΤΟΤΕ**

**ΓΡΑΨΕ** Προηγ\_A, S

Προηγ\_A  $\leftarrow$  A[ $i$ ]

$S \leftarrow 0$

**ΤΕΛΟΣ\_Αν**

$S \leftarrow S+1$

$i \leftarrow i+1$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** Προηγ\_A, S

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

και μία **δεύτερη Λύση:**

**ΠΡΟΓΡΑΜΜΑ** Συχνότητα2

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $i$ , S, Προηγ\_A, A[1000]

**ΑΡΧΗ**

$i \leftarrow 1$

$S \leftarrow 0$

Προηγ\_A  $\leftarrow$  A[ $i$ ]

**Όσο**  $i < 1001$  **ΕΠΑΝΑΛΑΒΕ**

**Αν** Προηγ\_A  $\neq$  A[ $i$ ] **ΤΟΤΕ**

$S \leftarrow S+1$

$i \leftarrow i+1$

**ΑΛΛΙΩΣ**

```
ΓΡΑΨΕ Προηγ_Α, S
Προηγ_Α ← A[i]
S ← 0
ΤΕΛΟΣ_ΑΝ
```

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  
ΓΡΑΨΕ Προηγ\_Α, S  
ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

**23.** Για τις ακόλουθες εντολές να γίνει πίνακας τιμών θεωρώντας ότι κάθε κελί του πίνακα Π αντιστοιχεί σε ένα γράμμα του ονόματός σου. Αν το όνομά σου έχει λιγότερα από 6 γράμματα, θεωρήσε ότι τα υπόλοιπα κελιά του Π περιέχουν το γράμμα "D". Για παράδειγμα, για κάποιον που τον λένε ΒΑΣΙΛΗ, θα θεωρήσει ότι Π[1]="Β", Π[2]="Α", Π[3]="Σ", Π[4]="Ι", Π[5]="Λ", Π[6]="Ε" ενώ για κάποια που τη λένε ΛΙΑ, θα θεωρήσει ότι Π[1]="Λ", Π[2]="Ι", Π[3]="Α", Π[4]="D", Π[5]="D".

**Λύση**

Σ ← 5

Για χ από 2 μέχρι 6

Εμφάνισε χ, Π[χ-1]

Αν Π[χ] > Π[χ-1] τότε

Σ ← Σ \* 10 + 9

αλλιώς\_Αν Π[χ] < Π[χ-1] τότε

Σ ← Σ \* 10 + 1

αλλιώς

Σ ← Σ \* 10

Τέλος\_Αν

Τέλος\_επανάληψης

Εμφάνισε Σ

**24.** Έστω ένα αρχείο με όνομα BOOKS το οποίο αποτελείται από 200 γραμμές. Κάθε γραμμή έχει τις πληροφορίες ISBN (μοναδικός κωδικός του βιβλίου), Τίτλος, Συγγραφέας, Πλήθος σελίδων, Ημερομηνία έκδοσης, Τιμή, Εκδοτικός οίκος.

Α. Αν θέλετε την αποθήκευση του συγκεκριμένου αρχείου, ποιο είδος μνήμης θα επιλέγατε; Γιατί;

Β. Πως ονομάζονται οι γραμμές του αρχείου και πως κάθε ανεξάρτητη πληροφορία της γραμμής;

Γ. Να περιγράψετε την ειδική λειτουργία που έχουν τα πεδία ISBN και Εκδοτικός οίκος στο παραπάνω αρχείο.

**Απάντηση**

Α. Αποθήκευση του αρχείου θα γίνει στη δευτερεύουσα μνήμη (πχ σκληρός δίσκος, DVD κ.λπ.). Επειδή σε αυτή τα δεδομένα παραμένουν μόνιμα σε αντίθεση με την κύρια μνήμη που χάνονται με τον τερματισμό εκτέλεσης του προγράμματος.

Β. Εγγραφές και πεδία αντίστοιχα.

Γ. Το ISBN ταυτοποιεί μοναδικά την κάθε εγγραφή (πρωτεύον κλειδί), ενώ το πεδίο Εκδοτικός οίκος περιγράφει χαρακτηριστικά της κάθε εγγραφής.

**25.** Το ράλλυ Βορείων Σποράδων είναι ένας αγώνας ιστιοπλοΐας ανοικτής θάλασσας που γίνεται κάθε χρόνο. Στην τελευταία διοργάνωση συμμετείχαν 35 σκάφη που διαγωνίστηκαν σε διαδρομή συνολικής απόστασης 70 μιλίων. Κάθε σκάφος ανήκει σε μια από τις κατηγορίες C1, C2, C3. Επειδή στον αγώνα συμμετέχουν σκάφη διαφορετικών δυνατοτήτων, η κατάταξη δεν προκύπτει από τον «πραγματικό» χρόνο τερματισμού, αλλά από ένα «σχετικό» χρόνο, που υπολογίζεται διαιρώντας τον «πραγματικό» χρόνο του σκάφους με τον «ιδανικό». Ο ιδανικός χρόνος είναι διαφορετικός για κάθε σκάφος και προκύπτει πολλαπλασιάζοντας την απόσταση της διαδρομής με τον δείκτη GPH του σκάφους. Ο δείκτης GPH αντιπροσωπεύει τον ιδανικό χρόνο που χρειάζεται το σκάφος για να καλύψει απόσταση ενός μιλίου. Να κατασκευάσετε αλγόριθμο ο οποίος:

1. Να ζητάει για κάθε σκάφος:

- το όνομά του.

- την κατηγορία του ελέγχοντας την ορθή καταχώρηση.

- το χρόνο (σε δευτερόλεπτα) που χρειάστηκε για να τερματίσει

- το δείκτη GPH (σε δευτερόλεπτα).

2. Να υπολογίζει το σχετικό χρόνο κάθε σκάφους.

3. Να εμφανίζει την κατηγορία στην οποία ανήκουν τα περισσότερα σκάφη.

4. Να εμφανίζει για κάθε κατηγορία καθώς και για την γενική κατάταξη τα ονόματα των σκαφών που κερδίζουν μετάλλιο. (Μετάλλια απονέμονται στους 3 πρώτους κάθε κατηγορίας και στους 3 πρώτους της γενικής κατάταξης). Σημείωση: Να θεωρήσετε ότι κάθε κατηγορία έχει διαφορετικό αριθμό σκαφών και τουλάχιστον τρία σκάφη. (Θέμα Πανελλαδικών).

### **Λύση**

#### **Αλγόριθμος Θέμα4**

! ερώτημα 1

! Εισαγωγή τιμών και έλεγχοι

Για  $i$  από 1 μέχρι 35

Διάβασε όνομα[ $i$ ]

    Αρχή\_επανάληψης

    Διάβασε κατηγορία[ $i$ ]

    Μέχρις\_ότου κατηγορία[ $i$ ] = 'C1' Η κατηγορία[ $i$ ] = 'C2' Η κατηγορία[ $i$ ] = 'C3'

Διάβασε χρόνος[ $i$ ], GPH[ $i$ ]

    Τέλος\_επανάληψης

! ερώτημα 2

! υπολογισμός σχετικού χρόνου

Για  $i$  από 1 μέχρι 35

ιδανικός\_χρόνος ←  $70 * GPH[i]$

Σ\_χρόνος[ $i$ ] ← χρόνος[ $i$ ] / ιδανικός\_χρόνος

    Τέλος\_επανάληψης

! ερώτημα 3

! μέτρηση των σκαφών για κάθε κατηγορία

Πλ\_C1 ← 0

Πλ\_C2 ← 0

Πλ\_C3 ← 0

Για  $i$  από 1 μέχρι 35

    Αν κατηγορία[ $i$ ] = 'C1' τότε

Πλ\_C1 ← Πλ\_C1 + 1  
**Αλλιώς\_Αν** κατηγορία[i] = 'C2' **τότε**  
Πλ\_C2 ← Πλ\_C2 + 1  
**Αλλιώς**  
Πλ\_C3 ← Πλ\_C3 + 1  
**Τέλος\_Αν**

Τέλος\_επανάληψης

**Αν** Πλ\_C1 > Πλ\_C2 ΚΑΙ Πλ\_C1 > Πλ\_C3 **τότε**

**Εμφάνισε** 'C1'

**Αλλιώς\_Αν** Πλ\_C2 > Πλ\_C1 ΚΑΙ Πλ\_C2 > Πλ\_C3 **τότε**

**Εμφάνισε** 'C2'

**Αλλιώς**

**Εμφάνισε** 'C3'

**Τέλος\_Αν**

! ερώτημα 4

! Ταξινόμηση των πινάκων ως προς τον σχετικό χρόνο με παράλληλη & αντιμετάθεση

**Για** i από 2 μέχρι 35

**Για** j από 35 μέχρι i με\_βήμα -1

        ! Αύξουσα ταξινόμηση, νικητής είναι ο μικρότερος σχετικός χρόνος

**Αν** Σ\_χρόνος[j] < Σ\_χρόνος[j - 1] **τότε**

**Αντιμετάθεσε** Σ\_χρόνος[j], Σ\_χρόνος[j - 1]

**Αντιμετάθεσε** όνομα[j], όνομα[j - 1]

**Αντιμετάθεσε** κατηγορία[j], κατηγορία [j - 1]

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

! Φυσικά, λύνεται και με την δομή επανάληψης ΓΙΑ, αλλά με την Όσο είναι & αποδοτικότερο (για τον υπολογιστή)

**Εμφάνισε** 'Καλύτεροι κατηγορίας C1'

Πλ ← 0

i ← 0

**Όσο** (i <= 35 και πλ <> 3) **επανάλαβε**

**Αν** κατηγορία[i] = 'C1' **τότε**

**Εμφάνισε** όνομα[i]

        Πλ ← πλ + 1

**Τέλος\_Αν**

i ← i + 1

Τέλος\_επανάληψης

**Εμφάνισε** 'Καλύτεροι κατηγορίας C2'

Πλ ← 0

i ← 0

**Όσο** (i <= 35 και πλ <> 3) **επανάλαβε**

**Αν** κατηγορία[i] = 'C2' **τότε**

**Εμφάνισε** όνομα[i]

        πλ ← πλ + 1

**Τέλος\_Αν**

i ← i + 1

Τέλος\_επανάληψης

**Εμφάνισε** 'Καλύτεροι κατηγορίας C3'

πλ ← 0

ι ← 0

**Όσο** (ι <= 35 και πλ <> 3) **επανάλαβε**

**Αν** κατηγορία[ι] = 'C3' **τότε**

**Εμφάνισε** όνομα[ι]

        πλ ← πλ + 1

**Τέλος\_Αν**

ι ← ι + 1

**Τέλος\_επανάληψης**

**Εμφάνισε** 'Καλύτεροι στην γενική κατάταξη'

**Εμφάνισε** όνομα[1], όνομα[2], όνομα[3]

**Τέλος** Θέμα4

**26.** Ερευνητές που ασχολούνται με μοντέλα προσομοίωσης εξάπλωσης επιδημιών, χρησιμοποιούν για τις μελέτες τους έναν αριθμητικό πίνακα  $M[5000]$ . Κάθε κελί του πίνακα αυτού αντιπροσωπεύει ένα άτομο σε μια περιοχή 5.000 κατοίκων στην οποία υπάρχουν εστίες μιας συγκεκριμένης μολυσματικής ασθένειας (επιδημίας). Από σύμβαση η τιμή μηδέν 0 σε ένα κελί αντιπροσωπεύει ένα υγιές άτομο, ενώ η τιμή -1 αντιπροσωπεύει ένα άτομο που έχει τη συγκεκριμένη ασθένεια (μολυσμένο άτομο). Κάθε άτομο έρχεται σε επαφή με τα γειτονικά του και η ασθένεια μπορεί να μεταδοθεί από τον ένα στον άλλο. (Γειτονικά χαρακτηρίζονται δύο άτομα, όταν τα κελιά του πίνακα που τα αντιπροσωπεύουν έχουν μια κοινή πλευρά). Θεωρήστε ότι δίνεται ο πίνακας  $M$  που περιέχει ήδη έναν αριθμό μολυσμένων ατόμων. Να υλοποιήσετε αλγόριθμο ο οποίος:

Δ1. Υπολογίζει και εμφανίζει με κατάλληλο μήνυμα τον συνολικό αριθμό των μολυσμένων ατόμων που υπάρχουν στο σύνολο του πληθυσμού.

Δ2. Αποθηκεύει σε κάθε κελί του πίνακα  $M$  που αντιπροσωπεύει ένα υγιές άτομο έναν αριθμό ο οποίος δείχνει με πόσα μολυσμένα άτομα γειτονεύει το υγιές.

Δ3. Βρίσκει αν υπάρχει έστω και μία «σημαντική» εστία μόλυνσης. Αν υπάρχει, εμφανίζει το μήνυμα «Υπάρχει σημαντική εστία μόλυνσης» μαζί με τη θέση του πρώτου κελιού της εστίας. Αν δεν υπάρχει, εμφανίζει το μήνυμα «Δεν υπάρχει σημαντική εστία μόλυνσης». (Μια εστία μόλυνσης χαρακτηρίζεται σημαντική, όταν δύο ή περισσότερα μολυσμένα άτομα βρίσκονται σε συνεχόμενα γειτονικά κελιά). (Θέμα Πανελλαδικών)

**Λύση**

**Αλγόριθμος** ΘέμαΔ

**Δεδομένα** //M//

! ερώτημα 1

μολυσμένοι ← 0

**Για** ι **από** 1 **μέχρι** 5000

**Αν**  $M[i] = -1$  **τότε**

        μολυσμένοι ← μολυσμένοι + 1

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Εμφάνισε** μολυσμένοι

! ερώτημα 2

**Αν**  $M[1] = 0$  **ΚΑΙ**  $M[2] = -1$  **τότε**

$M[1] \leftarrow 1$

**Τέλος\_Αν**

```

Αν  $M[5000] = 0$  ΚΑΙ  $M[4999] = -1$  τότε
 $M[5000] \leftarrow -1$ 
Τέλος_Αν
Για  $i$  από 2 μέχρι 4999
    Αν  $M[i] = 0$  τότε
        Αν  $M[i - 1] = -1$  τότε
             $M[i] \leftarrow M[i] + 1$ 
            Τέλος_Αν
            Αν  $M[i + 1] = -1$  τότε
                 $M[i] \leftarrow M[i] + 1$ 
                Τέλος_Αν
        Τέλος_Αν
Τέλος_επανάληψης
! ερώτημα 3
 $i \leftarrow 1$ 
βρέθηκε  $\leftarrow$  ΨΕΥΔΗΣ
Όσο βρέθηκε = ΨΕΥΔΗΣ και  $i \leq 4999$  επανάλαβε
    Αν  $M[i] = -1$  ΚΑΙ  $M[i+1] = -1$  τότε
        βρέθηκε  $\leftarrow$  ΑΛΗΘΗΣ
        θέση  $\leftarrow i$ 
        Τέλος_Αν
     $i \leftarrow i + 1$ 
    Τέλος_επανάληψης
Αν βρέθηκε = ΑΛΗΘΗΣ τότε
Εμφάνισε " Υπάρχει σημαντική εστία μόλυνσης"
Εμφάνισε θέση
Αλλιώς
Εμφάνισε " Δεν υπάρχει σημαντική εστία μόλυνσης"
Τέλος_Αν
Τέλος ΘέμαΔ

```

**27.** Σε μια διαδρομή τρένου υπάρχουν 20 σταθμοί (σε αυτούς περιλαμβάνονται η αφετηρία και ο τερματικός σταθμός). Το τρένο σταματά σε όλους τους σταθμούς. Σε κάθε σταθμό επιβιβάζονται και αποβιβάζονται επιβάτες. Οι πρώτοι επιβάτες επιβιβάζονται στην αφετηρία και στον τερματικό σταθμό αποβιβάζονται όλοι οι επιβάτες. Να κατασκευάσετε αλγόριθμο, ο οποίος να διαχειρίζεται την κίνηση των επιβατών. Συγκεκριμένα:

A. Να ζητάει από το χρήστη τον αριθμό των ατόμων που επιβιβάστηκαν σε κάθε σταθμό, εκτός από τον τερματικό, και να τον εισάγει σε πίνακα  $ΕΠΙΒ[19]$ .

B. Να εισάγει σε πίνακα  $ΑΠΟΒ[19]$  τον αριθμό των ατόμων που αποβιβάστηκαν σε κάθε σταθμό, εκτός από τον τερματικό, ως εξής:

Για την αφετηρία να εισάγει την τιμή μηδέν (0) και για τους υπόλοιπους σταθμούς να ζητάει από τον χρήστη τον αριθμό των ατόμων που αποβιβάστηκαν.

Γ. Να δημιουργεί πίνακα  $ΑΕ[19]$ , στον οποίο να καταχωρίζει τον αριθμό των επιβατών που βρίσκονται στο τρένο, μετά από κάθε αναχώρησή του.

Δ. Να βρίσκει και να εμφανίζει τον σταθμό από τον οποίο το τρένο αναχωρεί με τον μεγαλύτερο αριθμό επιβατών. (Να θεωρήσετε ότι από κάθε σταθμό το τρένο αναχωρεί με διαφορετικό αριθμό επιβατών). (Θέμα Πανελλαδικών)



## Λύση

Αλγόριθμος Θ\_3

Για  $i$  από 1 μέχρι 19

Εμφάνισε “Πόσα άτομα επιβιβάστηκαν στον “ $i$ ,” σταθμό;”

Διάβασε ΕΠΙΒ[ $i$ ]

Τέλος\_επανάληψης

ΑΠΟΒ[1]  $\leftarrow$  0

Για  $i$  από 2 μέχρι 19

Εμφάνισε “Πόσα άτομα αποβιβάστηκαν στον “ $i$ ,” σταθμό;”

Διάβασε ΑΠΟΒ[ $i$ ]

Τέλος\_επανάληψης

ΑΕ[1]  $\leftarrow$  ΕΠΙΒ[1]

Για  $i$  από 2 μέχρι 19

ΑΕ[ $i$ ]  $\leftarrow$  ΑΕ[ $i-1$ ] + ΕΠΙΒ[ $i$ ] - ΑΠΟΒ[ $i$ ]

Τέλος\_επανάληψης

max  $\leftarrow$  ΑΕ[1]

$\theta \leftarrow 1$

Για  $i$  από 2 μέχρι 19

    Αν ΑΕ[ $i$ ] > max τότε

        max  $\leftarrow$  ΑΕ[ $i$ ]

$\theta \leftarrow i$

    Τέλος\_Αν

Τέλος\_επανάληψης

Εμφάνισε  $\theta$

Τέλος Θ\_3

28. Για την ανάδειξη του επταμελούς (7) Διοικητικού Συμβουλίου ενός Πολιτιστικού Συλλόγου υπάρχουν 20 υποψήφιοι. Να γράψετε αλγόριθμο ο οποίος:

α. διαβάζει τα ονόματα των υποψηφίων και τα αποθηκεύει σε πίνακα.

β. διαβάζει για κάθε υποψήφιο τον αριθμό των ψήφων που έλαβε και τον αποθηκεύει σε πίνακα.

γ. εμφανίζει τα ονόματα των εκλεγέντων μελών του Διοικητικού Συμβουλίου κατά φθίνουσα σειρά ψήφων (να θεωρηθεί ότι δεν υπάρχουν περιπτώσεις ισοψηφίας).

δ. διαβάζει το όνομα ενός υποψηφίου και ελέγχει αν ο συγκεκριμένος εκλέγεται ή όχι, εμφανίζοντας κατάλληλο μήνυμα. (Θέμα Πανελλαδικών)

## Λύση

Αλγόριθμος Συμβούλιο

Για  $i$  από 1 μέχρι 20

    Διάβασε Ο[ $i$ ], Ψ[ $i$ ]

Τέλος\_επανάληψης

Για  $i$  από 2 μέχρι 20

    Για  $j$  από 20 μέχρι  $i$  με\_βήμα -1

        Αν Ψ[ $j-1$ ] < Ψ[ $j$ ] τότε

            Αντιμετάθεσε Ψ[ $j-1$ ], Ψ[ $j$ ]

            Αντιμετάθεσε Ο[ $j-1$ ], Ο[ $j$ ]

        Τέλος\_Αν

```

Τέλος_επανάληψης
Τέλος_επανάληψης
Για  $i$  από 1 μέχρι 7
  Εμφάνισε  $O[i]$ 
Τέλος_επανάληψης
Διάβασε όνομα
done  $\leftarrow$  ψευδής
pos  $\leftarrow$  0
 $i \leftarrow$  1
Όσο (done = ψευδής) και ( $i \leq 20$ ) επανάλαβε
  Αν  $O[i] = \text{όνομα}$  τότε
    done  $\leftarrow$  αληθής
    pos  $\leftarrow$   $i$ 
    Αλλιώς
       $i \leftarrow i + 1$ 
    Τέλος_Αν
Τέλος_επανάληψης
Αν done = αληθής τότε
  Αν pos  $\leq$  7 τότε
    Εμφάνισε "Εκλέγεται"
  Αλλιώς
    Εμφάνισε "Δεν εκλέγεται"
  Τέλος_Αν
Αλλιώς
  Εμφάνισε "Λάθος όνομα"
Τέλος_Αν
Τέλος Συμβούλιο

```

## 4.4 Θέματα Πανελληνίου Διαγωνισμού Πληροφορικής

1. Φιδάκι παιχνίδι: Μία παραλλαγή του παιχνιδιού "φιδάκι" είναι: Υπάρχει μια σειρά αριθμημένων τετραγώνων. Ένας παίχτης ξεκινά από το 1<sup>ο</sup> τετράγωνο και στόχος του για να κερδίσει είναι να φτάσει στο τελευταίο. Μέσα σε κάθε τετράγωνο είναι κρυμμένη η επόμενη κίνηση που επιτρέπεται να κάνει π.χ. να προχωρήσει εμπρός 3 τετράγωνα (+3) ή να πάει πίσω 2 τετράγωνα (-2). Να γραφεί αλγόριθμος όπου δίνονται:

α) Ένας αριθμός  $N$ ,  $1 < N < 1000$ , που δηλώνει το πλήθος των τετραγώνων στο φιδάκι.

β) Το περιεχόμενο κάθε θέσης που δείχνει την επιτρεπτή επόμενη κίνηση του παίχτη. Το περιεχόμενο δίνεται ως εξής: +α ή -α όπου α το πλήθος των βημάτων που θα κινηθεί ο παίκτης. Η τελευταία θέση έχει περιεχόμενο +0. Ζητείται να εμφανιστεί η σειρά όλων των θέσεων που ακολούθησε ένας παίχτης μέχρις ότου φθάσει στο τελευταίο τετράγωνο, περιλαμβανομένων της πρώτης και τελευταίας θέσης. Ο παίχτης πάντα φθάνει στην  $N$ -οστή τελευταία θέση. Μία παράσταση του προβλήματος είναι η ακόλουθη:

+3 +6 +7 +1 -3 +3 -2 -5 -2 +0

### Λύση

**Αλγόριθμος ΠΔΠ9**

**Αρχή\_επανάληψης**

Διάβασε  $N$

Μέχρις\_ότου ( $N > 1$  και  $N \leq 1000$ )

Για  $i$  από 1 μέχρι  $N - 1$

Διάβασε  $A[i]$

**Τέλος\_επανάληψης**

$A[N] \leftarrow 0$

$thesi \leftarrow 1$

$x \leftarrow A[1]$

**Αρχή\_επανάληψης**

**Εμφάνισε**  $thesi$

$thesi \leftarrow thesi + x$

$x \leftarrow A[thesi]$

**Μέχρις\_ότου**  $x = 0$

**Εμφάνισε**  $thesi$

**Τέλος ΠΔΠ9**

2. Δίνεται ένας ακέραιος τυχαίος αριθμός. Βρείτε το άθροισμα των παραγοντικών που προκύπτουν από καθένα από τα ψηφία του. Επαναλάβετε τη διαδικασία με τα ψηφία του αριθμού που προέκυψε σαν άθροισμα. Η επαναληπτική διαδικασία ολοκληρώνεται όταν ένας από τους αριθμούς που προκύπτουν από αυτή τη διαδικασία σαν άθροισμα έχει ήδη παραχθεί σε προηγούμενο βήμα. Για παράδειγμα δίνεται ο αριθμός :

25 2! + 5! = 2 + 120 = 122

122 1! + 2! + 2! = 5

$$5 \cdot 5! = 120$$

$$120 \cdot 1! + 2! + 0! = 4 \quad (0! = 1)$$

$$4 \cdot 4! = 24$$

$$24 \cdot 2! + 4! = 26$$

$$26 \cdot 2! + 6! = 722$$

$$722 \cdot 7! + 2! + 2! = 5044$$

$$5044 \cdot 5! + 0! + 4! + 4! = 169$$

$$169 \cdot 1! + 6! + 9! = 363601$$

$$363601 \cdot 3! + 6! + 3! + 6! + 0! + 1! = 1454$$

$$1454 \cdot 1! + 4! + 5! + 4! = 169$$

Εδώ σταματάει η επαναληπτική διαδικασία γιατί ο αριθμός 169 έχει παραχθεί ξανά στο 10ο βήμα. Γράψτε πρόγραμμα που να δέχεται έναν ακέραιο θετικό αριθμό μεταξύ 0 και 9999999, και να υπολογίζει τη λίστα των παραγοντικών, έως ότου παραχθεί εκ νέου ένας αριθμός που να υπάρχει ήδη στη λίστα.

### Λύση

**Αλγόριθμος** ΠΔΠ10|γυκείο !1998

**Αρχή\_επανάληψης**

Διάβασε  $x$

**Μέχρις\_ότου** ( $x \geq 0$  και  $x \leq 9999999$ )

**Εμφάνισε**  $x$

$table[1] \leftarrow x$

$k \leftarrow 1$

$vrethike \leftarrow$  Ψευδής

**Όσο**  $vrethike =$  Ψευδής **επανάλαβε**

$s \leftarrow 0$

**Όσο**  $x > 0$  **επανάλαβε**

$ps \leftarrow x \bmod 10$

$x \leftarrow x \operatorname{div} 10$

$par \leftarrow 1$

**Για**  $i$  **από** 1 **μέχρι**  $ps$

$par \leftarrow par * i$

**Τέλος\_επανάληψης**

$s \leftarrow s + par$

**Τέλος\_επανάληψης**

**Εμφάνισε**  $s$

$k \leftarrow k + 1$

$table[k] \leftarrow s$

$i \leftarrow 1$

**Όσο**  $vrethike =$  Ψευδής **και**  $i < k$  **επανάλαβε**

**Αν**  $s = table[i]$  **τότε**

$vrethike \leftarrow$  Αληθής

$thesi \leftarrow i$

**Εμφάνισε** "βρεθηκε στη θέση",  $thesi$

αλλιώς  
i ← i + 1  
Τέλος\_Αν  
Τέλος\_επανάληψης  
x ← s  
Τέλος\_επανάληψης  
Τέλος ΠΔΠ10

**3.** Είναι γεγονός ότι εάν τοποθετηθούν ώριμες ντομάτες ανάμεσα σε άγουρες ντομάτες, τότε οι άγουρες ντομάτες θα αρχίσουν να ωριμάζουν γρηγορότερα. Το πρόβλημα έχει ως εξής: υπάρχουν n ντομάτες τοποθετημένες σε μια γραμμή και είναι αριθμημένες από 1 έως n ( $2 < n < 70$ ). Μόνο μια ντομάτα από αυτές είναι κόκκινη, δηλαδή ώριμη. Ο αριθμός της θέσης της κόκκινης ντομάτας στη γραμμή είναι m, όπου  $1 < m < n$ . Αμφότερες οι γειτονικές ντομάτες της ώριμης ντομάτας θα ωριμάσουν, δηλαδή θα κοκκινίσουν, κατά τη διάρκεια της πρώτης μέρας. Κατά τη διάρκεια κάθε επόμενης μέρας θα ωριμάζουν οι γείτονες κάθε κόκκινης ντομάτας, εάν δεν είναι ήδη ώριμες. Προσοχή, κάθε ντομάτα στη γραμμή έχει δυο γείτονες εκτός της πρώτης και της τελευταίας που έχουν μόνο έναν. Γράψτε ένα πρόγραμμα που να υπολογίζει πόσες άγουρες ντομάτες θα παραμείνουν στη γραμμή μετά από d ( $1 < d < 30$ ) ημέρες.

#### **Λύση**

Αλγόριθμος ΠΔΠ15gymnasio  
Αρχή\_επανάληψης  
Διάβασε n  
Μέχρις\_ότου  $n \geq 2$  και  $n \leq 70$

Αρχή\_επανάληψης  
Διάβασε m  
Μέχρις\_ότου  $m \geq 1$  και  $m \leq n$

Αρχή\_επανάληψης  
Διάβασε d  
Μέχρις\_ότου  $d \geq 1$  και  $d \leq 30$

Για i από 1 μέχρι n  
NTOMATA[i] ← "α"  
Τέλος\_επανάληψης  
NTOMATA[m] ← "κ"

Για i από 1 μέχρι d  
left ← m - i  
right ← m + i  
Αν left > 0 τότε  
NTOMATA[left] ← "κ"  
Τέλος\_Αν  
Αν right ≤ n τότε  
NTOMATA[right] ← "κ"  
Τέλος\_Αν

## Τέλος\_επανάληψης

πλ ← 0

Για i από 1 μέχρι n

Αν ΝΤΟΜΑΤΑ[i] = "α" τότε

πλ ← πλ + 1

Τέλος\_Αν

Τέλος\_επανάληψης

Εμφάνισε πλ

Τέλος ΠΔΠ15

4. Οι ποδηλάτες έχουν αριθμούς από το 1 μέχρι το N και αγωνίζονται διαδοχικά από τον 1ο μέχρι τον N-οστό (από αυτόν που φέρει τον αριθμό 1 μέχρι αυτόν που φέρει τον αριθμό N). Μετά από την ολοκλήρωση της διαδρομής του κάθε ποδηλάτη, το ηλεκτρονικό σύστημα χρονομέτρησης, δίνει το χρόνο και ενημερώνει μια βάση δεδομένων με τη γενική σειρά κατάταξης. Το σύστημα τότε μας δίνει τη σειρά κατάταξης του ποδηλάτη σε σχέση με αυτούς που είχαν τερματίσει μέχρι εκείνη τη στιγμή. (Ο 1ος ποδηλάτης που θα αγωνιστεί θα έχει σειρά 1, ο 2ος 1 ή 2, ο 3ος 1 ή 2 ή 3, ... ο N-οστός 1 ή 2 ή 3 ή ... ή N). Να γραφεί πρόγραμμα το οποίο θα δέχεται τη σειρά κατάταξης του κάθε ποδηλάτη κατά τον τερματισμό του και θα τυπώνει αυτούς που κέρδισαν μετάλλιο. Θεωρούμε δεδομένο ότι όλοι οι ποδηλάτες τερμάτισαν και δεν υπάρχουν δύο ποδηλάτες με τον ίδιο χρόνο.

### Λύση

Αλγόριθμος ΠΔΠ16

Αρχή\_επανάληψης

Διάβασε N

Μέχρις\_ότου  $N \geq 3$  και  $N \leq 100$

Για i από 1 μέχρι N

Αρχή\_επανάληψης

Διάβασε ΘΕΣΗ[i]

Μέχρις\_ότου  $\Theta\text{ΕΣΗ}[i] \geq 1$  και  $\Theta\text{ΕΣΗ}[i] \leq N$

Τέλος\_επανάληψης

Για i από 1 μέχρι 3 με\_βήμα 1

ΣΕΙΡΑ[i] ← 0

Τέλος\_επανάληψης

Για i από 1 μέχρι N με\_βήμα 1

Αν  $\Theta\text{ΕΣΗ}[i] = 1$  τότε

ΣΕΙΡΑ[3] ← ΣΕΙΡΑ[2]

ΣΕΙΡΑ[2] ← ΣΕΙΡΑ[1]

ΣΕΙΡΑ[1] ← i

αλλιώς\_Αν  $\Theta\text{ΕΣΗ}[i] = 2$  τότε

ΣΕΙΡΑ[3] ← ΣΕΙΡΑ[2]

ΣΕΙΡΑ[2] ← i

αλλιώς\_Αν  $\Theta\text{ΕΣΗ}[i] = 3$  τότε

ΣΕΙΡΑ[3] ← i

Τέλος\_Αν  
Τέλος\_επανάληψης

Για i από 1 μέχρι 3  
Εμφάνισε ΣΕΙΡΑ[i]  
Τέλος\_επανάληψης  
Τέλος ΠΔΠ16

**5.** Ένα από τα πλέον όμορφα και ταυτόχρονα δυναμικά Ολυμπιακά Αγωνίσματα είναι το Δέκαθλο. Οι δεκαθλητές, δοκιμάζονται κυριολεκτικά σε δέκα αγωνίσματα στίβου. Στους Ολυμπιακούς Αγώνες της Αθήνας οι δεκαθλητές αγωνίστηκαν σε αυτά τα αγωνίσματα, στις 23 & 24 Αυγούστου 2004. Η τελική κατάταξη προέκυψε από το άθροισμα των βαθμών που συγκέντρωσαν οι αθλητές σε κάθε αγώνισμα. Για λόγους καθαρά στατιστικής, σε κάθε αγώνισμα βγήκε ένας «νικητής» αγωνίσματος. Στο άλμα επί κοντώ, ο νικητής προκύπτει από τους αθλητές που υπερπήδησαν το ίδιο ύψος με μικρότερο αριθμό συνολικών προσπαθειών. Αθλητές με ίδιο αριθμό προσπαθειών που υπερπήδησαν το ίδιο ύψος ανακηρύσσονται εξ ίσου νικητές. Να γραφεί πρόγραμμα το οποίο θα δέχεται για είσοδο τον αριθμό των αξιολογούμενων αθλητών, με βάση τη σειρά συμμετοχής στο άθλημα και για κάθε αθλητή, το τελικό ύψος υπερπήδησης και τον αριθμό των συνολικών προσπαθειών. Το πρόγραμμα θα επιστρέφει τον αριθμό των νικητών του αγωνίσματος και τη σειρά συμμετοχής, που αυτός/αυτοί είχε/είχαν.

**Λύση**

! Δέκαθλο 2004-05  
! Γνωστικοί στόχοι: Δομή επιλογής & επανάληψης, εμφύτευση, πίνακες, ταξινόμηση & αναζήτηση  
**Αλγόριθμος ΠΔΠ17**  
**Αρχή\_επανάληψης**  
  Διάβασε N  
  Μέχρις\_ότου N ≥ 1 και N ≤ 128

Για i από 1 μέχρι N  
  **Αρχή\_επανάληψης**  
    Διάβασε ΥΨΟΣ[i]  
    Μέχρις\_ότου ΥΨΟΣ[i] ≥ 400 και ΥΨΟΣ[i] ≤ 650  
    **Αρχή\_επανάληψης**  
      Διάβασε ΠΛ[i]  
      Μέχρις\_ότου ΠΛ[i] ≥ 3 και ΠΛ[i] ≤ 30  
      ΣΕΙΡΑ[i] ← i  
  **Τέλος\_επανάληψης**

Για i από 2 μέχρι N με\_βήμα 1  
  Για j από N μέχρι i με\_βήμα -1  
    Αν ΥΨΟΣ[j - 1] < ΥΨΟΣ[j] τότε  
      **Αντιμετάθεσε** ΥΨΟΣ[j - 1], ΥΨΟΣ[j]  
      **Αντιμετάθεσε** ΠΛ[j - 1], ΠΛ[j]  
      **Αντιμετάθεσε** ΣΕΙΡΑ[j - 1], ΣΕΙΡΑ[j]  
      **αλλιώς\_Αν** ΥΨΟΣ[j - 1] = ΥΨΟΣ[j] και ΠΛ[j - 1] > ΠΛ[j] τότε

```

    Αντιμετάθεσε ΠΛ[j - 1], ΠΛ[j]
    Αντιμετάθεσε ΣΕΙΡΑ[j - 1], ΣΕΙΡΑ[j]
    Τέλος_Αν
Τέλος_επανάληψης
Τέλος_επανάληψης

k ← 0
i ← 1
done ← Ψευδής
Όσο i ≤ N και done = Ψευδής επανάλαβε
    Αν ΥΨΟΣ[i] = ΥΨΟΣ[1] τότε
        Εμφάνισε ΣΕΙΡΑ[i]
        k ← k + 1
        done ← Αληθής
    αλλιώς
        i ← i + 1
    Τέλος_Αν
Τέλος_επανάληψης
Εμφάνισε k
Τέλος ΠΔΠ17

```

**6.** Η Ακρόπολη των Αθηνών αποτελεί, αν όχι το σημαντικότερο, ένα από τα σημαντικότερα δημιουργήματα της ανθρωπότητας. Η κατασκευή της Ακρόπολης από τη σύλληψη, τη σχεδίαση, τη μελέτη, το κτίσιμο και τη δημιουργία των αιώνιων γλυπτών της αποκαλύπτουν το μεγαλείο του Ελληνικού πολιτισμού. Ένα από τα πάρα πολλά προβλήματα που είχαν να αντιμετωπίσουν οι Αρχαίοι Έλληνες ήταν η ανύψωση των μαρμάρων στον ιερό βράχο. Για να το επιτύχουν χρησιμοποίησαν ένα σύστημα με τροχαλίες, έτσι ώστε όταν κατέρχονταν μια άδεια άμαξα, να χρησιμοποιείται σαν αντίβαρο για την ανερχόμενη. Για την καλύτερη επίτευξη του σκοπού αυτού, ήταν προτιμότερο οι ελαφρύτερες άμαξες να ανέλθουν πρώτες. Ωστόσο, κατά τη δημιουργία των γλυπτών της μετόπης του Παρθενώνα, μερικά ξεχωριστά κομμάτια πεντελικού μαρμάρου έπρεπε να μεταφερθούν άμεσα και σε συγκεκριμένη σειρά. Τα κομμάτια αυτά ήταν μικρά και πρακτικά μικρού βάρους (ενδεικτικό βάρος 1). Έργο σας είναι να κατασκευάσετε ένα πρόγραμμα που θα βοηθήσει τους προγόνους μας να προγραμματίσουν τη σειρά μεταφοράς των φορτίων. Τα βάρη με την ενδεικτική τιμή 1, αντιστοιχούν σε αυτά τα ξεχωριστά κομμάτια μαρμάρου, τα οποία πρέπει να μεταφερθούν με τη δεδομένη σειρά εμφάνισης. Αρχικά διαβάζουμε έναν ακέραιο αριθμό N,  $1 < N < 1000$  που εκφράζει τον αριθμό των φορτίων που πρέπει να μεταφερθούν. Έπειτα για κάθε φορτίο διαβάζουμε τα βάρη τους, που εκφράζονται με έναν ακέραιο αριθμό B,  $1 < B < 9000$ . Τα φορτία με  $B=1$  πρέπει να μεταφερθούν στη σειρά εμφάνισης.

### **Λύση**

! Γνωστικοί στόχοι: Δομή επιλογής & επανάληψης, εμφώλευση, πίνακες, &ταξινόμηση

**Αλγόριθμος ΠΔΠ18**

**Αρχή\_επανάληψης**

**Διάβασε** N

**Μέχρις\_ότου**  $N \geq 1$  και  $N \leq 1000$



Για  $i$  από 1 μέχρι  $N$   
Αρχή\_επανάληψης  
Διάβασε  $B[i]$   
Μέχρις\_ότου  $B[i] \geq 1$  και  $B[i] \leq 9000$   
Τέλος\_επανάληψης

$k \leftarrow 1$   
Για  $i$  από 1 μέχρι  $N$   
Αν  $B[i] > 1$  τότε  
     $MON[i] \leftarrow$  Ψευδής  
     $ΒΑΡΗ[k] \leftarrow B[i]$   
     $k \leftarrow k + 1$   
αλλιώς  
     $MON[i] \leftarrow$  Αληθής  
Τέλος\_Αν  
Τέλος\_επανάληψης

Για  $i$  από 2 μέχρι  $k - 1$  με\_βήμα 1  
Για  $j$  από  $k - 1$  μέχρι  $i$  με\_βήμα -1  
    Αν  $ΒΑΡΗ[j - 1] > ΒΑΡΗ[j]$  τότε  
        Αντιμετάθεσε  $ΒΑΡΗ[j - 1]$ ,  $ΒΑΡΗ[j]$   
    Τέλος\_Αν  
Τέλος\_επανάληψης  
Τέλος\_επανάληψης

$k \leftarrow 1$   
Για  $i$  από 1 μέχρι  $N$   
Αν  $MON[i] =$  Ψευδής τότε  
    Εμφάνισε  $ΒΑΡΗ[k]$   
     $k \leftarrow k + 1$   
αλλιώς  
    Εμφάνισε 1  
Τέλος\_Αν  
Τέλος\_επανάληψης  
Τέλος ΠΔΠ18

**7.** Ο αρχαίος Ελληνισμός από τη Μυκηναϊκή περίοδο, ανέπτυξε σημαντικές πόλεις γύρω από τον Αργοσαρωνικό κόλπο. Σε στρατηγικά σημεία του κόλπου κατασκευάστηκαν σημαντικά οχυρά. Στην ανατολική είσοδο του κόλπου, οι αρχαίοι Έλληνες κατασκεύασαν ένα απόρθητο ναυτικό οχυρό. Τα «Πατρόκλεια» τείχη προστάτευαν για αιώνες τον αντίστοιχο ναύσταθμο. Ο Πάτροκλος με τη βοήθεια Ατρείδων, χρησιμοποίησαν τεράστιους κυβόλιθους προκειμένου να ορθώσουν την οχύρωση. Έργο σας είναι να βοηθήσετε τους προγόνους σας να υπολογίσουν ποιους συγκεκριμένους κυβόλιθους θα πρέπει να χρησιμοποιήσουν ώστε να εξασφαλίσουν την Ελληνική κυριαρχία στο Αιγαίο για τα επόμενα 3500 χρόνια. Η τελική επιφάνεια των τειχών πρέπει να είναι όσο το δυνατόν κοντύτερα στην επιθυμητή. Αν αυτό μπορεί να γίνει με περισσότερους από έναν τρόπους, τότε πρέπει να γίνει με χρήση του μικρότερου δυνατού αριθμού κυβόλιθων. Ο αλγόριθμος διαβάζει τον αριθμό  $\Sigma$ ,  $100 < \Sigma < 10000$  που εκφράζει την επιφάνεια σε

τετραγωνικές μονάδες που πρέπει να έχουν τα τείχη και τον αριθμό  $N$ ,  $10 < N < 1000$  που εκφράζει τον αριθμό των κυβόλιθων που μπορούν να χρησιμοποιηθούν. Οι επόμενες  $N$  είσοδοι περιέχουν την επιφάνεια των κυβόλιθων, που εκφράζεται με έναν ακέραιο αριθμό  $E$ ,  $1 < E < 900$ . Υπολογίζονται-εμφανίζονται ο αριθμός των κυβόλιθων που χρησιμοποιήθηκαν και ο αύξοντας αριθμός του κυβόλιθου που χρησιμοποιήθηκε, με φθίνουσα σειρά μεγέθους.

### **Λύση**

**! Γνωστικοί στόχοι: Δομή επιλογής & επανάληψης, εμφώλευση, έλεγχος εγκυρότητας**

**Αλγόριθμος ΠΔΠ19**

**Αρχή\_επανάληψης**

**Διάβασε  $\Sigma$**

**Μέχρις\_ότου  $\Sigma \geq 100$  και  $\Sigma \leq 10000$**

**Αρχή\_επανάληψης**

**Διάβασε  $N$**

**Μέχρις\_ότου  $N \geq 10$  και  $N \leq 1000$**

**Για  $i$  από 1 μέχρι  $N$**

**Αρχή\_επανάληψης**

**Διάβασε  $Επ[i]$**

**Μέχρις\_ότου  $Επ[i] \geq 1$  και  $Επ[i] \leq 900$**

**$THESI[i] \leftarrow i$**

**Τέλος\_επανάληψης**

**Για  $i$  από 2 μέχρι  $N$  με\_βήμα 1**

**Για  $j$  από  $N$  μέχρι  $i$  με\_βήμα -1**

**Αν  $Επ[j - 1] < Επ[j]$  τότε**

**Αντιμετάθεσε  $Επ[j - 1]$ ,  $Επ[j]$**

**Αντιμετάθεσε  $THESI[j]$ ,  $THESI[j - 1]$**

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**$πλ \leftarrow 1$**

**$i \leftarrow 1$**

**Όσο  $\Sigma - Επ[i] > 0$  επανάλαβε**

**$πλ \leftarrow πλ + 1$**

**$\Sigma \leftarrow \Sigma - Επ[i]$**

**$i \leftarrow i + 1$**

**Τέλος\_επανάληψης**

**Εμφάνισε  $πλ$**

**Για  $i$  από 1 μέχρι  $πλ$**

**Εμφάνισε  $THESI[i]$**

**Τέλος\_επανάληψης**

**Τέλος ΠΔΠ19**

**8.** Οι αρχαίοι Έλληνες δεν άφησαν πίσω τους μόνο μια ασύλληπτη πνευματική κληρονομιά με τα θεωρητικά έργα τους: Αλλά, και με τα τεχνολογικά τους επιτεύγματα, παρέδωσαν στην ανθρωπότητα έναν τεχνολογικό πολιτισμό, που αν είχε αξιοποιηθεί, οι σημερινές μας δυνατότητες θα ήταν ασύγκριτα μεγαλύτερες. Τα κατασκευαστικά τους θαύματα, όπως ο χιλιομετρικής των Αθηνών, η ατμομηχανή του Ήρωνος, ο αστρολάβος των Αντικυθήρων, οι μηχανές του Αρχιμήδους κ.ά. αποτελούν μερικά από τα πολλά και πολύτιμα δημιουργήματά τους. Εξ' ίσου σημαντικά ήταν και τα επιτεύγματά τους στις επικοινωνίες. Χρησιμοποιώντας οπτικές ψηφιακές επικοινωνίες από το 12ο π.Χ. αιώνα μετέφεραν το μήνυμα της νίκης από την Τροία στις Μυκήνες μέσα σε λίγα 24ωρα. Από τα μέσα του 9ου π.Χ. χρησιμοποίησαν κωδικοποίηση του Ελληνικού αλφαβήτου (Καδμεία γραφή) για τη μετάδοση κειμένων με οπτική κωδικοποίηση σε Καρτεσιανές συντεταγμένες! Το εκπληκτικό είναι ότι κωδικοποίησαν το αλφάβητο με βάση την εντροπία του κάθε γράμματος. Η βασική αρχή αυτής της κωδικοποίησης είναι τα γράμματα να ταξινομούνται με βάση τη φθίνουσα σειρά εμφάνισής τους. Τα γράμματα με τη μεγαλύτερη **συχνότητα** εμφάνισης θα απαιτούν το άναμμα λιγότερων δαυλών και αντίστοιχα αυτά με τη μικρότερη συχνότητα εμφάνισης, περισσότερων. Να κωδικοποιήσετε το παραπάνω σενάριο.

### **Λύση**

! Συχνότητες λέξεων

! Γνωστικοί στόχοι: Δομή επιλογής & επανάληψης, εμφώλευση, πίνακες (παράλληλοι) συχνοτήτων, ταξινόμηση

**Αλγόριθμος ΠΔΠ20**

**Για i από 1 μέχρι 25 με\_βήμα 1**

**Διάβασε** ΓΡ[i]

  ΣΥΧΝ[i] ← 0

**Τέλος\_επανάληψης**

**Αρχή\_επανάληψης**

**Διάβασε** N

**Μέχρις\_ότου**  $N \geq 10$  και  $N \leq 100$

**Για i από 1 μέχρι N**

**Διάβασε** ΚΕΙΜ[i]

**Τέλος\_επανάληψης**

λέξεις ← 0

**Για i από 1 μέχρι N**

  key ← ΚΕΙΜ[i]

**Αν** key = ΓΡ[25] **τότε**

    λέξεις ← λέξεις + 1

**Τέλος\_Αν**

**Για j από 1 μέχρι 25**

**Αν** key = ΓΡ[j] **τότε**

    ΣΥΧΝ[j] ← ΣΥΧΝ[j] + 1

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

Για  $i$  από 2 μέχρι 24 με\_βήμα 1

Για  $j$  από 24 μέχρι  $i$  με\_βήμα -1

Αν  $ΣΥΧΝ[j - 1] < ΣΥΧΝ[j]$  τότε

Αντιμετάθεσε  $ΣΥΧΝ[j - 1]$ ,  $ΣΥΧΝ[j]$

Αντιμετάθεσε  $ΓΡ[j - 1]$ ,  $ΓΡ[j]$

αλλιώς\_Αν  $ΣΥΧΝ[j - 1] = ΣΥΧΝ[j]$  και  $ΓΡ[j - 1] > ΓΡ[j]$  τότε

Αντιμετάθεσε  $ΓΡ[j - 1]$ ,  $ΓΡ[j]$

Τέλος\_Αν

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Εμφάνισε λέξεις + 1

! το κείμενο τελειώνει σε γράμμα

Για  $i$  από 1 μέχρι 24

Εμφάνισε  $ΓΡ[i]$ ,  $ΣΥΧΝ[i]$

Τέλος\_επανάληψης

Τέλος ΠΔΠ20

**9.** Αποτελεί πλέον κοινή παραδοχή ότι το υφιστάμενο μοντέλο ανάπτυξης, πέραν όλων των άλλων στρεβλώσεων προκαλεί μια τρομακτική υποβάθμιση στο περιβάλλον του πλανήτη μας. Η μείωση της καύσης υδρογονανθράκων και ο περιορισμός της ποσότητας του εκπεμπόμενου διοξειδίου του άνθρακα ( $CO_2$ ), αποτελούν πρώτιστη προτεραιότητα για την ανθρωπότητα. Η χρήση υδρογόνου ( $H_2$ ), που μπορεί να παραχθεί φθηνά από το θαλασσινό νερό με ηλεκτρόλυση, τα ηλεκτρικά φορτία της οποίας μπορούν να μας τα παρέχουν φωτοβολταϊκά στοιχεία, είναι μια ελπιδοφόρα λύση. Η ένωση του  $H_2$  με το οξυγόνο ( $O_2$ ) γίνεται με ισχυρή εξώθερμη αντίδραση και μόνο κατάλοιπο το νερό! ( $2H_2 + O_2 \rightarrow 2H_2O$ ). Σύμπραξη Ελληνικών Πανεπιστημίων κατασκεύασαν μερικά δοκιμαστικά αυτοκίνητα Υδρογόνου με εξελιγμένα συστήματα μετατροπής ισχύος, μηχανολογικά αλλά και ηλεκτρονικά, για βέλτιστη συμπεριφορά. Προκειμένου να δοκιμαστεί η συμπεριφορά τους σε πραγματικές συνθήκες οδήγησης, με τη βοήθεια του δικτύου των Ελληνικών Πανεπιστημίων και ΤΕΙ, καταγράφεται σε κεντρικό πληροφοριακό σύστημα κάθε βλάβη που εντοπίζεται σε κάθε ένα από 10000 ξεχωριστά τμήματα του αυτοκινήτου. Οι βλάβες που δεν οφείλονται στον οδηγό, ελέγχονται και καταχωρίζονται ενώ οι άλλες (οφειλόμενες στον οδηγό) δεν καταχωρίζονται.

Να αναπτύξετε ένα πρόγραμμα το οποίο αφού διαβάσει τις αναφορές βλαβών για κάθε τμήμα, θα εμφανίζει τον αριθμό των τμημάτων που παρουσίασαν βλάβη (μη οφειλόμενη στον οδηγό) και τα αντίστοιχα τμήματα του αυτοκινήτου, ταξινομημένα κατά φθίνουσα σειρά αναφοράς βλαβών.

Διαβάζουμε τον αριθμό των τμημάτων για τα οποία υπήρξαν αναφορές βλάβης  $10 \leq C \leq 10000$ . Τα επόμενα  $C$  ζεύγη τιμών περιέχουν κάθε ένα δύο ακέραιους αριθμούς, τον αριθμό του τμήματος και τις συνολικές βλάβες που διαπιστώθηκαν.

Έξοδος: Τον αριθμό των τμημάτων τα οποία παρουσίασαν βλάβη ευθύνης του κατασκευαστή  $0 \leq L \leq 10000$ . Οι επόμενες  $L$  τιμές περιέχουν από έναν αριθμό. Τον αριθμό του τμήματος που παρουσίασε βλάβη με φθίνουσα όμως σειρά.

**Λύση**

! Αυτοκίνητα υδρογόνου

! Γνωστικοί στόχοι: Δομή επιλογής & επανάληψης, εμφώλευση

**Αλγόριθμος** πδπ22  
**Διάβασε** N  
**Για** i από 1 μέχρι N με\_βήμα 1  
  **Διάβασε** τμημα[i], βλαβες[i]  
**Τέλος\_επανάληψης**

πλ ← 0  
**Για** i από 1 μέχρι N  
  **Αν** βλαβες[i] > 0 τότε  
    πλ ← πλ + 1  
    τελ[πλ] ← τμημα[i]  
  **Τέλος\_Αν**  
**Τέλος\_επανάληψης**

**Εμφάνισε** πλ  
**Για** i από 1 μέχρι πλ  
  **Εμφάνισε** τελ[i]  
**Τέλος\_επανάληψης**  
**Τέλος** πδπ22

**10.** Οι τιμές κάποιων αγαθών ή τίτλων (π.χ. πετρελαίου, χρυσού, μετοχών αλλά και βασικών τροφίμων όπως των αλεύρων, της ζάχαρης κ.λπ.), διαμορφώνονται καθημερινά βάσει της προσφοράς και της ζήτησης, αλλά και με βάση την εκτίμηση για τη μελλοντική τους πορεία. Αποτέλεσμα αυτών των συναλλαγών είναι οι τιμές αυτές να αλλάζουν από μέρα σε μέρα. Κάποιοι εκμεταλλεύονται αυτήν την αυξομείωση των τιμών, αγοράζοντας μία ποσότητα (ή δικαίωμα σε ποσότητα) φθηνά, και έπειτα πουλούν την ίδια ποσότητα ή δικαίωμα ακριβότερα. Το κέρδος εκφράζεται από το λόγο της τιμής πώλησης προς την τιμή αγοράς. Έστω ότι γνωρίζουμε την τιμή που έχει κάποιο αγαθό κάθε μέρα για ένα μεγάλο χρονικό διάστημα. Θέλουμε να υπολογίσουμε το μέγιστο κέρδος που θα μπορούσε κάποιος να αποκομίσει με μία αγορά και στη συνέχεια μία πώληση.

Να αναπτύξετε ένα πρόγραμμα το οποίο να διαβάζει το πλήθος των ημερών για τις οποίες είναι γνωστή η τιμή του αγαθού, την τιμή του αγαθού για κάθε μία από αυτές τις ημέρες, και να υπολογίζει το μέγιστο δυνατό κέρδος από μία αγορά και στη συνέχεια μία πώληση.

### **Λύση**

! 23ος ΠΔΠ-Μεγιστοποίηση κέρδους

! Γνωστικοί στόχοι: Δομή επιλογής & επανάληψης, εμφώλευση - Μεγιστοποίηση κέρδους  
- Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του ΙΟΙ το οποίο να διαβάζει το πλήθος των ημερών για τις οποίες είναι γνωστή η τιμή του αγαθού, την τιμή του αγαθού για κάθε μία από αυτές τις ημέρες, και να υπολογίζει το μέγιστο δυνατό κέρδος από μία αγορά και στη συνέχεια μία πώληση.

**Αλγόριθμος** πδπ23  
**Διάβασε** N

**Για** i από 1 μέχρι N  
  **Διάβασε** A[i]  
**Τέλος\_επανάληψης**

P ← 1

Για i από 1 μέχρι N - 1

  Για j από i + 1 μέχρι N

    Αν  $A[j]/A[i] > P$  τότε

      P ←  $A[j]/A[i]$

  Τέλος\_Αν

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Εμφάνισε P

Τέλος\_πδπ23

**11.** Να γραφεί αλγόριθμος ο οποίος διαβάζει πίνακα μεγέθους N και εμφανίζει το πλήθος των χαρακτηριστικών συχνοτήτων και ποιες είναι αυτές σε αύξουσα σειρά. Η αναπαράσταση ενός σήματος στο πεδίο των συχνοτήτων αποτελεί το φάσμα του. Εξαιρετικά σημαντική είναι η παρατήρηση ότι τα τεχνητά σήματα (τα σήματα δηλαδή που παράγονται από τεχνητά κατασκευασμένα συστήματα), έχουν μια μοναδική χαρακτηριστική κατανομή φάσματος. Η κατανομή αυτή ονομάζεται και χαρακτηριστική τριπλέτα, επειδή συμμετρικά και εκατέρωθεν μιας δεδομένης συχνότητας, (χαρακτηριστική συχνότητα), εμφανίζονται δύο σήματα ίδιας ισχύος, και μάλιστα μικρότερης από το 50% της ισχύος του σήματος που αντιστοιχεί στη χαρακτηριστική συχνότητα. Το Πανεπιστήμιο του Berkeley έχει αναπτύξει ένα διεθνές πρόγραμμα επεξεργασίας σημάτων από εθελοντές χρήστες του Διαδικτύου, που αποσκοπεί στην αναζήτηση εξωγήινης νοημοσύνης και βασίζεται στην ανάλυση των σημάτων που συλλέγονται από ραδιοτηλεσκόπια. Το Berkeley συλλέγει τα σήματα από τα ραδιοτηλεσκόπια και, αφού κάνει μια αρχική επεξεργασία, τα διανέμει στους συμμετέχοντες στο πρόγραμμα για να τα επεξεργαστούν. Να αναπτύξετε αλγόριθμο ο οποίος θα διαβάζει τις N τιμές που αντιστοιχούν σε ένα σήμα και θα αναγνωρίζει τις χαρακτηριστικές συχνότητες, εμφανίζοντας το πλήθος τους και τις θέσεις που εντοπίστηκαν. Σημείωση: Μία τριπλέτα σε ένα σήμα είναι μία τριάδα τιμών του σήματος με τις εξής ιδιότητες: (α) οι δύο ακραίες τιμές ισαπέχουν από τη μεσαία τιμή, και (β) οι δύο ακραίες τιμές είναι ίσες μεταξύ τους και μικρότερες του μισού της μεσαίας τιμής. Η μεσαία τιμή μίας τριπλέτας μας δίνει τη χαρακτηριστική συχνότητα, η οποία ισούται με τη θέση της μεσαίας τιμής στο σήμα.

**Λύση**

! Γνωστικοί στόχοι: Δομή επιλογής & επανάληψης, εμφώλευση, flags & αλγοριθμική σκέψη

Αλγόριθμος πδπ25

Διάβασε N

Για i από 1 μέχρι N με\_βήμα 1

  Διάβασε συχ[i]

Τέλος\_επανάληψης

πλ ← 0

Για  $i$  από 2 μέχρι  $N - 1$

$L \leftarrow i - 1$

$R \leftarrow i + 1$

flag  $\leftarrow$  Ψευδής

Όσο ( $L \geq 1$  και  $R \leq N$ ) και flag = Ψευδής επανάλαβε

Αν  $\text{συχ}[L] = \text{συχ}[R]$  και  $\text{συχ}[L] < 0.5 * \text{συχ}[i]$  τότε

$\text{πλ} \leftarrow \text{πλ} + 1$

$\text{ιδ}[\text{πλ}] \leftarrow i$

flag  $\leftarrow$  Αληθής

Τέλος\_Αν

$L \leftarrow L - 1$

$R \leftarrow R + 1$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Εμφάνισε πλ

Για  $i$  από 1 μέχρι πλ

Εμφάνισε  $\text{ιδ}[i]$

Τέλος\_επανάληψης

Τέλος\_πδπ25

## 4.5 Ασκήσεις προς επίλυση

1. Να γραφεί αλγόριθμος σε "ΓΛΩΣΣΑ" ο οποίος:

A. Να διαβάξει το πλήθος των στοιχείων ενός μονοδιάστατου πίνακα με όνομα A. Το πλήθος δεν θα πρέπει να υπολείπεται του 2 αλλά ούτε και να υπερβαίνει το 100.

B. Να δέχεται από το πληκτρολόγιο όλα τα στοιχεία του πίνακα καθένα από τα οποία πρέπει να είναι θετικοί ακέραιοι αριθμοί.

Γ. Να δημιουργεί ένα μονοδιάστατο πίνακα B κάθε στοιχείο του οποίου πρέπει να είναι το γινόμενο των γειτονικών στοιχείων του A. Για παράδειγμα, το στοιχείο B[1] πρέπει να είναι το A[1] \* A[2], το B[2] είναι το A[2] \* A[3], κ.ο.κ. Για τον υπολογισμό του γινομένου να εφαρμοστεί η μέθοδος του πολλαπλασιασμού αλά ρωσικά.

2. Δίνεται ο μονοδιάστατος πίνακας Π με 8 στοιχεία.

6 8 5 20 14 7 12 10

A. Πώς συμβολίζεται το δεύτερο στοιχείο του πίνακα;

B. Ποιο είναι το περιεχόμενο του στοιχείου Π[5];

Γ. Σε ποια θέση του πίνακα βρίσκεται η τιμή 20;

Δ. Τι τύπου στοιχεία περιέχει ο πίνακας Π;

E. Ποιο είναι το περιεχόμενο του στοιχείου Π[Π[6]];

ΣΤ. Ποιο είναι το περιεχόμενο του στοιχείου Π[Π[4] mod 2 + Π[3]];

3. Να αναπτυχθεί πρόγραμμα που θα διαβάξει τους βαθμούς 120 μαθητών της Γ' τάξης στο μάθημα της Πληροφορικής και θα εκτυπώνει το βαθμό που εμφανίστηκε τις περισσότερες φορές.

4. Δίνεται ένας πίνακας γεμάτος με 50 άρτιους και 50 περιττούς ακέραιους αριθμούς ανακατεμένους μεταξύ τους. Να γίνει πρόγραμμα που να διαχωρίζει τον πίνακα σε δυο νέους πίνακες, ένα για τους άρτιους και ένα για τους περιττούς αριθμούς.

5. Να κατασκευαστεί ο εξής πίνακας με κομψό τρόπο:  $X = [1 \ 1 \ 1 \ 1 \ 2 \ 4 \ 6 \ 8 \ 0 \ 0]$ .

6. Να γραφεί πρόγραμμα που να συγχωνεύει τους πίνακες ΘΕΡΜ[28] και ΘΕΡΜ2[32] σε έναν πίνακα.

7. Να αναπτυχθεί αλγόριθμος που θα δημιουργεί μονοδιάστατο πίνακα τοποθετώντας την τιμή -1 στις περιττές θέσεις και 1 στις άρτιες.

8. Δίνεται η παρακάτω ακολουθία αριθμών: 25, 8, 12, 14, 71, 41, 1. Τοποθετούμε τους αριθμούς σε στοίβα και σε ουρά.

1. Ποια λειτουργία θα χρησιμοποιηθεί για την τοποθέτηση των αριθμών στη στοίβα και ποια για την τοποθέτησή τους στην ουρά;

2. Να σχεδιάσετε τις δύο δομές (στοίβα και ουρά) μετά την τοποθέτηση των αριθμών.

3. Ποια λειτουργία θα χρησιμοποιηθεί για την έξοδο αριθμών από τη στοίβα και ποια για την έξοδό τους από την ουρά;

4. Πόσες φορές θα πρέπει να γίνει η παραπάνω λειτουργία στη στοίβα και πόσες στην ουρά για να εξέλθει ο αριθμός 71; (Θέμα Πανελλαδικών)



**9.** Έστω πρόβλημα που αναφέρει: «...Να κατασκευάσετε αλγόριθμο που θα ζητάει τις ηλικίες 100 ανθρώπων και να εμφανίζει το μέσο όρο ηλικίας τους...». Για κάθε μία πρόταση να γράψετε αν θεωρείτε ότι είναι σωστή ή λανθασμένη αντίστοιχα.

- α. Πρέπει να χρησιμοποιηθεί πίνακας.
- β. Είναι δυνατόν να χρησιμοποιηθεί πίνακας.
- γ. Είναι δυνατόν να χρησιμοποιηθεί η εντολή Όσο.
- δ. Είναι δυνατόν να χρησιμοποιηθεί η εντολή Για.
- ε. Η εντολή Για είναι η καταλληλότερη.

**10.** Μια υπηρεσία για στατιστικούς λόγους θα καταμετρήσει τον πληθυσμό 150 μικρών χωριών. Έτσι σε έναν μονοδιάστατο πίνακα καταχωρήθηκαν τα ονόματα των χωριών και σε ένα δεύτερο πίνακα ο αντίστοιχος αριθμός των κατοίκων τους.

Να αναπτύξετε αλγόριθμο ο οποίος:

- (1) Διαβάζει τα ονόματα των χωριών και τα αποθηκεύει σε μονοδιάστατο πίνακα.
- (2) Διαβάζει τον αντίστοιχο αριθμό των κατοίκων των χωριών και τα αποθηκεύει σε μονοδιάστατο πίνακα.
- (3) Εμφανίζει το όνομα του μοναδικού ακατοίκητου χωριού.
- (4) Υπολογίζει και εμφανίζει το πλήθος και τα ονόματα των χωριών με τους περισσότερους κατοίκους.
- (5) Εμφανίζει ταξινομημένα κατά αύξουσα αλφαβητική σειρά τα ονόματα και τους κατοίκους των χωριών, των οποίων το αρχικό γράμμα είναι από "Ε" μέχρι και "Λ" και που δεν ξεπερνούν τους 100 κατοίκους.

**11.** Γράψτε αλγόριθμο που, με δεδομένο τον πραγματικό πίνακα  $A[50]$ , αντιγράφει τα στοιχεία των άρτιων θέσεων του πίνακα αυτού στον πραγματικό πίνακα  $B[25]$  και τα στοιχεία των περιττών θέσεων του  $A$  στον πραγματικό πίνακα  $\Gamma[25]$ .

**12.** Γράψτε αλγόριθμο που ελέγχει αν όλα τα στοιχεία του ακεραίου πίνακα  $X[60]$  είναι θετικά και εμφανίζει κατάλληλο ενημερωτικό μήνυμα.

**13.** Ας θεωρήσουμε τους ακεραίους πίνακες  $A[P,\Lambda]$  και  $B[P,M]$ . Γράψτε αλγόριθμο που δοθέντων των παραπάνω πινάκων, επιτυγχάνει τη συγχώνευσή τους στον ακεραίο πίνακα  $\Gamma[P,\Lambda+M]$

**14.** Μονοδιάστατος πίνακας 50 θέσεων περιέχει τις ποσότητες παραγωγής ενός προϊόντος. Να γραφεί πρόγραμμα που θα υλοποιεί τα παρακάτω:

- 1. Εισαγωγή των στοιχείων του πίνακα (περιοχή τιμών 0-100).
- 2. Εμφάνιση των στοιχείων του πίνακα.
- 3. Εύρεση της μέσης τιμής των στοιχείων του πίνακα.
- 4. Δημιουργία νέου πίνακα που θα περιέχει όλες τις ποσότητες των οποίων η διαφορά από τη μέση τιμή των στοιχείων του αρχικού πίνακα είναι μεγαλύτερη κατά 20% της μέσης τιμής.
- 5. Εμφάνιση των στοιχείων του νέου πίνακα καθώς και του πλήθους των στοιχείων του.

**15.** Να γραφεί πρόγραμμα που θα εναλλάσσει τα περιεχόμενα ενός μονοδιάστατου πίνακα ακεραίων  $N$  θέσεων ως εξής:

- Το 1ο στοιχείο με το  $N$ -οστό στοιχείο
- Το 2ο στοιχείο με το  $N-1$  στοιχείο κ.λπ.

Το πρόγραμμα θα εμφανίζει τον πίνακα πριν και μετά την εναλλαγή των στοιχείων.

**16.** Να γραφεί πρόγραμμα που θα διαχωρίζει ένα δεδομένο πίνακα ακεραίων τιμών  $N$  θέσεων σε δύο νέους πίνακες που θα περιέχουν αντίστοιχα τις άρτιες και τις περιττές τιμές του αρχικού πίνακα. Το πρόγραμμα θα εμφανίζει τον αρχικό πίνακα, τους δύο νέους πίνακες καθώς και το πλήθος των στοιχείων τους.

**17.** Να γραφεί πρόγραμμα που θα απομακρύνει από ένα μονοδιάστατο αριθμητικό πίνακα ακεραίων  $N$  θέσεων όλες τις πολλαπλές ίδιες τιμές δημιουργώντας ένα νέο πίνακα όπου κάθε αριθμός θα εμφανίζεται μόνο μία φορά.

**18.** Δίνεται μονοδιάστατος μη ταξινομημένος πίνακας  $T$  με  $N$  διαφορετικά στοιχεία. Να γράψετε τον αλγόριθμο σειριακής αναζήτησης της τιμής μιας μεταβλητής  $key$  στον πίνακα  $T$ .

**19.** Δίνεται μονοδιάστατος πίνακας  $\Pi$ ,  $N$  στοιχείων, που είναι ακέραιοι αριθμοί. Να αναπτύξετε αλγόριθμο, ο οποίος να ταξινομεί με τη μέθοδο της φυσαλίδας τα στοιχεία του πίνακα  $\Pi$ .

**20.** Να γραφεί πρόγραμμα που θα εισάγει σε μονοδιάστατο πίνακα 300 θέσεων τις ποσότητες παραγωγής ενός προϊόντος για ένα έτος. Κατά σειρά σε κάθε 25 θέσεις αντιστοιχεί ένας μήνας του έτους, δηλ. οι θέσεις 1-25 αντιστοιχούν στον Ιανουάριο, οι θέσεις 26-50 στον Φεβρουάριο κ.λπ. Να γίνει επίσης εισαγωγή της επιθυμητής μηνιαίας ποσότητας παραγωγής για κάθε έναν από τους 12 μήνες. Το πρόγραμμα θα βρίσκει και θα εμφανίζει: σε ποια ημέρα κάθε μήνα η συνολική μηνιαία ποσότητα παραγωγής υπερβαίνει την επιθυμητή και ποια είναι η μηνιαία πλεονάζουσα παραγωγή.

**21.** Να γραφεί αλγόριθμος για έναν πίνακα 10 θέσεων στον οποίο θα καταχωρούνται αριθμητικές τιμές και θα γίνεται έλεγχος για τη μοναδικότητα της κάθε τιμής. Θα πρέπει κάθε τιμή του πίνακα να ελέγχεται, Αν είναι μοναδική σε σχέση με όλες τις προηγούμενες να επιτρέπεται η καταχώρηση της στην αντίστοιχη θέση του πίνακα.

**22.** Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα πραγματικών αριθμών  $N$  στοιχείων ( $N = \text{γνωστό}$ ), έναν πραγματικό αριθμό  $X$  και θα επιστρέφει τη θέση εκείνου του στοιχείου του πίνακα στο οποίο το άθροισμα των στοιχείων του πίνακα, ξεκινώντας από το 1ο στοιχείο, γίνεται μεγαλύτερο του  $X$ .

**23.** Μετά τη μηχανοργάνωση στο Ίδρυμα Κοινωνικών Ασφαλίσεων, κάθε έτος θα εκδίδονται και θα αποστέλλονται στους ασφαλισμένους ετικέτες ασφαλιστικής ικανότητας, που θα πρέπει να επικολλούνται στα Ατομικά Βιβλιάρια Υγείας των ασφαλισμένων, ώστε να έχουν τη δυνατότητα ιατροφαρμακευτικής περίθαλψης. Δικαίωμα περίθαλψης έχουν όσοι ασφαλισμένοι είχαν το προηγούμενο έτος συνολικά πάνω από 50 ένσημα. Να γράψετε πρόγραμμα στη γλώσσα προγραμματισμού «ΓΛΩΣΣΑ» με το οποίο:

α) Θα εισάγονται σε έναν πίνακα Στοιχεία το όνομα, το επώνυμο και ο αριθμός μητρώου ασφαλισμένου για 100 ασφαλισμένους.

β) Θα εισάγονται σε ένα δεύτερο πίνακα Ένσημα, τα ένσημα που έχει κάθε ασφαλισμένος σε κάθε μήνα του προηγούμενου έτους.

γ) Θα υπολογίζει το σύνολο των ενσήμων του κάθε ασφαλισμένου για το έτος που

πέρασε.

δ) Θα τυπώνει ετικέτες με τα στοιχεία των ασφαλισμένων (όνομα, επώνυμο και αριθμό μητρώου ασφαλισμένου) και το έτος ισχύος για όσους ασφαλισμένους έχουν δικαίωμα περιθαλψής για το έτος 2003.

ε) Θα εμφανίζει το πλήθος των ετικετών που πρέπει να σταλούν και το κόστος για το ίδρυμα, αν για κάθε ετικέτα δαπανώνται 0,9 ευρώ για τη δημιουργία και αποστολή τους.

**24.** Να γραφεί αλγόριθμος με χρήση πινάκων, ο οποίος να διαβάζει-δέχεται ως είσοδο τις πωλήσεις και τα ονόματα πενήντα εταιρειών. Έπειτα να υπολογίζει και εμφανίζει:

α) το όνομα της εταιρείας με τις περισσότερες πωλήσεις.

β) το μέσο όρο των πωλήσεων όλων των εταιρειών.

γ) το πλήθος των εταιρειών που εμφάνισαν πωλήσεις πάνω από τα 3/5 του μέσου όρου.

δ) το πλήθος των εταιρειών με πωλήσεις από 13000 μέχρι και 25000 ευρώ.

**25.** Σ' ένα διαγωνισμό συμμετέχουν 5000 διαγωνιζόμενοι και εξετάζονται σε δύο μαθήματα.

Να γράψετε αλγόριθμο που:

1. να διαβάζει και να καταχωρίζει σε κατάλληλους πίνακες για κάθε διαγωνιζόμενο τον αριθμό μητρώου, το ονοματεπώνυμο και τους βαθμούς που πήρε στα δύο μαθήματα.

Οι αριθμοί μητρώου θεωρούνται μοναδικοί. Η βαθμολογική κλίμακα είναι από 0 έως και 100.

2. να εμφανίζει κατάσταση επιτυχόντων με την εξής μορφή:

Αριθ. Μητρώου Ονοματεπώνυμο Μέσος Όρος

Επιτυχών θεωρείται ότι είναι αυτός που έχει μέσο όρο βαθμολογίας μεγαλύτερο ή ίσο του 60.

3. να διαβάζει έναν αριθμό μητρώου και

α. σε περίπτωση που ο αριθμός μητρώου έχει καταχωρηθεί στον πίνακα, να εμφανίζεται ο αριθμός μητρώου, το ονοματεπώνυμο, ο μέσος όρος βαθμολογίας και η ένδειξη «ΕΠΙΤΥΧΩΝ» ή «ΑΠΟΤΥΧΩΝ», ανάλογα με το μέσο όρο.

β. σε περίπτωση που ο αριθμός μητρώου δεν είναι καταχωρισμένος στον πίνακα, να εμφανίζεται το μήνυμα «Ο αριθμός μητρώου δεν αντιστοιχεί σε διαγωνιζόμενο».

**26.** Να αναπτύξετε έναν αλγόριθμο, ώστε:

α) να διαβάζει το πλήθος των ασθενών ενός νοσοκομείου, το οποίο δεν μπορεί να δεχτεί περισσότερους από 500 ασθενείς,

β) για κάθε ασθενή να διαβάζει τις ημέρες νοσηλείας του, τον κωδικό του ασφαλιστικού του ταμείου και τη θέση νοσηλείας. Να ελέγχει την ορθότητα εισαγωγής των δεδομένων σύμφωνα με τα παρακάτω:

- οι ημέρες νοσηλείας είναι ακέραιος αριθμός μεγαλύτερος ή ίσος του 1,
- τα ασφαλιστικά ταμεία είναι 10 με κωδικούς από 1 μέχρι και 10,
- οι θέσεις νοσηλείας είναι Α ή Β ή Γ,

γ) να υπολογίζει και να εμφανίζει το μέσο όρο ημερών νοσηλείας των ασθενών στο νοσοκομείο,

δ) να υπολογίζει και να εμφανίζει για κάθε ασθενή το κόστος παραμονής που πρέπει να καταβάλει στο νοσοκομείο το ασφαλιστικό του ταμείο, σύμφωνα με τις ημέρες και τη θέση νοσηλείας. Το κόστος παραμονής στο νοσοκομείο ανά ημέρα και θέση νοσηλείας για κάθε ασθενή φαίνεται στον ακόλουθο πίνακα:

Θέση Νοσηλείας	Κόστος παραμονής ανά ημέρα νοσηλείας ανά ασθενή
----------------	---

A	125 €
B	90 €
Γ	60 €

ε) Να υπολογίζει και να εμφανίζει με τη χρήση πίνακα το συνολικό κόστος που θα καταβάλει το κάθε ασφαλιστικό ταμείο στο νοσοκομείο,  
στ) να υπολογίζει και να εμφανίζει το συνολικό ποσό που οφείλουν όλα τα ασφαλιστικά ταμεία στο νοσοκομείο. (Θέμα Πανελλαδικών)

**27.** Σε κάποια χώρα της Ευρωπαϊκής Ένωσης διεξάγονται εκλογές για την ανάδειξη των μελών του Ευρωπαϊκού Κοινοβουλίου. Θεωρήστε ότι μετέχουν 15 συνδυασμοί κομμάτων, οι οποίοι θα μοιραστούν 24 έδρες σύμφωνα με το ποσοστό των έγκυρων ψηφοδελτίων που έλαβαν. Κόμματα που δεν συγκεντρώνουν ποσοστό έγκυρων ψηφοδελτίων τουλάχιστον ίσο με το 3% του συνόλου των έγκυρων ψηφοδελτίων δεν δικαιούνται έδρα. Για κάθε κόμμα, εκτός του πρώτου κόμματος, ο αριθμός των εδρών που θα λάβει υπολογίζεται ως εξής: Το ποσοστό των έγκυρων ψηφοδελτίων πολλαπλασιάζεται επί 24 και στη συνέχεια το γινόμενο διαιρείται με το άθροισμα των ποσοστών όλων των κομμάτων που δικαιούνται έδρα. Το ακέραιο μέρος του αριθμού που προκύπτει είναι ο αριθμός των εδρών που θα λάβει το κόμμα. Το πρώτο κόμμα λαμβάνει τις υπόλοιπες έδρες. Να γράψετε αλγόριθμο ο οποίος:

- να διαβάζει και να αποθηκεύει σε μονοδιάστατους πίνακες τα ονόματα των κομμάτων και τα αντίστοιχα ποσοστά των έγκυρων ψηφοδελτίων τους.
- να εκτυπώνει τα ονόματα και το αντίστοιχο ποσοστό έγκυρων ψηφοδελτίων των κομμάτων που δεν έλαβαν έδρα.
- να εκτυπώνει το όνομα του κόμματος με το μεγαλύτερο ποσοστό έγκυρων ψηφοδελτίων.
- να υπολογίζει και να εκτυπώνει το άθροισμα των ποσοστών όλων των κομμάτων που δικαιούνται έδρα.
- να εκτυπώνει τα ονόματα των κομμάτων που έλαβαν έδρα και τον αντίστοιχο αριθμό των εδρών τους. (Θέμα Πανελλαδικών)

**28.** Να συμπληρώσετε το παρακάτω τμήμα αλγορίθμου, το οποίο αντιμεταθέτει τις τιμές των κελιών του πίνακα A, έτσι ώστε η τελική διάταξη των αριθμών να είναι από 1 μέχρι 10.

Για i από ... μέχρι ...

Αντιμετάθεσε A[...], A[...]

Τέλος\_επανάληψης

**29.** Σε έναν αγώνα δισκοβολίας συμμετέχουν 20 αθλητές. Κάθε αθλητής έκανε μόνο μία έγκυρη ρίψη που καταχωρείται ως επίδοση του αθλητή και εκφράζεται σε μέτρα. Να αναπτύξετε αλγόριθμο ο οποίος:

- να διαβάζει για κάθε αθλητή το όνομα και την επίδοσή του,
- να ταξινομεί τους αθλητές ως προς την επίδοσή τους,
- να εμφανίζει τα ονόματα και τις επιδόσεις των τριών πρώτων αθλητών, αρχίζοντας από εκείνον με την καλύτερη επίδοση,
- να εμφανίζει τα ονόματα και τις επιδόσεις των πέντε τελευταίων αθλητών, αρχίζοντας από εκείνον με την καλύτερη επίδοση.

**30.** Για την εύρεση πόρων προκειμένου οι μαθητές της Δ' τάξης Εσπερινού Λυκείου να συμμετάσχουν σε εκδρομή οργανώνεται λαχειοφόρος αγορά. Οι μαθητές του Λυκείου διαθέτουν λαχνούς στα σχολεία της περιοχής τους. Διακόσιοι μαθητές από δεκαπέντε διαφορετικά σχολεία αγόρασαν ο καθένας από έναν μόνο λαχνό. Μετά από κλήρωση ένας μαθητής κερδίζει τον πρώτο λαχνό.

Να γίνει τμήμα αλγορίθμου που:

α) για κάθε μαθητή που αγόρασε λαχνό να εισάγει σε μονοδιάστατο πίνακα A 200 θέσεων, το επώνυμό του και στην αντίστοιχη θέση μονοδιάστατου πίνακα B 200 θέσεων, το όνομα του σχολείου του,

β) να εισάγει σε μονοδιάστατο πίνακα Σ 15 θέσεων, τα ονόματα όλων των σχολείων της περιοχής και στις αντίστοιχες θέσεις μονοδιάστατου πίνακα Μ 15 θέσεων, τις ηλεκτρονικές διευθύνσεις των σχολείων,

γ) να διαβάζει το επώνυμο του μαθητή, που κέρδισε τον πρώτο λαχνό,

δ) χρησιμοποιώντας τον αλγόριθμο της σειριακής αναζήτησης να προσδιορίζει τη θέση του επωνύμου του τυχερού μαθητή στον πίνακα Α. Στη συνέχεια στον πίνακα Β να βρίσκει το όνομα του σχολείου που φοιτά,

ε) λαμβάνοντας υπόψη το όνομα του σχολείου που φοιτά ο τυχερός μαθητής και χρησιμοποιώντας τον αλγόριθμο της σειριακής αναζήτησης, να προσδιορίζει τη θέση του σχολείου στον πίνακα Σ. Στη συνέχεια στον πίνακα Μ να βρίσκει τη διεύθυνση του ηλεκτρονικού ταχυδρομείου του σχολείου αυτού,

στ) να εμφανίζει το επώνυμο του τυχερού μαθητή, το όνομα του σχολείου του και τη διεύθυνση του ηλεκτρονικού ταχυδρομείου του σχολείου του.

Να θεωρήσετε ότι δεν υπάρχουν μαθητές με το ίδιο επώνυμο και ότι κάθε μαθητής αγόρασε ένα μόνο λαχνό.

**31.** Να γίνουν 2 ξεχωριστά προγράμματα τα οποία θα διαβάζουν 2 αριθμούς Μ και Ν και καθένα από τα οποία θα επιτελεί τις παρακάτω λειτουργίες:

α) Θα διαβάζει τα στοιχεία ενός διδιάστατου πίνακα Α, με διαστάσεις  $M \times N$  και θα τα μεταφέρει γραμμή προς γραμμή σε ένα μονοδιάστατο πίνακα Β με  $M * N$  στοιχεία. Στη συνέχεια θα εμφανίζει τα στοιχεία του πίνακα Β σε μια γραμμή με ένα κενό χαρακτήρα μεταξύ τους.

β) Θα διαβάζει τα στοιχεία ενός μονοδιάστατου πίνακα Β και θα τα μεταφέρει στο διδιάστατο πίνακα Α κατά γραμμές. Στη συνέχεια θα εμφανίζει τον πίνακα Α κατά γραμμές.

**32.** Να ταξινομηθούν κατ' αύξουσα σειρά τα στοιχεία ενός πίνακα ακεραίων 10 θέσεων με τον αλγόριθμο τροποποιημένο Bubble-Sort - σύμφωνα με τον αλγόριθμο αυτό σαρώνουμε τα στοιχεία του πίνακα και αν ένα στοιχείο του είναι μεγαλύτερο από το επόμενο του, τότε ανταλλάσσουμε τις τιμές τους και δίνουμε την τιμή 1 σε μια μεταβλητή flag - η flag έχει την τιμή 0 κάθε φορά που μπαίνουμε στο βρόχο και αν πάρει τιμή 1 συνεχίζουμε να σαρώνουμε τον πίνακα μέχρις ότου διατηρήσει την τιμή 0 σε κάποιο πέρασμα του πίνακα - ένδειξη ότι ο πίνακας ταξινομήθηκε.

**33.** Να γραφεί πρόγραμμα που θα διαβάζει το πλήθος των στοιχείων Ν και τα στοιχεία ενός μονοδιάστατου πίνακα (μέγιστη διάσταση 50). Στη συνέχεια να υπολογίζει και να εμφανίζει το μέγιστο στοιχείο του και τις θέσεις στις οποίες αυτό εμφανίζεται. Προσπαθήστε να λύσετε το πρόβλημα με μία μόνο "σάρωση" του πίνακα.

**34.** Να συμπληρωθούν τα κενά έτσι ώστε να εμφανίζεται το μέγιστο του πίνακα Π[50].

Pos←1

Για ... από 2 μέχρι ...

Αν  $\Pi[\dots] > \Pi[\dots]$  τότε

...←...

Τέλος\_Αν

Τέλος\_επανάληψης

Εμφάνισε  $\Pi[\dots]$

**35.** Η τηλεοπτική σειρά Criminal Minds αποτελείται από 45 επεισόδια διάρκειας 48 λεπτών στην οποία έχουν πρωταγωνιστικούς ρόλους 11 ηθοποιοί. Γράψτε αλγόριθμο όπου:

1. Θα καταγράφονται τα ακόλουθα δεδομένα:

- πίνακας ΟΝ με τα ονόματα των 11 πρωταγωνιστών και
  - πίνακας Λ με τα λεπτά συμμετοχής του κάθε πρωταγωνιστή στο κάθε επεισόδιο, ελέγχοντας ώστε να αποδέχεται μόνο έγκυρους μη αρνητικούς αριθμούς.
2. Θα εμφανίζεται το όνομα κάθε πρωταγωνιστή, ακολουθούμενο από τα εξής ζητούμενα:
- το συνολικό χρόνο συμμετοχής του σε όλα τα επεισόδια μαζί.
  - τον αριθμό του τελευταίου επεισοδίου στο οποίο συμμετείχε.

**36.** Στη διάρκεια της χρονιάς ένας καθηγητής είχε 75 μαθητές, στους οποίους μοίρασε 35 φυλλάδια με σημειώσεις. Να γράψετε αλγόριθμο όπου:

(1) θα καταχωρούνται σε κατάλληλους πίνακες

το όνομα κάθε μαθητή, το πλήθος των σελίδων κάθε φυλλαδίου, που πρέπει να είναι τουλάχιστον 1 και το πλήθος των σελίδων που διάβασε ο κάθε μαθητής από το κάθε φυλλάδιο, που μπορεί να κυμαίνεται ανάμεσα στο 0 και στο πλήθος των σελίδων του φυλλαδίου.

(2) θα εμφανίζονται για κάθε φυλλάδιο, τα ονόματα όσων μαθητών το διάβασαν ολόκληρο.

(3) θα εμφανίζεται το όνομα του μαθητή που έχει διαβάσει τις λιγότερες σελίδες του μεγαλύτερου φυλλαδίου, θεωρώντας ότι είναι μοναδικός.

(4) θα εμφανίζονται ταξινομημένα σε αλφαβητική σειρά τα ονόματα των μαθητών που οι συνολικές σελίδες που έχουν διαβάσει ξεπερνάνε το 75% των σελίδων όλων των φυλλαδίων.

**37.** Σε έναν αγώνα F1 συμμετέχουν 10 ομάδες με 2 διαγωνιζόμενους η κάθε μία (δηλαδή συνολικά 20 αθλητές). Κάθε διαγωνιζόμενος τρέχει με το μονοθέσιό του σε μία διαδρομή και νικητής είναι αυτός που θα ολοκληρώσει την διαδρομή στον μικρότερο χρόνο. Να γίνει αλγόριθμος που με δεδομένα, τα ονόματα των 20 διαγωνιζόμενων, τις ομάδες στις οποίες αυτοί ανήκουν και τον χρόνο που ολοκλήρωσαν τη διαδρομή να βρίσκει και να εμφανίζει:

α) Τους τρεις πρώτους αθλητές και την ομάδα στην οποία ανήκουν.

β) Την ομάδα που είχε την μικρότερη διαφορά στην κατάταξη για τους 2 αθλητές της.

**38.** Το Υπουργείο Οικονομίας προκειμένου να περιορίσει τα έξοδα των Υπουργείων λόγω της οικονομικής κρίσης, κατέγραψε σε πίνακες τα ονόματα είκοσι υπουργείων και τα έξοδα τους για κάθε μήνα του προηγούμενου έτους. Οι περικοπές θα γίνουν στα υπουργεία που έχουν ετήσια έξοδα μεγαλύτερα από το μέσο όρο των είκοσι.

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού ΓΛΩΣΣΑ το οποίο:

1. Θα διαβάζει τα παραπάνω δεδομένα και θα τα αποθηκεύει στους πίνακες (να

ελέγχονται τα έξοδα να είναι θετικός αριθμός).

2. Θα εμφανίζει το όνομα του υπουργείου με τα μεγαλύτερα ετήσια έξοδα

3. Θα εμφανίζει τα υπουργεία στα οποία θα γίνουν περικοπές.

**39.** Ένας δρομέας πρέπει να πάρει μέρος σε δύο προκριματικές φάσεις για να συμμετέχει στον τελικό. Η πρώτη φάση έχει 3 αγώνες με όριο στο μέσο όρο τα 3 λεπτά. Η δεύτερη φάση αποτελείται από 5 αγώνες με όριο για το μέσο όρο τα 2 λεπτά. Η κάθε φάση είναι προαπαιτούμενη για την επόμενη. Να γραφτεί αλγόριθμος ο οποίος καταγράφει ανά φάση τις επιδόσεις σε πίνακα 8 θέσεων. Στο τέλος εμφανίζει μήνυμα αν έφτασε στον τελικό και με τι μέσο όρο σε κάθε φάση.

**40.** Να δοθεί η εκφώνηση της παρακάτω κωδικοποίησης.

**Αλγόριθμος** Αλγ2

**Δεδομένα** // A, N //

max ← A[1]

k ← 1

**Για** x από 2 μέχρι N

**Αν** A[x] = max τότε

        max ← A[x]

    k ← k+1

**αλλιώς\_αν** A[x] > max τότε

        k ← 0

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Αποτελέσματα** // k //

**Τέλος** Αλγ2

**41.** Στο τέλος του 2006 η Ευρωπαϊκή Επιτροπή (Commission) στην προσπάθεια της να συλλέξει στατιστικά στοιχεία σχετικά με τον τουρισμό στα 25 κράτη μέλη της, κατέγραψε τον αριθμό των τουριστών που επισκέφτηκαν την κάθε χώρα για τις χρονιές 2004, 2005 και 2006. Να γράψετε πρόγραμμα στη γλώσσα με το όνομα askisi12, το οποίο:

(α) να ζητά από το χρήστη το όνομα της κάθε χώρας και να το καταχωρεί σε μονοδιάστατο πίνακα με το όνομα countries. Παράλληλα, ζητά από το χρήστη τον αριθμό των τουριστών που επισκέφτηκαν την κάθε χώρα για κάθε μια από τις χρονιές 2004, 2005 και 2006. Τα στοιχεία αυτά τα καταχωρεί σε διδιάστατο πίνακα με το όνομα tourists.

(β) να υπολογίζει και να τυπώνει την αύξηση ή μείωση του αριθμού των τουριστών της κάθε χώρας, αφαιρώντας από τον αριθμό των τουριστών που επισκέφτηκαν την χώρα το 2006, αυτούς που την επισκέφτηκαν το 2004.

(γ) υποθέτοντας ότι ο κάθε τουρίστας ξοδεύει στη χώρα που επισκέπτεται 500 ευρώ, να τυπώνει το ποσό που κέρδισε η ΕΕ το 2005.

**42.** Ένα εργοστάσιο παρασκευάζει και πωλεί 6 διαφορετικά είδη χυμών. Τα ονόματα των 6 χυμών (π.χ. πορτοκάλι, μήλο, ροδάκινο, κ.λπ.) καταχωρούνται σε ένα μονοδιάστατο πίνακα με το όνομα Χυμοί. Σε έναν άλλο διδιάστατο παράλληλο πίνακα 6 γραμμών και 12 στηλών, με το όνομα Poliseis, καταχωρούνται οι μηνιαίες εισπράξεις ολόκληρου του έτους για κάθε είδος χυμού ξεχωριστά. Για παράδειγμα, η θέση [1,3] στον πίνακα Poliseis αντιπροσωπεύει τις εισπράξεις από τις πωλήσεις του πρώτου είδους χυμού για το μήνα Μάρτιο (3ος μήνας). Να γράψετε πρόγραμμα στη γλώσσα με το όνομα askisi43, το οποίο:

(α) Να ζητά από το χρήστη και να καταχωρεί τα στοιχεία στους πιο πάνω πίνακες.

(β) Να υπολογίζει και να τυπώνει τις συνολικές εισπράξεις του εργοστασίου από όλα τα προϊόντα για ολόκληρο το χρόνο.

(γ) Να υπολογίζει τις συνολικές εισπράξεις από το κάθε είδος χυμού ξεχωριστά, για ολόκληρο το χρόνο και να τις καταχωρεί σε μονοδιάστατο παράλληλο πίνακα με το όνομα Totals.

(δ) Να τυπώνει κατάλογο που να περιλαμβάνει το όνομα του κάθε χυμού και τις αντίστοιχες συνολικές εισπράξεις σε φθίνουσα σειρά με βάση τις συνολικές εισπράξεις. Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων.

**43.** Οι 150 μαθητές της Α΄ τάξης ενός Γυμνασίου της Κύπρου εξετάστηκαν σε ένα διαγώνισμα γενικών γνώσεων. Το διαγώνισμα περιλάμβανε 20 ερωτήσεις πολλαπλής επιλογής και για κάθε ερώτηση δίνονταν 5 επιλογές (Α, Β, C, D και E).

Να γράψετε πρόγραμμα σε γλώσσα με το όνομα askisi1, το οποίο:

α) να ζητά από το χρήστη τις σωστές απαντήσεις για τις 20 ερωτήσεις που δόθηκαν στο διαγώνισμα και να τις καταχωρεί στο μονοδιάστατο πίνακα Lyseis.

β) να ζητά από το χρήστη τις απαντήσεις που έδωσαν οι μαθητές στο διαγώνισμα και να τις καταχωρεί σε ένα διδιάστατο πίνακα 150 γραμμών και 20 στηλών με το όνομα Arantiseis, ο οποίος είναι παράλληλος με τον πίνακα Lyseis.

Κάθε γραμμή του πίνακα Arantiseis θα αντιστοιχεί με τις 20 απαντήσεις ενός μαθητή στο διαγώνισμα.

γ) να υπολογίζει το συνολικό βαθμό του κάθε μαθητή στο διαγώνισμα, δεδομένου ότι η κάθε σωστή απάντηση βαθμολογείται με 5 μονάδες. Οι βαθμοί όλων των μαθητών να καταχωρούνται σε μονοδιάστατο πίνακα με το όνομα Vathmoi, ο οποίος είναι παράλληλος με τον πίνακα Arantiseis.

δ) να ταξινομεί σε φθίνουσα σειρά τα στοιχεία του πίνακα Vathmoi και ακολούθως να τα τυπώνει.

ε) να ζητά από το χρήστη τον αριθμό μιας συγκεκριμένης ερώτησης. Ακολούθως, να καλεί μια διαδικασία με το όνομα erotisi, η οποία να δέχεται τους πίνακες Lyseis και Arantiseis, καθώς και τον αριθμό της ερώτησης τον οποίο έδωσε ο χρήστης. Η διαδικασία να υπολογίζει και να επιστρέφει στο κυρίως πρόγραμμα για εκτύπωση των αριθμό των μαθητών που απάντησαν σωστά σε αυτήν την ερώτηση. π.χ. δώσε αριθμό ερώτησης (1-20): 18.

Η ερώτηση αυτή απαντήθηκε σωστά από 73 μαθητές.

Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων. Θεωρείστε ότι τα δεδομένα δίνονται σωστά και δε χρειάζεται έλεγχος.

**44.** Η στατιστική υπηρεσία του champions league μελετά 30 γνωστούς ποδοσφαιριστές. Σε έναν πίνακα ON καταγράφει τα ονόματά τους και στον ΕΠΙΔ τις επιδόσεις τους ως εξής: Η πρώτη στήλη καταγράφει τις πάσες, η δεύτερη τα γκολ, η τρίτη τον δείκτη δύναμης, η τέταρτη τα κερδισμένα φάουλ, η πέμπτη τα λάθη και η έκτη το χρόνο συμμετοχής του (0-4000 λεπτά) κάθε αθλητή. Να γραφεί αλγόριθμος ο οποίος:

1. Διαβάζει τους πίνακες (να γίνει έλεγχος στο χρόνο).

2. Βρίσκει τον πρώτο σκόρερ (πιθανότητα ισότητας).

3. Εμφανίζει τον παίκτη με τον καλύτερο συντελεστή-λόγο γκολ/χρόνο συμμετοχής.

4. Να εμφανίσετε τους 4 καλύτερους παίκτες με βάση το εξής κριτήριο: (γκολ+πάσες- λάθη)/χρόνος συμμετοχής.



**45.** Στον τελικό ενός αγώνα ακοντισμού συμμετέχουν 12 αθλητές κάνοντας ο καθένας τους 6 προσπάθειες. Να γράψετε αλγόριθμο ο οποίος:

1. Διαβάζει τα ονόματα των αθλητών καθώς και τις επιδόσεις τους στις έξι προσπάθειες και τις καταχωρεί σε ανάλογους πίνακες.
2. Βρίσκει τα ονόματα των αθλητών καθώς και τις επιδόσεις τους στις έξι προσπάθειες και τις καταχωρεί σε ανάλογους πίνακες.
3. Βρίσκει για κάθε αθλητή τον αριθμό (1-6) της προσπάθειας (υποθέτουμε ότι είναι μόνο μία), στην οποία έκανε την καλύτερη επίδοση και τους καταχωρεί σε πίνακα.
4. Ταξινομεί τις επιδόσεις κάθε αθλητή από την καλύτερη προς τη χειρότερη.
5. Ταξινομεί τους αθλητές από τον καλύτερο προς τον χειρότερο. Υποθέτουμε ότι δεν υπάρχουν δύο αθλητές που οι επιδόσεις τους ενός είναι ίσες με τις επιδόσεις του άλλου.
6. Εμφανίζει την τελική κατάταξη του αγωνίσματος. Για κάθε αθλητή εμφανίζει: Τη σειρά κατάταξης του, το όνομά του και τον αριθμό της προσπάθειας που έκανε την καλύτερη επίδοση. (Θέμα Πανελλαδικών)

**46.** Ένα εμπορικό κατάστημα έχει καταγράψει τις μηνιαίες εισπράξεις του για τα έτη 2009 και 2010.

Να γράψετε αλγόριθμο ο οποίος:

1. Να διαβάζει τις μηνιαίες εισπράξεις για καθένα από τα δύο έτη και να τις καταχωρίζει σε αντίστοιχους μονοδιάστατους πίνακες.
2. Να υπολογίζει και να εμφανίζει τη μεγαλύτερη μηνιαία εισπράξη για κάθε έτος. Θεωρήστε ότι για κάθε έτος η τιμή αυτή είναι μοναδική.
3. Να εμφανίζει κατάλληλο μήνυμα στην περίπτωση που ο μήνας κατά τον οποίο σημειώθηκε η μεγαλύτερη μηνιαία εισπράξη ήταν ο ίδιος και για τα δύο έτη.
4. Να εμφανίζει το μέσο όρο των μηνιαίων εισπράξεων για κάθε έτος.
5. Να υπολογίζει και να εμφανίζει το πλήθος των μηνών του έτους 2009 κατά τους οποίους η μηνιαία εισπράξη ήταν μεγαλύτερη από αυτή του αντίστοιχου μήνα του έτους 2010.

**47.** Σε μια δημοτική δανειστική βιβλιοθήκη υπάρχουν 158 μέλη που δανείζονται βιβλία. Να γραφεί αλγόριθμος που:

1. Για κάθε μέλος διαβάζει το επώνυμο και το φύλο του (Α=άνδρας, Γ=γυναίκα) και τα αποθηκεύει στους πίνακες ΜΕΛΗ και ΦΥΛΟ, αντίστοιχα. Να γίνεται έλεγχος εγκυρότητας εισαγωγής του φύλου. Για κάθε μήνα ενός έτους διαβάζει το πλήθος των βιβλίων που δανείστηκε κάθε μέλος και το αποθηκεύει στον πίνακα δύο διαστάσεων ΒΙΒΛΙΑ.
2. Για κάθε μέλος υπολογίζει το συνολικό αριθμό των βιβλίων που δανείστηκε στο έτος και το αποθηκεύει στον πίνακα SUM.
3. α. Υπολογίζει το συνολικό αριθμό των βιβλίων που δανείστηκαν οι άνδρες.  
β. Υπολογίζει το συνολικό αριθμό των βιβλίων που δανείστηκαν οι γυναίκες.  
γ. Εμφανίζει κατάλληλο μήνυμα που δείχνει αν οι άνδρες ή οι γυναίκες έχουν δανειστεί τα περισσότερα βιβλία. Σε περίπτωση ίσων συνολικών αριθμών βιβλίων να εμφανίζει το μήνυμα "ΙΣΟΣ ΑΡΙΘΜΟΣ ΒΙΒΛΙΩΝ".
4. Να διαβάζει ένα επώνυμο και χρησιμοποιώντας τη σειριακή αναζήτηση, σε περίπτωση που το επώνυμο είναι αποθηκευμένο στον πίνακα ΜΕΛΗ, να εμφανίζει το σύνολο των βιβλίων που δανείστηκε στη διάρκεια του έτους. Σε περίπτωση που το επώνυμο δεν είναι αποθηκευμένο στον πίνακα να εμφανίζει το μήνυμα "ΤΟ ΕΠΩΝΥΜΟ ΑΥΤΟ ΔΕΝ ΥΠΑΡΧΕΙ".

**48.** Δίνεται το παρακάτω τμήμα αλγορίθμου που χρησιμοποιεί ένα μονοδιάστατο πίνακα Α[20]. Ο πίνακας περιέχει άρτιους και περιττούς θετικούς ακεραίους, σε τυχαίες θέσεις.

Το τμήμα αλγορίθμου δημιουργεί ένα νέο πίνακα B[20] στον οποίο υπάρχουν πρώτα οι άρτιοι και μετά ακολουθούν οι περιττοί. Να γράψετε τον αλγόριθμο συμπληρώνοντας τα κενά:

$K \leftarrow 0$

Για i από ..... μέχρι .....

    Αν  $A[i] \bmod 2 = 0$  τότε

$K \leftarrow \dots\dots\dots$

$B[\dots\dots\dots] \leftarrow A[i]$

    Τέλος\_Αν

Τέλος\_επανάληψης

Για i από ..... μέχρι .....

    Αν  $A[i] \bmod 2 = \dots\dots\dots$  τότε

        .....

$B[\dots\dots\dots] \leftarrow A[\dots\dots\dots]$

Τέλος\_Αν

Τέλος\_επανάληψης

**49.** Να γραφεί αλγόριθμος ο οποίος να βελτιώνει τον αλγόριθμο ταξινόμησης φουσαλίδας ενός πίνακα κατά φθίνουσα σειρά, ελέγχοντας αν έχουν γίνει αντιμεταθέσεις στοιχείων, οπότε συνεχίζει τις συγκρίσεις ή αντιθέτως αν δεν έχει γίνει καμία αντιμετάθεση σε αυτήν την περίπτωση σταματά.

**50.** Να γραφεί αλγόριθμος ο οποίος

A. Διαβάζει σε πίνακα τα ονόματα 20 οδηγών Enduro.

B. Γεμίζει διδιάστατο πίνακα με τη βαθμολογία των οδηγών σε 8 αγώνες.

Γ. Ταξινομεί σε φθίνουσα διάταξη τις βαθμολογίες του κάθε οδηγού.

Δ. Υπολογίζει την τελική βαθμολογία του κάθε οδηγού η οποία προκύπτει από το άθροισμα των τριών καλύτερων επιδόσεών του.

E. Εμφανίζει τον πρωταθλητή (μοναδικός) enduro και τη βαθμολογία του.

**51.** Να γραφεί αλγόριθμος ο οποίος προσομοιώνει ένα ιατρικό πείραμα ως εξής:

α. Διαβάζει τα ονόματα 50 ατόμων που συμμετέχουν εθελοντικά στο πείραμα σε πίνακα ON [50].

β. Διαβάζει τις μηνιαίες μικροβιολογικές μετρήσεις τους σε βιταμίνη B12 στον πίνακα B12 [50,12] κατά τη διάρκεια του 2012.

γ. Υπολογίζει και τυπώνει τον 7ο μήνα πόσοι και ποιοι είχαν μέτρηση κάτω από 500.

δ. Συνήθως η B12 αυξάνει σταθερά μηνιαία τους καλοκαιρινούς μήνες. Να υπολογιστεί σε πόσους επαληθεύεται αυτή η πεποίθηση.

**52.** Να γραφεί πρόγραμμα που θα απομακρύνει από ένα μονοδιάστατο αριθμητικό πίνακα ακεραίων N θέσεων, όλες τις πολλαπλές ίδιες τιμές δημιουργώντας ένα νέο πίνακα όπου κάθε αριθμός θα εμφανίζεται μόνο μία φορά.

**53.** Να γραφεί αλγόριθμος για έναν πίνακα 10 θέσεων στον οποίο θα καταχωρούνται αριθμητικές τιμές και θα γίνεται έλεγχος για τη μοναδικότητα της κάθε τιμής. Θα πρέπει κάθε τιμή του πίνακα να ελέγχεται, αφού είναι μοναδική σε σχέση με όλες τις προηγούμενες να επιτρέπεται η καταχώρηση της στην αντίστοιχη θέση του πίνακα.

**54.** Έστω δεδομένος πίνακας με τα εξής στοιχεία:

5 36 87 25 93 20 4 92 17 85 23 6 72 56 85 75 45 79.

Οι αριθμοί είναι γραμμένοι σε ομάδες. Ο 1ος αριθμός κάθε ομάδας δείχνει το πλήθος των μελών της ομάδας που ακολουθεί αμέσως μετά τον αριθμό αυτό. Να γραφεί αλγόριθμος ο οποίος υπολογίζει και εμφανίζει το μέσο όρο κάθε ομάδας.

**55.** Η επικρατούσα τιμή ενός συνόλου τιμών είναι η τιμή του στοιχείου, το οποίο εμφανίζει τη μέγιστη συχνότητα παρουσίας. Να γραφεί αλγόριθμος ο οποίος εμφανίζει την επικρατούσα τιμή (μοναδική) ενός δεδομένου πίνακα 100 ακεραίων.

**56.** Στις εξετάσεις της Πυροσβεστικής για 70 θέσεις παίρνουν μέρος 1500 υποψήφιοι οι οποίοι διαγωνίστηκαν σε 60 συνολικά ερωτήσεις τύπου Σωστό-Λάθος. Για την απλοποίηση της διόρθωσης των γραπτών οι σωστές απαντήσεις με Σ ή Λ καταχωρούνται σε έναν πίνακα ΑΠ[60]. Να γραφεί αλγόριθμος ο οποίος

1. θα καταχωρεί τα ονόματα των υποψηφίων σε έναν πίνακα ΟΝ[1500].
2. θα καταχωρεί τις απαντήσεις των υποψηφίων σε έναν πίνακα Β[1500,60] και τις σωστές απαντήσεις σε έναν πίνακα Α[60].
3. θα βρίσκει το πλήθος των σωστών απαντήσεων κάθε υποψηφίου συγκρίνοντας τα στοιχεία των δύο πινάκων και θα το αποθηκεύει σε έναν πίνακα ΠΛ[1500].
4. θα ταξινομεί τον πίνακα ΠΛ[1500] κατά φθίνουσα σειρά και θα εμφανίζει τα ονόματα των 70 επιτυχόντων. Σε περίπτωση ισοβαθμίας η σειρά των ονομάτων θα είναι αλφαβητική.
5. Αν υπάρχουν επιλαχόντες θα εμφανίζεται το πλήθος τους και το ονοματεπώνυμο του καθενός.







## 5.1 Λυμένες Ασκήσεις

1. Η κρυπτογράφηση χρησιμοποιείται για την προστασία των μεταδιδόμενων πληροφοριών. Ένας απλός αλγόριθμος κρυπτογράφησης χρησιμοποιεί την αντιστοίχιση κάθε γράμματος ενός κειμένου σε ένα άλλο γράμμα της αλφαβήτου. Για το σκοπό αυτό δίνεται πίνακας  $AB[2,24]$ , ο οποίος στην πρώτη γραμμή του περιέχει σε αλφαβητική σειρά τους χαρακτήρες από το Α έως και το Ω. Στη δεύτερη γραμμή του βρίσκονται οι ίδιοι χαρακτήρες, αλλά με διαφορετική σειρά. Κάθε χαρακτήρας της πρώτης γραμμής κρυπτογραφείται στον αντίστοιχο χαρακτήρα της δεύτερης γραμμής, που βρίσκεται στην ίδια στήλη. Επίσης, δίνεται πίνακας  $KEIM[500]$ , ο οποίος περιέχει αποθηκευμένο με κεφαλαία ελληνικά γράμματα το προς κρυπτογράφηση κείμενο. Κάθε χαρακτήρας του κειμένου βρίσκεται σε ένα κελί του πίνακα  $KEIM[500]$ . Οι λέξεις του κειμένου χωρίζονται με ένα χαρακτήρα κενό (' '), ενώ στο τέλος του κειμένου μπορεί να υπάρχουν χαρακτήρες κενό (' '), μέχρι να συμπληρωθεί ο πίνακας.

Να αναπτύξετε αλγόριθμο ο οποίος:

1. Να εμφανίζει το πλήθος των χαρακτήρων κενό (' '), που υπάρχουν μετά το τέλος του κειμένου στον πίνακα  $KEIM[500]$ . Αν δεν υπάρχει χαρακτήρας κενό μετά τον τελευταίο χαρακτήρα του μη κρυπτογραφημένου κειμένου, τότε να εμφανίζεται το μήνυμα: «Το μήκος του κειμένου είναι 500 χαρακτήρες». Θεωρήστε ότι ο πίνακας  $KEIM[500]$  περιέχει τουλάχιστον μία λέξη.

2. Να κρυπτογραφεί τους χαρακτήρες του πίνακα  $KEIM[500]$  στον πίνακα  $KRYΠ[500]$ , με βάση τον πίνακα  $AB[2,24]$ . Η κρυπτογράφηση να τερματίζεται με το τέλος του κειμένου. Δίνεται ότι κάθε χαρακτήρας κενό, που υπάρχει στον πίνακα  $KEIM[500]$ , παραμένει χαρακτήρας κενό στον πίνακα  $KRYΠ[500]$ .

3. Να εμφανίζει το πλήθος των λέξεων του κειμένου, καθώς και το πλήθος των χαρακτήρων που έχει η μεγαλύτερη λέξη του κειμένου στον πίνακα  $KRYΠ[500]$ .

Θεωρήστε ότι η μεγαλύτερη λέξη είναι μοναδική. (Θέμα Πανελλαδικών)

### Λύση

**Αλγόριθμος** ΘέμαΓ

**Δεδομένα** //AB, KEIM//

πλήθος ← 0

I ← 500

**Όσο**  $KEIM[I] = ' '$  **ΚΑΙ**  $I \geq 1$  **επανάλαβε**

πλήθος ← πλήθος + 1

I ← I - 1

Τέλος\_επανάληψης

**Αν** πλήθος = 0 **τότε**

**Εμφάνισε** 'Το μήκος του κειμένου είναι 500 χαρακτήρες'

αλλιώς

**Εμφάνισε** 'Το πλήθος των κενών είναι:', πλήθος

**Τέλος\_Αν**

πλ\_λεξ ← 0

max ← 0

πλ\_χαρ ← 0

**Για** I **από** 1 **μέχρι** 500 - πλήθος

**Αν**  $KEIM[I] = ' '$  **τότε**

```

ΚΡΥΠ[Ι] ← ‘ ‘
  Αν πλ_χαρ > max τότε max ← πλήθος
  πλ_λεξ ← πλ_λεξ + 1
  πλ_χαρ ← 0
  αλλιώς
  J ← 1
  flag ← ΨΕΥΔΗΣ
  pos ← 0
  Όσο J <= 24 ΚΑΙ flag = ΨΕΥΔΗΣ επανάλαβε
    Αν AB[1,J] = ΚΕΙΜ[Ι] τότε
      pos ← J
      flag ← ΑΛΗΘΗΣ
      αλλιώς
      J ← J + 1
      Τέλος_Αν
    Τέλος_επανάληψης
  ΚΡΥΠ[Ι] ← AB[2,pos]
  πλ_χαρ ← πλ_χαρ + 1
  Τέλος_Αν

```

**Τέλος\_επανάληψης**

**Εμφάνισε** ‘Πλήθος λέξεων κειμένου:’, πλ\_λεξ

**Εμφάνισε** ‘Πλήθος χαρακτήρων μεγαλύτερη λέξης:’, max

**Τέλος** ΘέμαΓ

**2.** Ας θεωρήσουμε τον πίνακα  $T[K, K]$  ( $K$  θετικός ακέραιος). Προσδιορίστε τις προϋποθέσεις που πρέπει να πληρούνται ώστε το στοιχείο  $T[\rho, \sigma]$  να ανήκει: α) στην πρώτη κύρια διαγώνιο β) στη δευτερεύουσα διαγώνιο.

**Απάντηση**

α)  $\rho = \sigma$

β)  $\rho + \sigma = K + 1$

**3.** Γράψτε αλγόριθμο που:

α) διαβάζει 150 άρτιους θετικούς ακέραιους και τους καταχωρεί στον ακέραιο πίνακα  $\Psi[10, 5, 3]$ ,

β) υπολογίζει: τη μικρότερη τιμή, το μέσο όρο όλου του πίνακα και το ποσοστό (%) των στοιχείων του, που είναι μικρότερα από το 88.

**Λύση**

**Αλγόριθμος** ΑΣΚ

Για κ από 1 μέχρι 10

  Για λ από 1 μέχρι 5

    Για μ από 1 μέχρι 3

**Αρχή\_επανάληψης**

**Διάβασε**  $\Psi[k, \lambda, \mu]$

**Μέχρις\_ότου** ( $\Psi[k, \lambda, \mu] > 0$ ) **ΚΑΙ**

          ( $\Psi[k, \lambda, \mu] \bmod 2 = 0$ )



```

    Τέλος_επανάληψης
  Τέλος_επανάληψης
Τέλος_επανάληψης
Μικ ← Ψ[1, 1, 1]
σ ← 0
φ ← 0
Για κ από 1 μέχρι 10
  Για λ από 1 μέχρι 5
    Για μ από 1 μέχρι 3
      Αν Ψ[κ, λ, μ] < Μικ τότε
        Μικ ← Ψ[κ, λ, μ]
      Τέλος_Αν
      σ ← σ+Ψ[κ, λ, μ]
      Αν Ψ[κ, λ, μ] < 88 τότε
        φ ← φ+1
      Τέλος_Αν
    Τέλος_επανάληψης
  Τέλος_επανάληψης
ΜΟ ← σ/150
ΠΟΣ ← 100*φ/150
Εμφάνισε Μικ, ΜΟ, ΠΟΣ
Τέλος ΑΣΚ

```

4. Γράψτε αλγόριθμο που, με δεδομένο τον πραγματικό πίνακα  $A[50,30]$ , αντιγράφει τα στοιχεία των άρτιων γραμμών του πίνακα αυτού στον πραγματικό πίνακα  $B[25,30]$  και τα στοιχεία των περιττών γραμμών του  $A$  στον πραγματικό πίνακα  $\Gamma[25,30]$ .

#### Λύση

```

Αλγόριθμος Διαχ
Δεδομένα // A //
μ ← 1
ξ ← 1
Για κ από 1 μέχρι 50
  Για λ από 1 μέχρι 30
    Αν κ mod 2 = 0 τότε
      Β[μ, λ] ← Α[κ, λ]
      Αν λ = 30 τότε
        μ ← μ+1
      Τέλος_Αν
    αλλιώς
      Γ[ξ, λ] ← Α[κ, λ]
      Αν λ = 30 τότε
        ξ ← ξ+1
      Τέλος_Αν
  Τέλος_Αν
Τέλος_επανάληψης
Τέλος_επανάληψης

```

5. Γράψτε αλγόριθμο που ελέγχει αν όλα τα στοιχεία του ακεραίου πίνακα  $X[6, 4]$  είναι θετικά και εμφανίζει κατάλληλο ενημερωτικό μήνυμα.

**Λύση**

! α' τρόπος

Αλγόριθμος Ιδιότητα

Δεδομένα // X //

$\Phi \leftarrow 0$

Για κ από 1 μέχρι 6

  Για λ από 1 μέχρι 4

    Αν  $X[k, \lambda] > 0$  τότε

$\Phi \leftarrow \Phi + 1$

    Τέλος\_Αν

  Τέλος\_επανάληψης

Τέλος\_επανάληψης

Αν  $\Phi = 24$  τότε

  Εμφάνισε 'Όλα τα στοιχεία είναι θετικά'

αλλιώς

  Εμφάνισε 'Δεν είναι όλα τα στοιχεία θετικά'

Τέλος\_Αν

Τέλος Ιδιότητα

! β' τρόπος

Αλγόριθμος Ιδιότητα

Δεδομένα // X //

Ελ  $\leftarrow$  Αληθής

κ  $\leftarrow$  1

Όσο (κ  $\leq$  6) ΚΑΙ (Ελ = Αληθής) επανάλαβε

  λ  $\leftarrow$  1

  Όσο (λ  $\leq$  4) ΚΑΙ (Ελ = Αληθής) επανάλαβε

    Αν  $X[k, \lambda] \leq 0$  τότε

      Ελ  $\leftarrow$  Ψευδής

    αλλιώς

      λ  $\leftarrow$  λ+1

    Τέλος\_Αν

  Τέλος\_επανάληψης

Αν Ελ = Αληθής τότε

  κ  $\leftarrow$  κ+1

Τέλος\_Αν

Τέλος\_επανάληψης

Αν Ελ = Αληθής τότε

  Εμφάνισε 'Όλα τα στοιχεία είναι θετικά'

αλλιώς

  Εμφάνισε 'Δεν είναι όλα τα στοιχεία θετικά'

Τέλος\_Αν

## Τέλος Ιδιότητα

6. Γράψτε αλγόριθμο που γεμίζει τον πίνακα  $A[3, 4]$  με το ακόλουθο περιεχόμενο:

12	15	18	21
21	24	27	30
30	33	36	39

### Λύση

**Αλγόριθμος ΑΣΚ**

$Y \leftarrow 12$

**Για κ από 1 μέχρι 3**

**Για λ από 1 μέχρι 4**

$A[k, \lambda] \leftarrow Y$

$Y \leftarrow Y+3$

**Τέλος\_επανάληψης**

$Y \leftarrow Y-3$

**Τέλος\_επανάληψης**

**Αποτελέσματα // A //**

**Τέλος ΑΣΚ**

7. Γράψτε αλγόριθμο που γεμίζει τον πίνακα  $A[3, 3]$  με το ακόλουθο περιεχόμενο

εφ(10)	ημ(20)	εφ(30)
ημ(40)	εφ(50)	ημ(60)
εφ(70)	ημ(80)	εφ(90)

### Λύση

**Αλγόριθμος Ασκ**

$Y \leftarrow 10$

**Για κ από 1 μέχρι 3**

**Για λ από 1 μέχρι 3**

**Αν  $(κ+λ) \bmod 2 = 1$  τότε**

$A[k, \lambda] \leftarrow ΕΦ(Y)$

**αλλιώς**

$A[k, \lambda] \leftarrow ΗΜ(Y)$

**Τέλος\_Αν**

$Y \leftarrow Y+10$

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Αποτελέσματα // A //**

**Τέλος Ασκ**

8. Ας θεωρήσουμε τους ακέραιους πίνακες  $A[P, \Lambda]$  και  $B[P, M]$ . Γράψτε αλγόριθμο που δοθέντων των παραπάνω πινάκων, επιτυγχάνει τη συγχώνευσή τους στον ακέραιο πίνακα  $\Gamma[P, \Lambda+M]$ .

### Λύση

**Αλγόριθμος** Συγχώνευση  
**Δεδομένα** // A, B, P, Λ, M //  
**Για** κ από 1 μέχρι P  
     **Για** ξ από 1 μέχρι Λ  
          $\Gamma[k, \xi] \leftarrow A[k, \xi]$   
     **Τέλος\_επανάληψης**  
**Τέλος\_επανάληψης**  
**Για** κ από 1 μέχρι P  
     **Για** ξ από 1 μέχρι M  
          $\Gamma[k, \Lambda + \xi] \leftarrow B[k, \xi]$   
     **Τέλος\_επανάληψης**  
**Τέλος\_επανάληψης**  
**Αποτελέσματα** // Γ //  
**Τέλος** Συγχώνευση

9. Θεωρείστε τον πίνακα Βαθμ[28,6] που περιέχει τους βαθμούς που έγραψαν οι 28 μαθητές της Γ' τάξης ενός γενικού λυκείου στα 6 πανελλαδικώς εξεταζόμενα μαθήματα. Γράψτε αλγόριθμο που με δεδομένο τον παραπάνω πίνακα, εμφανίζει τους τρεις καλύτερους βαθμούς επίδοσης για κάθε μαθητή ξεχωριστά.

**Λύση**

**Αλγόριθμος** Μαθητές  
**Δεδομένα** // Βαθμ //  
**Για** Μαθητής από 1 μέχρι 28  
     ! αντιγράφω τη γραμμή Μαθητής στον A[6]  
     **Για** κ από 1 μέχρι 6  
          $A[k] \leftarrow \text{Βαθμ}[\text{Μαθητής}, k]$   
     **Τέλος\_επανάληψης**  
     ! Ταξινομώ τον A κατά φθίνουσα σειρά  
     **Για** κ από 2 μέχρι 6  
         **Για** λ από 6 μέχρι κ με\_βήμα -1  
             **Αν**  $A[\lambda - 1] < A[\lambda]$  τότε  
                 Βοηθ  $\leftarrow A[\lambda - 1]$   
                  $A[\lambda - 1] \leftarrow A[\lambda]$   
                  $A[\lambda] \leftarrow \text{Βοηθ}$   
         **Τέλος\_Αν**  
     **Τέλος\_επανάληψης**  
     **Τέλος\_επανάληψης**  
     ! Εμφανίζω τα τρία πρώτα στοιχεία του A  
     **Για** κ από 1 μέχρι 3  
         Εμφάνισε  $A[k]$   
     **Τέλος\_επανάληψης**  
**Τέλος\_επανάληψης**  
**Τέλος** Μαθητές

10. Θεωρήστε τον ακέραιο πίνακα A[6, 6] που περιέχει 18 άρτιες και 18 περιττές τιμές. Γράψτε αλγόριθμο που, με δεδομένο τον πίνακα A, αντιγράφει όλες τις άρτιες τιμές του A

στον ακέραιο πίνακα  $B[6, 3]$  και όλες τις περιττές τιμές του  $A$  στον ακέραιο πίνακα  $\Gamma[6, 3]$ .

### Λύση

**Αλγόριθμος Διαμερισμός**

**Δεδομένα** //  $A$  //

$\Gamma_{\rho B} \leftarrow 1$

$\Sigma_{\tau B} \leftarrow 1$

$\Gamma_{\rho \Gamma} \leftarrow 1$

$\Sigma_{\tau \Gamma} \leftarrow 1$

**Για**  $K$  **από** 1 **μέχρι** 6

**Για**  $\Lambda$  **από** 1 **μέχρι** 6

**Αν**  $A[K, \Lambda] \bmod 2 = 0$  **τότε**

$B[\Gamma_{\rho B}, \Sigma_{\tau B}] \leftarrow A[K, \Lambda]$

$\Sigma_{\tau B} \leftarrow \Sigma_{\tau B} + 1$

**Αν**  $\Sigma_{\tau B} > 3$  **τότε**

$\Sigma_{\tau B} \leftarrow 1$

$\Gamma_{\rho B} \leftarrow \Gamma_{\rho B} + 1$

**αλλιώς**

$\Sigma_{\tau B} \leftarrow \Sigma_{\tau B} + 1$

**Τέλος\_Αν**

**αλλιώς**

$\Gamma[\Gamma_{\rho \Gamma}, \Sigma_{\tau \Gamma}] \leftarrow A[K, \Lambda]$

$\Sigma_{\tau \Gamma} \leftarrow \Sigma_{\tau \Gamma} + 1$

**Αν**  $\Sigma_{\tau \Gamma} > 3$  **τότε**

$\Sigma_{\tau \Gamma} \leftarrow 1$

$\Gamma_{\rho \Gamma} \leftarrow \Gamma_{\rho \Gamma} + 1$

**αλλιώς**

$\Sigma_{\tau \Gamma} \leftarrow \Sigma_{\tau \Gamma} + 1$

**Τέλος\_Αν**

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Αποτελέσματα** //  $B, \Gamma$  //

**Τέλος Διαμερισμός**

**11.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο:

α. Θα διαβάζει τα ονόματα 20 εταιριών υψηλής κεφαλαιοποίησης που είναι εισηγμένες στο Χρηματιστήριο Αξιών Αθηνών, καθώς και τα καθαρά τους κέρδη για κάθε έτος ξεχωριστά από το 1990 μέχρι και το 1999 (υποθέστε ότι καμία χρονιά δε σημείωσαν ζημία).

β. Θα υπολογίζει για κάθε εταιρία την τετραετία εκείνη που είχε τα υψηλότερα κέρδη.

γ. Για κάθε εταιρία θα εμφανίζει μήνυμα του τύπου «Η εταιρία όνομα εταιρίας σημείωσε τα υψηλότερα κέρδη της, την τετραετία από το 199\_μέχρι το 199\_».

### Λύση

**ΠΡΟΓΡΑΜΜΑ ΧΑΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΧΑΡΑΚΤΗΡΕΣ:** ΟΝΟΜΑ [20]

ΑΚΕΡΑΙΕΣ : I, J, K, SUM , ΚΕΡΔΗ [20,10], FOUR [20], YEAR [20]

ΑΡΧΗ

! Εισαγωγή στοιχείων

Για I ΑΠΟ 1 ΜΕΧΡΙ 20

ΓΡΑΨΕ ' ΔΩΣΕ ΤΟ ΟΝΟΜΑ ΤΗΣ ', I, ' ης ΕΤΑΙΡΙΑΣ: '

ΔΙΑΒΑΣΕ ΟΝΟΜΑ[I]

Για J ΑΠΟ 1 ΜΕΧΡΙ 10

ΓΡΑΨΕ ' ΔΩΣΕ ΤΑ ΚΕΡΔΗ ΤΗΣ ', I, ' ης ΕΤΑΙΡΙΑΣ ΤΟ ΕΤΟΣ ', 1989 + J

ΔΙΑΒΑΣΕ ΚΕΡΔΗ [I, J]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

Για I ΑΠΟ 1 ΜΕΧΡΙ 20

FOUR [I] ← 0

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

Για K ΑΠΟ 1 ΜΕΧΡΙ 7

Για I ΑΠΟ 1 ΜΕΧΡΙ 20

SUM ← 0

Για J ΑΠΟ K ΜΕΧΡΙ K+3

SUM ← SUM + ΚΕΡΔΗ [I, J]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

Αν SUM > FOUR [I] ΤΟΤΕ

FOUR [I] ← SUM

YEAR [I] ← K

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

Για I ΑΠΟ 1 ΜΕΧΡΙ 20

ΓΡΑΨΕ ' Η εταιρία ', ΟΝΟΜΑ [I], ' σημείωσε τα υψηλότερα κέρδη της, την τετραετία από το ', 1989 + YEAR [I], ' μέχρι το ', 1992 + YEAR [I]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

12. Να γραφτεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο θα διαβάζει τα στοιχεία ενός πίνακα ακέραιων 100x100 και θα βρίσκει τυπώνοντας αντίστοιχο μήνυμα αν ο πίνακας είναι «Μαγικός» ή όχι. Σημείωση: «Μαγικός» είναι ο πίνακας που το άθροισμα κάθε γραμμής και κάθε στήλης αλλά και των δύο διαγωνίων του είναι σταθερό

**Λύση**

ΠΡΟΓΡΑΜΜΑ ΜΑΓΙΚΟΣ\_ΠΙΝΑΚΑΣ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: I, J, A[100,100], ROW[100], COL[100], SUM 1, SUM 2

ΛΟΓΙΚΕΣ: DONE

ΑΡΧΗ

!Εισαγωγή στοιχείων

ΓΡΑΨΕ ' ΔΩΣΕ ΤΑ ΣΤΟΙΧΕΙΑ ΤΟΥ ΠΙΝΑΚΑ '

Για I ΑΠΟ 1 ΜΕΧΡΙ 100

Για J ΑΠΟ 1 ΜΕΧΡΙ 100

ΔΙΑΒΑΣΕ A[I,J]

```

                ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
! Εύρεση αθροίσματος γραμμών-στηλών
Για Ι ΑΠΟ 1 ΜΕΧΡΙ 100
ROW [I] ←0
COL [I] ←0
ΤΕΛΟΣ_ ΕΠΑΝΑΛΗΨΗΣ
        Για Ι ΑΠΟ 1 ΜΕΧΡΙ 100
                Για J ΑΠΟ 1 ΜΕΧΡΙ 100
                        ROW [I]←ROW[I] +A [I, J]
                        COL [J]← COL [J] + A [I, J]
                ΤΕΛΟΣ_ ΕΠΑΝΑΛΗΨΗΣ
        ΤΕΛΟΣ_ ΕΠΑΝΑΛΗΨΗΣ
! Εύρεση αθροισμάτων κύριας και δευτερεύουσας διαγωνίου
SUM1 ←0
SUM2 ←0
Για Ι ΑΠΟ 1 ΜΕΧΡΙ 100
SUM1 ← SUM1 + A[I,I]
SUM2 ← SUM2 + A [I, 101-I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
! Έλεγχος αν όλα τα στοιχεία είναι ίσα
DONE ←ΑΛΗΘΗΣ
Αν (SUM1<>SUM2) ΤΟΤΕ
DONE ←ΨΕΥΔΗΣ
ΑΛΛΙΩΣ
I←1
        Όσο (I<=100) ΚΑΙ (DONE = ΑΛΗΘΗΣ) ΕΠΑΝΑΛΑΒΕ
                Αν (SUM1<>ROW[I] Ή (SUM1<>COL[I])ΤΟΤΕ
                        DONE←ΨΕΥΔΗΣ
                ΑΛΛΙΩΣ
                        I←I+1
                ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ
Αν DONE= ΑΛΗΘΗΣ ΤΟΤΕ
ΓΡΑΨΕ ' Ο ΠΙΝΑΚΑΣ ΕΙΝΑΙ ΜΑΓΙΚΟΣ'
ΑΛΛΙΩΣ
ΓΡΑΨΕ ' Ο ΠΙΝΑΚΑΣ ΔΕΝ ΕΙΝΑΙ ΜΑΓΙΚΟΣ'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

**13.** Μια σκακιέρα αποτελείται από 8 οριζόντιες και 8 κάθετες γραμμές δημιουργώντας έτσι 64 νταμάκια. Σε κάθε ένα από τα νταμάκια αυτά, αναγράφουμε την βαθμολογία [1-10] που θα παίρνει ένα πιόνι όταν βρεθεί πάνω του. Στην αρχή τοποθετούμε ένα πιόνι στο νταμάκι της πρώτης γραμμής και της πρώτης στήλης παίρνοντας έτσι τους πρώτους του βαθμούς. Το πιόνι μπορεί να κινηθεί μόνο πάνω (Π), κάτω (Κ), δεξιά (Δ), αριστερά (Α) εφόσον δεν βγαίνει εκτός σκακιέρας. Να γραφεί πρόγραμμα το οποίο:α. Διαβάζει τους βαθμούς για κάθε νταμάκι ελέγχοντας ώστε κάθε βαθμός να βρίσκεται εντός ορίων,β.

Διαβάζει τις διαδοχικές κινήσεις που κάνει το πιόνι, γ. Εμφανίζει το σύνολο των κινήσεων (μετάβαση σε νέο νταμάκι) που έκανε το πιόνι μέχρι να ξεπεράσει τους 100 βαθμούς καθώς και πόσους βαθμούς τελικά συγκέντρωσε.

### **Λύση**

**ΠΡΟΓΡΑΜΜΑ** ΣΚΑΚΙΕΡΑ

ΜΕΤΑΒΛΗΤΕΣ

**ΑΚΕΡΑΙΕΣ:** I, J COUNT, POSX, POSY

**ΠΡΑΓΜΑΤΙΚΕΣ:** Π[8,8], SUM

**ΧΑΡΑΚΤΗΡΕΣ :** MOVE

ΑΡΧΗ

**Για I ΑΠΟ 1 ΜΕΧΡΙ 8**

**Για J από 1 ΜΕΧΡΙ 8**

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

**ΔΙΑΒΑΣΕ** Π[I,J]

**ΜΕΧΡΙΣ\_ ΟΤΟΥ** (Π[I,J]>=1) ΚΑΙ (Π [I,J]<=10)

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

COUNT←0

SUM←Π[1,1]

POSX←-1

POSY←-1 ! απόλυτη θέση πιονιού

**ΟΣΟ** SUM <=100 **ΕΠΑΝΑΛΑΒΕ**

X← 0

Y←-0 ! Σχετική μετατόπιση πιονιού

**ΔΙΑΒΑΣΕ** MOVE

**Αν** (MOVE= 'Π' ) ΚΑΙ (POSX >1) **ΤΟΤΕ**

X← -1

**ΑΛΛΙΩΣ\_Αν** (MOVE= 'Κ') ΚΑΙ (POSX<8) **ΤΟΤΕ**

X←-1

**ΑΛΛΙΩΣ\_Αν** (MOVE='Δ') ΚΑΙ (POSY<8) **ΤΟΤΕ**

Y←-1

**ΑΛΛΙΩΣ\_Αν** (MOVE= 'Α') ΚΑΙ (POSY>1) **ΤΟΤΕ**

Y← -1

**ΤΕΛΟΣ\_ΑΝ**

**Αν** X<>Y **ΤΟΤΕ**

POSX←POSX +X

POSY←POSY +Y

SUM←SUM + Π [POSX,POSY]

COUNT←COUNT +1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** SUM, COUNT

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**14.** Στο πρωτάθλημα μπάσκετ της Α1 κατηγορίας ανδρών, μια ομάδα συμμετέχει με 10 παίκτες δίνοντας συνολικά 30 αγώνες (15 αγώνες στα πλαίσια του α' γύρου του πρωταθλήματος και αντίστοιχα 15 αγώνες στα πλαίσια του β' γύρου). Γράψτε ένα



πρόγραμμα που:

1. καταχωρεί σ' έναν πίνακα ΠΟΝΤΟΙ[10,30] τους πόντους που πέτυχε ο κάθε παίκτης σε κάθε αγώνα του πρωταθλήματος με τους παρακάτω κανόνες:

α. όταν ένας παίκτης δε συμμετείχε σ' έναν αγώνα τότε καταχωρούμε την τιμή -1 στον πίνακα.

β. Αν ο χρήστης πληκτρολογήσει τιμή πάνω από 30 να ρωτάει για επιβεβαίωση αν η τιμή είναι σωστή οπότε και την καταχωρεί, αλλιώς να του ζητάει να την ξανακαταχωρήσει.

2. καταχωρεί σ' έναν πίνακα ΠΑΙΚΤΕΣ[10] τα ονοματεπώνυμα των παικτών της ομάδας.

3. δημιουργεί έναν πίνακα ΑΠΟΤ[10,2] ο οποίος θα περιέχει στην πρώτη του στήλη το σύνολο των πόντων που πέτυχε ο κάθε παίκτης στον α' γύρο του πρωταθλήματος, ενώ στη δεύτερη στήλη αντίστοιχα το σύνολο των πόντων στον β' γύρο του πρωταθλήματος.

4. εμφανίζει μια λίστα με τα ονόματα των παικτών που πέτυχαν περισσότερους πόντους στο β' γύρο του πρωταθλήματος από ότι στον α' γύρο.

5. εμφανίζει τα ονόματα των παικτών που δεν έχασαν κανέναν αγώνα στο πρωτάθλημα.

### Λύση

#### ΠΡΟΓΡΑΜΜΑ ΜΠΑΣΚΕΤ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:Ι,Ι, ΠΟΝΤΟΙ[10,30],ΑΠΟΤ[10,2],Π

ΧΑΡΑΚΤΗΡΕΣ: ΠΑΙΚΤΕΣ[10],ΑΠ

ΛΟΓΙΚΕΣ:F

ΑΡΧΗ

! Καταχώρηση στοιχείων πίνακα ΠΟΝΤΟΙ[10,30]

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10

    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 30

        ΓΡΑΨΕ 'ΔΩΣΕ ΤΟΥΣ ΠΟΝΤΟΥΣ ΤΟΥ ΠΑΙΚΤΗ', Ι, 'ΣΤΟ ΑΓΩΝΑ', J

        ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

        F←ΨΕΥΔΗΣ

        ΔΙΑΒΑΣΕ Π

            ΑΝ Π > 30 ΤΟΤΕ

                ΓΡΑΨΕ 'ΕΠΙΒΕΒΑΙΩΣΤΕ ΑΝ ΕΙΝΑΙ ΣΩΣΤΗ Η ΤΙΜΗ',

                Π,'ΠΛΗΚΤΡΟΛΟΓΗΣΤΕ ΝΑΙ ή ΟΧΙ'

                ΔΙΑΒΑΣΕ ΑΠ

                    ΑΝ ΑΠ='ΝΑΙ' ΤΟΤΕ

                        ΠΟΝΤΟΙ[Ι,Ι]←Π

                        ΑΛΛΙΩΣ

                        ΓΡΑΨΕ 'ΠΑΡΑΚΑΛΩ ΞΑΝΑΔΩΣΤΕ ΤΗΝ ΤΙΜΗ'

                        F←ΑΛΗΘΗΣ

                        ΤΕΛΟΣ\_ΑΝ

                ΑΛΛΙΩΣ

                ΠΟΝΤΟΙ[Ι,Ι]←Π

                ΤΕΛΟΣ\_ΑΝ

            ΜΕΧΡΙΣ\_ΟΤΟΥ F=ΨΕΥΔΗΣ

        ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

! Καταχώρηση ονομάτων στον πίνακα ΠΑΙΚΤΕΣ[10]

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10

    ΓΡΑΨΕ 'ΔΩΣΕ ΤΟ ΟΝΟΜΑ ΤΟΥ ΠΑΙΚΤΗ', Ι

    ΔΙΑΒΑΣΕ ΠΑΙΚΤΕΣ[Ι]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*! Δημιουργία του πίνακα ΑΠΟΤ[10,2]*

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10**

**ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 2**

ΑΠΟΤ[Ι, J] ← 0

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10**

**ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 30**

**ΑΝ ΠΟΝΤΟΙ[Ι, J] <> -1 ΤΟΤΕ**

**ΑΝ J <= 15 ΤΟΤΕ**

ΑΠΟΤ[Ι, 1] ← ΑΠΟΤ[Ι, 1] + ΠΟΝΤΟΙ[Ι, J]

ΑΛΛΙΩΣ

ΑΠΟΤ[Ι, 2] ← ΑΠΟΤ[Ι, 2] + ΠΟΝΤΟΙ[Ι, J]

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*! Δημιουργία λίστας παικτών που πέτυχαν περισσότερους πόντους το β' γύρο & από τον α' γύρο*

**ΓΡΑΨΕ 'ΛΙΣΤΑ ΠΑΙΚΤΩΝ ΜΕ ΠΕΡΙΣΣΟΤΕΡΟΥΣ ΠΟΝΤΟΥΣ ΣΤΟ Β ΓΥΡΟ'**

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10**

**ΑΝ ΑΠΟΤ[Ι, 2] > ΑΠΟΤ[Ι, 1] ΤΟΤΕ**

**ΓΡΑΨΕ ΠΑΙΚΤΕΣ[Ι]**

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

*! Εύρεση παικτών που δεν έχασαν αγώνα*

**ΓΡΑΨΕ 'ΟΙ ΠΑΡΑΚΑΤΩ ΠΑΙΚΤΕΣ ΔΕΝ ΕΧΑΣΑΝ ΑΓΩΝΑ'**

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10**

J ← 1

F ← ΨΕΥΔΗΣ

**ΟΣΟ F = ΨΕΥΔΗΣ ΚΑΙ J <= 30 ΕΠΑΝΑΛΑΒΕ**

**ΑΝ ΠΟΝΤΟΙ[Ι, J] = -1 ΤΟΤΕ**

F ← ΑΛΗΘΗΣ

ΑΛΛΙΩΣ

J ← J + 1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ F = ΨΕΥΔΗΣ ΤΟΤΕ**

**ΓΡΑΨΕ ΠΑΙΚΤΕΣ[Ι]**

ΤΕΛΟΣ\_ΑΝ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**15.** Ο σκάουτερ της Λίβερπουλ καταγράφει τα ονόματα 20 επιθετικών σε ένα πίνακα ΟΝ και τις επιδόσεις τους την περσινή χρονιά στον πίνακα ΕΠ. Ο πίνακας έχει 6 στήλες: Η πρώτη αντικατοπτρίζει τα γκολ που σημείωσε ο παίκτης, η δεύτερη τις ασίστ που έδωσε, η τρίτη τα πέναλτι που κέρδισε, η τέταρτη ένα δείκτη που υποδηλώνει την ταχύτητα τους,

η πέμπτη τα λάθη που έκανε και η έκτη το χρόνο συμμετοχής τους, ο οποίος πρέπει να είναι από 0 - 2700 λεπτά.

1. Να γεμίσετε τους παραπάνω πίνακες.

2. Να βρείτε το όνομα του παίκτη που σημείωσε τα περισσότερα γκολ και είχε τα λιγότερα λάθη. Αν δεν υπάρχει να εμφανίζεται κατάλληλο μήνυμα.

3. Να βρείτε τους παίκτες που έχουν τα περισσότερα γκολ σε σχέση με το χρόνο συμμετοχής τους.

4. Να εμφανίσετε τους τρεις καλύτερους παίκτες με βάση την αξιολόγηση από το σύστημα RANKEX, που υπολογίζεται ως εξής: RANKEX = (αρ. γκολ + ασίστ + πέναλτι - λάθη) / χρόνο.

### **Λύση**

#### **ΑΛΓΟΡΙΘΜΟΣ ΣΚΑΟΥΤΕΡ**

**ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20**

**ΔΙΑΒΑΣΕ ΟΝ[i]**

**ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 5**

**ΔΙΑΒΑΣΕ ΕΠ[i, j]**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ ΕΠ[i, 6]**

**ΜΕΧΡΙΣ\_ΟΤΟΥ ((ΕΠ[i, 6] > 0) ΚΑΙ (ΕΠ[i, 6] <= 2700))**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**max ← ΕΠ[1,1]**

**min ← ΕΠ[1,5]**

**ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 20**

**ΑΝ ΕΠ[i,1] > max ΤΟΤΕ**

**max ← ΕΠ[i,1]**

**ΤΕΛΟΣ\_ΑΝ**

**ΑΝ ΕΠ[i,5] < min ΤΟΤΕ**

**min ← ΕΠ[i,5]**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ**

**ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20**

**ΑΝ max = ΕΠ[i, 1] ΚΑΙ min = ΕΠ[i, 5] ΤΟΤΕ**

**ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ**

**ΕΜΦΑΝΙΣΕ ΟΝ[i]**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ ΤΟΤΕ**

**ΕΜΦΑΝΙΣΕ "Δεν υπάρχει τέτοιος παίκτης"**

**ΤΕΛΟΣ\_ΑΝ**

**ΜΕΓ ← ΕΠ [1, 1] / ΕΠ [1, 6]**

**ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 20**

**Λ ← ΕΠ [i, 1] / ΕΠ [i, 6]**

**ΑΝ Λ > ΜΕΓ ΤΟΤΕ**

**ΜΕΓ ← Λ**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20

ΑΝ ΕΠ [i, 1] / ΕΠ [i, 6] = ΜΕΓ ΤΟΤΕ

ΕΜΦΑΝΙΣΕ ΟΝ[i]

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20

PANK[i] ← (ΕΠ[i, 1] + ΕΠ[i, 2] + ΕΠ[i, 3] - ΕΠ[i, 5]) / ΕΠ[i,6]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 20

ΓΙΑ j ΑΠΟ 20 ΜΕΧΡΙ i ΜΕ\_ΒΗΜΑ -1

ΑΝ PANK[j - 1] < PANK[j] ΤΟΤΕ

temp1 ← PANK[j-1]

PANK[j-1] ← PANK[j]

PANK[j] ← temp1

temp2 ← ΟΝ[j-1]

ΟΝ[j-1] ← ΟΝ[j]

ΟΝ[j] ← temp2

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 3

ΕΜΦΑΝΙΣΕ ΟΝ[i]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ ΣΚΑΟΥΤΕΡ

16. Ένα ταξιδιωτικό γραφείο ζήτησε προσφορές από 5 αεροπορικές εταιρίες για μαζική αγορά εισιτηρίων για το δρομολόγιο Θεσσαλονίκη - Βαρκελώνη. Όλες οι αεροπορικές εταιρίες έστειλαν την προσφορά τους, με κλιμακωτές χρεώσεις ανάλογα με τον αριθμό των επιβατών. Η 1η κλίμακα είναι από 1-20 επιβάτες, η δεύτερη από 21-50 και η τελευταία είναι άνω των 50 επιβατών. Ένα παράδειγμα της προσφοράς (με τυχαίους αριθμούς) φαίνεται πιο κάτω:

	1 - 20 επιβάτες	21 - 50 επιβάτες	άνω των 50 επιβατών
1 Airlines	210€	190€	180€
2 Airlines	220€	175€	140€
3 Airlines	200€	185€	180€
4 Airlines	199€	189€	179€
5 Airlines	205€	186€	175€

Να γίνει αλγόριθμος που με δεδομένο τον πίνακα ΧΡΕΩΣΕΙΣ[5,3] που περιέχει τις κλιμακωτές χρεώσεις κάθε αεροπορικής εταιρίας και τον πίνακα ΟΝ με τα ονόματα των αεροπορικών εταιριών, να διαβάξει τον αριθμό των επιβατών που το ταξιδιωτικό γραφείο διαθέτει και να τυπώνει το όνομα της αεροπορικής με την πιο συμφέρουσα (πιο φθηνή) προσφορά. Υποθέστε, πως υπάρχει μόνο μία προσφορά η οποία είναι η φθηνότερη.

Γνωρίζουμε πως έχουμε 5 προσφορές, κάθε μία από τις οποίες βρίσκεται σε μία γραμμή του πίνακα χρεώσεις. Άρα, επαναληπτικά θα υπολογίζουμε κλιμακωτά την χρέωση για

κάθε γραμμή, και στη συνέχεια θα βρίσκουμε την ελάχιστη τιμή. Υποθέστε πως υπάρχει μόνο μία προσφορά η οποία είναι η φθηνότερη.

### Λύση

#### Αλγόριθμος Πτήσεις

Δεδομένα //ΧΡΕΩΣΕΙΣ, ΟΝ//

Διάβασε αρ\_επιβατών

Για i από 1 μέχρι 5

Αν αρ\_επιβατών <= 20 τότε

    χρε\_εταιρείας ← ΧΡΕΩΣΕΙΣ[i, 1] \* αρ\_επιβατών

αλλιώς\_Αν αρ\_επιβατών <= 50 τότε

    χρε\_εταιρείας ← ΧΡΕΩΣΕΙΣ[i, 1] \* 20 + ΧΡΕΩΣΕΙΣ[i, 2] \* (αρ\_επιβατών - 20)

αλλιώς

    χρε\_εταιρείας ← ΧΡΕΩΣΕΙΣ[i, 1] \* 20 + ΧΡΕΩΣΕΙΣ[i, 2] \* 30 + ΧΡΕΩΣΕΙΣ[i, 3] \*  
    (αρ\_επιβατών - 50)

Τέλος\_Αν

ΤΕΛΙΚΗ\_ΧΡΕΩΣΗ[i] ← χρε\_εταιρείας

Τέλος\_επανάληψης

min ← 1

Για i από 2 μέχρι 5

Αν ΤΕΛΙΚΗ\_ΧΡΕΩΣΗ[i] < ΤΕΛΙΚΗ\_ΧΡΕΩΣΗ[min] τότε

    min ← i

Τέλος\_Αν

Τέλος\_επανάληψης

Εμφάνισε ΟΝ[min]

Τέλος Πτήσεις

**17.** Για την 5ήμερη εκδρομή της Γ Λυκείου ενός σχολείου, οι 60 μαθητές της, αποφάσισαν να διεξάγουν μια λαχειοφόρο αγορά, πουλώντας λαχνούς. Κάθε μαθητής πούλησε από 30 λαχνούς. Σε έναν πίνακα ΑΡΙΘΜΟΙ[60,30] καταγράφηκαν οι αριθμοί από τους λαχνούς των 60 μαθητών και σε ένα πίνακα ΟΝΟΜΑ[60] τα ονόματα των 60 μαθητών. Να γίνει αλγόριθμος, που με δεδομένο τον πίνακα ΑΡΙΘΜΟΙ και τον πίνακα ΟΝΟΜΑ, θα διαβάζει τον τυχερό αριθμό που κληρώθηκε και θα εμφανίζει ποιος μαθητής πούλησε τον τυχερό λαχνό.

### Λύση

#### Αλγόριθμος Λαχειοφόρος\_αγορά

Δεδομένα //ΑΡΙΘΜΟΙ, ΟΝΟΜΑ//

Διάβασε αριθμός\_κλήρωσης

! Θα αναζητήσουμε στον πίνακα ΑΡΙΘΜΟΙ. Ο πίνακας ωστόσο είναι διδιάστατος συνεπώς θα εφαρμόσουμε τη γνωστή μας σειριακή αναζήτηση, για κάθε & γραμμή και στήλη του πίνακα ΑΡΙΘΜΟΙ

βρέθηκε ← ΨΕΥΔΗΣ

i ← 1

Όσο  $i \leq 60$  ΚΑΙ βρέθηκε = ΨΕΥΔΗΣ επανάλαβε

$j \leftarrow 1$

Όσο  $j \leq 30$  ΚΑΙ βρέθηκε = ΨΕΥΔΗΣ επανάλαβε

Αν  $\text{ΑΡΙΘΜΟΙ}[i,j] = \text{αριθμός\_κλήρωσης}$  τότε

βρέθηκε  $\leftarrow$  ΑΛΗΘΗΣ

Εμφάνισε "Ο Αριθμός που κληρώθηκε ανήκει σε λαχνό που πουλήθηκε από  
&τον μαθητή",  $\text{ΟΝΟΜΑ}[i]$

Τέλος\_Αν

$j \leftarrow j + 1$

Τέλος\_επανάληψης

$i \leftarrow i + 1$

Τέλος\_επανάληψης

Αν βρέθηκε = ΨΕΥΔΗΣ τότε

Εμφάνισε "Ο αριθμός που κληρώθηκε δεν ανήκει σε κάποιο τυχερό, Τζακποτ!"

Τέλος\_Αν

Τέλος Λαχειοφόρος\_αγορά

**18.** Ας υποθέσουμε ότι μας δίνεται ένας πίνακας δύο διαστάσεων 3 γραμμών και 4 στηλών. Ζητούμενο είναι η δημιουργία αλγορίθμου που θα ταξινομή τα στοιχεία του, σε αύξουσα σειρά ώστε το μικρότερο να βρίσκεται στην θέση 1,1 και το μεγαλύτερο στην θέση 3,4. Δηλαδή τα στοιχεία να διατάσσονται γραμμή-προς-γραμμή από το μικρότερο στο μεγαλύτερο.

**Λύση**

Δημιουργία ενός μονοδιάστατου πίνακα με όλα τα στοιχεία του δισδιάστατου, ταξινόμηση του μονοδιάστατου και στη συνέχεια αντιγραφή των στοιχείων από το μονοδιάστατο στο δισδιάστατο.

**Αλγόριθμος** ταξινόμηση\_διδιάστατου

Δεδομένα // A //

! αντιγραφή των στοιχείων σε πίνακα μιας διάστασης (B)

$k \leftarrow 1$

Για  $i$  από 1 μέχρι 3

Για  $j$  από 1 μέχρι 4

$B[k] \leftarrow A[i,j]$

$k \leftarrow k + 1$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

! ταξινόμηση του B

Για  $i$  από 2 μέχρι 12

Για  $j$  από 12 μέχρι  $i$  με\_βήμα -1

Αν  $B[j] < B[j-1]$  τότε

Αντιμετάθεσε  $B[j], B[j-1]$

Τέλος\_Αν

Τέλος\_επανάληψης

Τέλος\_επανάληψης

! επανατοποθέτηση των στοιχείων στον πίνακα A

$k \leftarrow 1$

Για  $i$  από 1 μέχρι 3

Για  $j$  από 1 μέχρι 4

$A[i,j] \leftarrow B[k]$

$k \leftarrow k + 1$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Τέλος ταξινόμηση\_διδιάστατου

**19.** Δίνεται δισδιάστατος πίνακας  $A$  με  $\mu$  γραμμές και  $\nu$  στήλες. Να αντιγραφούν τα στοιχεία του γραμμή προς γραμμή σε ένα μονοδιάστατο πίνακα  $B$  (εννοείται  $\mu \cdot \nu$  στοιχείων).

**Λύση**

$k \leftarrow 1$

Για  $i$  από 1 μέχρι  $\mu$

Για  $j$  από 1 μέχρι  $\nu$

$B[k] \leftarrow A[i, j]$

$k \leftarrow k+1$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

**20.** Να γραφεί αλγόριθμος σε "ΓΛΩΣΣΑ" ο οποίος:

A. Να διαβάξει το πλήθος των στοιχείων ενός μονοδιάστατου πίνακα με όνομα  $A$ . Το πλήθος δεν θα πρέπει να υπολείπεται του 2 αλλά ούτε και να υπερβαίνει το 100.

B. Να δέχεται από το πληκτρολόγιο όλα τα στοιχεία του πίνακα, καθένα από τα οποία πρέπει να είναι θετικοί ακέραιοι αριθμοί.

Γ. Να δημιουργεί ένα μονοδιάστατο πίνακα  $B$  κάθε στοιχείο του οποίου πρέπει να είναι το γινόμενο των γειτονικών στοιχείων του  $A$ . Για παράδειγμα, το στοιχείο  $B[1]$  πρέπει να είναι το  $A[1] * A[2]$ , το  $B[2]$  είναι το  $A[2] * A[3]$ , κ.ο.κ. Για τον υπολογισμό του γινομένου να εφαρμοστεί η μέθοδος του πολλαπλασιασμού αλά ρωσικά.

Δ. Να εμφανίζει (με κατάλληλα διαμορφωμένο μήνυμα) κάθε ζευγάρι στοιχείων του πίνακα  $A$  καθώς και το αντίστοιχο γινόμενο του.

Ε. Να υπολογίζει το μεγαλύτερο γινόμενο και να εμφανίζει (με κατάλληλα διαμορφωμένο μήνυμα) ποιο ζευγάρι από τα στοιχεία του  $A$  το είχε, καθώς ποιοι ήταν οι δείκτες τους στον πίνακα.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ** Θέμα1

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $A[100]$

**ΑΚΕΡΑΙΕΣ:**  $B[99]$ ,  $\Gamma[99,2]$ ,  $I$ ,  $MAX$ ,  $\Theta E \Sigma H$ ,  $N$ ,  $\Gamma I N$ ,  $K$ ,  $M1$ ,  $M2$ ,  $L$

**ΑΡΧΗ**

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Δώστε το πλήθος των στοιχείων του πίνακα  $A$ . Να είναι  $\geq 2$  και  $\leq 100$ '

**ΔΙΑΒΑΣΕ**  $N$

**ΜΕΧΡΙΣ\_ΟΤΟΥ** ( $N \geq 2$ ) ΚΑΙ ( $N \leq 100$ )

```

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
ΓΡΑΨΕ 'Δώστε στοιχεία πίνακα Α'
      ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
      ΔΙΑΒΑΣΕ Α[Ι]
      ΜΕΧΡΙΣ_ΟΤΟΥ (Α[Ι] > 0) ΚΑΙ (Α_Μ(Α[Ι])) = Α[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΝ ← 0
Κ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν -1
Μ1 ← Α[Ι]
Μ2 ← Α[Ι+1]
      ΟΣΟ Μ2 > 0 ΕΠΑΝΑΛΑΒΕ
        ΑΝ Μ2 MOD 2 = 1 ΤΟΤΕ
          ΓΙΝ ← ΓΙΝ + Μ1
          ΤΕΛΟΣ_ΑΝ
        Μ1 ← 2 * Μ1
        Μ2 ← Μ2 DIV 2
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Κ ← Κ + 1
Β[Κ] ← ΓΙΝ
Γ[Κ, 1] ← Ι
Γ[Κ, 2] ← Ι + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Λ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν -1
Λ ← Λ + 1
ΓΡΑΨΕ 'Στοιχεία Α[', Α[Ι], ', ', Α[Ι+1], Α[Ι+1], ', ' Γινόμενο:', Β[Λ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ ← Β[1]
ΘΕΣΗ ← 1
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Κ
      ΑΝ Β[Ι] > ΜΑΧ ΤΟΤΕ
        ΜΑΧ ← Β[Ι]
        ΘΕΣΗ ← Ι
      ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Στοιχεία με το μεγαλύτερο γινόμενο: Α[', Γ[ΘΕΣΗ, 1], ', ' Α[', Γ[ΘΕΣΗ, 2], ',']'
ΓΡΑΨΕ 'Με δείκτες:', Γ[ΘΕΣΗ, 1], Γ[ΘΕΣΗ, 2]
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

**21.** Να γραφεί αλγόριθμος σε "ΓΛΩΣΣΑ" ο οποίος:

Α. Να διαβάσει (μετά την εμφάνιση κατάλληλου ενημερωτικού μηνύματος) τις επωνυμίες 100 επιχειρήσεων και να τις καταχωρίζει στον πίνακα ΕΠΩΝΥΜΙΑ.

Β. Να διαβάσει τις μηνιαίες εισπράξεις (μετά την εμφάνιση κατάλληλου ενημερωτικού μηνύματος) που πραγματοποίησε καθεμία από τις επιχειρήσεις κατά τη διάρκεια του προηγούμενου έτους και να τις καταχωρίζει στον πίνακα ΕΙΣΠΡΑΞΕΙΣ.

Γ. Να ταξινομή κατά φθίνουσα σειρά τις εισπράξεις κάθε επιχείρησης και να εμφανίζει τη μέγιστη μηνιαία εισπράξη κάθε επιχείρησης.



Δ. Να βρίσκει και να εμφανίζει (με κατάλληλα διαμορφωμένο μήνυμα) την επωνυμία (ή τις επωνυμίες) της επιχείρησης της οποίας η μέγιστη μηνιαία εισπραξη ξεπέρασε το ποσό των 50.000 ευρώ.

Παρατήρηση: Κάθε εισπραξη δεν πρέπει να υπολείπεται των 25.000 ευρώ, αλλά ούτε και να ξεπερνά τα 250.000 ευρώ. Επίσης, θεωρήστε ότι όλες οι εισπράξεις είναι μεταξύ τους διαφορετικές.

### **Λύση**

**ΠΡΟΓΡΑΜΜΑ** Θέμα2

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** ΕΙΣΠΡΑΞΕΙΣ[100,12], TEMP

**ΑΚΕΡΑΙΕΣ:** I, J, K

**ΧΑΡΑΚΤΗΡΕΣ:** ΕΠΩΝΥΜΙΑ[100]

**ΑΡΧΗ**

**Για I ΑΠΟ 1 ΜΕΧΡΙ 75**

**ΓΡΑΨΕ** 'Δώστε επωνυμία ', I, 'ης επιχείρησης:'

**ΔΙΑΒΑΣΕ** ΕΠΩΝΥΜΙΑ[I]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Για I ΑΠΟ 1 ΜΕΧΡΙ 100**

**Για J ΑΠΟ 1 ΜΕΧΡΙ 12**

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Δώστε εισπραξη ', I, 'ης επιχείρησης **Για** το ', J, 'ο μήνα:'

**ΔΙΑΒΑΣΕ** ΕΙΣΠΡΑΞΕΙΣ[I,J]

**ΜΕΧΡΙΣ\_ΟΤΟΥ** (ΕΙΣΠΡΑΞΕΙΣ[I,J] >= 25000) **ΚΑΙ** (ΕΙΣΠΡΑΞΕΙΣ[I,J] <= 250000)

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Για K ΑΠΟ 1 ΜΕΧΡΙ 100**

**Για I ΑΠΟ 2 ΜΕΧΡΙ 12**

**Για J ΑΠΟ 12 ΜΕΧΡΙ I ΜΕ\_ΒΗΜΑ -1**

**Αν** ΕΙΣΠΡΑΞΕΙΣ[K,J-1] < ΕΙΣΠΡΑΞΕΙΣ[K,J] **ΤΟΤΕ**

TEMP ← ΕΙΣΠΡΑΞΕΙΣ[K, J-1]

ΕΙΣΠΡΑΞΕΙΣ[K,J-1] ← ΕΙΣΠΡΑΞΕΙΣ[K,J]

ΕΙΣΠΡΑΞΕΙΣ[K,J] ← TEMP

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Η μέγιστη εισπραξη της επιχείρησης ', ΕΠΩΝΥΜΙΑ[I], ' είναι: ', ΕΙΣΠΡΑΞΕΙΣ[I,1]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Για I ΑΠΟ 1 ΜΕΧΡΙ 100**

**Αν** ΕΙΣΠΡΑΞΕΙΣ[I,1] > 50000 **ΤΟΤΕ**

**ΓΡΑΨΕ** 'Η μέγιστη εισπραξη της επιχείρησης ', ΕΠΩΝΥΜΙΑ[I], ' ξεπέρασε τα 50000 ευρώ'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**22.** Μια αλυσίδα κινηματογράφων έχει δέκα αίθουσες. Τα ονόματα των αιθουσών καταχωρούνται σε ένα μονοδιάστατο πίνακα και οι μηνιαίες εισπράξεις κάθε αίθουσας για ένα έτος καταχωρούνται σε πίνακα δύο διαστάσεων. Να γράψετε αλγόριθμο ο οποίος:

α. να διαβάζει τα ονόματα των αιθουσών

β. να διαβάζει τις μηνιαίες εισπράξεις των αιθουσών αυτού του έτους

γ. να υπολογίζει τη μέση μηνιαία τιμή των εισπράξεων για κάθε αίθουσα

δ. να βρίσκει και να εμφανίζει τη μικρότερη μέση μηνιαία τιμή

ε. να βρίσκει και να εμφανίζει το όνομα ή τα ονόματα των αιθουσών που έχουν την ανωτέρω μικρότερη μέση μηνιαία τιμή. (Θέμα Πανελλαδικών)

### Λύση

**Αλγόριθμος** Θέμα\_4

**Για** i από 1 μέχρι 10

**Διάβασε** A[i]

**Τέλος\_επανάληψης**

**Για** i από 1 μέχρι 10

**Για** j από 1 μέχρι 12

**Διάβασε** E[i, j]

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Για** i από 1 μέχρι 10

$\Sigma \leftarrow 0$

**Για** j από 1 μέχρι 12

$\Sigma \leftarrow \Sigma + E[i, j]$

**Τέλος\_επανάληψης**

$MO[i] \leftarrow \Sigma / 12$

**Τέλος\_επανάληψης**

    min  $\leftarrow MO[1]$

**Για** i από 2 μέχρι 10

**Αν**  $MO[i] < min$  **τότε** min  $\leftarrow MO[i]$

**Τέλος\_επανάληψης**

**Εμφάνισε** min

**Για** i από 1 μέχρι 10

**Αν**  $MO[i] = min$  **τότε** **Εμφάνισε** A[i]

**Τέλος\_επανάληψης**

**Τέλος** Θέμα\_4

**23.** Για την πρώτη φάση της Ολυμπιάδας Πληροφορικής, δήλωσαν συμμετοχή 500 μαθητές. Οι μαθητές διαγωνίζονται σε τρεις γραπτές εξετάσεις και βαθμολογούνται με ακέραιους βαθμούς στη βαθμολογική κλίμακα από 0 έως και 100.

Να γράψετε αλγόριθμο ο οποίος:

α. Να διαβάζει τα ονόματα των μαθητών και να τα αποθηκεύει σε μονοδιάστατο πίνακα.

β. Να διαβάζει τους τρεις βαθμούς που έλαβε κάθε μαθητής και να τους αποθηκεύει σε δισδιάστατο πίνακα.

γ. Να υπολογίζει το μέσο όρο των βαθμών του κάθε μαθητή.

δ. Να εκτυπώνει τα ονόματα των μαθητών και δίπλα τους το μέσο όρο των βαθμών τους ταξινομημένα με βάση το μέσο όρο κατά φθίνουσα σειρά. Σε περίπτωση ισοβαθμίας η

σειρά ταξινόμησης των ονομάτων να είναι αλφαβητική.

ε. Να υπολογίζει και να εκτυπώνει το πλήθος των μαθητών με το μεγαλύτερο μέσο όρο.  
(Θέμα Πανελλαδικών)

### **Λύση**

**Αλγόριθμος** Θέμα4

**Για**  $i$  **από** 1 **μέχρι** 500

**Διάβασε**  $O[i]$

**Τέλος\_επανάληψης**

**Για**  $i$  **από** 1 **μέχρι** 500

**Για**  $j$  **από** 1 **μέχρι** 3

**Διάβασε**  $B[i, j]$

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Για**  $i$  **από** 1 **μέχρι** 500

$\Sigma \leftarrow 0$

**Για**  $j$  **από** 1 **μέχρι** 3

$\Sigma \leftarrow \Sigma + B[i, j]$

**Τέλος\_επανάληψης**

$MO[i] \leftarrow \Sigma / 3$

**Τέλος\_επανάληψης**

**Για**  $i$  **από** 2 **μέχρι** 500

**Για**  $j$  **από** 500 **μέχρι**  $i$  **με\_βήμα** -1

**Αν**  $MO[j - 1] < MO[j]$  **τότε**

**Αντιμετάθεσε**  $MO[j - 1], MO[j]$

**Αντιμετάθεσε**  $O[j - 1], O[j]$

**αλλιώς\_Αν**  $MO[j - 1] = MO[j]$  **τότε**

**Αν**  $O[j - 1] > O[j]$  **τότε**

**Αντιμετάθεσε**  $O[j - 1], O[j]$

**Τέλος\_Αν**

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Για**  $i$  **από** 1 **μέχρι** 500

**Εμφάνισε**  $O[i], MO[i]$

**Τέλος\_επανάληψης**

$πλ \leftarrow 0$

**Για**  $i$  **από** 1 **μέχρι** 500

**Αν**  $MO[i] = MO[1]$  **τότε**

$πλ \leftarrow πλ + 1$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Εμφάνισε**  $πλ$

**Τέλος** Θέμα4

**24.** Δίνεται πίνακας  $A[N]$  ακέραιων και θετικών αριθμών, καθώς και πίνακας  $B[N-1]$  πραγματικών και θετικών αριθμών. Να γραφεί αλγόριθμος, ο οποίος να ελέγχει αν κάθε στοιχείο  $B[i]$  είναι ο μέσος όρος των στοιχείων  $A[i]$  και  $A[i+1]$ , δηλαδή αν  $B[i] = (A[i] +$

$A[i+1])/2$ . Σε περίπτωση που ισχύει, τότε να εμφανίζεται το μήνυμα «Ο πίνακας Β είναι ο τρέχων μέσος του Α», διαφορετικά να εμφανίζεται το μήνυμα «Ο πίνακας Β δεν είναι ο τρέχων μέσος του Α». (Θέμα Πανελλαδικών)

### Λύση

**Αλγόριθμος** Θέμα\_3

**Δεδομένα** //A, B, N//

f ← **Αληθής**

Για **από 1 μέχρι** N-1

**Αν**  $B[i] <> (A[i] + A[i+1]) / 2$  **τότε**

        f ← **Ψευδής**

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Αν** f = **Αληθής** **τότε**

**Εμφάνισε** "Ο πίνακας Β είναι ο τρέχων μέσος του Α"

**αλλιώς**

**Εμφάνισε** "Ο πίνακας Β δεν είναι ο τρέχων μέσος του Α"

**Τέλος\_Αν**

**Τέλος** Θέμα\_3

**25.** Σε ένα Μετεωρολογικό Σταθμό καταγράφονται ανά ημέρα και ώρα η θερμοκρασία του περιβάλλοντος για μία εβδομάδα. Να γράψετε αλγόριθμο που:

α. Διαβάζει τα ονόματα των επτά ημερών της εβδομάδας και τα καταχωρεί σε μονοδιάστατο πίνακα και τη θερμοκρασία για κάθε ημέρα της εβδομάδας και κάθε ώρα της ημέρας και την καταχωρεί σε δισδιάστατο πίνακα, ελέγχοντας οι τιμές της θερμοκρασίας να είναι από -20 μέχρι και 50.

β. Υπολογίζει για κάθε ημέρα τη μέση θερμοκρασία και την καταχωρεί σε μονοδιάστατο πίνακα.

γ. Βρίσκει και εμφανίζει τη μέγιστη μέση θερμοκρασία της εβδομάδας από τον πίνακα των μέσων θερμοκρασιών.

δ. Βρίσκει και εμφανίζει την ημέρα της εβδομάδας με τη μέγιστη μέση θερμοκρασία (να θεωρήσετε ότι υπάρχει μόνο μία τέτοια ημέρα).

ε. Υπολογίζει και εμφανίζει το πλήθος των ημερών της εβδομάδας που είχαν μέση θερμοκρασία μεγαλύτερη των 20. (Θέμα Πανελλαδικών)

### Λύση

**ΑΛΓΟΡΙΘΜΟΣ** ΘΕΡΜΟΚΡΑΣΙΕΣ

Για i **από 1 μέχρι** 7

**Διάβασε** Ημ[i]

    Για j **από 1 μέχρι** 24

**Αρχή\_επανάληψης**

**Διάβασε** Θ[ i, j ]

**Μέχρις\_ότου**  $(\Theta[ i, j ] \geq -20)$  **και**  $(\Theta[ i, j ] \leq 50)$

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

Για i **από 1 μέχρι** 7

Σ ← 0

    Για j **από 1 μέχρι** 24

```

    Σ ← Σ + Θ[i,j]
    Τέλος_επανάληψης
ΜΟ[i] ← Σ / 24
Τέλος_επανάληψης
ΜΑΧ ← ΜΟ[1]
ΗΜΕΡΑ ← 1
Για i από 2 μέχρι 7
    Αν ΜΟ[i] > ΜΑΧ τότε
        ΜΑΧ ← ΜΟ[i]
        ΗΜΕΡΑ ← i
    Τέλος_Αν
Τέλος_επανάληψης
ΕΜΦΑΝΙΣΕ " Η μεγαλύτερη μέση θερμοκρασία της εβδομάδας είναι ", ΜΑΧ
ΕΜΦΑΝΙΣΕ " και σημειώθηκε την & ημέρα ", Ημ[ΗΜΕΡΑ]
Πλήθος ← 0
Για i από 1 μέχρι 7
    Αν ΜΟ[i] > 20 τότε
        Πλήθος ← Πλήθος + 1
    Τέλος_Αν
Τέλος_επανάληψης
ΕΜΦΑΝΙΣΕ " Οι ημέρες της εβδομάδας με μέση θερμοκρασία > 20°C είναι & ", Πλήθος
ΤΕΛΟΣ_Θερμοκρασίες

```

**26.** Σε ένα Δήμο υπάρχουν 4 σταθμοί μέτρησης ενός συγκεκριμένου ατμοσφαιρικού ρύπου. Η καταγραφή της τιμής του ρύπου γίνεται ανά ώρα και σε 24ωρη βάση. Οι αποδεκτές τιμές του ρύπου κυμαίνονται από 0 έως και 100. Να γραφεί αλγόριθμος, ο οποίος:

- για κάθε σταθμό και για κάθε ώρα του 24ώρου διαβάζει την τιμή του ρύπου και την καταχωρίζει σε πίνακα διαστάσεων 4x24, ελέγχοντας την εγκυρότητα κάθε τιμής.
- για κάθε ώρα του 24ώρου υπολογίζει και εμφανίζει τη μέση τιμή του ρύπου από τους 4 σταθμούς.
- για κάθε σταθμό βρίσκει και εμφανίζει τη μέγιστη και την ελάχιστη τιμή του ρύπου στο 24ωρο.
- βρίσκει και εμφανίζει τη μέγιστη τιμή του ρύπου στη διάρκεια του 24ώρου, καθώς και την ώρα και τον αριθμό του σταθμού που σημειώθηκε η τιμή αυτή. (Να θεωρήσετε ότι η τιμή αυτή είναι μοναδική στον πίνακα). (Θέμα Πανελλαδικών)

#### **Λύση**

```

Αλγόριθμος Μετρήσεις
Για i από 1 μέχρι 4
    Για j από 1 μέχρι 24
        Αρχή_επανάληψης
            Διάβασε Μ[i , j ]
            Μέχρις_ότου Μ[i , j ] >= 0 και Μ[i , j ] <= 100
        Τέλος_επανάληψης
    Τέλος_επανάληψης
Για j από 1 μέχρι 24

```

```

Σ ← 0
Για i από 1 μέχρι 4
    Σ ← Σ + M[i, j]
Τέλος_επανάληψης
μο ← Σ / 4
Εμφάνισε μο
Τέλος_επανάληψης
Για i από 1 μέχρι 4
    max ← M[i, 1]
    min ← M[i, 1]
    Για j από 1 μέχρι 24
        Αν M[i, j] > max τότε
            max ← M[i, j]
        Τέλος_Αν
        Αν M[i, j] < min τότε
            min ← M[i, j]
        Τέλος_Αν
    Τέλος_επανάληψης
    Εμφάνισε max, min
Τέλος_επανάληψης
max ← M[1, 1]
σταθμός ← 1
ώρα ← 1
Για i από 1 μέχρι 4
    Για j από 1 μέχρι 24
        Αν M[i, j] > max τότε
            max ← M[i, j]
            σταθμός ← i
            ώρα ← j
        Τέλος_Αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Εμφάνισε max, σταθμός, ώρα
Τέλος Μετρήσεις

```

**27.** Στους προκριματικούς αγώνες ιππικού τριάθλου συμμετέχουν 16 αθλητές. Τα αγωνίσματα είναι: ιππική δεξιοτεχνία, υπερπήδηση εμποδίων και ελεύθερη ιππασία. Ο κάθε αθλητής βαθμολογείται ξεχωριστά σε κάθε ένα από τα τρία αγωνίσματα.

Να σχεδιάσετε αλγόριθμο ο οποίος:

- καταχωρίζει σε πίνακα τις ονομασίες των τριών αγωνισμάτων, όπως αυτές δίνονται παραπάνω.
- διαβάζει για κάθε αθλητή όνομα, επίθετο, όνομα αλόγου με το οποίο αγωνίζεται και τους βαθμούς του σε κάθε αγώνισμα και θα καταχωρίζει τα στοιχεία σε πίνακες.
- διαβάζει το όνομα και το επίθετο ενός αθλητή και θα εμφανίζει το όνομα του αλόγου με το οποίο αγωνίστηκε και τη συνολική του βαθμολογία στα τρία αγωνίσματα. Αν δεν υπάρχει ο αθλητής, θα εμφανίζει κατάλληλα διαμορφωμένο μήνυμα.
- εμφανίζει την ονομασία του αγωνίσματος (ή των αγωνισμάτων) με το μεγαλύτερο «άνοιγμα βαθμολογίας». Ως «άνοιγμα βαθμολογίας» να θεωρήσετε τη διαφορά ανάμεσα

στην καλύτερη και στη χειρότερη βαθμολογία του αγωνίσματος. (Θέμα Πανελλαδικών)

### **Λύση**

**Αλγόριθμος** thema\_4

**Για** J από 1 μέχρι 3

**Διάβασε** ΑΘΛΗΜΑ[J]

**Τέλος\_επανάληψης**

**Για** I από 1 μέχρι 16

**Διάβασε** Ο[I], Ε[I], ΑΛ[I]

**Για** J από 1 μέχρι 3

**Διάβασε** Β[I, J]

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Διάβασε** όνομα, επίθετο

Done ← ΨΕΜΑ

I ← 1

Position ← 0

ΣΒ ← 0

**Όσο** I ≤ 16 **ΚΑΙ** done = ΨΕΜΑ **επανάλαβε**

**Αν** όνομα = Ο[I] **ΚΑΙ** Ε[I] = επίθετο **τότε**

**Για** J από 1 μέχρι 3

        ΣΒ ← ΣΒ + Β[I, J]

**Τέλος\_επανάληψης**

    Done ← ΑΛΗΘΕΙΑ

    Position ← I

**Αλλιώς**

    I ← I + 1

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Αν** done = ΑΛΗΘΕΙΑ **τότε**

**Εμφάνισε** ΑΛ[position], ΣΒ

**Αλλιώς**

**Εμφάνισε** 'ο αθλητής δεν υπάρχει'

**Τέλος\_Αν**

**Για** J από 1 μέχρι 3

    Max ← Β[1, J]

    min ← Β[1, J]

**Για** I από 2 μέχρι 16

**Αν** Β[I, J] > max **τότε**

    max ← Β[I, J]

**Τέλος\_Αν**

**Αν** Β[I, J] < min **τότε**

    min ← Β[I, J]

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

    D[J] ← max - min

**Τέλος\_επανάληψης**

    maxim ← D[1]

**Για** J από 2 μέχρι 3

**Αν** D[J] > maxím τότε  
 maxím ← D[J]  
**Τέλος\_Αν**  
**Τέλος\_επανάληψης**  
**Για** J από 1 μέχρι 3  
**Αν** D[J] = maxím τότε  
**Εμφάνισε** ΑΘΛΗΜΑ[J]  
**Τέλος\_Αν**  
**Τέλος\_επανάληψης**  
**Τέλος** thema\_4

**28.** Οι εκατό (100) υπάλληλοι μιας εταιρείας εργάζονται 40 ώρες την εβδομάδα. Κάθε ώρα υπερωρίας αμείβεται με 5 € (ευρώ). Να γράψετε αλγόριθμο ο οποίος:

A. για καθένα από τους υπαλλήλους της εταιρείας

α. διαβάζει το όνομά του και για κάθε μέρα από τις πέντε (5) εργάσιμες της εβδομάδας διαβάζει τις ώρες εργασίας του.

β. υπολογίζει τις εβδομαδιαίες ώρες εργασίας του.

γ. εάν έχει εργαστεί περισσότερο από 40 ώρες την εβδομάδα, εμφανίζει το όνομά του και υπολογίζει και εμφανίζει την αμοιβή του για τις υπερωρίες του.

B. υπολογίζει και εμφανίζει, το πλήθος των υπαλλήλων που έχουν εργαστεί λιγότερο από 40 ώρες την εβδομάδα. (Θέμα Πανελλαδικών)

**Λύση**

**Αλγόριθμος** thema\_3

K ← 0

**Για** I από 1 μέχρι 100

**Διάβασε** ON

    Sum ← 0

**Για** J από 1 μέχρι 5

**Διάβασε** Ω[I,J]

        Sum ← sum + Ω[I,J]

**Τέλος\_επανάληψης**

**Αν** sum > 40 τότε

        ΥΠΕΡ ← (sum - 40)\*5

**Εμφάνισε** ON, ΥΠΕΡ

    Αλλιώς

        K ← K + 1

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Εμφάνισε** K, 'υπάλληλοι με λιγότερες από 40 ώρες'

**Τέλος** thema\_3

**29.** Στην αρχή της ποδοσφαιρικής περιόδου οι 22 παίκτες μιας ομάδας, οι οποίοι αριθμούνται από 1 έως 22, ψηφίζουν για τον αρχηγό που θα τους εκπροσωπεί. Κάθε παίκτης μπορεί να ψηφίσει όσους συμπαίκτες του θέλει, ακόμα και τον εαυτό του. Τα αποτελέσματα της ψηφοφορίας καταχωρίζονται σε έναν πίνακα ΨΗΦΟΣ με 22 γραμμές και 22 στήλες, έτσι ώστε το στοιχείο ΨΗΦΟΣ[i,j] να έχει την τιμή 1, όταν ο παίκτης με



- αριθμό  $i$  έχει ψηφίσει τον παίκτη με αριθμό  $j$ , και τιμή 0 στην αντίθετη περίπτωση.  
 Να γράψετε αλγόριθμο ο οποίος:  
 Δ1. Να διαβάζει τα στοιχεία του πίνακα ΨΗΦΟΣ και να ελέγχει την ορθότητά τους, με αποδεκτές τιμές 0 ή 1.  
 Δ2. Να εμφανίζει για κάθε παίκτη το πλήθος των ψήφων που έδωσε.  
 Δ3. Να εμφανίζει για κάθε παίκτη το πλήθος των ψήφων που έλαβε.  
 Δ4. Να εμφανίζει τον αριθμό του παίκτη που έλαβε τις περισσότερες ψήφους. Θεωρήστε ότι είναι μοναδικός.  
 Δ5. Να εμφανίζει τον αριθμό κάθε παίκτη που δεν ψηφίσει τον εαυτό του. (Θέμα Πανελλαδικών)

### Λύση

**ΑΛΓΟΡΙΘΜΟΣ ΘΕΜΑ\_4**

**ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 22**

**ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 22**

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ ΨΗΦΟΣ [I, J]**

**ΜΕΧΡΙΣ\_ΟΤΟΥ ((ΨΗΦΟΣ [I, J] = 0) Η (ΨΗΦΟΣ [I, J] = 1))**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Για I από 1 μέχρι 22**

**ΚΕ[I]←0**

**Για J από 1 μέχρι 22**

**ΚΕ[I]←ΚΕ[I] + ΨΗΦΟΣ[I,J]**

**Τέλος\_επανάληψης**

**Εμφάνισε “ ψήφοι που έδωσε”, ΚΕ[I]**

**Τέλος\_επανάληψης**

**Για J από 1 μέχρι 22**

**ΚΕΛ[J]←0**

**Για I από 1 μέχρι 22**

**ΚΕΛ[J]←ΚΕΛ[J]+ ΨΗΦΟΣ[I,J]**

**Τέλος\_επανάληψης**

**Εμφάνισε “ψήφοι που έλαβε”, ΚΕΛ[J]**

**Τέλος\_επανάληψης**

**Pos←-1, max←ΚΕΛ[J]**

**Για J από 2 μέχρι 22**

**Αν ΚΕΛ[J] > max τότε**

**max←ΚΕΛ[J]**

**Pos←J**

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Εμφάνισε pos**

<p><b>Για I από 1 μέχρι 22</b>  <b>Για J από 1 μέχρι 22</b>  <b>Αν I = J τότε</b>  <b>Αν ΨΗΦΟΣ[I,J] = 0 τότε</b>  <b>Εμφάνισε I</b>  <b>Τέλος_Αν</b></p>	<p><b>Β' τρόπος</b>   <b>Για I από 1 μέχρι 22</b>  <b>Αν ΨΗΦΟΣ[I,I] = 0 τότε</b>  <b>Εμφάνισε I</b>  <b>Τέλος_Αν</b></p>
--	--

Τέλος_Αν Τέλος_επανάληψης	Τέλος_επανάληψης
------------------------------	------------------

Τέλος\_επανάληψης

Τέλος Θέμα\_4

**30.** Μια εταιρεία ασχολείται με εγκαταστάσεις φωτοβολταϊκών συστημάτων, με τα οποία οι πελάτες της έχουν τη δυνατότητα αφενός να παράγουν ηλεκτρική ενέργεια για να καλύπτουν τις ανάγκες της οικίας τους, αφετέρου να πωλούν την πλεονάζουσα ενέργεια προς 0,55€/kWh, εξασφαλίζοντας επιπλέον έσοδα. Η εταιρεία αποφάσισε να ερευνήσει τις εγκαταστάσεις που πραγματοποίησε την προηγούμενη χρονιά σε δέκα (10) πελάτες. Να αναπτύξετε αλγόριθμο ο οποίος:

Δ1. Να διαβάζει τα ονόματα των πελατών και να τα αποθηκεύει σε πίνακα ΟΝΟΜΑ[10].

Δ2. Να διαβάζει το ποσό της ηλεκτρικής ενέργειας σε kWh που παρήγαγαν τα φωτοβολταϊκά συστήματα κάθε πελάτη, καθώς και το ποσό της ηλεκτρικής ενέργειας που κατανάλωσε κάθε πελάτης ανά μήνα του έτους, και να τα αποθηκεύει στους πίνακες Π[10,12] για την παραγωγή και Κ[10,12] για την κατανάλωση αντίστοιχα. Θεωρήστε ότι δεν απαιτείται έλεγχος εγκυρότητας για τα δεδομένα εισόδου.

Δ3. Με βάση τα στοιχεία του διδιάστατου πίνακα Π[10,12], να αποθηκεύει σε μονοδιάστατο πίνακα ΕΤΗΣΙΑ\_Π[10] τις ετήσιες αποδόσεις σε kWh για κάθε πελάτη. Με βάση τα στοιχεία του διδιάστατου πίνακα Κ[10,12], να αποθηκεύει σε μονοδιάστατο πίνακα ΕΤΗΣΙΑ\_Κ[10] τις ετήσιες καταναλώσεις σε kWh που αντιστοιχούν σε κάθε πελάτη.

Δ4. Σε μονοδιάστατο πίνακα ΕΣΟΔΑ[10] να αποθηκεύει τα ετήσια έσοδα σε ευρώ αν η ετήσια παραγόμενη ηλεκτρική ενέργεια είναι μεγαλύτερη από την ενέργεια που έχει καταναλωθεί για κάθε πελάτη, αλλιώς να αποθηκεύει την τιμή 0.

Δ5. Να εμφανίζει τα ετήσια έσοδα σε ευρώ κατά φθίνουσα σειρά.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ** ΘέμαΔ

! Ερώτημα Δ1α

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** Π[10, 12], Κ[10, 12], ΣΠ, ΣΚ, Εσοδα[10], Max, Min

**ΑΚΕΡΑΙΕΣ:** I, J, Posmin

**ΧΑΡΑΚΤΗΡΕΣ:** ON[10, 2]

ΑΡΧΗ

! Ερωτήματα Δ1β, Δ1γ

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10

    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 2

        ΔΙΑΒΑΣΕ ON[I, J]

        ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10

    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12

        ΔΙΑΒΑΣΕ Π[I, J], Κ[I, J]

        ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

! Ερώτημα Δ2

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10

$\Sigma\text{Π} \leftarrow 0$

$\Sigma\text{Κ} \leftarrow 0$

**ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12**

$\Sigma\text{Π} \leftarrow \Sigma\text{Π} + \text{Π}[I, J]$

$\Sigma\text{Κ} \leftarrow \Sigma\text{Κ} + \text{Π}[I, J]$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

$\text{Εσοδα}[I] \leftarrow (\Sigma\text{Π} - \Sigma\text{Κ}) * 0.55$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*! Ερώτημα Δ3*

$\text{Max} \leftarrow \text{Εσοδα}[1]$

**ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10**

**ΑΝ**  $\text{Εσοδα}[I] > \text{Max}$  **ΤΟΤΕ**

$\text{Max} \leftarrow \text{Εσοδα}[I]$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10**

**ΑΝ**  $\text{Εσοδα}[I] = \text{Max}$  **ΤΟΤΕ**

**ΓΡΑΨΕ**  $\text{ΟΝ}[I, 2]$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*! Ερώτημα Δ4 - Κλήση υποπρογράμματος*

**ΚΑΛΕΣΕ** Εμφάνιση (Εσοδα)

*! Ερώτημα Δ5*

$\text{Min} \leftarrow \text{Π}[1, 1]$

$\text{Posmin} \leftarrow 1$

**ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10**

**ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 12**

**ΑΝ**  $\text{Π}[I, J] < \text{Min}$  **ΤΟΤΕ**

$\text{Min} \leftarrow \text{Π}[I, J]$

$\text{Posmin} \leftarrow J$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ**  $\text{Posmin}$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**31.** Ένας επενδυτής διέθεσε 10.000 € για την αγορά ορισμένων τεμαχίων 10 διαφορετικών μετοχών. Να γράψετε αλγόριθμο ο οποίος:

α. Για καθεμία από τις 10 μετοχές διαβάζει το όνομα της μετοχής και το πλήθος των τεμαχίων της μετοχής, που κατέχει ο επενδυτής, ελέγχοντας το πλήθος να είναι θετικός αριθμός, και καταχωρίζει τα δεδομένα αυτά σε σχετικούς πίνακες.

β. Για καθεμία από τις 10 μετοχές και για καθεμία από τις πέντε (5) εργάσιμες ημέρες της εβδομάδας, διαβάζει την τιμή ενός τεμαχίου της μετοχής και την αποθηκεύει σε κατάλληλο πίνακα δύο διαστάσεων, ελέγχοντας η τιμή του τεμαχίου να είναι θετικός αριθμός.

γ. Για καθεμία από τις 10 μετοχές υπολογίζει τη μέση εβδομαδιαία τιμή του τεμαχίου της και την αποθηκεύει σε μονοδιάστατο πίνακα.

δ. Υπολογίζει και εμφανίζει τη συνολική αξία όλων των τεμαχίων όλων των μετοχών του

επενδυτή, την τελευταία ημέρα της εβδομάδας.

ε. Υπολογίζει εάν ο επενδυτής στο τέλος της εβδομάδας έχει κέρδος ή ζημία ή καμία μεταβολή σε σχέση με το αρχικό ποσό που διέθεσε, εμφανίζοντας κατάλληλα μηνύματα. (Θέμα Πανελλαδικών)

### Λύση

**Αλγόριθμος Χρηματιστήριο**

```
Για i από 1 μέχρι 10
    Διάβασε M[i]
    Αρχή_επανάληψης
        Διάβασε Π[i]
        Μέχρις_ότου Π[i] > 0
        Για j από 1 μέχρι 5
            Αρχή_επανάληψης
                Διάβασε T[i, j]
                Μέχρις_ότου T[i, j] > 0
            Τέλος_επανάληψης
        Τέλος_επανάληψης
    Τέλος_επανάληψης
```

```
Για i από 1 μέχρι 10
    Σ ← 0
    Για j από 1 μέχρι 5
        Σ ← Σ + T[i, j]
    Τέλος_επανάληψης
    ΜΤ[i] ← Σ / 5
Τέλος_επανάληψης
```

```
Σμ ← 0
Για i από 1 μέχρι 10
    Σμ ← Σμ + Π[i] * T[i, 5]
Τέλος_επανάληψης
```

```
Αν Σμ > 10000 τότε
    Εμφάνισε "Κέρδος"
Αλλιώς_Αν Σμ < 10000 τότε
    Εμφάνισε "Ζημία"
Αλλιώς
    Εμφάνισε "Καμία μεταβολή"
Τέλος_Αν
Τέλος Χρηματιστήριο
```

**32.** Μια εταιρεία αποθηκεύει είκοσι (20) προϊόντα σε δέκα (10) αποθήκες. Να γράψετε πρόγραμμα στη γλώσσα προγραμματισμού "ΓΛΩΣΣΑ", το οποίο:

- περιέχει τμήμα δήλωσης των μεταβλητών του προγράμματος
- εισάγει σε μονοδιάστατο πίνακα τα ονόματα των είκοσι προϊόντων
- εισάγει σε πίνακα δύο διαστάσεων Π[20,10] την πληροφορία που αφορά στην παρουσία ενός προϊόντος σε μια αποθήκη (καταχωρούμε την τιμή 1 στην περίπτωση που υπάρχει το προϊόν στην αποθήκη και την τιμή 0, αν το προϊόν δεν υπάρχει στην αποθήκη).

δ. υπολογίζει σε πόσες αποθήκες βρίσκεται το κάθε προϊόν  
ε. τυπώνει το όνομα κάθε προϊόντος και το πλήθος των αποθηκών στις οποίες υπάρχει το προϊόν. (Θέμα Πανελλαδικών)

### Λύση

**Πρόγραμμα** Αποθήκες

**Ακέραιες:** A[20,10], I, J, M[20]

**Χαρακτήρες:** ON[20]

Αρχή

Για I από 1 μέχρι 20

Εμφάνισε ' Δώσε το όνομα του προϊόντος'

Διάβασε ON[I]

Τέλος\_επανάληψης

Για I από 1 μέχρι 20

    Για J από 1 μέχρι 10

        Εμφάνισε ' Δώσε 1 Αν το 'I,' προϊόν βρίσκεται στην '

        Εμφάνισε J, 'αποθήκη'

        Εμφάνισε ' Δώσε 0 Αν το 'I,' προϊόν δεν βρίσκεται '

        Εμφάνισε 'στην' ,J, 'αποθήκη'

        Διάβασε A[I, J]

        Τέλος\_επανάληψης

Τέλος\_επανάληψης

Για I από 1 μέχρι 20

M[I] ← 0

Τέλος\_επανάληψης

Για I από 1 μέχρι 20

    Για J από 1 μέχρι 10

        M[I] ← M[I]+A[I,J]

    Τέλος\_επανάληψης

Τέλος\_επανάληψης

Για I από 1 μέχρι 20

Εμφάνισε 'Το Προϊόν ' ON[I],' υπάρχει σε ' , M[I], 'αποθήκες'

Τέλος\_επανάληψης

Τέλος\_προγράμματος

**33.** Για την παρακολούθηση των θερμοκρασιών της επικράτειας κατά το μήνα Μάιο, καταγράφεται κάθε μέρα η θερμοκρασία στις 12:00 το μεσημέρι για 20 πόλεις. Να σχεδιάσετε αλγόριθμο που:

α. θα διαβάζει τα ονόματα των 20 πόλεων και τις αντίστοιχες θερμοκρασίες για κάθε μία από τις ημέρες του μήνα και θα καταχωρεί τα στοιχεία σε πίνακες.

β. θα διαβάζει το όνομα μίας πόλης και θα εμφανίζει τη μέγιστη θερμοκρασία της στη διάρκεια του μήνα. Αν δεν υπάρχει η πόλη στον πίνακα, θα εμφανίζει κατάλληλα διαμορφωμένο μήνυμα.

γ. θα εμφανίζει το πλήθος των ημερών που η μέση θερμοκρασία των 20 πόλεων ξεπέρασε τους 20 C, αλλά όχι τους 30 C. (Θέμα Πανελλαδικών)

### Λύση

**Αλγόριθμος** θερμοκρασίες

Για I από 1 μέχρι 20

Διάβασε ON [i]

Τέλος\_επανάληψης

Για I από 1 μέχρι 20

Για J από 1 μέχρι 31

Διάβασε Θ [I, J]

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Διάβασε πόλη

K ← 0

I ← 1

F ← ψευδής

Όσο F = ψευδής και I ≤ 20 επανάλαβε

Αν ON [I] = πόλη τότε

K ← I

F ← αληθής

Αλλιώς

I ← I + 1

Τέλος\_Αν

Τέλος\_επανάληψης

Αν F = ψευδής τότε

Γράψε “δεν υπάρχει τέτοια πόλη”

Αλλιώς

MAX ← Θ [K, 1]

Για J από 2 μέχρι 31

Αν Θ [K, J] > MAX τότε

MAX ← Θ [K, J]

Τέλος\_Αν

Τέλος\_επανάληψης

Γράψε “ η μεγαλύτερη θερμοκρασία της πόλης ”, ON [K], “είναι”, MAX, “βαθμοί.”

Τέλος\_Αν

Για J από 1 μέχρι 31

Σ ← 0

Για I από 1 μέχρι 20

Σ ← Σ + Θ [I, J]

Τέλος\_επανάληψης

MO[J] ← Σ / 20

Τέλος\_επανάληψης

M ← 0

Για J από 1 μέχρι 31

Αν MO [J] > 20 και MO [J] ≤ 30 τότε

M ← M + 1

Τέλος\_Αν

Τέλος\_επανάληψης

Γράψε “Οι ημέρες που ξεπέρασαν τη μέση θερμοκρασία είναι:”, M

Τέλος\_θερμοκρασίες

34. Μια δισκογραφική εταιρεία καταγράφει στοιχεία για ένα έτος για κάθε ένα από τα 20

CDs που κυκλοφόρησε. Τα στοιχεία αυτά είναι ο τίτλος του CD, ο τύπος της μουσικής που περιέχει και οι μηνιαίες του πωλήσεις (ποσά σε ευρώ) στη διάρκεια του έτους. Οι τύποι μουσικής είναι δύο: «ορχηστρική» και «φωνητική».

Να αναπτυχθεί πρόγραμμα το οποίο:

α. Για κάθε ένα από τα 20 CDs, να διαβάζει τον τίτλο, τον τύπο της μουσικής και τις πωλήσεις του για κάθε μήνα, ελέγχοντας την έγκυρη καταχώριση του τύπου της μουσικής.

β. Να εμφανίζει τον τίτλο ή τους τίτλους των CDs με τις περισσότερες πωλήσεις τον 3ο μήνα του έτους.

γ. Να εμφανίζει τους τίτλους των ορχηστρικών CDs με ετήσιο σύνολο πωλήσεων τουλάχιστον 5000 ευρώ.

δ. Να εμφανίζει πόσα από τα CDs είχαν σύνολο πωλήσεων στο δεύτερο εξάμηνο μεγαλύτερο απ' ότι στο πρώτο. (Θέμα Πανελλαδικών)

## Λύση

### ΠΡΟΓΡΑΜΜΑ CD

#### ΜΕΤΑΒΛΗΤΕΣ

**ΧΑΡΑΚΤΗΡΕΣ:** τίτλος[20], τύπος[20]

**ΠΡΑΓΜΑΤΙΚΕΣ:** πωλήσεις[20,12], μαξ, σύνολο[20], σ, σ1, σ2

**ΑΚΕΡΑΙΕΣ:** ι,μ, πλήθος

#### ΑΡΧΗ

! Ερώτημα α.

**ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 20**

**ΓΡΑΨΕ** "Δώσε τον τίτλο του ", ι, " CD:"

**ΔΙΑΒΑΣΕ** τίτλος[ι]

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

**ΓΡΑΨΕ** "Δώσε τον τύπο του ", ι, " CD:"

**ΔΙΑΒΑΣΕ** τύπος[ι]

**ΜΕΧΡΙΣ\_ΟΤΟΥ** τύπος[ι]="ορχηστρική" Η τύπος[ι]="φωνητική"

**ΓΡΑΨΕ** "Δώσε τις πωλήσεις του ",ι," **Για** τον κάθε μήνα"

**ΓΙΑ μ ΑΠΟ 1 ΜΕΧΡΙ 12**

**ΔΙΑΒΑΣΕ** πωλήσεις [ι,μ]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

#### ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

! Ερώτημα β.

! Πρώτα βρίσκουμε τις περισσότερες πωλήσεις, μαξ

μαξ ← πωλήσεις[1,3]

**ΓΙΑ μ ΑΠΟ 2 ΜΕΧΡΙ 20**

**ΑΝ** πωλήσεις[μ,3] > μαξ **ΤΟΤΕ**

μαξ ← πωλήσεις[μ,3]

**ΤΕΛΟΣ\_ΑΝ**

#### ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

! Τυπώνουμε τους τίτλους των CD με τις περισσότερες πωλήσεις

**ΓΙΑ μ ΑΠΟ 1 ΜΕΧΡΙ 20**

**ΑΝ** πωλήσεις[μ,3] = μαξ **ΤΟΤΕ**

**ΓΡΑΨΕ** τίτλος[μ]

**ΤΕΛΟΣ\_ΑΝ**

#### ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

! Ερώτημα γ.

! Υπολογίζουμε το ετήσιο σύνολο για κάθε τίτλο

**ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 20**

$\sigma \leftarrow 0$

**ΓΙΑ μ ΑΠΟ 1 ΜΕΧΡΙ 12**

$\sigma \leftarrow \sigma + \text{πωλήσεις}[ι,μ]$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

$\text{σύνολο}[ι] \leftarrow \sigma$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*! Βρίσκουμε ποιοί τίτλοι έχουν ετήσιο σύνολο  $\geq 5000$  και είναι*

*δαορηστρικοί*

**ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 20**

**ΑΝ**  $\text{σύνολο}[ι] \geq 5000$  **ΚΑΙ**  $\text{τύπος}[ι] = \text{"ορηστρική"}$  **ΤΟΤΕ**

**ΓΡΑΨΕ**  $\text{τίτλος}[ι]$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*! Ερώτημα δ.*

$\text{πλήθος} \leftarrow 0$

**ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 20**

*! Υπολογίζουμε το σύνολο πωλήσεων α' και β' εξαμήνου*

$\sigma_1 \leftarrow 0$

$\sigma_2 \leftarrow 0$

**ΓΙΑ μ ΑΠΟ 1 ΜΕΧΡΙ 6**

$\sigma_1 \leftarrow \sigma_1 + \text{πωλήσεις}[ι,μ]$

$\sigma_2 \leftarrow \sigma_2 + \text{πωλήσεις}[ι,μ+6]$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*! Συγκρίνουμε τα δύο σύνολα*

**ΑΝ**  $\sigma_2 > \sigma_1$  **ΤΟΤΕ**

$\text{πλήθος} \leftarrow \text{πλήθος} + 1$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** "Περισσότερες πωλήσεις στο β' εξάμηνο είχαν: ",  $\text{πλήθος}$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**35.** Στο ευρωπαϊκό πρωτάθλημα ποδοσφαίρου συμμετέχουν 16 ομάδες. Κάθε ομάδα συμμετέχει σε 30 αγώνες. Να γράψετε αλγόριθμο ο οποίος:

α. Διαβάζει σε μονοδιάστατο πίνακα  $ΟΝ[16]$  τα ονόματα των ομάδων.

β. Διαβάζει σε δισδιάστατο πίνακα  $ΑΠ[16,30]$  τα αποτελέσματα σε κάθε αγώνα ως εξής: Το χαρακτήρα «N» για ΝΙΚΗ, το χαρακτήρα «I» για ΙΣΟΠΑΛΙΑ και το χαρακτήρα «H» για ΗΤΤΑ και κάνει τον απαραίτητο έλεγχο εγκυρότητας των δεδομένων.

γ. Για κάθε ομάδα υπολογίζει και καταχωρεί σε δισδιάστατο πίνακα  $ΠΛ[16,3]$  το πλήθος των νικών στην πρώτη στήλη, το πλήθος των ισοπαλιών στη δεύτερη στήλη, και το πλήθος των ηττών στην τρίτη στήλη του πίνακα. Ο πίνακας αυτός πρέπει προηγουμένως να έχει μηδενισθεί.

δ. Με βάση τα στοιχεία του πίνακα  $ΠΛ[16,3]$  υπολογίζει και καταχωρεί σε νέο πίνακα  $ΒΑΘ[16]$  τη συνολική βαθμολογία κάθε ομάδας, δεδομένου ότι για κάθε νίκη η ομάδα παίρνει τρεις βαθμούς, για κάθε ισοπαλία έναν βαθμό και για κάθε ήττα κανένα βαθμό.

ε. Εμφανίζει τα ονόματα και τη βαθμολογία των ομάδων, ταξινομημένα σε φθίνουσα σειρά με βάση τη βαθμολογία. (Θέμα Πανελλαδικών)

**Λύση**



**Αλγόριθμος** Πρωτάθλημα

**Για**  $i$  από 1 μέχρι 16

**Διάβασε**  $ON[i]$

**Τέλος\_επανάληψης**

**Για**  $i$  από 1 μέχρι 16

**Για**  $j$  από 1 μέχρι 30

**Αρχή\_επανάληψης**

**Διάβασε**  $AP[i, j]$

**Μέχρις\_ότου**  $AP[i, j] = "N"$  ή  $AP[i, j] = "I"$  ή  $AP[i, j] = "H"$

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Για**  $i$  από 1 μέχρι 16

**Για**  $j$  από 1 μέχρι 3

$PL[i, j] \leftarrow 0$

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Για**  $i$  από 1 μέχρι 16

$mn \leftarrow 0$

$mi \leftarrow 0$

$mj \leftarrow 0$

**Για**  $j$  από 1 μέχρι 30

**Αν**  $AP[i, j] = "N"$  τότε

$mn \leftarrow mn + 1$

**Αλλιώς\_Αν**  $AP[i, j] = "I"$  τότε

$mi \leftarrow mi + 1$

**Αλλιώς**

$mj \leftarrow mj + 1$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

$PL[i, 1] \leftarrow mn$

$PL[i, 2] \leftarrow mi$

$PL[i, 3] \leftarrow mj$

$B[i] \leftarrow PL[i, 1] * 3 + PL[i, 2] * 1$

**Τέλος\_επανάληψης**

**Για**  $i$  από 2 μέχρι 16

**Για**  $j$  από 16 μέχρι  $i$  με\_βήμα -1

**Αν**  $B[j-1] < B[j]$  τότε

**Αντιμετάθεσε**  $B[j-1], B[j]$

**Αντιμετάθεσε**  $ON[j-1], ON[j]$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Για**  $i$  από 1 μέχρι 16

**Εμφάνισε**  $ON[i], B[i]$

**Τέλος\_επανάληψης**

**Τέλος** Πρωτάθλημα

**36.** Σε ένα πανελλήνιο σχολικό διαγωνισμό μετέχουν 20 σχολεία. Κάθε σχολείο αξιολογεί 5 άλλα σχολεία και δεν αυτοαξιολογείται. Η βαθμολογία κυμαίνεται από 1 έως και 10. Να γραφεί αλγόριθμος που:

α) να διαβάζει τα ονόματα των σχολείων και να τα αποθηκεύει σε μονοδιάστατο πίνακα A 20 θέσεων.

β) να εισάγει αρχικά την τιμή 0 σε όλες τις θέσεις ενός δισδιάστατου πίνακα B 20 γραμμών και 20 στηλών.

γ) Να καταχωρίζει στον πίνακα B τη βαθμολογία που δίνει κάθε σχολείο για 5 άλλα σχολεία. Σημείωση: Στη θέση  $i, j$  του πίνακα B αποθηκεύεται ο βαθμός που το σχολείο  $i$  δίνει στο σχολείο  $j$ , όπως φαίνεται στο παράδειγμα που ακολουθεί.

δ) να υπολογίζει τη συνολική βαθμολογία του κάθε σχολείου και να την καταχωρίζει σε μονοδιάστατο πίνακα 20 θέσεων με όνομα SUM.

ε) να εμφανίζει τα ονόματα και τη συνολική βαθμολογία όλων των σχολείων κατά φθίνουσα σειρά της συνολικής βαθμολογίας. (Θέμα Πανελλαδικών)

### **Λύση**

**Αλγόριθμος** Θέμα\_4

Για  $i$  από 1 μέχρι 20

Διάβασε A[i]

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι 20

Για  $j$  από 1 μέχρι 20

$B[i, j] \leftarrow 0$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι 20 ! κάθε σχολείο δίνει βαθμό

Για  $j$  από 1 μέχρι 5 ! σε 5 άλλα

Αρχή\_επανάληψης

Διάβασε  $\delta$  ! σε ποιο σχολείο δίνει βαθμός εκτός του εαυτού του

Μέχρις\_ότου ( $\delta \geq 1$  και  $\delta \leq 20$ ) και  $\delta \neq i$

Αρχή\_επανάληψης

Διάβασε  $B[i, \delta]$

Μέχρις\_ότου  $B[i, \delta] \geq 1$  και  $B[i, \delta] \leq 20$  ! τι βαθμό δίνει, στην κατάλληλη θέση B

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Για  $j$  από 1 μέχρι 20 ! άθροισμα κατά στήλη, ερώτημα  $\delta$

$\acute{\alpha}\theta\rho \leftarrow 0$

Για  $i$  από 1 μέχρι 20

$\acute{\alpha}\theta\rho \leftarrow \acute{\alpha}\theta\rho + B[i, j]$

Τέλος\_επανάληψης

$SUM[j] \leftarrow \acute{\alpha}\theta\rho$

Τέλος\_επανάληψης

Για  $i$  από 2 μέχρι 20

Για j από 20 μέχρι i με\_βήμα -1  
Αν  $SUM[j-1] < SUM[j]$  τότε  
  Αντιμετάθεσε  $SUM[j-1], SUM[j]$   
  Αντιμετάθεσε  $A[j-1], A[j]$   
Τέλος\_Αν  
Τέλος\_επανάληψης  
Τέλος\_επανάληψης  
Για i από 1 μέχρι 20  
  Εκτύπωσε  $A[i]$   
Τέλος\_επανάληψης  
Τέλος Θέμα\_4

## 5.2 Θέματα Πανελληνίου Διαγωνισμού Πληροφορικής

37. Ζητείται να γίνει ο αλγόριθμος ο οποίος:

α) Να δέχεται τις τιμές των στοιχείων  $a$  με  $i=1, 2, \dots, n$  και  $j=1, 2, \dots, n$  ενός τετραγωνικού πίνακα  $A(n \times n)$ .

β) Να υπολογίζει και να τυπώνει :

(I) Το άθροισμα των στοιχείων των δύο διαγωνίων του πίνακα.

(II) Το άθροισμα των στοιχείων κάθε διαγωνίου που βρίσκεται πάνω και κάτω από την κύρια διαγώνιο.

### Λύση

**Αλγόριθμος ΠΔΠ1 !1989 1ος**

$N \leftarrow 4$

**Για**  $i$  από 1 μέχρι  $N$

**Για**  $j$  από 1 μέχρι  $N$

        Διάβασε  $table[i, j]$

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

$s1 \leftarrow 0$

$s2 \leftarrow 0$

$s3 \leftarrow 0$

$s4 \leftarrow 0$

**Για**  $i$  από 1 μέχρι  $N$

**Για**  $j$  από 1 μέχρι  $N$

**Αν**  $i = j$  τότε

$s1 \leftarrow s1 + table[i, j]$

**Τέλος\_Αν**

**Αν**  $j = i + N - 1$  τότε

$s2 \leftarrow s2 + table[i, j]$

**Τέλος\_Αν**

**Αν**  $j = j+1$  τότε

$s3 \leftarrow s2 + table[i, j]$

**Τέλος\_Αν**

**Αν**  $i = j-1$  τότε

$s2 \leftarrow s2 + table[i, j]$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Εμφάνισε**  $s1, s2, s3, s4$

**Τέλος ΠΔΠ1**

38. Οι Μακεδόνες, Δωρικό Ελληνικό φύλο με ανθρώπους έχοντες μάκος (μήκος = ψηλούς), έχοντας την ίδια γλώσσα, την ίδια θρησκεία, τα ίδια ήθη και έθιμα αλλά και την

ίδια ιστορική διαδρομή με τα άλλα Ελληνικά φύλα, ενοποίησαν επί Φιλίππου του Β' (382 - 336 π.Χ.) τα Ελληνικά Βασίλεια (Πανελλήνια Συμμαχία). Αμέσως μετά, υπό το γιο του, Αλέξανδρο τον Μέγα (356 - 323 π.Χ.) εγκατέστησαν τη μεγαλύτερη αυτοκρατορία όλων των εποχών. Οι Ρωμαίοι των οποίων η αυτοκρατορία, διαδέχθηκε αυτή του Μ. Αλεξάνδρου, δεν έκρυψαν ποτέ το θαυμασμό τους για το μεγάλο στρατηλάτη. Στην Πομπηία διατηρήθηκε στο μεγαλύτερο μέρος του, ένα από τα αρχαιότερα ψηφιδωτά που Μεγάλου Αλεξάνδρου, που απεικονίζει τον Μ. Αλέξανδρο να νικά τον Δαρείο στη μάχη της Ισού. Το Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, επί πολλές δεκαετίες διεξάγει συστηματικά αρχαιολογική έρευνα σχετικά με την περίοδο της ακμής του Μακεδονικού Πολιτισμού στον ευρύτερο Ελλαδικό αλλά και διεθνή χώρο. Έκπληκτοι οι ερευνητές του ΑΠΘ, διαπίστωσαν ότι οι χρωματιστές ψηφίδες που χρησιμοποιήθηκαν για την κατασκευή του ψηφιδωτού, είχαν την ίδια χημική σύσταση και με δύο διαφορετικές συγκεντρώσεις Μαγνησίου (Mg) που έχουν διαπιστωθεί μόνο στην ευρύτερη περιοχή της Πέλλας (P) και της Βεργίνας (V) αντίστοιχα! Θραύσματα από άλλα ψηφιδωτά της Ελληνιστικής περιόδου, από όλο το γνωστό τότε κόσμο, επιβεβαίωσαν τα ίδια αποτελέσματα Είναι λοιπόν φανερό, ότι οι Έλληνες βασιλείς και οι Ρωμαίοι αργότερα, θεωρούσαν στοιχείο ισχύος να διακοσμήσουν τα βασίλεια τους με χρωματιστές ψηφίδες από την μητροπολιτική Ελλάδα και μάλιστα από την περιοχή της Μακεδονίας. Το ενδιαφέρον επίσης από τεχνική άποψη, είναι ότι καμία ψηφίδα δεν εφάπτονταν κάθετα ή οριζόντια με ψηφίδα αντίστοιχης προέλευσης. Στο παρακάτω σχήμα δίνεται ένα μεγεθυμένο τμήμα ψηφιδωτού. P V P V P V P

Πρόβλημα: Να αναπτύξετε έναν αλγόριθμο ο οποίος αφού διαβάσει τις διαστάσεις του ψηφιδωτού και τον τύπο της πρώτης ψηφίδας θα επιστρέφει τη διάταξη των ψηφίδων από τις δύο τοποθεσίες

### **Λύση**

! 21ος ΠΔΠ, Ψηφιδωτό Μεγάλου Αλεξάνδρου

! Γνωστικοί στόχοι: Δομή επιλογής & επανάληψης, εμφώλευση, Πίνακες 2d & αλγοριθμική σκέψη...

**Αλγόριθμος** πδπ21

**Διάβασε** L, C

**Διάβασε** Ψ[1, 1]

X ← Ψ[1, 1]

**Αν** X = 'P' τότε

    γ ← 'V'

**αλλιώς**

    γ ← 'P'

**Τέλος\_Αν**

**Για** i από 1 μέχρι L με\_βήμα 1

**Αν** i mod 2 = 1 τότε

**Για** j από 1 μέχρι C με\_βήμα 1

**Αν** j mod 2 = 1 τότε

                Ψ[i, j] ← X

**αλλιώς**

                Ψ[i, j] ← γ

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**αλλιώς**

**Για**  $j$  από 1 μέχρι  $C$  με\_βήμα 1

**Αν**  $j \bmod 2 = 1$  τότε

$\Psi[i, j] \leftarrow \gamma$

**αλλιώς**

$\Psi[i, j] \leftarrow \chi$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από 1 μέχρι  $L$  με\_βήμα 1

**Για**  $j$  από 1 μέχρι  $C$  με\_βήμα 1

Εμφάνισε  $\Psi[i, j]$

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

**Τέλος\_πδπ21**

## 5.3 Ασκήσεις προς επίλυση

1. Η Ε.Μ.Υ. καταγράφει ανά 8 ώρες τις θερμοκρασίες 10 πόλεων της Ελλάδος (τα ονόματα των πόλεων είναι κωδικοποιημένα με αριθμούς: 1=Θεσσαλονίκη, 2=Αθήνα κ.λπ.). Να γραφεί πρόγραμμα που θα διαβάσει τις θερμοκρασίες ενός εικοσιτετραώρου για κάθε πόλη και υπολογίζει-εμφανίζει: τον εθνικό μέσο όρο θερμοκρασίας και για κάθε πόλη τη μέση θερμοκρασία της και τη μέγιστη απόκλισή της από τον εθνικό μέσο όρο.

2. Στο διαγωνισμό PISA που έγινε το Μάρτιο του 2012 συμμετείχαν, 49 σχολεία της Κύπρου. Ο αριθμός συμμετοχής για κάθε σχολείο ήταν 80 μαθητές στη γραπτή εξέταση και 35 μαθητές στην εξέταση στον Η/Υ.

Τα ονόματα των σχολείων αποθηκεύονται σε πίνακα schools και οι βαθμολογίες σε δύο δισδιάστατους πίνακες (παράλληλους) με όνομα grapt και electro αντίστοιχα (49\*80). Να γραφεί πρόγραμμα το οποίο αφού διαβάσει τους πίνακες,

- Υπολογίζει τους μέσους όρους των βαθμολογιών που πήρε το κάθε σχολείο στα γραπτά και στους Η/Υ και τους καταχωρεί αντίστοιχα στους μονοδιάστατους mesosg και mesose.
- Βρίσκει και τυπώνει το σχολείο με το μεγαλύτερο άθροισμα των 2 μέσων όρων.
- Υπολογίζει και τυπώνει πόσοι μαθητές από κάθε σχολείο έχουν βαθμολογία μεγαλύτερη από το μέσο όρο του σχολείου τους στα γραπτά.

3. Να γραφεί αλγόριθμος ο οποίος:

- διαβάζει σε πίνακα A[10,2] τη μάρκα και την κατηγορία ενός οχήματος: στην πρώτη στήλη αποθηκεύεται η μάρκα και στη δεύτερη στήλη με έλεγχο εγκυρότητας η κατηγορία (ΙΧ, Δίκυκλο, Επαγγελματικό)
- διαβάζει σε πίνακα B[10,12] τις μηνιαίες πωλήσεις κάθε μάρκας
- υπολογίζει στον πίνακα MO[50] το μέσο όρο πωλήσεων κάθε μάρκας οχήματος
- Εμφανίζει το πλήθος και τη μάρκα των οχημάτων ΙΧ με μέσο όρο πωλήσεων πάνω από 900000 ευρώ.

4. Σε ένα Εσπερινό Γυμνάσιο φοιτούν 80 μαθητές. Να γραφεί αλγόριθμος ο οποίος:

- Διαβάζει για κάθε μαθητή το ονοματεπώνυμό του, την τάξη του και τον τελικό βαθμό του και τα καταχωρεί σε μονοδιάστατους πίνακες, ελέγχοντας την ορθότητα εισαγωγής των δεδομένων σύμφωνα με τα παρακάτω:
  - Οι τάξεις είναι Α ή Β ή Γ.
  - Ο τελικός βαθμός είναι από 1 μέχρι και 20.
- Εμφανίζει τα ονόματα των μαθητών της Β τάξης που έχουν τελικό βαθμό μεγαλύτερο ή ίσο του 18,5.
- Υπολογίζει και εμφανίζει το πλήθος των μαθητών κάθε τάξης.
- Υπολογίζει και εμφανίζει το μέσο όρο των τελικών βαθμών των μαθητών της Γ τάξης.
- Εμφανίζει ταξινομημένα κατά αλφαβητική σειρά τα ονοματεπώνυμα και τους αντίστοιχους τελικούς βαθμούς των μαθητών της Α τάξης.

5. Ένας όμιλος αποτελείται από 20 εταιρίες. Να γράψετε αλγόριθμο ο οποίος:

- να διαβάσει τα ονόματα των εταιριών του ομίλου και τα κέρδη τους για κάθε ένα από τα έτη 2001 έως και 2005. (Θεωρήστε ότι τα κέρδη είναι θετικοί αριθμοί).
- να υπολογίζει για κάθε εταιρία το συνολικό κέρδος της στην πενταετία.
- να εμφανίζει το όνομα της εταιρίας με τα περισσότερα κέρδη στην πενταετία.

(Θεωρήστε ότι η εταιρία αυτή είναι μοναδική).

Δ4. να διαβάξει το όνομα μιας εταιρίας και, αν η εταιρία αυτή δεν ανήκει στον όμιλο, να εμφανίζει κατάλληλο μήνυμα. Διαφορετικά να υπολογίζει και να εμφανίζει το έτος με τα λιγότερα κέρδη για την εταιρία αυτή. (Θεωρήστε ότι το έτος αυτό είναι μοναδικό για κάθε εταιρία).

**6.** Εταιρεία, που ασχολείται με μετρήσεις τηλεθέασης καταγράφει στοιχεία, ανά ημέρα και για χρονικό διάστημα μίας εβδομάδας, τα οποία αφορούν την τηλεθέαση των κεντρικών δελτίων ειδήσεων που προβάλλονται από πέντε (5) τηλεοπτικούς σταθμούς. Για τη διευκόλυνση της στατιστικής επεξεργασίας των δεδομένων, να αναπτύξετε αλγόριθμο, ο οποίος:

Δ1. Για κάθε έναν από τους τηλεοπτικούς σταθμούς να δέχεται το όνομά του και το πλήθος των τηλεθεατών, που παρακολούθησαν το κεντρικό δελτίο ειδήσεων κάθε μέρα της εβδομάδας, από Δευτέρα έως και Κυριακή, χωρίς έλεγχο εγκυρότητας, δηλαδή θεωρήστε ότι οι τιμές που εισάγονται είναι θετικοί αριθμοί και η εισαγωγή των δεδομένων γίνεται χωρίς λάθη.

Δ2. Να εμφανίζει τα ονόματα των σταθμών, για τους οποίους ο μέσος όρος τηλεθέασης του Σαββατοκύριακου (2 ημέρες) ήταν τουλάχιστον 10% μεγαλύτερος από το μέσο όρο τηλεθέασης στις καθημερινές (Δευτέρα έως Παρασκευή).

Δ3. Να εμφανίζει τα ονόματα των τηλεοπτικών σταθμών, οι οποίοι κάθε ημέρα από Δευτέρα έως και Κυριακή παρουσιάζουν συνεχώς, δηλαδή από ημέρα σε ημέρα, αύξηση τηλεθέασης. Αν δεν υπάρχουν τέτοιοι σταθμοί, να εμφανίζει το μήνυμα «κανένας σταθμός δεν έχει συνεχή αύξηση τηλεθέασης».

**7.** Μια αλυσίδα ξενοδοχείων έχει 5 ξενοδοχεία. Σε ένα μονοδιάστατο πίνακα ΞΕΝΟΔΟΧΕΙΑ[5] καταχωρούνται τα ονόματα των ξενοδοχείων. Σε έναν άλλο δισδιάστατο πίνακα ΕΙΣΠΡΑΞΕΙΣ[5,12] καταχωρούνται οι εισπράξεις κάθε ξενοδοχείου για κάθε μήνα του έτους 2001, έτσι ώστε στην *i* γραμμή καταχωρούνται οι εισπράξεις του *j* ξενοδοχείου. Να αναπτύξετε αλγόριθμο, ο οποίος:

α. διαβάξει τα στοιχεία των δύο πινάκων.

β. εκτυπώνει το όνομα κάθε ξενοδοχείου και τις ετήσιες εισπράξεις του για το έτος 2001.

γ. εκτυπώνει το όνομα του ξενοδοχείου με τις μεγαλύτερες εισπράξεις για το έτος 2001. (Θέμα Πανελλαδικών)

**8.** Κατά τη διάρκεια πρωταθλήματος μπάσκετ μια ομάδα που αποτελείται από δώδεκα (12) παίκτες έδωσε είκοσι (20) αγώνες, στους οποίους συμμετείχαν όλοι οι παίκτες. Να αναπτύξετε στο τετράδιό σας αλγόριθμο ο οποίος:

α. Να διαβάξει τα ονόματα των παικτών και να τα αποθηκεύει σε μονοδιάστατο πίνακα.

β. Να διαβάξει τους πόντους που σημείωσε κάθε παίκτης σε κάθε αγώνα και να τους αποθηκεύει σε πίνακα δύο διαστάσεων.

γ. Να υπολογίζει για κάθε παίκτη το συνολικό αριθμό πόντων του σε όλους τους αγώνες και το μέσο όρο πόντων ανά αγώνα.

δ. Να εκτυπώνει τα ονόματα των παικτών της ομάδας και το μέσο όρο πόντων του κάθε παίκτη ταξινομημένα με βάση το μέσο όρο τους κατά φθίνουσα σειρά.

Παρατήρηση: Σε περίπτωση ισοβαθμίας δεν μας ενδιαφέρει η σχετική σειρά των παικτών. (Θέμα Πανελλαδικών)

**9.** Ένα εμπορικό κατάστημα έχει καταγράψει τις μηνιαίες εισπράξεις του για τα έτη 2009 και 2010.



Να γράψετε αλγόριθμο ο οποίος:

- Γ1. Να διαβάσει τις μηνιαίες εισπράξεις για καθένα από τα δύο έτη και να τις καταχωρίζει σε αντίστοιχους μονοδιάστατους πίνακες.
- Γ2. Να υπολογίζει και να εμφανίζει τη μεγαλύτερη μηνιαία εισπράξη για κάθε έτος. Θεωρήστε ότι για κάθε έτος η τιμή αυτή είναι μοναδική.
- Γ3. Να εμφανίζει κατάλληλο μήνυμα στην περίπτωση που ο μήνας κατά τον οποίο σημειώθηκε η μεγαλύτερη μηνιαία εισπράξη ήταν ο ίδιος και για τα δύο έτη.
- Γ4. Να εμφανίζει τον μέσο όρο των μηνιαίων εισπράξεων για κάθε έτος.
- Γ5. Να υπολογίζει και να εμφανίζει το πλήθος των μηνών του έτους 2009 κατά τους οποίους η μηνιαία εισπράξη ήταν μεγαλύτερη από αυτή του αντίστοιχου μήνα του έτους 2010.

**10.** Σ' ένα διαγωνισμό συμμετέχουν 100 υποψήφιοι. Κάθε υποψήφιος διαγωνίζεται σε 50 ερωτήσεις πολλαπλής επιλογής. Να αναπτύξετε αλγόριθμο που να κάνει τα παρακάτω:

α. Να καταχωρεί σε πίνακα ΑΠ[100,50] τα αποτελέσματα των απαντήσεων του κάθε υποψηφίου σε κάθε ερώτηση. Κάθε καταχώρηση μπορεί να είναι μόνο μία από τις παρακάτω:

i. Σ αν είναι σωστή η απάντηση

ii. Λ αν είναι λανθασμένη η απάντηση και

iii. Ξ αν ο υποψήφιος δεν απάντησε. Να γίνεται έλεγχος των δεδομένων εισόδου.

β. Να βρίσκει και να τυπώνει τους αριθμούς των ερωτήσεων που παρουσιάζουν το μεγαλύτερο βαθμό δυσκολίας, δηλαδή έχουν το μικρότερο πλήθος σωστών απαντήσεων.

γ. Αν κάθε Σ βαθμολογείται με 2 μονάδες, κάθε Λ με -1 μονάδα και κάθε Ξ με 0 μονάδες τότε:

i. Να δημιουργεί ένα μονοδιάστατο πίνακα ΒΑΘ[100], κάθε στοιχείο του οποίου θα περιέχει αντίστοιχα τη συνολική βαθμολογία ενός υποψηφίου.

ii. Να τυπώνει το πλήθος των υποψηφίων που συγκέντρωσαν βαθμολογία μεγαλύτερη από 50.

**11.** Να γραφεί αλγόριθμος ο οποίος σε διδιάστατο πίνακα 500\*4 καταγράφει το βαθμό ικανοποίησης 500 χρηστών Η/Υ σε σχέση με τα λειτουργικά συστήματα Android, Windows, Linux, MacOS αντίστοιχα, με αποδεκτούς χαρακτηρισμούς (καλό, μέτριο, κακό). Έπειτα υπολογίζει-εμφανίζει το πιο δημοφιλές-αποδεκτό (το ΛΣ με τα περισσότερα "καλό").

**12.** Να γραφεί αλγόριθμος ο οποίος για δεδομένη εικόνα - διδιάστατο πίνακα 800\*600 (με διαβαθμίσεις του γκρι-φωτεινότητας 0-255), να την μετατρέπει σε δίτονη (δηλαδή τιμή pixel - στοιχείου πίνακα μέχρι 127 να γίνεται 0 και πάνω από 127 να γίνεται 255).

**13.** Στον Πανελλήνιο διαγωνισμό Πληροφορικής στα πλαίσια της Β φάσης έδωσαν σωστή λύση 340 μαθητές. Σε δισδιάστατο πίνακα 340\*2, αποθηκεύονται τα επώνυμα και οι νομοί καταγωγής του κάθε διαγωνιζόμενου και σε άλλο μονοδιάστατο πίνακα οι χρόνοι εκτέλεσης σε sec του κώδικά τους. Στην τελική φάση προκρίνονται οι διαγωνιζόμενοι με χρόνο εκτέλεσης λιγότερο των 3 sec. Να εμφανισθούν σε φθίνουσα σειρά οι 10 πρώτοι που προκρίνονται (αν υπάρχουν τόσοι). Θεωρείστε ότι προκρίνεται τουλάχιστον ένας.

**14.** Να γραφεί αλγόριθμος που:

α) θα διαβάσει τα ονόματα των 20 ποδοσφαιριστών μιας ομάδας καθώς και τα λεπτά συμμετοχής τους σε καθένα από τα 30 παιχνίδια του πρωταθλήματος και θα τα

αποθηκεύει σε πίνακες  $on[20]$ ,  $\lambda[20,30]$

β) Θα βρίσκει για κάθε παίκτη τα συνολικά λεπτά συμμετοχής και έπειτα θα τα τοποθετεί σε φθίνουσα σειρά. Σε περίπτωση ισοβαθμίας, η ταξινόμηση γίνεται κατά αλφαβητική σειρά (α -> ω).

γ) Θα δέχεται από το χρήστη ένα όνομα και σε περίπτωση που υπάρχει παίκτης με αυτό το όνομα θα εμφανίζει το πλήθος των ποδοσφαιριστών των οποίων τα λεπτά συμμετοχής είναι λιγότερα από τα λεπτά συμμετοχής του συγκεκριμένου παίκτη. Σε περίπτωση που δεν υπάρχει τέτοιος παίκτης θα εμφανίζει κατάλληλο μήνυμα.

δ) Θα εμφανίζει για κάθε ποδοσφαιριστή το μέγιστο αριθμό συνεχόμενων παιχνιδιών στα οποία δεν είχε ούτε λεπτό συμμετοχής.

**15.** Να γραφεί πρόγραμμα το οποίο θα δημιουργεί μια εικόνα RGB ανάλυσης, 1024 x 1024 pixels, με τον τρόπο που περιγράφεται παρακάτω:

1) Δημιουργία 3 διδιάστατων πινάκων 1024x1024 pixels ( $R[1024,1024]$ ,  $G[1024,1024]$ ,  $B[1024,1024]$ ), όπου σε κάθε κελί του κάθε πίνακα θα αναγράφεται η τιμή του χρώματος κόκκινου, πράσινου και μπλε ( $R[1024,1024]$ ,  $G[1024,1024]$ ,  $B[1024,1024]$ ) αντίστοιχα, με επιτρεπόμενα πεδία από 0 έως 255. Να γίνει έλεγχος επιτρεπόμενων τιμών.

2) Απόδοση θολούρας πρώτης τάξης κατά 10% σε πεδίο 9 pixel 3 x 3.

Κάθε pixel παίρνει 10% χρώματος από τα 8 pixel που το περικυκλώνουν.

Έστω ότι στο πεδίο του R έχουμε τα εξής χρώματα:

155 200 100

000 015 023

150 111 255

τότε το κεντρικό χρώμα 015 θα επηρεαστεί κατά 10% από τον μέσο όρο των 8 γειτονικών του (ακέραια διαίρεση). Να εφαρμοστεί θολούρα σε κάθε pixel της εικόνας εκτός από την πρώτη και τελευταία γραμμή και στήλη.

Να αποθηκευτεί η θολούρα σε 3 πίνακες  $R\_BLUR$ ,  $G\_BLUR$  και  $B\_BLUR$ .

Τέλος, να μετατραπεί και να εμφανισθεί η τελική εικόνα σε έναν τελικό  $RGB[1024,1024]$  πίνακα μορφοποίηση 255255255.

**16.** Το παιχνίδι αποτελείται από μια συλλογή κυττάρων τα οποία συμβολίζονται με X στο πλέγμα. Κάθε κύτταρο γειτονεύει με τα οκτώ κελιά που πρόσκεινται σε αυτό (δεξιά, αριστερά, πάνω, κάτω, διαγώνια). Υπάρχουν τέσσερις κανόνες που καθορίζουν το πότε γεννιέται ένα νέο κύτταρο (στα άδεια κελιά) και πότε πεθαίνει ή πολλαπλασιάζεται κάποιο ζωντανό:

1. Κάθε ζωντανό κύτταρο με ένα ή κανένα ζωντανό γείτονα θα πεθάνει στην επόμενη γενεά, ως απομονωμένο.

2. Κάθε ζωντανό κύτταρο με δύο ή τρεις ζωντανούς γείτονες θα παραμένει ζωντανό και στην επόμενη γενεά.

3. Κάθε ζωντανό κύτταρο με τέσσερις ή περισσότερους γείτονες θα πεθαίνει στην επόμενη γενεά, λόγω ανταγωνισμού-υπερπληθυσμού.

4. Αν υπάρχει μια κενή θέση που γειτονεύει με ακριβώς τρία ζωντανά κύτταρα τότε στην επόμενη γενεά θα γεννιέται εκεί ένα νέο κύτταρο.

**17.** Οι εικόνες σε έναν υπολογιστή χωρίζονται σε ένα πίνακα από μικρές κουκκίδες που ονομάζονται εικονοστοιχεία (pixels). Σε μια ασπρόμαυρη εικόνα, κάθε εικονοστοιχείο είναι είτε μαύρο είτε άσπρο. Ο υπολογιστής μπορεί να παρουσιάζει ασπρόμαυρες εικόνες με τη βοήθεια ακολουθίας αριθμών, όπως για παράδειγμα:

Ο πρώτος αριθμός δίνει πάντα τον αριθμό των λευκών εικονοστοιχείων, ο επόμενος

αριθμός δίνει τον αριθμό των μαύρων εικονοστοιχείων και ούτω καθεξής. Για παράδειγμα, η πρώτη γραμμή αποτελείται από 0 λευκά εικονοστοιχεία ακολουθούμενη από 5 μαύρα εικονοστοιχεία. Η δεύτερη γραμμή αποτελείται από 2 λευκά εικονοστοιχεία, 1 μαύρο εικονοστοιχείο και 2 λευκά εικονοστοιχεία.

Ποιο γράμμα παρουσιάζει η παρακάτω ακολουθία αριθμών;

0,1,3,1

0,1,3,1

0,5

0,1,3,1

0,1,3,1

**18.** Ανάμεσα στις πληροφορίες που καταγράφονται στις εφημερίδες υπάρχουν και οι ημερήσιες πωλήσεις των 10 μεγαλύτερων σε κυκλοφορία εφημερίδων. Κάποιος παρακολουθεί τα στοιχεία αυτά κατά τη διάρκεια ενός έτους. Να γίνει αλγόριθμος ο οποίος

α) Καταγράφει σε πίνακα τα ονόματα των 10 εφημερίδων και για κάθε μία από αυτές καταγράφει σε δεύτερο πίνακα τις ημερήσιες πωλήσεις της. Στα στοιχεία που αφορούν τις ημερήσιες πωλήσεις θα γίνεται έλεγχος εγκυρότητας ώστε να είναι θετικά.

β) Υπολογίζει και τυπώνει τις ετήσιες πωλήσεις κάθε εφημερίδας καθώς και το όνομα αυτής.

γ) Με βάση τις ετήσιες πωλήσεις τυπώνει τις 3 πρώτες σε πωλήσεις εφημερίδες.

δ) Υπολογίζει και τυπώνει για κάθε μία από τις 10 εφημερίδες ποιος ήταν ο μήνας που πραγματοποίησε τις περισσότερες πωλήσεις.

**19.** Μια εταιρία διαθέτει 100 πωλητές σε όλη την Ελλάδα. Να γίνει πρόγραμμα το οποίο:

- να καταχωρεί σε πίνακα ΟΝ[100] τα ονόματα τους και σε πίνακα ΠΩΛ[100, 12] τις μηνιαίες πωλήσεις έτσι ώστε αυτές να είναι μεγαλύτερες ή ίσες με 0.
- να εμφανίζει τα ονόματα αυτών που έχουν μηδενικές πωλήσεις τις περισσότερες φορές.
- να ζητάει το όνομα ενός πωλητή και να υπολογίζει τον μέσο όρο των πωλήσεων του αγνοώντας τις μηδενικές πωλήσεις.

**20.** Στο κέντρο της Αθήνας υπάρχουν 50 θέατρα, τα οποία για κάθε ένα από τα 8 τελευταία χρόνια και για κάθε μήνα κάθε θεατρικού έτους (9 μήνες έχει το θεατρικό έτος) συγκεντρώνουν τα έσοδα από τις παραστάσεις τους. Το υπουργείο πολιτισμού αποφάσισε να σταματήσει την επιδότηση στα 25 θέατρα που θα έχουν τα παρουσιάσει τα λιγότερα έσοδα την τελευταία θετία. Να γράψετε αλγόριθμο ο οποίος για κάθε θέατρο

1) Διαβάζει το όνομα του.

2) Για κάθε χρόνο διαβάζει τα μηνιαία έσοδά του, ελέγχοντας την εγκυρότητα τους ώστε να είναι θετικά και τα τοποθετεί σε πίνακα Έσοδα [8,9].

3) Υπολογίζει και τυπώνει για κάθε χρόνο, τον μήνα με τα περισσότερα έσοδα

4) Υπολογίζει τα συνολικά έσοδα του θεάτρου και τα αποθηκεύει σε πίνακα.

Β) Υπολογίζει και τυπώνει τα ονόματα των θεάτρων που δεν θα πάρουν επιδότηση.

**21.** Σε πίνακα Π[100, 10] εισάγονται οι 10 βαθμοί στα μαθήματα κατεύθυνσης 100 μαθητών της Γ' Λυκείου. Οι βαθμοί είναι ακέραιοι στο διάστημα [0, 100] και δεν θα πρέπει να γίνεται έλεγχος. Το τελευταίο μάθημα είναι οι «Αρχές Οικονομικής Θεωρίας». Αν κάποιος μαθητής δεν έχει εξεταστεί Πανελλαδικώς στο μάθημα, τότε εισάγεται στην αντίστοιχη θέση του πίνακα η τιμή -1. Σε δεύτερο πίνακα Α[100] εισάγονται τα επίθετα

των μαθητών. Να γραφεί πρόγραμμα το οποίο:

- Θα διαβάξει τα στοιχεία των πινάκων.
- Θα υπολογίζει και θα εμφανίζει το μέσο όρο για κάθε μαθητή στα 9 πρώτα μαθήματα κατεύθυνσης μαζί με το όνομά του.
- Θα ταξινομεί τα ονόματα των μαθητών κατά φθίνουσα σειρά ως προς τον μέσο όρο στα 9 πρώτα μαθήματα κατεύθυνσης και θα εμφανίζει τους 10 πρώτους.
- Θα εμφανίζει το σύνολο των μαθητών που συμμετείχαν πανελλαδικώς στο μάθημα «Αρχές Οικονομικής θεωρίας».

**22.** Να γραφεί αλγόριθμος ο οποίος δημιουργεί και εμφανίζει στην οθόνη τον ακόλουθο πίνακα:

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

**23.** Η στατιστική υπηρεσία του champions league μελετά 30 γνωστούς ποδοσφαιριστές. Σε έναν πίνακα ΟΝ καταγράφει τα ονόματά τους και στον ΕΠΙΔ τις επιδόσεις τους ως εξής: Η πρώτη στήλη καταγράφει τις πάσες, η δεύτερη τα γκολ, η τρίτη τον δείκτη δύναμης, η τέταρτη τα κερδισμένα φάουλ, η πέμπτη τα λάθη και η έκτη το χρόνο συμμετοχής του (0-4000 λεπτά) κάθε αθλητή. Να γραφεί αλγόριθμος ο οποίος:

1. Διαβάζει τους πίνακες (να γίνει έλεγχος στο χρόνο)
2. Βρίσκει τον πρώτο σκόρερ (πιθανότητα ισότητας)
3. Εμφανίζει τον παίκτη με τον καλύτερο συντελεστή-λόγο γκολ/χρόνο συμμετοχής
4. Να εμφανίσει τους 4 καλύτερους παίκτες με βάση το εξής κριτήριο: (γκολ+πάσες-λάθη)/χρόνος συμμετοχής.

**24.** Μία εταιρία δημοσκοπήσεων διεξάγει μία έρευνα σχετικά με τις προτιμήσεις των πολιτών για το ιστορικότερο πρόσωπο του 20ου αιώνα. Η έρευνα γίνεται σε δύο πόλεις και σε αυτή συμμετέχουν από 500 άτομα. Να κάνετε έναν αλγόριθμο, ο οποίος:

- i. Να αποθηκεύει σε έναν πίνακα  $A[500, 2]$ , το πρόσωπο που ψήφισε κάθε ένας από τους πολίτες κάθε πόλης που συμμετείχε στην έρευνα.
- ii. Να αποθηκεύει σε κατάλληλους πίνακες  $\Pi_1, \Psi_1$  και  $\Pi_2, \Psi_2$  και να εμφανίζει ταξινομημένα αλφαβητικά τα πρόσωπα που ψηφίστηκαν στις δύο πόλεις καθώς και τις ψήφους που πήρε το καθένα.
- iii. Να εμφανίζει τα ονόματα των 5 πρώτων (καθώς και όλων όσων ισοψηφούν με τον 5ο).

**25.** Στους αγώνες F1 συμμετέχουν 15 ομάδες με δυο οδηγούς η κάθε μια. Το πρωτάθλημα ολοκληρώνεται μετά από 20 grand prix στα οποία οι 8 πρώτοι βαθμολογούνται με 10,8,6,5,4,3,2 και 1 βαθμό αντίστοιχα, ενώ η κατασκευάστρια εταιρεία παίρνει το άθροισμα των βαθμών των οδηγών της. Γράψτε έναν αλγόριθμο, ο οποίος:

- a. Σε έναν πίνακα ΟΝ  $[15,3]$  να καταχωρεί τα ονόματα των κατασκευαστών και των

οδηγών τους.

β. Σε έναν πίνακα  $K [15,40]$  να καταχωρεί τη θέση που κατέλαβαν οι οδηγοί κάθε ομάδας (στις στήλες 1, 2, 3, 5,..... καταχωρεί την θέση του πρώτου οδηγού και στις υπόλοιπες του δεύτερου).

γ. Να δημιουργεί έναν πίνακα  $B [15,3]$  όπου θα καταχωρούνται οι συνολικοί βαθμοί κάθε οδηγού και της ομάδας του και στη συνέχεια να εμφανίζει το όνομα (ή τα ονόματα) των οδηγών με τις περισσότερες νίκες.

ε. Να εμφανίζει το όνομα του παγκόσμιου πρωταθλητή (όταν δύο οδηγοί έχουν τους ίδιους βαθμούς, νικητής είναι αυτός με τις περισσότερες πρώτες θέσεις).

στ. Να εμφανίζει το όνομα του πρωταθλητή κατασκευαστών (θεωρείστε ότι οι συνολικές βαθμολογίες των ομάδων είναι διαφορετικές).

**26.** Σε ένα σινεμά υπάρχουν 20 σειρές καθισμάτων (Η 1η σειρά είναι η κοντινότερη στην οθόνη) και κάθε σειρά περιλαμβάνει 40 καθίσματα. Να γραφεί αλγόριθμος που:

α) Να διαβάσει τον πίνακα ΚΡΑΤΗΣΕΙΣ (20Χ40) στον οποίο η τιμή 1 ή -1 σημαίνει ότι η αντίστοιχη θέση του σινεμά είναι διαθέσιμη ή μη διαθέσιμη, αντίστοιχα.

β) Όσο υπάρχουν διαθέσιμες θέσεις, να διαβάσει τον αριθμό των εισιτηρίων που επιθυμεί να αγοράσει ο θεατής, ο οποίος θα πρέπει να ελέγχεται ώστε να είναι από 1 μέχρι 5.

γ) Στη συνέχεια, να εμφανίζει όλους τους αριθμούς σειρών στις οποίες υπάρχει διαθέσιμος αριθμός συνεχόμενων καθισμάτων τουλάχιστον ίσος με αυτόν που ζήτησε ο θεατής. Οι σειρές θα πρέπει να εμφανίζονται με αύξουσα σειρά ως προς την απόστασή τους από τη σκηνή.

**27.** Ένα σούπερ μάρκετ έχει 10 ταμεία. Ο διευθυντής του σούπερ μάρκετ θέλει να παρακολουθήσει της πωλήσεις των ταμείων για τους 12 μήνες ενός χρόνου. Να γραφεί πρόγραμμα το οποίο:

α. Να γίνει τμήμα δηλώσεων.

β. Να διαβάσει τις πωλήσεις των ταμείων για τους 12 μήνες του χρόνου και να τις αποθηκεύει στον πίνακα ΠΩΛΗΣΕΙΣ[10,12]. Κατά την καταχώρηση να γίνεται έλεγχος για αρνητικές τιμές.

γ. Να υπολογίζει και να εμφανίζει το άθροισμα των πωλήσεων ανά ταμείο.

δ. Να υπολογίζει και να εμφανίζει το ταμείο με το μεγαλύτερο άθροισμα πωλήσεων.

ε. Να υπολογίζει και να εμφανίζει τους μήνες των οποίων οι συνολικές εισπράξεις είναι μεγαλύτερες από τον μέσο όρο των εισπράξεων όλων των μηνών.

στ. Να υπολογίζει και να εμφανίζει ποια εποχή είχε περισσότερες και ποια λιγότερες εισπράξεις. (Εποχές θεωρούμε τον Χειμώνα, την Άνοιξη, το Καλοκαίρι και το Φθινόπωρο).

**28.** Μια ποδοσφαιρική ομάδα διαθέτει 22 ποδοσφαιριστές, για κάθε έναν από τους οποίους αποθηκεύουμε το ονοματεπώνυμό τους, τα λεπτά συμμετοχής τους και τη θέση στην οποία παίζουν ("Ε" για επίθεση, "Α" για άμυνα, "Κ" για κέντρο και "Τ" όταν ο παίκτης είναι τερματοφύλακας). Να γίνει αλγόριθμος που θα διαβάσει τα παραπάνω δεδομένα και θα εντοπίζει τους παίκτες που έχουν τα περισσότερα λεπτά συμμετοχής για κάθε θέση.

**29.** Μια εταιρεία ασχολείται με εγκαταστάσεις φωτοβολταϊκών συστημάτων, με τα οποία οι πελάτες της έχουν τη δυνατότητα αφενός να παράγουν ηλεκτρική ενέργεια για να καλύπτουν τις ανάγκες της οικίας τους, αφετέρου να πωλούν την πλεονάζουσα ενέργεια προς 0,55€/kWh, εξασφαλίζοντας επιπλέον έσοδα. Η εταιρεία αποφάσισε να ερευνήσει τις εγκαταστάσεις που πραγματοποίησε την προηγούμενη χρονιά σε δέκα (10) πελάτες.

Να αναπτύξετε αλγόριθμο ο οποίος:

- Δ1. Να διαβάζει τα ονόματα των πελατών και να τα αποθηκεύει σε πίνακα ΟΝΟΜΑ[10].
- Δ2. Να διαβάζει το ποσό της ηλεκτρικής ενέργειας σε kWh που παρήγαγαν τα φωτοβολταϊκά συστήματα κάθε πελάτη, καθώς και το ποσό της ηλεκτρικής ενέργειας που κατανάλωσε κάθε πελάτης ανά μήνα του έτους, και να τα αποθηκεύει στους πίνακες Π[10,12] για την παραγωγή και Κ[10,12] για την κατανάλωση αντίστοιχα. Θεωρήστε ότι δεν απαιτείται έλεγχος εγκυρότητας για τα δεδομένα εισόδου.
- Δ3. Με βάση τα στοιχεία του δισδιάστατου πίνακα Π[10,12], να αποθηκεύει σε μονοδιάστατο πίνακα ΕΤΗΣΙΑ\_Π[10] τις ετήσιες αποδόσεις σε kWh για κάθε πελάτη. Με βάση τα στοιχεία του δισδιάστατου πίνακα Κ[10,12], να αποθηκεύει σε μονοδιάστατο πίνακα ΕΤΗΣΙΑ\_Κ[10] τις ετήσιες καταναλώσεις σε kWh που αντιστοιχούν σε κάθε πελάτη.
- Δ4. Σε μονοδιάστατο πίνακα ΕΣΟΔΑ[10] να αποθηκεύει τα ετήσια έσοδα σε ευρώ αν η ετήσια παραγόμενη ηλεκτρική ενέργεια είναι μεγαλύτερη από την ενέργεια που έχει καταναλωθεί για κάθε πελάτη, αλλιώς να αποθηκεύει την τιμή 0.
- Δ5. Να εμφανίζει τα ετήσια έσοδα σε Ευρώ κατά φθίνουσα σειρά.

**30.** Σε ένα Εσπερινό Γυμνάσιο φοιτούν 80 μαθητές. Να γραφεί αλγόριθμος ο οποίος:

- α) Διαβάζει για κάθε μαθητή το ονοματεπώνυμό του, την τάξη του και τον τελικό βαθμό του και τα καταχωρεί σε μονοδιάστατους πίνακες, ελέγχοντας την ορθότητα εισαγωγής των δεδομένων σύμφωνα με τα παρακάτω:
- Οι τάξεις είναι Α ή Β ή Γ.
  - Ο τελικός βαθμός είναι από 1 μέχρι και 20.
- β) Εμφανίζει τα ονόματα των μαθητών της Β τάξης που έχουν τελικό βαθμό μεγαλύτερο ή ίσο του 18,5.
- γ) Υπολογίζει και εμφανίζει το πλήθος των μαθητών κάθε τάξης.
- δ) Υπολογίζει και εμφανίζει το μέσο όρο των τελικών βαθμών των μαθητών της Γ τάξης.
- ε) Εμφανίζει ταξινομημένα κατά αλφαβητική σειρά τα ονοματεπώνυμα και τους αντίστοιχους τελικούς βαθμούς των μαθητών της Α τάξης.

**31.** Σε ένα σχολείο με 100 μαθητές ο καθηγητής της Πληροφορικής έβαλε ένα διαγώνισμα στους μαθητές του. Επιθυμεί να αναπροσαρμόσει τις βαθμολογίες τους τόσο ώστε ο καλύτερος βαθμός να είναι το 20, αυξάνοντας τις βαθμολογίες των μαθητών το πολύ κατά 3 βαθμούς.

Να γράψετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: για κάθε μαθητή θα διαβάζει το όνομα και τη βαθμολογία του, ελέγχοντας αυτή να ανήκει στο διάστημα [0,20].

Θα ελέγχει αν οι βαθμολογίες των μαθητών χρειάζονται αναπροσαρμογή, εμφανίζοντας τα κατάλληλα μηνύματα, και αν χρειάζονται θα την πραγματοποιεί.

Θα εμφανίζει αν ο μαθητής Ζαμπέτας είναι μαθητής του σχολείου ή όχι, και αν είναι θα εμφανίζει αν όντως είναι ο καλύτερος μαθητής, αλλιώς θα εμφανίζει το μήνυμα «Δεν είναι μαθητής του σχολείου».

Θα μπορούσε να λυθεί χωρίς πίνακες; Αιτιολογήστε την απάντησή σας.

**32.** Δίνεται πίνακας ΕΣ 30\*80 ο οποίος περιέχει για κάθε μήνα τα έσοδα 20 εταιρειών για τα έτη 2009, 2010, 2011, 2012. Οι πρώτες 20 στήλες αντιστοιχούν στα έσοδα της πρώτης χρονιάς κ.ο.κ. Δίνεται και ο πίνακας Ον με τα ονόματα των 30 εταιρειών. Να αναπτυχθεί αλγόριθμος ο οποίος:

Α. Να εμφανίζει τα έσοδα της εταιρείας BEST για το μήνα Αύγουστο ανά έτος

Β. Να εμφανίζει για την εταιρεία BEST ταξινομημένα τα έσοδα σε φθίνουσα σειρά για το 2010.

**33.** Σε έναν πίνακα ΟΝ 150 κελιών αποθηκεύουμε τα ονόματα των εργαζομένων μιας εταιρείας. Σε άλλον πίνακα ΩΡΕΣ[150,6] αποθηκεύουμε τις ώρες που εργάστηκε ο κάθε εργαζόμενος από Δευτέρα έως Σάββατο. Να γραφεί πρόγραμμα το οποίο:

α. διαβάζει το πίνακα ΟΝ.

β. γεμίζει τον πίνακα ΩΡΕΣ όπου αποθηκεύει μηδέν αν απουσιάζει ο εργαζόμενος και επιπλέον δεν μπορεί να ξεπερνά τις 11 ώρες ημερησίως.

γ. Αν ο εργαζόμενος πληρώνεται 8 ευρώ την ώρα μέχρι τις 8 ώρες και 11 ευρώ για κάθε επιπλέον, υπολογίστε σε νέο πίνακα ΜΙΣΘΟΔΟΣΙΑ[150] το ποσό που θα πληρωθεί κάθε εργαζόμενος την εβδομάδα και να το εμφανίζει με το όνομά του.

δ. να εμφανίζει το συνολικό ποσό που θα πληρώσει η εταιρεία για ώρες μέχρι το 8ωρο και το συνολικό για υπερωρίες.

ε. να εμφανίζει τους εργαζόμενους που έλειψαν περισσότερες από 3 μέρες την εβδομάδα.

**34.** Να γραφεί πρόγραμμα το οποίο:

A. Διαβάζει τα ονόματα 20 μετοχών στον πίνακα Ον

B. Για κάθε μετοχή διαβάζει την τιμή κλεισίματος για έναν χρόνο στον πίνακα ΑΞΙΑ [20,360] ελέγχοντας να μην αυξομειώνεται πάνω από 12% σε δύο διαδοχικά κλεισίματα

Γ. Δημιουργεί τον πίνακα ΜΕΣΟΣ[20,12] με τις μέσες μηνιαίες τιμές

Δ. Εμφανίζει πόσες μετοχές και ποιες είχαν τη μέγιστη τιμή τους μέσα στο πρώτο τρίμηνο.

**35.** Σε ένα λύκειο της Καστοριάς φοιτούν 270 μαθητές και βαθμολογούνται σε 14 μαθήματα. Να γραφεί πρόγραμμα το οποίο αποθηκεύει σε κατάλληλους πίνακες τα ονόματα και τους βαθμούς.

1. Δημιουργεί τους πίνακες ΠΡΟΑΓΟΝΤΑΙ και ΠΑΡΑΠΕΜΠΟΝΤΑΙ με τα ονόματα των μαθητών που έχουν μέσο όρο τουλάχιστον 9,5 ή κάτω αντίστοιχα.

2. Εμφανίζει το μέσο όρο των προαγόμενων και το μέσο όρο των παραπεμπόμενων.

**36.** Μια εταιρεία παράγει 50 προϊόντα ηλεκτρονικών και τα πουλά σε 20 χώρες. Να γραφεί πρόγραμμα το οποίο:

α. αποθηκεύει την ονομασία κάθε προϊόντος στον πίνακα ΕΙΔΗ[50]

β. αποθηκεύει τα ονόματα των χωρών στον πίνακα ΧΩΡΑ[20]

γ. καταχωρεί τις πωλήσεις του κάθε είδους προϊόντος για κάθε χώρα σε πίνακα ΠΩΛ[50,20].

δ. εμφανίζει σε ποιες και πόσες χώρες πωλείται ένα είδος που θα διαβάζεται από το χρήστη (να ελέγχεται αν υπάρχει το είδος).

ε. θα εμφανίζει τις χώρες που είναι εξαιρετικοί πελάτες (νν έχει αγοράσει τουλάχιστον 10000 κομμάτια από το κάθε προϊόν).

**37.** Για τις 30 μέρες του μήνα Ιουνίου, με τη βοήθεια ειδικών αισθητήρων-καταγραφών καταχωρούμε σε πίνακα ΑΥΤ[30,24] το πλήθος αυτοκινήτων που περνούν από μία λεωφόρο της Αθήνας. Ο αλγόριθμος.

α. Εμφανίζει ποια/ποιες μέρες υπήρξε η μεγαλύτερη κίνηση.

β. Για κάθε μέρα θα επιβεβαιώνει ή όχι η πεποίθηση ότι από τις 9:00 μέχρι τις 15:00 η διέλευση των οχημάτων αυξάνεται 'ώρα με την ώρα.

γ. Για κάθε ώρα θα εμφανίζει τις διελεύσεις των αυτοκινήτων σε αύξουσα σειρά μαζί με την ημέρα που σημειώθηκαν.

**38.** Μια ποδοσφαιρική ομάδα διαθέτει 20 παίκτες και συμμετέχει σε πρωτάθλημα 32 αγωνιστικών.

Να γραφεί αλγόριθμος ο οποίος:

- α. Διαβάζει τα ονόματα των παικτών και τα γκολ που πέτυχαν ανά αγώνα
- β. Εμφανίζει ανά αγώνα τα γκολ που πέτυχε η ομάδα
- γ. Εμφανίζει τον/τους αγώνες με τα περισσότερα γκολ στην επίθεση
- δ. Εμφανίζει τον πρώτο σκόρερ της ομάδας στο πρωτάθλημα.









### Στόχοι:

- ➔ Να κατανοήσετε τις έννοιες του πηγαίου, αντικείμενου και εκτελέσιμου αρχείου.
- ➔ Να κατανοήσετε το ρόλο των διαδικασιών μεταγλώττισης και σύνδεσης.
- ➔ Να αποκτήσετε μια αίσθηση της δομής των προγραμμάτων.
- ➔ Να εκτιμήσετε τη σπουδαιότητα των βιβλιοθηκών.
- ➔ Να κατανοήσετε ότι πολλά απλά προγράμματα αποτελούνται από τρεις φάσεις: είσοδο, υπολογισμό και έξοδο.
- ➔ Να εντοπίζετε συντακτικά και λογικά λάθη.

## 6.1 Θεωρία-Οδηγίες

Κάθε γλώσσα προγραμματισμού έχει το δικό της λεξιλόγιο και τα προγράμματα της ακολουθούν αυστηρούς γραμματικούς και συντακτικούς κανόνες. Για τη δημιουργία σωστών προγραμμάτων είναι απαραίτητη η γνώση των εντολών και του τρόπου σύνταξής τους.

### Ορισμοί:

**Ανάλυση analysis:** Η μεθοδική μελέτη ενός προβλήματος και η διαδικασία της διάσπασης του σε μικρότερες μονάδες για περαιτέρω έρευνα σε λεπτομέρεια.

**Αντικείμενο πρόγραμμα:** Πρόγραμμα υπολογιστή σε μια τελική γλώσσα που έχει μεταφραστεί από πηγαίο πρόγραμμα.

**Γεγονός:** Ενέργεια του χρήστη μιας εφαρμογής που αναγκάζει την εφαρμογή να ανταποκριθεί. Τα γεγονότα συχνά σχετίζονται με κινήσεις του ποντικιού ή πληκτρολογήσεις.

**Γλώσσα μηχανής machine language:** Γλώσσα χαμηλού επιπέδου που οι εντολές της αποτελούνται μόνο από τα δυαδικά ψηφία.

**Γλώσσα προγραμματισμού programming language:** Τεχνητή γλώσσα σχεδιασμένη για να δημιουργεί ή να εκφράζει προγράμματα.

**Διερμηνευτής interpreter:** Πρόγραμμα που μεταφράζει και εκτελεί κάθε εντολή μια γλώσσας προγραμματισμού υψηλού επιπέδου πριν τη μετάφραση και εκτέλεση της επόμενης.

**Δομημένος προγραμματισμός:** Τρεις τύποι ελέγχου χρησιμοποιούνται στο δομημένο προγραμματισμό: ακολουθιακός, υπό συνθήκη και επαναληπτικός.

**Μεταγλωττιστής compiler:** Πρόγραμμα υπολογιστή που χρησιμοποιείται για τη μετάφραση σε γλώσσα χαμηλού επιπέδου ενός προγράμματος εκφρασμένου σε γλώσσα.

**Οδηγούμενο από γεγονότα event driven:** Η ιδιότητα ενός λειτουργικού συστήματος ή περιβάλλοντος, κατά την οποία όταν συμβεί ένα γεγονός εκτελείται κατάλληλο τμήμα κώδικα για την εξυπηρέτησή του.

**Πρόγραμμα (υπολογιστή) program:** Ακολουθία εντολών κατάλληλων για επεξεργασία. Η επεξεργασία περιλαμβάνει τη χρήση μεταφραστικού προγράμματος για να προετοιμάσει το πρόγραμμα για εκτέλεση, καθώς και την ίδια την εκτέλεση του προγράμματος.

**Προγραμματισμός programming:** Η διαδικασία δημιουργίας προγραμμάτων υπολογιστή.

**Συμβολική γλώσσα assembly language:** Γλώσσα χαμηλού επιπέδου εξαρτώμενη από το υλικό και η οποία έχει άμεση αντιστοιχία με τη γλώσσα μηχανής. Αποτελεί συμβολική αναπαράσταση του δυαδικού κώδικα της γλώσσας μηχανής και χρειάζεται συμβολομετάφραση.

**Συμβολομεταφραστής assembler:** Πρόγραμμα που μεταφράζει συμβολική γλώσσα σε γλώσσα μηχανής του δεδομένου υπολογιστή.

**Κώδικας code:** Ένα ή περισσότερα προγράμματα ή τμήμα προγράμματος.

**Περιβάλλον ανάπτυξης λογισμικού:** Σύνολο μεταφραστικών προγραμμάτων και άλλων εργαλείων ανάπτυξης λογισμικού που χρησιμοποιούνται στη δημιουργία προγραμμάτων εφαρμογών.

**Πληροφορία information:** Γνώση που αφορά πράγματα όπως πράξεις, έννοιες, αντικείμενα, γεγονότα, ιδέες ή διεργασίες που μέσα σε συγκεκριμένο κείμενο έχουν μια ιδιαίτερη σημασία.

Η **επίλυση** ενός προβλήματος με τον υπολογιστή περιλαμβάνει τρία εξίσου σημαντικά στάδια.

- ✓ Τον ακριβή προσδιορισμό του προβλήματος.
- ✓ Την ανάπτυξη του αντίστοιχου αλγορίθμου.
- ✓ Τη διατύπωση του αλγορίθμου σε κατανοητή μορφή από τον υπολογιστή.

Ο **προγραμματισμός** ασχολείται με το τρίτο αυτό στάδιο, τη δημιουργία του προγράμματος, δηλαδή του συνόλου των εντολών που πρέπει να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος. Ο προγραμματισμός είναι αυτός που δίνει την εντύπωση ότι, οι υπολογιστές είναι έξυπνες μηχανές που επιλύουν τα πολύπλοκα προβλήματα. Η εντύπωση αυτή όμως είναι απλώς μία ψευδαίσθηση.

Οι γλώσσες προγραμματισμού αναπτύχθηκαν με σκοπό την επικοινωνία του ανθρώπου (προγραμματιστή) με τη μηχανή (υπολογιστή).

Ένα πρόγραμμα σε **γλώσσα μηχανής** είναι μια ακολουθία δυαδικών ψηφίων, που αποτελούν εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες. Οι εντολές σε **συμβολική γλώσσα** αποτελούνται από συμβολικά ονόματα που αντιστοιχούν σε εντολές της γλώσσας μηχανής. Οι συμβολικές γλώσσες είναι συνδεδεμένες με την αρχιτεκτονική κάθε υπολογιστή.

Οι γλώσσες **υψηλού επιπέδου** χρησιμοποιούν ως εντολές απλές λέξεις της αγγλικής γλώσσας ακολουθώντας αυστηρούς κανόνες σύνταξης, οι οποίες μεταφράζονται από τον ίδιο τον υπολογιστή σε εντολές σε γλώσσα μηχανής.

Στα **πλεονεκτήματα** των γλωσσών προγραμματισμού υψηλού επιπέδου σε σχέση με τις συμβολικές μπορούν να αναφερθούν:

1. Ο φυσικότερος και πιο “**ανθρώπινος**” **τρόπος έκφρασης** των προβλημάτων. Τα προγράμματα σε γλώσσα υψηλού επιπέδου είναι πιο κοντά στα προβλήματα που επιλύουν.
2. Η **ανεξαρτησία από** τον τύπο του υπολογιστή. Προγράμματα σε μία γλώσσα υψηλού επιπέδου μπορούν να εκτελεστούν σε οποιονδήποτε υπολογιστή με ελάχιστες ή καθόλου μετατροπές. Η δυνατότητα της **μεταφρασιμότητας** των προγραμμάτων είναι σημαντικό προσόν.
3. Η ευκολία της **εκμάθησης** και εκπαίδευσης ως απόρροια των προηγουμένων.
4. Η **διόρθωση λαθών** και η **συντήρηση** προγραμμάτων σε γλώσσα υψηλού επιπέδου είναι πολύ ευκολότερο έργο.

Συνολικά οι γλώσσες υψηλού επιπέδου ελάττωσαν σημαντικά το **χρόνο** και το **κόστος**

παραγωγής νέων προγραμμάτων, αφού λιγότεροι προγραμματιστές μπορούν σε μικρότερο χρόνο να αναπτύξουν προγράμματα που χρησιμοποιούνται σε περισσότερους υπολογιστές.

Μπορούμε να ισχυριστούμε με βεβαιότητα ότι μία γλώσσα προγραμματισμού που να είναι αντικειμενικά καλύτερη από τις άλλες δεν υπάρχει, ούτε πρόκειται να υπάρξει.

Κάθε γλώσσα προσδιορίζεται από το αλφάβητο της, το λεξιλόγιο της, τη γραμματική της και τη σημασιολογία της.

✓ **Αλφάβητο** μιας γλώσσας καλείται το σύνολο των στοιχείων που χρησιμοποιούνται από τη γλώσσα.

✓ Το **Λεξιλόγιο** αποτελείται από ένα υποσύνολο όλων των ακολουθιών που δημιουργούνται από τα στοιχεία του αλφαβήτου, τις λέξεις που είναι δεκτές από την γλώσσα.

✓ Η **Γραμματική** αποτελείται από το τυπικό ή τυπολογικό και το συντακτικό.

**Τυπικό** είναι το σύνολο των κανόνων που ορίζει τις μορφές με τις οποίες μία λέξη είναι αποδεκτή.

**Συντακτικό** είναι το σύνολο των κανόνων που καθορίζει τη νομιμότητα της διάταξης και της σύνδεσης των λέξεων της γλώσσας για τη δημιουργία προτάσεων.

✓ Η **Σημασιολογία** (Semantics) είναι το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και κατά επέκταση των εκφράσεων και προτάσεων που χρησιμοποιούνται σε μία γλώσσα. Στις γλώσσες προγραμματισμού οι οποίες είναι τεχνητές γλώσσες, ο **δημιουργός** της γλώσσας αποφασίζει τη σημασιολογία των λέξεων της γλώσσας.

Ο **δομημένος** προγραμματισμός στηρίζεται στη χρήση τριών και μόνο στοιχειωδών λογικών δομών, τη δομή της ακολουθίας, τη δομή της επιλογής και τη δομή της επανάληψης. Όλα τα προγράμματα μπορούν να γραφούν χρησιμοποιώντας μόνο αυτές τις τρεις δομές καθώς και συνδυασμό τους. Κάθε πρόγραμμα όπως και κάθε ενότητα προγράμματος έχει μόνο μία είσοδο και μόνο μία έξοδο.

Ο **Δομημένος** Προγραμματισμός αποτελεί μία μεθοδολογία σύνταξης προγραμμάτων που έχει σκοπό να βοηθήσει τον προγραμματιστή στην ανάπτυξη σύνθετων προγραμμάτων. Οι γλώσσες προγραμματισμού εξελίσσονται ενσωματώνοντας νέα χαρακτηριστικά και δυνατότητες, όμως η βασική τους φιλοσοφία εξακολουθεί να στηρίζεται στις αρχές του Δομημένου Προγραμματισμού.

Επιγραμματικά μπορούμε να αναφέρουμε τα εξής **πλεονεκτήματα** του δομημένου προγραμματισμού.

✓ Δημιουργία απλούστερων προγραμμάτων.

✓ Άμεση μεταφορά των αλγορίθμων σε προγράμματα.

✓ Διευκόλυνση ανάλυσης του προγράμματος σε τμήματα.

✓ Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος.

✓ Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους.

✓ Ευκολότερη διόρθωση και συντήρηση.

Εντολή **Goto**: ρητή διακλάδωση σε συγκεκριμένη θέση. Η χρήση της είναι αντίθετη με τις αρχές του δομημένου προγραμματισμού.

Κάθε πρόγραμμα που γράφτηκε σε οποιαδήποτε γλώσσα προγραμματισμού, πρέπει να

μετατραπεί σε μορφή αναγνωρίσιμη και εκτελέσιμη από τον υπολογιστή, δηλαδή σε εντολές **γλώσσας μηχανής**.

Το αρχικό πρόγραμμα λέγεται **πηγαίο πρόγραμμα** (source), ενώ το πρόγραμμα που παράγεται από το μεταγλωττιστή λέγεται **αντικείμενο** πρόγραμμα (object).

Το αντικείμενο πρόγραμμα είναι μεν σε μορφή κατανοητή από τον υπολογιστή, αλλά συνήθως δεν είναι σε θέση να εκτελεστεί. Χρειάζεται να συμπληρωθεί και να συνδεθεί με άλλα τμήματα προγράμματος απαραίτητα για την εκτέλεσή του, τμήματα που είτε τα γράφει ο προγραμματιστής είτε βρίσκονται στις **βιβλιοθήκες** (libraries) της γλώσσας. Το πρόγραμμα που επιτρέπει αυτή τη σύνδεση ονομάζεται **συνδέτης - φορτωτής** (linker loader).

Το αποτέλεσμα του συνδέτη είναι η παραγωγή του **εκτελέσιμου προγράμματος** (executable), το οποίο είναι το τελικό πρόγραμμα που εκτελείται από τον υπολογιστή. Για το λόγο αυτό η συνολική διαδικασία αποκαλείται μεταγλώττιση και σύνδεση.

## **Διερμηνευτής vs μεταγλωττιστής**

### **Ομοιότητες:**

Και οι δύο μεταφράζουν το πηγαίο πρόγραμμα (υψηλού επιπέδου) σε γλώσσα μηχανής.

Και οι δύο ανιχνεύουν τα συντακτικά λάθη.

### **Διαφορές:**

Ο μεταγλωττιστής μεταγλωττίζει όλο το πρόγραμμα και με τη βοήθεια του συνδέτη - φορτωτή παράγεται το εκτελέσιμο.

Ο διερμηνευτής εκτελεί μία μία τις εντολές και δε χρειάζεται συνδέτη-φορτωτή.

Ο διερμηνευτής αφού εκτελεί τις εντολές μία μία έχει το πλεονέκτημα της **άμεσης διόρθωσης** των λαθών. Για το λόγο αυτό χρησιμοποιείται συνήθως κατά την συγγραφή-διόρθωση ενός προγράμματος.

Η εκτέλεση ενός προγράμματος με τον διερμηνευτή είναι πιο αργή, γιατί για να εκτελεστεί το πρόγραμμα, πρέπει κάθε φορά να ξαναγίνεται η διερμηνεία από την αρχή, ενώ ο μεταγλωττιστής παράγει μια φορά το αντικείμενο πρόγραμμα και δεν χρειάζεται ξανά μεταγλώττιση.

Για να εκτελεστεί ένα πρόγραμμα με τον διερμηνευτή είναι απαραίτητη η παρουσία του πηγαίου προγράμματος ενώ με τον μεταγλωττιστή μόνο την πρώτη φορά.

Τα λάθη του προγράμματος είναι γενικά δύο ειδών, **λογικά** και **συντακτικά**. Τα λογικά λάθη εμφανίζονται μόνο στην **εκτέλεση**, ενώ τα συντακτικά λάθη στο στάδιο της **μεταγλώττισης**.

### **Συντακτικά λάθη**

Οφείλονται σε αναγραμματισμούς γραμμάτων εντολών, παράληψη δήλωσης δεδομένων κ.λπ. Εντοπίζονται από τον μεταγλωττιστή ή τον διερμηνευτή. Πρέπει να διορθωθούν ώστε να δημιουργηθεί το εκτελέσιμο πρόγραμμα. Οι γλώσσες προγραμματισμού διαθέτουν ένα σύνολο **συντακτικών κανόνων** (syntax rules), οι οποίοι καθορίζουν αν ένα πρόγραμμα είναι σωστά δομημένο. Ο μεταγλωττιστής ελέγχει το πρόγραμμά σας και αν κάποιος ή κάποιιοι από αυτούς τους συντακτικούς κανόνες παραβιάζονται τότε αναφέρει ένα συντακτικό λάθος. Σε αυτή την περίπτωση τα λάθη θα πρέπει να διορθωθούν και το πρόγραμμα να μεταγλωττιστεί ξανά.

## Λογικά λάθη

Οφείλονται σε σφάλματα κατά την υλοποίηση του αλγορίθμου.

Εντοπίζονται μόνο κατά την εκτέλεση του προγράμματος.

Είναι τα πλέον σοβαρά και δύσκολα στην διόρθωση.

Ο σημαντικότερος τύπος προγραμματιστικού λάθους δεν είναι τα συντακτικά, αλλά τα λάθη που έχουν ως αποτέλεσμα το πρόγραμμα να παράγει λάθος αποτελέσματα ή καθόλου αποτελέσματα. Τα λάθη αυτά, τα οποία εμποδίζουν το πρόγραμμά σας να επιλύσει ένα πρόβλημα εξαιτίας ενός λάθους στη λογική σας, ονομάζονται σφάλματα (bugs) ή **λογικά λάθη**.

Για τη δημιουργία, τη μετάφραση και την εκτέλεση ενός προγράμματος απαιτούνται τουλάχιστον τρία προγράμματα: ο **συντάκτης**, ο **μεταγλωττιστής** και ο **συνδέτης**. Τα σύγχρονα προγραμματιστικά περιβάλλοντα παρέχουν αυτά τα προγράμματα με ενιαίο τρόπο.



## 6.2 Ερωτήσεις Σωστού-Λάθους

1. Ένα πρόγραμμα σε γλώσσα μηχανής είναι μια ακολουθία δυαδικών ψηφίων.
2. Ο μεταγλωττιστής δέχεται στην είσοδό του ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής.
3. Το πηγαίο πρόγραμμα εκτελείται από τον υπολογιστή χωρίς μεταγλώττιση.
4. Ο διερμηνευτής διαβάζει μία προς μία τις εντολές του πηγαίου προγράμματος και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.
5. Ένα πρόγραμμα σε γλώσσα μηχανής χρειάζεται μεταγλώττιση.
6. Οι συμβολικές γλώσσες είναι ανεξάρτητες από την αρχιτεκτονική των υπολογιστών.
7. Η χρήση διερμηνευτή επιτρέπει την άμεση διόρθωση του προγράμματος
8. Σε μια αριθμητική έκφραση ο αριθμός των αριστερών παρενθέσεων ισούται πάντα με τον δεξιών
9. Η χρήση των συμβολικών γλωσσών οδηγεί στη συγγραφή σύντομων προγραμμάτων που είναι εύκολο να συντηρηθούν
10. Η JAVA χρησιμοποιείται ιδιαίτερα για προγραμματισμό στο Διαδίκτυο (Internet).
11. Για την εκτέλεση μιας εντολής συμβολικής γλώσσας απαιτείται η μετάφρασή της σε γλώσσα μηχανής.
12. Ο ιεραρχικός προγραμματισμός χρησιμοποιεί τη στρατηγική συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα.
13. Τα συντακτικά λάθη οδηγούν σε λάθη που παράγονται κατά την εκτέλεση του προγράμματος.
14. Οι εντολές σε γλώσσα μηχανής εκτελούνται αμέσως από τον υπολογιστή.
15. Η FORTRAN αναπτύχθηκε ως γλώσσα κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων.
16. Η εντολή GOTO που αλλάζει τη ροή εκτέλεσης ενός προγράμματος είναι απαραίτητη στο δομημένο προγραμματισμό.
17. Τα συντακτικά λάθη στον πηγαίο κώδικα εμφανίζονται κατά το στάδιο της μεταγλώττισής του.
18. Η JAVA είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών που εκτελούνται σε καταναμημένα περιβάλλοντα, δηλαδή σε διαφορετικούς υπολογιστές οι οποίοι είναι συνδεδεμένοι στο διαδίκτυο.
19. Κατά τη μεταγλώττιση, το εκτελέσιμο πρόγραμμα παράγεται πριν το αντικείμενο.
20. Το αποτέλεσμα του συνδέτη-φορτωτή είναι το πηγαίο πρόγραμμα.
21. Η συγγραφή του πηγαίου προγράμματος και η διόρθωση των λαθών του γίνεται με την βοήθεια του συντάκτη.
22. Στη δομή ενός προγράμματος το τμήμα δήλωσης των σταθερών ακολουθεί το τμήμα δήλωσης των μεταβλητών.
23. Η Fortran είναι μία γλώσσα, που ειδικεύεται στον προγραμματισμό εφαρμογών Τεχνητής Νοημοσύνης.
24. Το αποτέλεσμα του συνδέτη-φορτωτή είναι το πηγαίο πρόγραμμα
25. Το τυπικό μιας γλώσσας είναι το σύνολο των κανόνων που ορίζει τις μορφές με τις οποίες μία λέξη είναι αποδεκτή.
26. Οι γλώσσες μηχανής είναι στενά συνδεδεμένες με την αρχιτεκτονική του υπολογιστή.
27. Η C είναι μία γλώσσα, που ειδικεύεται στον προγραμματισμό συστημάτων.
28. Η γλώσσα ALGOL είναι κατάλληλη για προγραμματισμό σε καταναμημένα περιβάλλοντα.

29. Ο δομημένος προγραμματισμός επιτρέπει την ευκολότερη συντήρηση και διόρθωση ενός προγράμματος.

30. Ο δομημένος προγραμματισμός δεν περιλαμβάνει εντολές τύπου GOTO.

31. Η γλώσσα JAVA είναι 4ης γενιάς.

32. Ο προγραμματισμός με συμβολική γλώσσα χαρακτηρίζεται από μεταφερισιμότητα.

33. Η ανεξέλεγκτη χρήση της εντολής goto οδηγεί στη δημιουργία δυσνόητων προγραμμάτων.

34. Όταν ένας χρήστης διαπιστώσει την ύπαρξη ενός λογικού λάθους σε ένα εκτελέσιμο πρόγραμμα τότε έχει τη δυνατότητα να το διορθώσει μόνος του, ακόμα κι αν δε διαθέτει το αρχικό πηγαίο πρόγραμμα σε γλώσσα υψηλού επιπέδου.

35. Η έλλειψη της δυνατότητας σύνθετων εκφράσεων και εντολών στις γλώσσες χαμηλού επιπέδου, οδηγεί στη δημιουργία μακροσκελών προγραμμάτων.

### Απαντήσεις

1.Σ, 2.Σ, 3.Λ, 4.Σ, 5.Λ, 6.Λ, 7.Σ, 8.Σ, 9.Λ, 10.Σ, 11.Σ, 12.Σ, 13.Λ, 14.Σ, 15.Σ, 16.Λ, 17.Σ, 18.Σ, 19.Λ, 20.Λ, 21.Σ, 22.Λ, 23.Λ, 24.Λ, 25.Σ, 26.Σ, 27.Σ, 28.Λ, 29.Σ, 30.Σ, 31.Λ, 32.Λ, 33.Σ, 34.Λ, 35.Σ.

### Ερωτήσεις Ανάπτυξης

- ✓ Για ποιο λόγο αναπτύχθηκαν οι συμβολικές γλώσσες;
- ✓ Ποιος ο ρόλος του συμβολομεταφραστή;
- ✓ Να εξηγήσετε τι εννοούμε με τον όρο μεταφερισιμότητα των προγραμμάτων.
- ✓ Τι καλείται οπτικός προγραμματισμός;
- ✓ Τι καλείται προγραμματισμός οδηγούμενος από γεγονότα;
- ✓ Τι ονομάζουμε αλφάβητο και τι λεξιλόγιο μίας γλώσσας προγραμματισμού;
- ✓ Τι είναι το τυπικό και το συντακτικό μίας γλώσσας προγραμματισμού;

### Ερωτήσεις πολλαπλής επιλογής

1. Ποια από τις παρακάτω δεν είναι γλώσσα προγραμματισμού υψηλού επιπέδου;

- α) Pascal                    β) Java                    γ) C                    δ) Assembly

2. Ποια ή ποιες από τις παρακάτω κατηγορίες γλωσσών προσφέρουν αυτή τη δυνατότητα στα προγράμματα:

- α. γλώσσες μηχανής                    β. συμβολικές γλώσσες                    γ. γλώσσες υψηλού επιπέδου.

3. Ποιο από τα παρακάτω δεν είναι πλεονέκτημα μίας γλώσσας υψηλού επιπέδου σε σχέση με μία συμβολική γλώσσα;

- α) Ανεξαρτησία από τον τύπο του υπολογιστή.                    β) Ευκολία εκμάθησης.  
γ) Γρηγορότερα προγράμματα.                    δ) Ευκολότερη συντήρηση.

## 6.3 Λυμένες Ασκήσεις

1. Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
σ ← 0
μ ← 1
5: Αν μ > 20 ΤΟΤΕ GOTO 15
α ← 1
10: ΔΙΑΒΑΣΕ λ
α ← α*λ
Αν α < 1890 ΤΟΤΕ GOTO 10
σ ← σ+α
μ ← μ+1
GOTO 5
15: ΓΡΑΨΕ σ
```

### Λύση

```
σ ← 0
μ ← 1
Όσο μ <= 20 ΕΠΑΝΑΛΑΒΕ
α ← 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ λ
α ← α*λ
ΜΕΧΡΙΣ_ΟΤΟΥ α >= 1890
σ ← σ+α
μ ← μ+1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ σ
```

2. Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
A ← 0
10: ΔΙΑΒΑΣΕ X
Αν X <= 0 ΤΟΤΕ GOTO 10
A ← A+X
ΑΝ A <= 10000 ΤΟΤΕ GOTO 10
ΑΝ A < 15000 ΤΟΤΕ GOTO 20
ΑΝ A < 25000 ΤΟΤΕ GOTO 30
ΓΡΑΨΕ 'γ'
GOTO 40
20: ΓΡΑΨΕ 'α'
GOTO 40
30: ΓΡΑΨΕ 'β'
40: ΓΡΑΨΕ 'ΤΕΛΟΣ'
```

### Λύση

$A \leftarrow 0$

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ Χ
ΜΕΧΡΙΣ_ΟΤΟΥ Χ > 0
Α ← Α+Χ
ΜΕΧΡΙΣ_ΟΤΟΥ Α > 10000
Αν Α < 15000 ΤΟΤΕ
  ΓΡΑΨΕ 'α'
ΑΛΛΙΩΣ_Αν Α < 25000 ΤΟΤΕ
  ΓΡΑΨΕ 'β'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'γ'
ΤΕΛΟΣ_Αν
ΓΡΑΨΕ 'ΤΕΛΟΣ'
```

3. Τα παρακάτω σύνολα εντολών φιλοδοξούν να υπολογίσουν το μέσο όρο ύψους 20 ατόμων, ωστόσο έχουν κάποια λάθη. Ποια από αυτά είναι συντακτικά και ποια λογικά;

Για κ από 1 μέχρι 20 (1)

Διάβασε Υ

$\sigma \leftarrow \sigma + Y$

Τέλος\_επανάληψης

Εμφάνισε  $\sigma/20$

$\sigma \leftarrow 0$  (2)

Για κ από 1 μέχρι 20

Διάβασε σ

$\sigma \leftarrow \sigma + Y$

Τέλος\_επανάληψης

Εμφάνισε  $\sigma/20$

$\sigma \leftarrow 0$  (3)

Για κ από 1 μέχρι 20

Διάβασε Υ

$\sigma = \sigma + Y$

Τέλος\_επανάληψης

Εμφάνισε  $\sigma/20$

$\sigma \leftarrow 0$  (4)

Για κ από 1 μέχρι 20 κάνε

Διάβασε Υ

$\sigma \leftarrow \sigma + Y$

Τέλος\_επανάληψης

Εμφάνισε  $\sigma/20$

$\sigma \leftarrow 0$  (5)

Για κ από 1 μέχρι 20

Διάβασε Y

$\sigma \leftarrow \sigma + Y$

Τέλος\_επανάληψης

Εμφάνισε  $\sigma$

**Λύση**

1. Λογικό λάθος, γιατί δε μηδενίσαμε έξω από την εντολή Για το  $\sigma$ .
2. Λογικό λάθος, γιατί αντί να διαβάζουμε την τιμή της Y, διαβάζουμε το  $\sigma$ .
3. Συντακτικό λάθος, γιατί η εντολή  $\sigma = \sigma + Y$  έπρεπε να ήταν  $\sigma \leftarrow \sigma + Y$
4. Συντακτικό λάθος, γιατί η λέξη κάνε είναι περιττή.
5. Λογικό λάθος, γιατί στο τέλος δεν έχουμε διαιρέσει το  $\sigma$  με το 20.

**4.** Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

$\sigma \leftarrow 0$

$d \leftarrow 1$

5: **AN**  $d > 20$  **ΤΟΤΕ** GOTO 30

$\alpha \leftarrow 1$

3: **ΔΙΑΒΑΣΕ** λ

$\alpha \leftarrow \alpha * \lambda$

**AN**  $\alpha < 1890$  **ΤΟΤΕ** GOTO 3

$\sigma \leftarrow \sigma + \alpha$

$d \leftarrow d + 1$

GOTO 5

30: **ΓΡΑΨΕ**  $\sigma$





### Στόχοι:

- ➔ Να εκτιμήσετε τις συναρτήσεις ως εργαλείο απλοποίησης της δομής των προγραμμάτων.
- ➔ Να κατανοήσετε την έννοια της κλήσης μιας συνάρτησης και το λόγο για τον οποίο μεταβιβάζονται παράμετροι κατά την κλήση.
- ➔ Να κατανοήσετε τα πρωτότυπα των συναρτήσεων και πώς να τα γράφετε.
- ➔ Να είστε σε θέση να υλοποιείτε απλές συναρτήσεις.
- ➔ Να κατανοήσετε τη σχέση μεταξύ των τυπικών παραμέτρων μιας συνάρτησης και των πραγματικών που μεταβιβάζονται κατά την κλήση της.
- ➔ Να καταλάβετε τη σημασία του όρου διαδικασία.

## 7.1 Θεωρία-Οδηγίες

Έχοντας πλέον εξασκηθεί σε πολλά σημεία της αλγοριθμικής επίλυσης προβλημάτων μπορείτε να αναγνωρίσετε τότε μία **ενέργεια**:

- ✓ **επαναλαμβάνεται** ώστε να χρησιμοποιήσετε τη **δομή επανάληψης** ή
- ✓ **εφαρμόζεται** σε πολλά **όμοια δεδομένα** ώστε να χρησιμοποιήσετε **πίνακες**.

Προχωρώντας σε πιο πολύπλοκα προβλήματα, θα δείτε ότι συχνά κάποια **ενέργεια** (ή και ολόκληρο αλγοριθμικό κομμάτι), επαναλαμβάνεται σχεδόν αυτούσια σε πολλά **διαφορετικά σημεία** του αλγόριθμου, αφού επεξεργάζεται **διαφορετικά δεδομένα** αλλά με τον ίδιο ακριβώς τρόπο. "Επαναλαμβάνεται" δηλαδή, αλλά με τρόπο που δεν ταιριάζει ώστε να χρησιμοποιηθεί απλά η δομή επανάληψης (με ή χωρίς τη χρήση πίνακα). Σε τέτοιες περιπτώσεις δεν είναι καλή λύση να ξαναγράψουμε το ίδιο "αλγοριθμικό κομμάτι" σε όλα τα σημεία που χρειάζεται.

Θα μάθετε πώς μπορείτε να αντιμετωπίσετε συστηματικά και αποτελεσματικά τέτοια προβλήματα εντοπίζοντας σε αυτά αυτόνομες λειτουργίες και κατασκευάζοντας τα αντίστοιχα αλγοριθμικά κομμάτια που θα τα ονομάσουμε **υποπρογράμματα**. Θα δείτε πώς μπορείτε να χρησιμοποιήσετε (καλέσετε) αυτά τα αλγοριθμικά κομμάτια σε πολλά διαφορετικά σημεία του αλγόριθμου χωρίς να τα ξαναγράψετε.

Η **βιβλιοθήκη** (library) είναι μια συλλογή εργαλείων που έχουν γραφτεί από άλλους προγραμματιστές προκειμένου να εκτελούν συγκεκριμένες λειτουργίες. Οι βιβλιοθήκες είναι πολύ σημαντικές για τον προγραμματισμό, γιατί μας δίνουν τη δυνατότητα να χρησιμοποιούμε τα εργαλεία που περιλαμβάνονται σε αυτές και μας απαλλάσσουν από τον κόπο να τα γράφουμε μόνοι μας.

**Τμηματικός** προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.

Για να αποφύγεις τα πλέον κοινά λάθη να **προσέχεις** ιδιαίτερα:

- ✓ Πριν ξεκινήσεις να γράφεις το πρόγραμμα σου, να μελετήσεις πώς το πρόγραμμα μπορεί να αναλυθεί σε επιμέρους τμήματα και να αποφασίσεις για τα αντίστοιχα υποπρογράμματα. Χρήσιμο είναι να κάνεις ένα διάγραμμα που θα δείχνει την ιεραρχία ανάμεσα στα υποπρογράμματα, τότε να αναπτύσσεις τους αλγόριθμους για το κάθε υποπρόγραμμα και στη συνέχεια να γράφεις το πρόγραμμα.



- ✓ Να μελετάς αν ένα υποπρόγραμμα πρέπει να υλοποιηθεί με διαδικασία ή μπορεί να υλοποιηθεί με συνάρτηση.
- ✓ Εξέτασε αν κάποια υποπρογράμματα τα οποία έχεις ήδη γράψει ή υπάρχουν σε έτοιμες βιβλιοθήκες προγραμμάτων μπορούν να χρησιμοποιηθούν. Θα γλιτώσεις χρόνο και κόπο.
- ✓ Κάθε υποπρόγραμμα να προσπαθείς να είναι όσο το δυνατόν πιο ανεξάρτητο από τα άλλα. Αυτό σε προφυλάσσει από λάθη στο πρόγραμμα σου και σου επιτρέπει τη χρήση του σε άλλα προγράμματα αργότερα.
- ✓ Να ορίζεις τον τύπο της συνάρτησης. Οι συναρτήσεις παράγουν μόνο ένα αποτέλεσμα συγκεκριμένου τύπου, ακεραίου, πραγματικού κ.λπ. που πρέπει να ορίζεται.
- ✓ Να μην υπάρχουν λάθη στην αντιστοίχιση τυπικών και πραγματικών παραμέτρων. Πρόσεξε ότι οι λίστες πρέπει να περιέχουν τον ίδιο αριθμό παραμέτρων και κάθε τυπική παράμετρος με την αντίστοιχη πραγματική πρέπει να είναι του ίδιου τύπου.

Οι μαθητές ενθαρρύνονται να χρησιμοποιούν υποπρογράμματα. Ιδιαίτερη έμφαση πρέπει να δοθεί στη σωστή ανάλυση του προγράμματος σε υποπρογράμματα και στην επιλογή του κατάλληλου τύπου συνάρτησης ή διαδικασίας. Η ανάλυση αυτή πρέπει να γίνεται πάντα πριν αρχίσουν να γράφουν το πρόγραμμα. Πρέπει να αποφασίζουν για κάθε υποπρόγραμμα το είδος του, τη λειτουργία του, καθώς και τον αριθμό των παραμέτρων που χρειάζεται.

Όταν ένα τμήμα προγράμματος επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα, τότε αναφερόμαστε σε υποπρόγραμμα (subprogram) Υπάρχουν πάντως τρεις **ιδιότητες** που πρέπει να διακρίνουν τα υποπρογράμματα:

- ✓ Κάθε υποπρόγραμμα έχει μόνο **μία είσοδο** και **μία έξοδο**. Στην πραγματικότητα κάθε υποπρόγραμμα ενεργοποιείται με την είσοδο σε αυτό, που γίνεται πάντοτε από την αρχή του, εκτελεί ορισμένες ενέργειες, και απενεργοποιείται με την έξοδο από αυτό που γίνεται πάντοτε από το τέλος του.
- ✓ Κάθε υποπρόγραμμα πρέπει να είναι **ανεξάρτητο** από τα άλλα. Αυτό σημαίνει ότι κάθε υποπρόγραμμα μπορεί να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί αυτόνομα χωρίς να επηρεαστούν άλλα υποπρογράμματα. Στην πράξη βέβαια η απόλυτη ανεξαρτησία είναι δύσκολο να επιτευχθεί.
- ✓ Κάθε υποπρόγραμμα πρέπει να **μην είναι πολύ μεγάλο**. Η έννοια του μεγάλου προγράμματος είναι υποκειμενική, αλλά πρέπει κάθε υποπρόγραμμα να είναι τόσο, ώστε να είναι εύκολα κατανοητό για να μπορεί να ελέγχεται. Γενικά κάθε υποπρόγραμμα πρέπει να εκτελεί μόνο μία λειτουργία. αν εκτελεί περισσότερες λειτουργίες, τότε συνήθως μπορεί και πρέπει να διασπαστεί σε ακόμη μικρότερα υποπρογράμματα.

## Πλεονεκτήματα τμηματικού προγραμματισμού

### 1. Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντιστοίχου προγράμματος.

Επιτρέπει την εξέταση και την επίλυση απλών προβλημάτων και όχι την αντιμετώπιση του συνολικού προβλήματος. Με τη σταδιακή επίλυση των υποπροβλημάτων και τη δημιουργία των αντιστοιχών υποπρογραμμάτων τελικά επιλύεται το συνολικό πρόβλημα.

### 2. Διευκολύνει την κατανόηση και διόρθωση του προγράμματος.

Ο χωρισμός του προγράμματος σε μικρότερα αυτοτελή τμήματα επιτρέπει τη γρήγορη διόρθωση ενός συγκεκριμένου τμήματος του, χωρίς οι αλλαγές αυτές να επηρεάσουν όλο το υπόλοιπο πρόγραμμα. Επίσης διευκολύνει οποιοδήποτε χρειαστεί να διαβάσει και να

κατανοήσει τον τρόπο που λειτουργεί το πρόγραμμα. Όπως έχει πολλές φορές τονιστεί, αυτό είναι πολύ σημαντικό χαρακτηριστικό του σωστού προγραμματισμού, αφού ένα μεγάλο πρόγραμμα στον κύκλο της ζωής του χρειάζεται να συντηρηθεί από διαφορετικούς προγραμματιστές.

### 3. Απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος.

Πολύ συχνά χρειάζεται η ίδια λειτουργία σε διαφορετικά σημεία ενός προγράμματος. Από τη στιγμή που ένα υποπρόγραμμα έχει γραφεί, μπορεί το ίδιο να καλείται από πολλά σημεία του προγράμματος. Έτσι μειώνονται το μέγεθος του προγράμματος, ο χρόνος που απαιτείται για τη συγγραφή του και οι πιθανότητες λάθους, ενώ ταυτόχρονα το πρόγραμμα γίνεται πιο εύληπτο και κατανοητό.

### 4. Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.

Ένα υποπρόγραμμα που έχει γραφεί μπορεί να χρησιμοποιηθεί πολύ εύκολα και σε άλλα προγράμματα. Από τη στιγμή που έχει δημιουργηθεί, η χρήση του δεν διαφέρει από τη χρήση των ενσωματωμένων συναρτήσεων που παρέχει η γλώσσα προγραμματισμού, όπως για τον υπολογισμό του ημίτονου ή του συνημίτονου ή την εντολή με την οποία εκτελεί μία συγκεκριμένη διαδικασία, για παράδειγμα γράφει στην οθόνη (εντολή ΓΡΑΨΕ). Αν λοιπόν χρειάζεται συχνά κάποια λειτουργία που δεν υποστηρίζεται απευθείας από τη γλώσσα, όπως για παράδειγμα η εύρεση του μικρότερου δύο αριθμών, τότε μπορεί να γραφεί το αντίστοιχο υποπρόγραμμα. Η συγγραφή πολλών υποπρογραμμάτων και η δημιουργία βιβλιοθηκών με αυτά, ουσιαστικά επεκτείνουν την ίδια τη γλώσσα προγραμματισμού.

Οι **παράμετροι** είναι σαν τις κοινές μεταβλητές ενός προγράμματος με μία ουσιαστική διαφορά, χρησιμοποιούνται για να περνούν τιμές στα υποπρογράμματα. Μία παράμετρος είναι μία μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε ένα άλλο.

Υπάρχουν δύο είδη υποπρογραμμάτων: οι **διαδικασίες και οι συναρτήσεις**. Κάθε είδος υποπρογράμματος καθορίζεται από το είδος της λειτουργίας που καλείται να επιτελέσει, ενώ ο τρόπος κλήσης και σύνταξης των δύο αυτών ειδών είναι διαφορετικός.

Οι **διαδικασίες** μπορούν να εκτελέσουν οποιαδήποτε λειτουργία από αυτές που μπορεί να εκτελέσει ένα πρόγραμμα. Να εισάγουν δεδομένα, να εκτελέσουν υπολογισμούς, να μεταβάλλουν τις τιμές των μεταβλητών και να τυπώσουν αποτελέσματα. Με τη χρήση των παραμέτρων, αυτές τις τιμές μπορούν να τις μεταφέρουν και στα άλλα υποπρογράμματα. Η εντολή **ΚΑΛΕΣΕ** μας επιτρέπει να καλέσουμε μία διαδικασία (όχι όμως συναρτήσεις, αυτές καλούνται μέσα από εκφράσεις). Οι διαδικασίες είναι υποπρογράμματα που γράφονται μετά το κυρίως πρόγραμμα. Μπορούν να κληθούν με την εντολή **ΚΑΛΕΣΕ** από οποιοδήποτε σημείο του προγράμματος. Οι παράμετροι αναφέρονται ονομαστικά κατά τη δήλωση της διαδικασίας και στη συνέχεια ο τύπος τους δηλώνεται στο τμήμα «Μεταβλητές».

Σαν παράμετρος μπορεί να περαστεί οτιδήποτε έχει τιμή, όπως μεταβλητές, πίνακες, σταθερές, εκφράσεις.

Αν μία διαδικασία κληθεί με μεταβλητή σαν παράμετρο και αλλαχθεί η τιμή της μέσα στην διαδικασία, τότε η τιμή της μεταβλητής στο καλών υποπρόγραμμα θα ενημερωθεί (αλλαχθεί) μετά την εκτέλεση της εντολής **ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**.

Αν θέλουμε να περάσουμε σε διαδικασία κάποια μεταβλητή-πραγματική παράμετρο χωρίς να επιστραφεί αλλαγμένη, τότε πρέπει να κάνουμε κάποια «πράξη» με αυτήν ώστε να πάψει να είναι μεταβλητή (π.χ.  $1 \cdot \chi$ ,  $\chi + 0$ ).

Αντίθετα η λειτουργία των **συναρτήσεων** είναι πιο περιορισμένη. Οι συναρτήσεις υπολογίζουν μόνο μία τιμή, αριθμητική, χαρακτήρα ή λογική και μόνο αυτήν επιστρέφουν στο υποπρόγραμμα που την κάλεσε. Η συνάρτηση είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μία τιμή με το όνομά της (όπως οι μαθηματικές συναρτήσεις). Η ΓΛΩΣΣΑ επιτρέπει τη δημιουργία συναρτήσεων από το χρήστη, οι οποίες μπορούν στη συνέχεια να χρησιμοποιηθούν όπως και οι **ενσωματωμένες** συναρτήσεις. Το αποτέλεσμα κάθε συνάρτησης έχει πάντα τιμή κάποιου τύπου δεδομένων και δηλώνεται μετά τις παραμέτρους της συνάρτησης (ΑΚΕΡΑΙΑ, ΠΡΑΓΜΑΤΙΚΗ, ΧΑΡΑΚΤΗΡΑΣ ή ΛΟΓΙΚΗ). Οι συναρτήσεις δεν επιτρέπεται να επιστρέφουν πίνακες. Όλες οι συναρτήσεις πρέπει να περιέχουν μία εντολή του τύπου ΌνομαΣυνάρτησης ← τιμή, ώστε να επιστρέφουν κάποια τιμή στο καλών υποπρόγραμμα.

Οι παράμετροι στις συναρτήσεις περνιούνται έτσι ώστε οποιαδήποτε αλλαγή στις τυπικές παραμέτρους μιας συνάρτησης δεν επηρεάζει τις πραγματικές παραμέτρους της συνάρτησης. Μια συνάρτηση που αλλάζει το περιεχόμενο των παραμέτρων της δεν επιστρέφει τις αλλαγές αυτές στις πραγματικές παραμέτρους.

Η λίστα των **τυπικών** παραμέτρων καθορίζει τις παραμέτρους στη δήλωση του υποπρογράμματος. Η λίστα των **πραγματικών παραμέτρων** καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.

#### **Κανόνες περάσματος παραμέτρων**

1. Ο αριθμός των πραγματικών και των τυπικών παραμέτρων πρέπει να είναι ίδιος.
2. Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση. Για παράδειγμα, η πρώτη της λίστας των τυπικών παραμέτρων στην πρώτη της λίστας των πραγματικών παραμέτρων κ.ο.κ.
3. Η τυπική παράμετρος και η αντίστοιχη της πραγματική πρέπει να είναι του ίδιου τύπου.

Όλες οι μεταβλητές είναι γνωστές, έχουν ισχύ όπως λέγεται, μόνο για το τμήμα προγράμματος στο οποίο έχουν δηλωθεί, ισχύουν δηλαδή **τοπικά** για το συγκεκριμένο υποπρόγραμμα ή κυρίως πρόγραμμα.

**Επιστροφόμενη τιμή:** η πληροφορία που επιστρέφει μια συνάρτηση όταν επιστρέφει τον έλεγχο ροής-εκτέλεσης στο σημείο από το οποίο κλήθηκε (διεύθυνση επιστροφής).

**Επικεφαλίδα συνάρτησης-διαδικασίας:** προσδιορίζει τη διεπαφή της συνάρτησης και άρα τον τρόπο αναφοράς-χρήσης του υποπρογράμματος.

Αν στις πραγματικές παραμέτρους υπάρχει συγκεκριμένη θέση του πίνακα, τότε η αντίστοιχη τυπική παράμετρος πρέπει να είναι μεταβλητή.

Άρα, ένα υποπρόγραμμα δεν διαβάσει, ούτε εμφανίζει στην οθόνη, εκτός αν ορίζεται ρητά στην εκφώνηση.

Αν το αποτέλεσμα είναι ένα, μπορεί να υλοποιηθεί με Συνάρτηση η οποία "κουβαλάει" η ίδια το αποτέλεσμα.

Σε Διαδικασία, όσα και να είναι τα αποτελέσματα, είναι παράμετροι.

Οι μεταβλητές πάντα δηλώνονται τοπικά (εμβέλεια) και όλες, άρα οι μεταβλητές του προγράμματος δηλώνονται μόνο σε αυτό και των υποπρογραμμάτων μόνο σε αυτά.

Η έννοια της **στοίβας** είναι πολύ χρήσιμη στο ίδιο το λογισμικό των γλωσσών προγραμματισμού. Όταν μία διαδικασία ή συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η αμέσως επόμενη διεύθυνση του κύριου προγράμματος, που ονομάζεται **διεύθυνση επιστροφής** (return address), αποθηκεύεται από το μεταφραστή σε μία στοίβα που ονομάζεται **στοίβα χρόνου εκτέλεσης** (execution time stack). Μετά την εκτέλεση της διαδικασίας ή της συνάρτησης, η διεύθυνση επιστροφής απωθείται από τη στοίβα και έτσι ο έλεγχος του προγράμματος μεταφέρεται και πάλι στο κύριο πρόγραμμα. Η **τεχνική** αυτή εφαρμόζεται και γενικότερα, δηλαδή οποτεδήποτε μία διαδικασία ή συνάρτηση καλεί μία διαδικασία ή συνάρτηση.

## 7.2 Ερωτήσεις Σωστού - Λάθους

1. Σε μια συνάρτηση μπορούμε να χρησιμοποιήσουμε την εντολή ΓΡΑΨΕ, ενώ σε μια διαδικασία όχι.
2. Κάθε υποπρόγραμμα θα πρέπει να είναι ανεξάρτητο από άλλα υποπρογράμματα.
3. Μία συνάρτηση μπορεί να δεχτεί μία μόνο παράμετρο.
4. Ένα υποπρόγραμμα μπορεί κατά την εκτέλεσή του να καλέσει το κυρίως πρόγραμμα.
5. Τα υποπρογράμματα ενεργοποιούνται από μόνα τους.
6. Στον κώδικα υλοποίησης μιας Συνάρτησης, το όνομά της πρέπει οπωσδήποτε να βρίσκεται τουλάχιστον μια φορά στο αριστερό τμήμα εντολής εκχώρησης.
7. Ένα υποπρόγραμμα δεν μπορεί να κληθεί περισσότερες από δυο φορές από το κυρίως πρόγραμμα.
8. Μια συνάρτηση υπολογίζει και επιστρέφει παραπάνω από μία τιμές με το όνομά της.
9. Ένα υποπρόγραμμα μπορεί να καλείται από ένα άλλο υποπρόγραμμα ή από το κύριο πρόγραμμα.
10. Ο τρόπος κλήσης των διαδικασιών και των συναρτήσεων είναι ίδιος, ενώ ο τρόπος σύνταξής τους είναι διαφορετικός.
11. Κατά την κλήση ενός υποπρογράμματος, η πραγματική παράμετρος και η αντίστοιχη τυπική της είναι δυνατόν να έχουν το ίδιο όνομα
12. Όταν ένα υποπρόγραμμα καλείται από διαφορετικά σημεία του προγράμματος, οι πραγματικές παράμετροι πρέπει να είναι οι ίδιες.
13. Ο αριθμός των τυπικών παραμέτρων κατά την κλήση ενός υποπρογράμματος πρέπει να είναι ίδιος με τον αριθμό των πραγματικών παραμέτρων στη δήλωση ενός προγράμματος.
14. Η λίστα των πραγματικών παραμέτρων καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.
15. Μία συνάρτηση μπορεί να δεχτεί μία μόνο παράμετρο.
16. Η στοιβία χρόνου εκτέλεσης δημιουργείται από το μεταγλωττιστή.
17. Κατά την εκτέλεση μιας εντολής εκχώρησης του κύριου προγράμματος σε ΓΛΩΣΣΑ, η στοιβία χρόνου εκτέλεσης είναι κενή.
18. Κατά την κλήση μιας διαδικασίας οι πραγματικές παράμετροι μπορούν να είναι μεταβλητές μόνο πραγματικού τύπου.
19. Μια συνάρτηση μπορεί να ενεργοποιηθεί μέσα από μια εντολή εκχώρησης ή μια εντολή ΓΡΑΨΕ.
20. Κάθε διαδικασία μπορεί να υλοποιηθεί και ως συνάρτηση.
21. Η παράμετρος είναι μια ιδιαίτερη περίπτωση μεταβλητής.
22. Μια συνάρτηση μπορεί να επιτελέσει οποιαδήποτε λειτουργία.
23. Όλα τα υποπρογράμματα ενεργοποιούνται με τον ίδιο τρόπο.
24. Το όνομα μιας διαδικασίας υπόκειται στους κανόνες ονοματολογίας των μεταβλητών και των συμβολικών σταθερών.
25. Πραγματικές ονομάζονται οι παράμετροι που χρησιμοποιούνται κατά την κλήση ενός υποπρογράμματος
26. Όταν σ' ένα υποπρόγραμμα έχουμε παράμετρο πίνακα, τότε η τυπική και η αντίστοιχη πραγματική παράμετρος, πρέπει να είναι ίδιου τύπου και ίδιου μεγέθους
27. Κατά την κλήση ενός υποπρογράμματος, η πραγματική παράμετρος και η αντίστοιχη τυπική της είναι δυνατόν να έχουν το ίδιο όνομα.
28. Μία συνάρτηση μπορεί να εμφανίσει στην οθόνη την τιμή που υπολόγισε.

**29.** Μία διαδικασία επιστρέφει στο πρόγραμμα που την έχει καλέσει τις τιμές όλων των παραμέτρων της.

**30.** Η συνάρτηση είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελέσει όλες τις λειτουργίες ενός προγράμματος.

**31.** Κατά την κλήση των υποπρογραμμάτων απαιτείται μία στοίβα, η λεγόμενη «στοίβα χρόνου εκτέλεσης» του προγράμματος.

**32.** Την ώρα που εκτελούνται οι εντολές ενός υποπρογράμματος, η στοίβα χρόνου εκτέλεσης είναι κενή.

### **Απαντήσεις**

1.Λ, 2.Σ, 3.Λ, 4.Λ, 5.Λ, 6.Σ, 7.Λ, 8.Λ, 9.Σ, 10.Λ, 11.Σ, 12.Λ, 13.Σ, 14.Σ, 15.Λ, 16.Σ, 17.Σ, 18.Λ, 19.Σ, 20.Λ, 21.Σ, 22.Λ, 23.Λ, 24.Σ, 25.Σ, 26.Σ, 27.Σ, 28.Λ, 29.Σ, 30.Λ, 31.Σ, 32.Λ

## 7.3 Λυμένες Ασκήσεις

1. Ας θεωρήσουμε την παρακάτω λανθασμένη συνάρτηση:

**ΣΥΝΑΡΤΗΣΗ**  $APX(\kappa, \lambda)$  : ΠΡΑΓΜΑΤΙΚΕΣ

**ΣΤΑΘΕΡΕΣ**

$\mu \leftarrow 3.2$

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $\kappa$

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $\lambda$

**ΑΡΧΗ**

$APX \leftarrow \mu$

**Αν**  $\lambda > \kappa$  **ΤΟΤΕ**

$APX \leftarrow APX + (\lambda - \kappa)$

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

Διορθώστε τα λάθη και ξαναγράψτε τη συνάρτηση από την αρχή.

**Λύση**

**ΣΥΝΑΡΤΗΣΗ**  $APX(\kappa, \lambda)$  : ΠΡΑΓΜΑΤΙΚΗ

**ΣΤΑΘΕΡΕΣ**

$\mu = 3.2$

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $\kappa$

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $\lambda, TEMP$

**ΑΡΧΗ**

$TEMP \leftarrow \mu$

**Αν**  $\lambda > \kappa$  **ΤΟΤΕ**

$TEMP \leftarrow TEMP + (\lambda - \kappa)$

**ΤΕΛΟΣ\_Αν**

$APX \leftarrow TEMP$

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

2. α) Να γραφεί συνάρτηση η οποία δέχεται ένα χαρακτήρα και ελέγχει αν είναι φωνήεν ή όχι και επιστρέφει αυτή την πληροφορία.

β) Να γραφεί πρόγραμμα το οποίο δέχεται άγνωστο πλήθος χαρακτήρων και υπολογίζει:

i) το συνολικό αριθμό των φωνηέντων που πληκτρολογήθηκαν χρησιμοποιώντας την παραπάνω συνάρτηση

ii) το πλήθος των λέξεων που πληκτρολογήθηκαν

Το πρόγραμμα θα τερματίζει όταν δοθεί ως χαρακτήρας η τελεία και θα εμφανίζει το πλήθος των λέξεων που πληκτρολογήθηκαν και το ποσοστό των φωνηέντων σε σχέση με όλους τους άλλους χαρακτήρες που πληκτρολογήθηκαν.

**Λύση**

**Πρόγραμμα** Κειμενογράφος

**Μεταβλητές**

Ακέραιες: Πλήθος\_φωνηέντων, Πλήθος\_κενών, Πλήθος

**Πραγματικές** : Ποσοστό

**Χαρακτήρες**: X

**Λογική**: Φ

**Αρχή**

Πλήθος\_φωνηέντων  $\leftarrow 0$

Πλήθος\_κενών  $\leftarrow 0$

Πλήθος  $\leftarrow 0$

**Διάβασε X**

**Όσο X < > ' ' επανάλαβε**

Φ  $\leftarrow$  Φωνήεν(X)

**Αν Φ = αληθής τότε**

Πλήθος\_φωνηέντων  $\leftarrow$  Πλήθος\_φωνηέντων + 1

**Τέλος\_Αν**

**Αν X = ' ' τότε**

Πλήθος\_κενών  $\leftarrow$  Πλήθος\_κενών + 1

**Τέλος\_Αν**

Πλήθος  $\leftarrow$  Πλήθος + 1

**Διάβασε X**

**Τέλος\_επανάληψης**

**Γράψε** 'Το πλήθος των λέξεων που πληκτρολογήθηκαν είναι', Πλήθος\_κενών

Ποσοστό\_φωνηέντων  $\leftarrow$  Πλήθος\_φωνηέντων \* 100 / Πλήθος

**Γράψε** 'Το ποσοστό των φωνηέντων που πληκτρολογήθηκαν είναι',

Ποσοστό\_φωνηέντων

**Τέλος\_Προγράμματος**

**Συνάρτηση Φωνήεν(X): Λογική**

Μεταβλητές

**Χαρακτήρες**: X

**Αρχή**

**Αν** X="α" ή X="ο" ή X="ι" ή X="η", ή X="υ" ή X="ω" ή X="ε" ή X="Α" ή X="Ο" ή X="Ι", ή X="Η,  
ή X="Υ, ή X="Ω, ή X="Ε" **τότε**

Φωνήεν  $\leftarrow$  αληθής

**αλλιώς**

Φωνήεν  $\leftarrow$  ψευδής

**Τέλος\_Αν**

**Τέλος\_Συνάρτησης**

**3.** Γράψτε συνάρτηση που δέχεται ως παράμετρο τον ακέραιο πίνακα A[76, 76] και υπολογίζει το αποτέλεσμα της παράστασης:

$(MAX+(MIN-MO)) / 2$  όπου:

MAX: η μέγιστη τιμή της κύριας διαγωνίου,

MIN: η ελάχιστη τιμή της 76ης γραμμής,

MO: ο μέσος όρος της 76ης στήλης.

**Λύση**

**ΣΥΝΑΡΤΗΣΗ ΥΠΟΛ(Α) : ΠΡΑΓΜΑΤΙΚΗ  
ΜΕΤΑΒΛΗΤΕΣ**



**ΑΚΕΡΑΙΕΣ:** A[76, 76], κ, σ, MAX, MIN  
**ΠΡΑΓΜΑΤΙΚΕΣ:** ΜΟ

**ΑΡΧΗ**

MAX ← A[1, 1]

**Για κ ΑΠΟ 2 ΜΕΧΡΙ 76**

**Αν** A[κ, κ] > MAX **ΤΟΤΕ**

MAX ← A[κ, κ]

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

MIN ← A[76, 1]

**Για κ ΑΠΟ 2 ΜΕΧΡΙ 76**

**Αν** A[76, κ] < MIN **ΤΟΤΕ**

MIN ← A[76, κ]

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

σ ← 0

**Για κ ΑΠΟ 1 ΜΕΧΡΙ 76**

σ ← σ+A[κ, 76]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΜΟ ← σ/76

ΥΠΟΛ ← (MAX+(MIN-ΜΟ))/2

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**4.** Ένα μαγαζί ηλεκτρονικών ειδών παρέχει στους πελάτες του τη δυνατότητα προμήθειας παιχνιδιών για υπολογιστές, χρεώνοντας το κάθε παιχνίδι με τον ακόλουθο τρόπο: τα πρώτα 5 παιχνίδια κοστολογούνται με 10 ευρώ και κάθε άλλο επιπλέον PC game κοστολογείται με 8 ευρώ.

I. Γράψτε συνάρτηση που δέχεται ως παράμετρο το πλήθος των PC games που θέλει να αγοράσει ένας πελάτης και υπολογίζει το συνολικό κόστος της αγοράς του.

II. Γράψτε πρόγραμμα το οποίο, για καθένα από τους 23 πελάτες που εκμεταλλεύτηκαν την προσφορά, διαβάζει το πλήθος των PC games που αγόρασε, καλώντας τη συνάρτηση που γράψατε στο I, υπολογίζει το κόστος της αγοράς του και εμφανίζει το κόστος αυτό με κατάλληλα διαμορφωμένο μήνυμα. Το πρόγραμμα επιπρόσθετα υπολογίζει και εμφανίζει: τη συνολική είσπραξη από την πώληση των PC games στους παραπάνω πελάτες και πόσοι πελάτες ξόδεψαν περισσότερα από 80 ευρώ για την αγορά PC games.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ ΜΑΓΑΖΙ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** κ, Φ, σ, Πληθ, Κοστ

**ΑΡΧΗ**

σ ← 0

Φ ← 0

**Για κ ΑΠΟ 1 ΜΕΧΡΙ 23**

**ΔΙΑΒΑΣΕ** Πληθ

Κοστ ← ΚΟΣΤΟΣ(Πληθ)

**ΓΡΑΨΕ** 'Κόστος πελάτη =', Κοστ

σ ← σ+Κοστ

```

    Αν Κοστ > 80 ΤΟΤΕ
    Φ ← Φ+1
    ΤΕΛΟΣ_Αν
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Συνολική είσπραξη =', σ
ΓΡΑΨΕ Φ, ' πελάτες ξόδεψαν > 80 ευρώ'
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

```

ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΣ(Πληθ) : ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Πληθ
ΑΡΧΗ
    Αν Πληθ <= 5 ΤΟΤΕ
        ΚΟΣΤΟΣ ← Πληθ*10
    ΑΛΛΙΩΣ
        ΚΟΣΤΟΣ ← 5*10+(Πληθ-5)*8
    ΤΕΛΟΣ_Αν
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

5. Γράψτε συνάρτηση που δέχεται ως παράμετρο έναν ακεραίο πίνακα  $A[9,9]$  και δύο ακεραίες τιμές  $K, \Lambda$  και επιστρέφει την τιμή:

- 0, αν το  $A[K, \Lambda]$  ανήκει στην κύρια διαγώνιο του  $A$ ,
- 1, αν το  $A[K, \Lambda]$  βρίσκεται πάνω από την κύρια διαγώνιο του  $A$ ,
- 2, αν το  $A[K, \Lambda]$  βρίσκεται κάτω από την κύρια διαγώνιο του  $A$ .

**Λύση**

```

ΣΥΝΑΡΤΗΣΗ ΕΛΕΓΧΟΣ(A, K, Λ) : ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: A[9, 9], K, Λ
ΑΡΧΗ
    Αν K = Λ ΤΟΤΕ
        ΕΛΕΓΧΟΣ ← 0
    ΑΛΛΙΩΣ_Αν K < Λ ΤΟΤΕ
        ΕΛΕΓΧΟΣ ← 1
    ΑΛΛΙΩΣ
        ΕΛΕΓΧΟΣ ← 2
    ΤΕΛΟΣ_Αν
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

6. Επισημάνετε τα συντακτικά λάθη της ακόλουθης διαδικασίας:

```

ΔΙΑΔΙΚΑΣΙΑ ΤΕΛΟΣ [B[2, 3], M, N]
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: M, N, B[2, 3]
ΣΤΑΘΕΡΕΣ
    A = 21
ΑΡΧΗ

```

$M \leftarrow 100$

$N \leftarrow 0$

**Για κ ΑΠΟ 1 ΜΕΧΡΙ Α ΜΕΒΗΜΑ 3**

**Για λ ΑΠΟ 4 ΜΕΧΡΙ 2Α ΜΕ\_ΒΗΜΑ 2**

$M \leftarrow M - B[k, \lambda]$

$N \leftarrow N - B[k, \lambda]^2$

**ΤΕΛΟΣ\_Για**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ**

### **Λύση**

- Το όνομα της διαδικασίας είναι λανθασμένο, γιατί το ΤΕΛΟΣ είναι δεσμευμένη λέξη.
- Οι παράμετροι της διαδικασίας πρέπει να περικλείονται από παρενθέσεις και όχι αγκύλες.
- Στην επικεφαλίδα της διαδικασίας πρέπει να αναγράφεται μόνο Β αντί του Β[2, 3].
- Το τμήμα δήλωσης των συμβολικών σταθερών πρέπει να προηγείται του τμήματος δήλωσης των μεταβλητών.
- Δεν έχουν δηλωθεί οι μεταβλητές κ, λ, στο τμήμα δήλωσης των μεταβλητών της διαδικασίας
- Στην εξωτερική Για η λέξη ΜΕΒΗΜΑ πρέπει να αντικατασταθεί από τη λέξη ΜΕ\_ΒΗΜΑ.
- Στην εσωτερική Για
  - α. το 2Α πρέπει να γίνει 2\*Α
  - β. το ΤΕΛΟΣ\_Για να γίνει ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ.
- Η διαδικασία πρέπει να τελειώσει με τη δεσμευμένη λέξη ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ αντί της λέξης ΤΕΛΟΣ.

**7.** Να αναπτυχθεί πρόγραμμα (με χρήση υποπρογραμμάτων) που θα διαβάζει δύο ακέραιους αριθμούς α και β και ένα θετικό αριθμό Ν και θα υπολογίζει και θα εκτυπώνει την τιμή της παράστασης:

$$S = 1 + (\alpha+1)\beta + (\alpha+2)\beta^2 + (\alpha+3)\beta^3 + \dots + (\alpha+N)\beta^N$$

### **Λύση**

**ΠΡΟΓΡΑΜΜΑ ΠΡΟΟΔΟΣ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** α,β,Ν,Ν

**ΑΡΧΗ**

**ΓΡΑΨΕ** 'Δώσε δύο αριθμούς'

**ΔΙΑΒΑΣΕ** α,β

**ΚΑΛΕΣΕ** ΘΕΤΙΚΟ (Ν)

$S \leftarrow 1$

**Για i ΑΠΟ 1 ΜΕΧΡΙ Ν**

$S \leftarrow S + \text{ΟΡΟΣ}(\alpha, \beta, i)$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** S

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** ΠΡΟΟΔΟΣ

**ΔΙΑΔΙΚΑΣΙΑ** ΘΕΤΙΚΟ(num)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** num

**ΑΡΧΗ**

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Δώσε ένα θετικό αριθμό'

**ΔΙΑΒΑΣΕ** num

**ΜΕΧΡΙΣ\_ΟΤΟΥ** num > 0

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**ΣΥΝΑΡΤΗΣΗ** ΟΡΟΣ(a,b,k):**ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** a,b,k

**ΑΡΧΗ**

**ΟΡΟΣ**  $\leftarrow (a+k)*b^k$

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**8.** Μετατρέψτε την παρακάτω συνάρτηση σε διαδικασία.

**ΣΥΝΑΡΤΗΣΗ** Φ(A, B) : **ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** A, B

**ΑΡΧΗ**

A  $\leftarrow$  A+1

B  $\leftarrow$  B\*2

Φ  $\leftarrow$  A^B

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**Λύση**

**ΔΙΑΔΙΚΑΣΙΑ** Φ(A, B, Γ)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** K, Λ, A, B, Γ

**ΑΡΧΗ**

K  $\leftarrow$  A+1

Λ  $\leftarrow$  B\*2

Γ  $\leftarrow$  K^Λ

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**9.** Γράψτε διαδικασία που δέχεται ως παράμετρο μια γωνία Θ και υπολογίζει το ημίτονο, συνημίτονο, εφαπτομένη και συνεφαπτομένη αυτής της γωνίας (Αν δεν είναι εφικτός ο υπολογισμός της εφαπτομένης ή συνεφαπτομένης, αποδίδεται σε αυτή η τιμή -9999.9). Υπενθυμίζεται ότι η εφαπτομένη μια γωνίας ισούται με το αποτέλεσμα της διαίρεσης του ημιτόνου με το συνημίτονο της γωνίας. Η συνεφαπτομένη της γωνίας ισούται με το αποτέλεσμα της διαίρεσης του συνημιτόνου με το ημίτονο της γωνίας.

**Λύση**

**ΔΙΑΔΙΚΑΣΙΑ ΓΩΝΙΑ(Θ, ΗΜΘ, ΣΥΝΘ, ΕΦΘ, ΣΦΘ)  
ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** Θ, ΗΜΘ, ΣΥΝΘ, ΕΦΘ, ΣΦΘ

**ΑΡΧΗ**

ΗΜΘ ← ΗΜ(Θ)

ΣΥΝΘ ← ΣΥΝ(Θ)

**Αν ΣΥΝΘ < > 0 ΤΟΤΕ**

ΕΦΘ ← ΗΜΘ/ΣΥΝΘ

**ΑΛΛΙΩΣ**

ΕΦΘ ← -1

**ΤΕΛΟΣ\_Αν**

**Αν ΗΜΘ < > 0 ΤΟΤΕ**

ΣΦΘ ← ΣΥΝΘ/ΗΜΘ

**ΑΛΛΙΩΣ**

ΣΦΘ ← -1

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**10.** Η ταξινόμηση ευθείας ανταλλαγής (αλγόριθμος φυσαλίδας), όπως αναφέρεται στο σχολικό βιβλίο, ταξινομεί τα στοιχεία ενός πίνακα κατά αύξουσα ή φθίνουσα σειρά. Τι γίνεται όμως στην περίπτωση που έχουμε δυο ταξινομημένους πίνακες και θέλουμε να τους συγχωνεύσουμε σε έναν τρίτο διατηρώντας την ταξινόμηση; Η πρώτη λύση είναι να τους συγχωνεύσουμε σε έναν τρίτο και μετά να τον ταξινομήσουμε εφαρμόζοντας τον αλγόριθμο της φυσαλίδας. Μήπως όμως υπάρχει καλύτερη λύση εκμεταλλευόμενη την ήδη ταξινομημένη διάταξη τους, κάνοντας με αυτόν τον τρόπο λιγότερες πράξεις;

**Λύση**

**ΠΡΟΓΡΑΜΜΑ ΤΑΞΙΝΟΜΗΣΗ  
ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** Α[5], Β[5], Γ[10], Ι

**ΑΡΧΗ**

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5**

**ΔΙΑΒΑΣΕ Α[Ι]**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5**

**ΔΙΑΒΑΣΕ Β[Ι]**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΚΑΛΕΣΕ ΤΑΞ(Α, Β, Γ)**

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10**

**ΓΡΑΨΕ Γ[Ι]**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΔΙΑΔΙΚΑΣΙΑ ΤΑΞ(Α, Β, Γ)**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** Α[5], Β[5], Γ[10], Ι, Κ, Τ

```

ΑΡΧΗ
T ← 1
K ← 1
I ← 1
ΟΣΟ I <= 5 ΚΑΙ K <= 5 ΕΠΑΝΑΛΑΒΕ
    ΑΝ A[I] <= B[K] ΤΟΤΕ
        Γ[T] ← A[I]
        I ← I + 1
    ΑΛΛΙΩΣ
        Γ[T] ← B[K]
        K ← K + 1
    ΤΕΛΟΣ_ΑΝ
    T ← T + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ I = 6 ΤΟΤΕ
    ΟΣΟ (K <= 5) ΕΠΑΝΑΛΑΒΕ
        Γ[T] ← B[K]
        K ← K + 1
        T ← T + 1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ
    ΟΣΟ (I <= 5) ΕΠΑΝΑΛΑΒΕ
        Γ[T] ← A[I]
        I ← I + 1
        T ← T + 1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

**11.** Αναπτύξτε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: να διαβάζει έναν πίνακα Y το πολύ 50 στοιχείων, να διαβάζει ένα δεύτερο πίνακα X το πολύ 20 στοιχείων και μικρότερου μεγέθους του Y και με χρήση υποπρογραμμάτων να εντοπίζει πόσες φορές υπάρχει ο πίνακας Y στον πίνακα X.

### **Λύση**

```

ΠΡΟΓΡΑΜΜΑ ΠΙΝΑΚΕΣ
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: I, ΧΜ, ΥΜ, ΠΛ
ΧΑΡΑΚΤΗΡΕΣ : Χ[20], Υ[50]
ΑΡΧΗ
ΓΡΑΨΕ 'ΔΩΣΕ ΜΗΚΟΣ ΠΡΩΤΟΥ ΠΙΝΑΚΑ'
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ ΥΜ
ΜΕΧΡΙΣ_ΟΤΟΥ ΥΜ <= 50 ΚΑΙ ΥΜ >= 0
ΓΡΑΨΕ 'ΤΩΡΑ ΔΩΣΕ ΤΑ ΣΤΟΙΧΕΙΑ ΤΟΥ ΕΝΑ ΕΝΑ'
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ ΥΜ
ΔΙΑΒΑΣΕ Υ[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΓΡΑΨΕ 'ΔΩΣΕ ΜΗΚΟΣ ΤΟΥ ΠΙΝΑΚΑ ΠΟΥ ΘΕΣ ΝΑ ΑΝΑΖΗΤΗΣΕΙΣ'
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ ΧΜ
ΜΕΧΡΙΣ_ΟΤΟΥ ΧΜ > 0 ΚΑΙ ΧΜ <= ΥΜ
ΓΡΑΨΕ 'ΤΩΡΑ ΔΩΣΕ ΤΑ ΣΤΟΙΧΕΙΑ ΤΟΥ ΕΝΑ ΕΝΑ'
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ ΧΜ
ΔΙΑΒΑΣΕ Χ[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛ ← ΥΠ_ΠΛ(Χ, Υ, ΧΜ, ΥΜ)
ΓΡΑΨΕ 'Ο ΔΕΥΤΕΡΟΣ ΠΙΝΑΚΑΣ ΥΠΑΡΧΕΙ', ΠΛ, 'ΦΟΡΕΣ ΜΕΣΑ ΣΤΟΝ ΠΡΩΤΟ'
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

**ΣΥΝΑΡΤΗΣΗ ΥΠ\_ΠΛ (Χ, Υ, ΧΜ, ΥΜ) : ΑΚΕΡΑΙΑ** *!Επικεφαλίδα συνάρτησης*

**ΜΕΤΑΒΛΗΤΕΣ**

**ΧΑΡΑΚΤΗΡΕΣ : Χ[20], Υ[50]**

**ΑΚΕΡΑΙΕΣ : ΧΜ, ΥΜ, Ι, Κ, ΕΜΦ**

**ΛΟΓΙΚΕΣ:ΤΑΙΡΙΑΖΕΙ**

**ΑΡΧΗ**

**ΥΠ\_ΠΛ ← 0**

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ ΥΜ - ΧΜ + 1**

**ΑΝ Υ[Ι] = Χ[1] ΤΟΤΕ**

**Κ ← 2**

**ΤΑΙΡΙΑΖΕΙ ← ΑΛΗΘΗΣ**

**ΟΣΟ Κ<=ΧΜ ΚΑΙ ΤΑΙΡΙΑΖΕΙ=ΑΛΗΘΗΣ ΕΠΑΝΑΛΑΒΕ**

**ΑΝ Υ[Ι + Κ - 1] <> Χ[Κ] ΤΟΤΕ**

**ΤΑΙΡΙΑΖΕΙ ← ΨΕΥΔΗΣ**

**ΤΕΛΟΣ\_ΑΝ**

**Κ ← Κ+1**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ (ΤΑΙΡΙΑΖΕΙ=ΑΛΗΘΗΣ) ΤΟΤΕ**

**ΥΠ\_ΠΛ ← ΥΠ\_ΠΛ + 1**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**12.** Να γράψετε πρόγραμμα που γεμίζει έναν πίνακα 100 θέσεων με ακεραίους και να καλεί υποπρόγραμμα για κάθε θέση του πίνακα, το οποίο να αποφασίζει αν υπάρχει «συμμετρία» σε αυτή τη θέση. Αν η υποψήφια θέση είναι πιο κοντά στην αριστερή άκρη του πίνακα, τότε όλα τα στοιχεία αριστερά του να «καθρεπτίζονται» στο δεξί του τμήμα. Και αντίστοιχα για τις θέσεις κοντύτερα στο δεξί. π.χ. για τον πίνακα | 0 | 9 | 7 | 2 | 3 | 2 | 7 | παρατηρείται η παραπάνω συμμετρία στη θέση 5, όπου τα στοιχεία δεξιά του 3, υπάρχουν συμμετρικά αριστερά του. Να εμφανίζει τις θέσεις που παρατηρείται συμμετρία ή κατάλληλο μήνυμα σε αντίθετη περίπτωση.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ ΣΥΜΜΕΤΡΙΑ\_ΠΙΝΑΚΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

```

ΑΚΕΡΑΙΕΣ: Ι,ΠΛ
ΠΡΑΓΜΑΤΙΚΕΣ: Χ[100]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
ΔΙΑΒΑΣΕ Χ[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛ ← 0
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 99
    ΑΝ ΣΥΜ(Χ,Ι)=ΑΛΗΘΗΣ ΤΟΤΕ
        ΓΡΑΨΕ 'ΕΙΝΑΙ ΣΥΜΜΕΤΡΙΚΟΣ ΣΤΗΝ', Ι, 'ΘΕΣΗ'
        ΠΛ ← ΠΛ+1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ ΠΛ>0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ο ΠΙΝΑΚΑΣ ΕΙΝΑΙ ΣΥΜΜΕΤΡΙΚΟΣ ΣΕ', ΠΛ,'ΣΗΜΕΙΑ'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΔΕΝ ΒΡΕΘΗΚΕ ΚΑΝΕΝΑ ΣΥΜΜΕΤΡΙΚΟ ΣΗΜΕΙΟ'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

```

ΣΥΝΑΡΤΗΣΗ ΣΥΜ(Π,Θ):ΛΟΓΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: Θ,Ι
ΠΡΑΓΜΑΤΙΚΕΣ: Π[100]
ΛΟΓΙΚΕΣ: Κ
ΑΡΧΗ
Κ ← ΑΛΗΘΗΣ
Ι ← 1
ΟΣΟ Ι<Θ ΚΑΙ Ι<=100-Θ ΚΑΙ Κ=ΑΛΗΘΗΣ ΕΠΑΝΑΛΑΒΕ
    ΑΝ Π[Θ-Ι]<>Π[Θ+Ι] ΤΟΤΕ
        Κ ← ΨΕΥΔΗΣ
    ΑΛΛΙΩΣ
        Ι ← Ι+1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΣΥΜ ← Κ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

**13.** Ένας πρωταθλητής για ένα μήνα προσπαθεί κάθε μέρα να βελτιώσει το ρεκόρ του. Κάθε μέρα κάνει το μέγιστο 20 προσπάθειες, σταματώντας νωρίτερα αν έχει ήδη επιτύχει καλύτερο χρόνο. Να αναπτύξετε ΠΡΟΓΡΑΜΜΑ σε ΓΛΩΣΣΑ το οποίο:

Θα διαβάσει το ρεκόρ του αθλητή ελέγχοντας να είναι θετικός αριθμός, εμφανίζοντας μήνυμα σε αντίθετη περίπτωση. Εμφανίζει για κάθε μέρα, πόσες προσπάθειες χρειάστηκαν για να ξεπεράσει το ρεκόρ του. Τους χρόνους της κάθε προσπάθειας θα τους διαβάζει ελέγχοντας να είναι θετικός αριθμός. Για τις ημέρες που ξεπέρασε το ρεκόρ του, να υπολογίζει και να εμφανίζει το μέσο όρο των προσπαθειών που χρειάζεται. Ποια μέρα χρειάστηκε τις λιγότερες προσπάθειες.

**Λύση**



**ΠΡΟΓΡΑΜΜΑ ΠΡΩΤΑΘΛΗΤΗΣ**

**ΜΕΤΑΒΛΗΤΕΣ**

ΑΚΕΡΑΙΕΣ: ΜΙΝΠΛ, Σ, ΗΜΚ, ΜΙΝΜΕΡΑ, Ι, ΠΛ

**ΠΡΑΓΜΑΤΙΚΕΣ:** ΜΙΝ, ΧΡΟΝΟΣ, ΜΟ

**ΑΡΧΗ**

ΜΙΝΠΛ ← 21

Σ ← 0

ΗΜΚ ← 0

**ΔΙΑΒΑΣΕ ΜΙΝ**

**ΟΣΟ ΜΙΝ<=0 ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ 'ΛΑΘΟΣ ΧΡΟΝΟΣ'**

**ΔΙΑΒΑΣΕ ΜΙΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30**

ΠΛ ← 1

**ΓΡΑΨΕ 'ΔΩΣΤΕ ΤΟΝ ΧΡΟΝΟ ΤΗΣ', Ι, 'ΗΜΕΡΑΣ ΚΑΙ', ΠΛ, 'ΠΡΟΣΠΑΘΕΙΑΣ'**

**ΔΙΑΒΑΣΕ ΧΡΟΝΟΣ**

**ΟΣΟ ΧΡΟΝΟΣ <=0 ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ 'ΛΑΘΟΣ ΧΡΟΝΟΣ'**

**ΔΙΑΒΑΣΕ ΧΡΟΝΟΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΟΣΟ ΠΛ<20 ΚΑΙ ΧΡΟΝΟΣ>=ΜΙΝ ΕΠΑΝΑΛΑΒΕ**

ΠΛ ← ΠΛ+1

**ΓΡΑΨΕ 'ΔΩΣΤΕ ΤΟΝ ΧΡΟΝΟ ΤΗΣ', Ι, 'ΗΜΕΡΑΣ ΚΑΙ', ΠΛ, 'ΠΡΟΣΠΑΘΕΙΑΣ'**

**ΔΙΑΒΑΣΕ ΧΡΟΝΟΣ**

**ΟΣΟ ΧΡΟΝΟΣ <=0 ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ 'ΛΑΘΟΣ ΧΡΟΝΟΣ'**

**ΔΙΑΒΑΣΕ ΧΡΟΝΟΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ ΧΡΟΝΟΣ<ΜΙΝ ΤΟΤΕ**

**ΓΡΑΨΕ 'ΜΠΡΑΒΟ, ΤΑ ΚΑΤΑΦΕΡΑΤΕ ΣΗΜΕΡΑ ΣΕ', ΠΛ, 'ΠΡΟΣΠΑΘΕΙΕΣ'**

ΜΙΝ ← ΧΡΟΝΟΣ

Σ ← Σ+ΠΛ

ΗΜΚ ← ΗΜΚ+1

**ΑΝ ΠΛ<ΜΙΝΠΛ ΤΟΤΕ**

ΜΙΝΠΛ ← ΠΛ

ΜΙΝΜΕΡΑ ← Ι

**ΤΕΛΟΣ\_ΑΝ**

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ 'ΛΥΠΑΜΑΙ ΑΛΛΑ ΣΗΜΕΡΑ ΔΕΝ ΤΑ ΚΑΤΑΦΕΡΑΤΕ'**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ ΗΜΚ=0 ΤΟΤΕ**

**ΓΡΑΨΕ 'ΛΥΠΑΜΑΙ ΑΛΛΑ ΔΕΝ ΤΑ ΚΑΤΑΦΕΡΑΤΕ ΚΑΜΙΑ ΗΜΕΡΑ'**

**ΑΛΛΙΩΣ**

ΜΟ ← Σ/ΗΜΚ

**ΓΡΑΨΕ 'Ο ΜΕΣΟΣ ΟΡΟΣ ΠΡΟΣΠΑΘΕΙΩΝ ΕΙΝΑΙ', ΜΟ**

**ΓΡΑΨΕ 'Η ΜΕΡΑ ΠΟΥ ΧΡΕΙΑΣΤΗΚΕ ΤΙΣ ΛΙΓΟΤΕΡΕΣ ΠΡΟΣΠΑΘΕΙΕΣ ΗΤΑΝ Η', ΜΙΝΜΕΡΑ  
ΤΕΛΟΣ\_ΑΝ  
ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**14.** Να γραφεί υποπρόγραμμα που θα δέχεται στην είσοδό του ένα πραγματικό αριθμό και θα επιστρέφει την τιμή του στρογγυλοποιημένη στον πλησιέστερο ακέραιο (π.χ. 5.68 σε 6, 3.17 σε 3, 6.5 σε 7).

**Λύση**

**ΣΥΝΑΡΤΗΣΗ ROUND(a):ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ: a**

**ΑΡΧΗ**

**ΑΝ a - A\_M(a) >= 0.5 ΤΟΤΕ**

**ROUND ← A\_M(a) + 1**

**ΑΛΛΙΩΣ**

**ROUND ← A\_M(a)**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**!2η Λύση**

**ΑΝ A\_M(a \* 100) mod 100 >= 50 ΤΟΤΕ**

**ROUND ← A\_M(a+1)**

**ΑΛΛΙΩΣ**

**ROUND ← A\_M(a)**

**ΤΕΛΟΣ\_ΑΝ**

**15.** Ο αριθμός 368 έχει την ιδιότητα τα ψηφία του να αυξάνουν από τα αριστερά προς τα δεξιά (3,6,8). Ζητείται να γραφεί πρόγραμμα που να εμφανίζει όλους τους τριψήφιους ακέραιους αριθμούς που έχουν την ιδιότητα αυτή, καθώς επίσης και το πλήθος τους. Το πρόγραμμα θα κάνει κλήση μιας διαδικασίας η οποία θα δέχεται έναν οποιοδήποτε τριψήφιο ακέραιο αριθμό και θα τον διασπά στα τρία ψηφία του τα οποία και θα επιστρέφει στο κυρίως πρόγραμμα. Επίσης θα καλεί μια συνάρτηση που θα δέχεται τα τρία ψηφία ενός οποιουδήποτε τριψήφιου αριθμού και θα αποφασίζει αν αυτά πληρούν την παραπάνω ιδιότητα (δηλαδή είναι σε αύξουσα σειρά).

**Λύση**

**ΠΡΟΓΡΑΜΜΑ ΤΡΙΨΗΦΙΟΙ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ: i,x,y,z**

**ΑΡΧΗ**

**ΓΙΑ i ΑΠΟ 100 ΜΕΧΡΙ 999**

**ΚΑΛΕΣΕ SPLIT (i,x,y,z)**

**ΑΝ ΑΥΞΟΥΣΑ(x,y,z) = ΑΛΗΘΗΣ ΤΟΤΕ**

**ΓΡΑΨΕ 'Ο αριθμός',i,'είναι σε αύξουσα σειρά'**

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ 'Ο αριθμός',i,'δεν είναι σε αύξουσα σειρά'**

ΤΕΛΟΣ\_ΑΝ  
ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  
ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ ΤΡΙΨΗΦΙΟΙ

**ΔΙΑΔΙΚΑΣΙΑ** SPLIT(num,a,b,c)

! Επικεφαλίδα διαδικασίας

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** num,a,b,c

**ΑΡΧΗ**

a ← num **div** 100

b ← num **mod** 100 **div** 10

c ← num **mod** 10

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**ΣΥΝΑΡΤΗΣΗ** ΑΥΞΟΥΣΑ(a,b,c):ΛΟΓΙΚΗ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** a,b,c

**ΑΡΧΗ**

**ΑΝ** (a < b) **ΚΑΙ** (b < c) **ΤΟΤΕ**

ΑΥΞΟΥΣΑ ← **ΑΛΗΘΗΣ**

**ΑΛΛΙΩΣ**

ΑΥΞΟΥΣΑ ← **ΨΕΥΔΗΣ**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**16.** Υπολογισμός του παραγοντικού με επαναληπτική διαδικασία.

**Λύση**

**ΣΥΝΑΡΤΗΣΗ** Παραγοντικο(N):**ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, N

**ΑΡΧΗ**

Fact ← 1

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** N

Fact ← Fact\*i

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**17.** Έλεγχος ημερομηνίας: Ένα έτος λέγεται δίσεκτο, αν ο αριθμός του είναι πολλαπλάσιο του 4, αλλά όχι του 100, εκτός αν είναι πολλαπλάσιο του 400. Για παράδειγμα: το 1984 είναι δίσεκτο (πολλαπλάσιο του 4), το 1900 δεν είναι (πολλαπλάσιο του 4 αλλά και του 100), το 2000 είναι δίσεκτο (πολλαπλάσιο του 4, του 100, αλλά και του 400), τέλος το 1993 δεν είναι.

**Λύση**

**ΔΙΑΔΙΚΑΣΙΑ** Έλεγχος\_2(η, μ, ε, c)

! Ιουλιανό ημερολόγιο

c ← 1

AN (10<η ΚΑΙ η<23) ΚΑΙ μ=3 ΚΑΙ ε=1923 ΤΟΤΕ c ← 0  
ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

**18.** Ένα σύστημα υπολογιστή χρησιμοποιεί για τον έλεγχο πρόσβασης των χρηστών του έναν πίνακα 1000 γραμμών και 3 στηλών με τα στοιχεία τους. Σε κάθε γραμμή του αποθηκεύει, στην πρώτη στήλη το όνομα πρόσβασης του χρήστη, στη δεύτερη στήλη το συνθηματικό του και στην τρίτη έναν από τους χαρακτήρες «Σ» ή «Α». (Ο χαρακτήρας «Σ» δηλώνει ότι το συνθηματικό συνεχίζει να ισχύει, ενώ ο χαρακτήρας «Α» δηλώνει ότι το συνθηματικό πρέπει να αλλάξει). Θεωρήστε ότι υπάρχει ένα κύριο πρόγραμμα που υλοποιεί τα παραπάνω και καλεί τη διαδικασία ΕΛΕΓΧΟΣ η οποία ελέγχει την πρόσβαση του χρήστη στο σύστημα. Να γράψετε τη διαδικασία ΕΛΕΓΧΟΣ η οποία να περιλαμβάνει:

Γ1. Τμήμα δηλώσεων. Κύριο τμήμα το οποίο:

Γ2. Διαβάζει το όνομα και το συνθηματικό του χρήστη. Ελέγχει αν το όνομα πρόσβασης και το συνθηματικό είναι έγκυρα, δηλαδή υπάρχουν στον πίνακα χρηστών και αναφέρονται στον ίδιο χρήστη. Αν υπάρχουν, εμφανίζει το μήνυμα «ΚΑΛΩΣ ΗΡΘΑΤΕ», διαφορετικά εμφανίζει το μήνυμα «ΛΑΘΟΣ ΟΝΟΜΑ ΠΡΟΣΒΑΣΗΣ Η ΣΥΝΘΗΜΑΤΙΚΟ» και ζητά εκ νέου την εισαγωγή των δύο αυτών στοιχείων (ονόματος πρόσβασης και συνθηματικού) μέχρι να δοθούν έγκυρα στοιχεία.

Γ3. Μετά την εμφάνιση του μηνύματος «ΚΑΛΩΣ ΗΡΘΑΤΕ» ελέγχει αν το συνθηματικό χρειάζεται αλλαγή. Αν χρειάζεται, ζητά από τον χρήστη την εισαγωγή νέου συνθηματικού δύο φορές (η δεύτερη ως επιβεβαίωση) μέχρις ότου το συνθηματικό και η επιβεβαίωση του ταυτιστούν. Όταν ταυτιστούν, η διαδικασία αντικαθιστά το παλιό συνθηματικό με το νέο και τον αντίστοιχο χαρακτήρα «Α» της τρίτης στήλης με το «Σ». (Θέμα Πανελλαδικών)

### **Λύση**

**ΔΙΑΔΙΚΑΣΙΑ ΕΛΕΓΧΟΣ(ΣΤΟΙΧΕΙΑ)**

ΜΕΤΑΒΛΗΤΕΣ

**ΑΚΕΡΑΙΕΣ:** i, θέση

**ΧΑΡΑΚΤΗΡΕΣ:** ΣΤΟΙΧΕΙΑ[1000,3], όνομα\_χρήστη, συνθηματικό,

&νέο\_συνθηματικό, επιβεβαίωση

**ΛΟΓΙΚΕΣ:** βρέθηκε

ΑΡΧΗ

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

**ΓΡΑΨΕ** 'Εισάγετε το όνομα χρήστη: '

**ΔΙΑΒΑΣΕ** όνομα\_χρήστη

**ΓΡΑΨΕ** 'Εισάγετε το συνθηματικό: '

**ΔΙΑΒΑΣΕ** συνθηματικό

*!αναζήτηση του ονόματος χρήστη και του συνθηματικού στην πρώτη στήλη και δεύτερη στήλη του πίνακα ΣΤΟΙΧΕΙΑ*

i ← 1

βρέθηκε ← ΨΕΥΔΗΣ

**ΟΣΟ** βρέθηκε = ΨΕΥΔΗΣ και i <= 1000 **ΕΠΑΝΑΛΑΒΕ**

**ΑΝ** ΣΤΟΙΧΕΙΑ[i, 1] = όνομα\_χρήστη ΚΑΙ

&ΣΤΟΙΧΕΙΑ[i, 2] = συνθηματικό **ΤΟΤΕ**

βρέθηκε ← **ΑΛΗΘΗΣ**

θέση ← i

**ΤΕΛΟΣ\_ΑΝ**

i ← i + 1

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ βρέθηκε = ΑΛΗΘΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'ΚΑΛΩΣ ΗΡΘΑΤΕ'

ΑΝ ΣΤΟΙΧΕΙΑ[θέση, 3] = 'Α' ΤΟΤΕ

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Εισάγετε νέο συνθηματικό: '

ΔΙΑΒΑΣΕ νέο\_συνθηματικό

ΓΡΑΨΕ 'Επιβεβαιώστε: '

ΔΙΑΒΑΣΕ επιβεβαιώση

ΜΕΧΡΙΣ\_ΟΤΟΥ νέο\_συνθηματικό = επιβεβαιώση

ΤΕΛΟΣ\_ΑΝ

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'ΛΑΘΟΣ ΟΝΟΜΑ ΠΡΟΣΒΑΣΗΣ Η ΣΥΝΘΗΜΑΤΙΚΟ'

ΤΕΛΟΣ\_ΑΝ

ΜΕΧΡΙΣ\_ΟΤΟΥ βρέθηκε = ΑΛΗΘΗΣ

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

**19.** Για ποιες από τις παρακάτω περιπτώσεις μπορεί να χρησιμοποιηθεί συνάρτηση:

- α. εισαγωγή ενός δεδομένου
- β. υπολογισμός του μικρότερου από πέντε ακεραίου
- γ. υπολογισμός των δύο μικρότερων από πέντε ακεραίους
- δ. έλεγχος αν δύο αριθμοί είναι ίσοι
- ε. ταξινόμηση πέντε αριθμών
- στ. έλεγχος αν ένας χαρακτήρας είναι φωνήεν ή σύμφωνο.

**Απάντηση**

β, δ, στ

**20.** Εταιρεία που ασχολείται με μετρήσεις τηλεθέασης καταγράφει στοιχεία, ανά ημέρα και για χρονικό διάστημα μίας εβδομάδας, τα οποία αφορούν στην τηλεθέαση των κεντρικών δελτίων ειδήσεων που προβάλλονται από πέντε (5) τηλεοπτικούς σταθμούς. Για τη διευκόλυνση της στατιστικής επεξεργασίας των δεδομένων να αναπτύξετε πρόγραμμα το οποίο:

Δ1. Να περιλαμβάνει τμήμα δηλώσεων.

Δ2. Για κάθε έναν από τους τηλεοπτικούς σταθμούς να δέχεται το όνομά του και το πλήθος των τηλεθεατών που παρακολούθησαν το κεντρικό δελτίο ειδήσεων κάθε μέρα της εβδομάδας, από Δευτέρα έως και Κυριακή. Να μη γίνει έλεγχος εγκυρότητας.

Δ3. Να καλεί για κάθε έναν από τους τηλεοπτικούς σταθμούς κατάλληλο υποπρόγραμμα, το οποίο να υπολογίζει και να επιστρέφει το μέσο πλήθος τηλεθεατών, που παρακολούθησαν το κεντρικό δελτίο ειδήσεών του, τη συγκεκριμένη εβδομάδα. Να αναπτύξετε το κατάλληλο υποπρόγραμμα.

Δ4. Να εμφανίζει τα ονόματα των σταθμών για τους οποίους ο μέσος όρος τηλεθέασης του Σαββατοκύριακου (2 ημέρες) ήταν τουλάχιστον 10% μεγαλύτερος από το μέσο όρο τηλεθέασης στις καθημερινές (Δευτέρα έως και Παρασκευή).

Δ5. Να εμφανίζει τα ονόματα των τηλεοπτικών σταθμών, οι οποίοι κάθε ημέρα, από Δευτέρα έως και Κυριακή, παρουσιάζουν συνεχώς, από ημέρα σε ημέρα, αύξηση τηλεθέασης. Αν δεν υπάρχουν τέτοιοι σταθμοί, να εμφανίζει το μήνυμα: «Κανένας

σταθμός δεν είχε συνεχή αύξηση τηλεθέασης». (Θέμα Πανελλαδικών)

### **Λύση**

```
ΠΡΟΓΡΑΜΜΑ τηλεθέαση
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ : I, J, Π[5,7], Σ1, Σ2, K, station
ΠΡΑΓΜΑΤΙΚΕΣ : ΜΟ[5], ΜΟΣ[5], ΜΟΣΚ[5]
ΧΑΡΑΚΤΗΡΕΣ : Ο[5]
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5
    ΔΙΑΒΑΣΕ Ο[I]
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 7
    ΔΙΑΒΑΣΕ Π[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5
    ΜΟ[I] ← μεσοσ(Π,I)
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5
    Σ1←0
    Σ2←0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 7
        ΑΝ j <= 5 ΤΟΤΕ
            Σ1←Σ1+Π[I,J]
        ΑΛΛΙΩΣ
            Σ2←Σ2+Π[I,J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟΚ←Σ1/5
    ΜΟΣΚ←Σ2/2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5
    ΑΝ ΜΟΣΚ > ΜΟΚ + ΜΟΚ*10/100 ΤΟΤΕ
        ΓΡΑΨΕ Ο[I]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Station ←0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5
    Κ ← 0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 6
        ΑΝ Π[I,J+1] > Π[I,J] ΤΟΤΕ
            Κ ← Κ +1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ Κ = 6 ΤΟΤΕ
        ΓΡΑΨΕ Ο[I]
        station ← station +1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

AN station = 0 ΤΟΤΕ

ΓΡΑΨΕ 'κανένας σταθμός δεν είχε συνεχή αύξηση τηλεθέασης'

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ μεσος(θ,Κ) : ΠΡΑΓΜΑΤΙΚΗ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ : θ[5,7], Κ, sum, J

ΑΡΧΗ

Sum ← 0

ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 7

Sum ← sum + θ[Κ,J]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

μεσος ← sum/7

ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ

**21.** Ξενοδοχείο διαθέτει 25 δωμάτια. Τα δωμάτια αριθμούνται από 1 μέχρι 25. Ο συνολικός αριθμός υπαλλήλων που απασχολούνται ημερησίως καθορίζεται από τα κατειλημμένα δωμάτια και δίνεται από τον παρακάτω πίνακα:

ΔΩΜΑΤΙΑ	ΑΡΙΘΜΟΣ ΥΠΑΛΛΗΛΩΝ
0 μέχρι 4	3
5 μέχρι 8	4
9 μέχρι 12	5
άνω των 12	6

Η ημερήσια χρέωση για κάθε δωμάτιο είναι 75 ευρώ. Το ημερομίσθιο κάθε υπαλλήλου είναι 45 ευρώ. Να κατασκευάστε πρόγραμμα που: περιλαμβάνει το τμήμα δηλώσεων και διαβάζει πίνακα σε ΚΡΑΤ[25,7] την κατάσταση κάθε δωματίου για κάθε μέρα της εβδομάδας, ελέγχοντας την ορθή καταχώρηση. Το πρόγραμμα δέχεται μόνο τους χαρακτήρες "Κ" για κατειλημμένο και "Δ" για διαθέσιμο. Να υπολογίζει το συνολικό κέρδος ή τη συνολική ζημιά κατά τη διάρκεια της εβδομάδας και να εμφανίζει κατάλληλο μήνυμα. Για το σκοπό αυτό να καλεί υποπρόγραμμα ΚΕΡΔΟΣ που περιγράφεται παρακάτω.

Να αναπτύξετε υποπρόγραμμα κέρδος το οποίο να δέχεται τον πίνακα των κρατήσεων και έναν αριθμό ημέρας από 1 μέχρι 7. Το υποπρόγραμμα υπολογίζει και επιστρέφει το κέρδος της συγκεκριμένης μέρας. Το κέρδος κάθε μέρας προκύπτει αν από τα ημερήσια έσοδα ενοικιάσεων αφαιρεθούν τα ημερομίσθια των υπαλλήλων της συγκεκριμένης μέρας, αν τα έσοδα είναι μικρότερα τότε το κέρδος είναι αρνητικό (ζημιά). (Θέμα Πανελλαδικών)

**Λύση**

ΠΡΟΓΡΑΜΜΑ HOTEL

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ : I,J,ΗΜΕΡΗΣΙΟ\_ΚΕΡΔΟΣ,ΕΒΔΟΜΑΔΙΑΙΟ\_ΚΕΡΔΟΣ

ΧΑΡΑΚΤΗΡΕΣ : ΚΡΑΤ[25,7]

ΑΡΧΗ

**ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 25**  
**ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 7**  
**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**  
**ΔΙΑΒΑΣΕ ΚΡΑΤ[I,J]**  
**ΜΕΧΡΙΣ\_ΟΤΟΥ (ΚΡΑΤ[I,J] = 'Κ') Η (ΚΡΑΤ[I,J]= 'Δ')**  
**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  
**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΗΜΕΡΗΣΙΟ\_ΚΕΡΔΟΣ ← 0**  
**ΕΒΔΟΜΑΔΙΑΙΟ\_ΚΕΡΔΟΣ ← 0**  
**ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 7**  
**ΗΜΕΡΙΣΙΟ\_ΚΕΡΔΟΣ ← ΚΕΡΔΟΣ(ΚΡΑΤ,J)**  
**ΕΒΔΟΜΑΔΙΑΙΟ\_ΚΕΡΔΟΣ ← ΕΒΔΟΜΑΔΙΑΙΟ\_ΚΕΡΔΟΣ + ΗΜΕΡΗΣΙΟ\_ΚΕΡΔΟΣ**  
**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  
**ΑΝ ΕΒΔΟΜΑΔΙΑΙΟ\_ΚΕΡΔΟΣ > 0 ΤΟΤΕ**  
**ΓΡΑΨΕ 'ΚΕΡΔΟΣ = ',ΕΒΔΟΜΑΔΙΑΙΟ\_ΚΕΡΔΟΣ**  
**ΑΛΛΙΩΣ\_ΑΝ ΕΒΔΟΜΑΔΙΑΙΟ\_ΚΕΡΔΟΣ < 0 ΤΟΤΕ**  
**ΓΡΑΨΕ 'ΖΗΜΙΑ = ', A\_T(ΕΒΔΟΜΑΔΙΑΙΟ\_ΚΕΡΔΟΣ)**  
**ΑΛΛΙΩΣ**  
**ΓΡΑΨΕ " ΙΣΟ ΖΥΓΙΟ "**  
**ΤΕΛΟΣ\_ΑΝ**  
**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΣΥΝΑΡΤΗΣΗ ΚΕΡΔΟΣ(ΚΡΑΤ,J):ΑΚΕΡΑΙΑ**  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ : Ι,ΕΣΟΔΑ[7],ΕΞΟΔΑ[7],ΚΑΠΠΑ[7]**

**ΑΡΧΗ**

**ΕΣΟΔΑ[J] ← 0**

**ΕΞΟΔΑ[J] ← 0**

**I ← 1**

**ΟΣΟ I <= 25 ΕΠΑΝΑΛΑΒΕ**

**ΑΝ ΚΡΑΤ[I,J] = 'Κ' ΤΟΤΕ**

**ΚΑΠΠΑ[J] ← ΚΑΠΠΑ[J] + 1**

**ΕΣΟΔΑ[J] ← ΕΣΟΔΑ[J] + 75**

**ΤΕΛΟΣ\_ΑΝ**

**I ← I + 1**

**ΑΝ I > 25 ΤΟΤΕ**

**ΑΝ ΚΑΠΠΑ[J] <= 4 ΤΟΤΕ**

**ΕΞΟΔΑ[J] ← 3 \* 45**

**ΑΛΛΙΩΣ\_ΑΝ ΚΑΠΠΑ[J] <= 8 ΤΟΤΕ**

**ΕΞΟΔΑ[J] ← 4 \* 45**

**ΑΛΛΙΩΣ\_ΑΝ ΚΑΠΠΑ[J] <= 12 ΤΟΤΕ**

**ΕΞΟΔΑ[J] ← 5 \* 45**

**ΑΛΛΙΩΣ**

**ΕΞΟΔΑ[J] ← 6 \* 45**

**ΤΕΛΟΣ\_ΑΝ**

**ΚΕΡΔΟΣ ← ΕΣΟΔΑ[J] - ΕΞΟΔΑ[J]**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**



## ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ

### 22. ΠΡΟΓΡΑΜΜΑ Κύριο

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, B, Γ

ΑΡΧΗ

ΔΙΑΒΑΣΕ A, B, Γ

ΚΑΛΕΣΕ Διαδ1(A, B, Γ)

ΓΡΑΨΕ A, B, Γ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

### ΔΙΑΔΙΚΑΣΙΑ Διαδ1(B, A, Γ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, B, Γ

ΑΡΧΗ

$A \leftarrow A + 2$

$B \leftarrow B - 3$

$\Gamma \leftarrow A + B$

ΓΡΑΨΕ A, B, Γ

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

Τι θα εμφανιστεί κατά την εκτέλεση του προγράμματος, αν ως τιμές εισόδου δοθούν οι αριθμοί 5, 7, 10;

23. Σε ένα διαγωνισμό του ΑΣΕΠ εξετάζονται 1500 υποψήφιοι. Ως εξεταστικό κέντρο χρησιμοποιείται ένα κτίριο με αίθουσες διαφορετικής χωρητικότητας. Ο αριθμός των επιτηρητών που απαιτούνται ανά αίθουσα καθορίζεται αποκλειστικά με βάση τη χωρητικότητα της αίθουσας ως εξής:

ΧΩΡΗΤΙΚΟΤΗΤΑ	ΑΡΙΘΜΟΣ ΕΠΙΤΗΡΗΤΩΝ
μέχρι και 15 θέσεις	1
από 16 μέχρι και 23 θέσεις	2
πάνω από 23 θέσεις	3

Να γίνει πρόγραμμα σε γλώσσα προγραμματισμού «ΓΛΩΣΣΑ» το οποίο:

α. Για κάθε αίθουσα θα διαβάζει τη χωρητικότητά της, θα υπολογίζει και θα εμφανίζει τον αριθμό των επιτηρητών που χρειάζονται. Ο υπολογισμός του αριθμού των επιτηρητών να γίνεται από συνάρτηση που θα κατασκευάσετε για το σκοπό αυτό.

β. Θα σταματάει όταν εξασφαλισθεί ο απαιτούμενος συνολικός αριθμός θέσεων.

Σημείωση: Να θεωρήσετε ότι η συνολική χωρητικότητα των αιθουσών του κτιρίου επαρκεί για τον αριθμό των υποψηφίων. (Θέμα Πανελλαδικών)

### Λύση

ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ3

ΜΕΤΑΒΛΗΤΕΣ

**ΑΚΕΡΑΙΕΣ:** ΥΠΟΨ, ΘΕΣΕΙΣ, ΑΡ\_ΕΠΙΤΗΡ

ΑΡΧΗ

ΥΠΟΨ ← 0

**ΟΣΟ** ΥΠΟΨ < 1500 **ΕΠΑΝΑΛΑΒΕ**

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Δώσε χωρητικότητα ατόμων ανά αίθουσα'

**ΔΙΑΒΑΣΕ** ΘΕΣΕΙΣ

**ΜΕΧΡΙΣ\_ΟΤΟΥ** ΘΕΣΕΙΣ > 0

ΑΡ\_ΕΠΙΤΗΡ ← ΕΠΙΤΗΡΗΤΕΣ (ΘΕΣΕΙΣ)

**ΓΡΑΨΕ** 'Ο ΑΡΙΘΜΟΣ ΕΠΙΤΗΡΗΤΩΝ ΠΟΥ ΧΡΕΙΑΖΟΝΤΑΙ ΕΙΝΑΙ', ΑΡ\_ΕΠΙΤΗΡ

ΥΠΟΨ ← ΥΠΟΨ + ΘΕΣΕΙΣ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** ΘΕΜΑ3

**ΣΥΝΑΡΤΗΣΗ** ΕΠΙΤΗΡΗΤΕΣ (X): **ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** X, K

**ΑΡΧΗ**

**ΑΝ** X < = 15 **ΤΟΤΕ**

K ← 1

**ΑΛΛΙΩΣ\_ΑΝ** X < = 23 **ΤΟΤΕ**

K ← 2

**ΑΛΛΙΩΣ**

K ← 3

**ΤΕΛΟΣ\_ΑΝ**

ΕΠΙΤΗΡΗΤΕΣ ← K

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**24.** Μια έρευνα διεξάγεται για την πρόοδο των μαθητών της Γ' Λυκείου κατά τη διάρκεια της χρονιάς. Έτσι διεξάγονται δύο εξετάσεις, η μια κατά την περίοδο των διακοπών των Χριστουγέννων και η άλλη κατά την περίοδο των διακοπών του Πάσχα σε αντίστοιχα κομμάτια της ύλης. Η κάθε εξέταση περιλαμβάνει τρία μαθήματα και συμμετέχουν 1000 μαθητές. Να γραφεί πρόγραμμα που:

1. Διαβάζει τα ονόματα των 1000 μαθητών σε μονοδιάστατο πίνακα ΟΝΟΜΑΤΑ[1000].

2. Καταχωρεί για κάθε μαθητή τη βαθμολογία (1-20) για τα αποτελέσματα της πρώτης και δεύτερης εξέτασης σε πίνακες ΑΠΟΤ\_1[1000,3] και ΑΠΟΤ\_2[1000,3] με χρήση υποπρογράμματος, ελέγχοντας για την ακεραιότητα των δεδομένων.

3. Υπολογίζει το μέσο όρο του κάθε μαθητή στα 3 μαθήματα για την πρώτη και δεύτερη εξέταση αντίστοιχα καταχωρώντας τα αποτελέσματα στους πίνακες ΜΟ\_1[1000] και ΜΟ\_2[1000] με χρήση υποπρογράμματος.

4. Υπολογίζει και εμφανίζει το ποσοστό των μαθητών που:

α. αύξησαν ή διατήρησαν.

β. μείωσαν την απόδοσή τους στην δεύτερη εξέταση τους σε σχέση με την πρώτη εμφανίζοντας κατάλληλα μηνύματα.

5. Θεωρώντας ότι το πρώτο μάθημα που εξετάζεται είναι η ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ, σε κάθε εξέταση εμφανίστε το όνομα ή τα ονόματα των μαθητών που έγραψαν το μεγαλύτερο βαθμό στο μάθημα ανεξαρτήτως εξέτασης.

**Λύση**

**ΠΡΟΓΡΑΜΜΑ** ΕΡΕΥΝΑ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** I, N1, N2

**ΠΡΑΓΜΑΤΙΚΕΣ:** ΑΠΟΤ\_1[1000], ΑΠΟΤ\_2[1000], ΜΟ\_1[1000], ΜΟ\_2[1000], ΜΑΧ

**ΧΑΡΑΚΤΗΡΕΣ:** ΟΝΟΜΑΤΑ[1000]

**ΑΡΧΗ**

**ΓΙΑ** I **ΑΠΟ** 1 **ΜΕΧΡΙ** 1000

**ΓΡΑΨΕ** 'ΔΩΣΕ ΤΟ ΟΝΟΜΑ ΤΟΥ', I, 'ου ΜΑΘΗΤΗ'

**ΔΙΑΒΑΣΕ** ΟΝΟΜΑΤΑ[I]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΚΑΛΕΣΕ** ΚΑΤΑΧΩΡΗΣΗ(ΑΠΟΤ\_1, ΟΝΟΜΑΤΑ)

**ΚΑΛΕΣΕ** ΚΑΤΑΧΩΡΗΣΗ(ΑΠΟΤ\_2, ΟΝΟΜΑΤΑ)

**ΚΑΛΕΣΕ** ΑΠΟΤΕΛΕΣΜΑΤΑ(ΑΠΟΤ\_1, ΜΟ\_1)

**ΚΑΛΕΣΕ** ΑΠΟΤΕΛΕΣΜΑΤΑ(ΑΠΟΤ\_2, ΜΟ\_2)

N1←0

N2←0

**ΓΙΑ** I **ΑΠΟ** 1 **ΜΕΧΡΙ** 1000

**ΑΝ** ΜΟ\_2[I] >= ΜΟ\_1[I] **ΤΟΤΕ**

N1←N1+1

**ΑΛΛΙΩΣ**

N2←N2+1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'ΠΟΣΟΣΤΟ ΜΑΘΗΤΩΝ ΠΟΥ ΑΥΞΗΣΑΝ Η ΔΙΑΤΗΡΗΣΑΝ ΤΗΝ ΑΠΟΔΟΣΗ &ΤΟΥΣ',  
N1/1000\*100, '%'

**ΓΡΑΨΕ** 'ΠΟΣΟΣΤΟ ΜΑΘΗΤΩΝ ΠΟΥ ΜΕΙΩΣΑΝ ΤΗΝ ΑΠΟΔΟΣΗ ΤΟΥΣ', N2/1000\*100, & '%'  
ΜΑΧ←0

**ΓΙΑ** I **ΑΠΟ** 1 **ΜΕΧΡΙ** 2000

**ΑΝ** I <= 1000 **ΤΟΤΕ**

**ΑΝ** ΑΠΟΤ\_1[I, 1] > ΜΑΧ **ΤΟΤΕ**

ΜΑΧ← ΑΠΟΤ\_1[I, 1]

**ΤΕΛΟΣ\_ΑΝ**

**ΑΛΛΙΩΣ**

**ΑΝ** ΑΠΟΤ\_2[I-1000, 1] > ΜΑΧ **ΤΟΤΕ**

ΜΑΧ← ΑΠΟΤ\_1[I-1000, 1]

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'ΜΑΘΗΤΗΣ/ΕΣ ΜΕ ΜΑΧ ΒΑΘΜΟ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΕΙΝΑΙ &Ο/ΟΙ:'

**ΓΙΑ** I **ΑΠΟ** I **ΜΕΧΡΙ** 1000

**ΑΝ** ΑΠΟΤ\_1[I, 1] = ΜΑΧ **Η** ΑΠΟΤ\_2[I, 1] = ΜΑΧ **ΤΟΤΕ**

**ΓΡΑΨΕ** ΟΝΟΜΑΤΑ[I]

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΔΙΑΔΙΚΑΣΙΑ** ΚΑΤΑΧΩΡΗΣΗ(Π, ΟΝ)

**ΜΕΤΑΒΛΗΤΕΣ**

```

ΠΡΑΓΜΑΤΙΚΕΣ: Π[1000,3]
ΑΚΕΡΑΙΕΣ: Ι, J
ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[1000]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 1000
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 3
        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
            ΓΡΑΨΕ 'ΔΩΣΕ ΤΟ ΒΑΘΜΟ ΤΟΥ ΜΑΘΗΤΗ', ΟΝΟΜΑΤΑ[Ι] , 'ΣΤΟ ΜΑΘΗΜΑ', J
            ΔΙΑΒΑΣΕ Π[Ι, J]
            ΜΕΧΡΙΣ_ΟΤΟΥ Π[Ι, J] >= 0 ΚΑΙ Π[Ι, J] <= 20
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΠΟΤΕΛΕΣΜΑΤΑ(Π, ΜΟ)
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: Π[1000,3], ΜΟ[1000], ΑΘΡ
ΑΚΕΡΑΙΕΣ: Ι, J
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 1000
    ΑΘΡ ← 0
        ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 3
            ΑΘΡ ← ΑΘΡ + Π[Ι, J]
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟ[Ι] ← ΑΘΡ / 3
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

## 7.4 Ασκήσεις προς επίλυση

1. Να γραφεί διαδικασία η οποία θα δέχεται την ώρα, τα λεπτά και τα δευτερόλεπτα και επιστρέφει το χρόνο σε ώρες, λεπτά, δευτερόλεπτα αυξημένο κατά ένα δευτερόλεπτο.

2. Δίνεται παρακάτω ένα πρόγραμμα με ένα υποπρόγραμμα:

**ΠΡΟΓΡΑΜΜΑ** Υπολογισμοί

ΜΕΤΑΒΛΗΤΕΣ

**ΠΡΑΓΜΑΤΙΚΕΣ:** α, β, γ

ΑΡΧΗ

**ΔΙΑΒΑΣΕ** α, β

γ ← α + Πράξη (α, β)

**ΓΡΑΨΕ** γ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΣΥΝΑΡΤΗΣΗ** Πράξη (χ, ψ): ΠΡΑΓΜΑΤΙΚΗ

ΜΕΤΑΒΛΗΤΕΣ

**ΠΡΑΓΜΑΤΙΚΕΣ:** χ, ψ

ΑΡΧΗ

**ΑΝ** χ >= ψ **ΤΟΤΕ**

Πράξη ← χ - ψ

**ΑΛΛΙΩΣ**

Πράξη ← χ + ψ

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

α. Να ξαναγράψετε το πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία χρησιμοποιώντας διαδικασία αντί συνάρτησης.

β. Να ξαναγράψετε το πρόγραμμα που δόθηκε αρχικά, ώστε να επιτελεί την ίδια λειτουργία χωρίς τη χρήση υποπρογράμματος.

γ. Να γράψετε στο τετράδιό σας τις τιμές που θα εμφανιστούν κατά την εκτέλεση του αρχικού προγράμματος που δόθηκε, αν ως τιμές εισόδου δοθούν οι αριθμοί: α = 10 και β = 5. (Θέμα Πανελλαδικών)

3. Ο αριθμός 368 έχει την ιδιότητα τα ψηφία του να αυξάνουν από τα αριστερά προς τα δεξιά (3,6,8). Ζητείται να γραφεί πρόγραμμα που να εμφανίζει όλους τους τριψήφιους ακέραιους αριθμούς που έχουν την ιδιότητα αυτή, καθώς επίσης και το πλήθος τους. Το πρόγραμμα θα κάνει κλήση μιας διαδικασίας η οποία θα δέχεται έναν οποιοδήποτε τριψήφιο ακέραιο αριθμό και θα τον διασπά στα τρία ψηφία του τα οποία και θα επιστρέφει στο κυρίως πρόγραμμα. Επίσης θα καλεί μια συνάρτηση που θα δέχεται τα τρία ψηφία ενός οποιουδήποτε τριψήφιου αριθμού και θα αποφασίζει αν αυτά πληρούν την παραπάνω ιδιότητα

4. Να γραφεί μια συνάρτηση με όνομα YesNo που θα διαβάζει την απάντηση του χρήστη στο (N/O) και θα επιστρέφει τη λογική τιμή αληθής ή ψευδής αν πατήσει το πλήκτρο N ή O αντίστοιχα.

**5.** Να γραφεί πρόγραμμα που θα διαβάζει τις τιμές τριών μεταβλητών Κεφαλαίο, Εριτοκίο και Ετί και στη συνέχεια θα υπολογίζει και θα εμφανίζει το συνολικό κεφάλαιο. Η εισαγωγή των στοιχείων θα γίνεται με τη βοήθεια διαδικασίας ενώ ο υπολογισμός του κεφαλαίου με τη βοήθεια συνάρτησης.

**6.** Να γραφεί ένα πρόγραμμα που θα διαβάζει έναν ακέραιο αριθμό και θα περιλαμβάνει μια συνάρτηση με όνομα Reverse, η οποία θα δέχεται τον ακέραιο αυτό και θα επιστρέφει τον αριθμό που προκύπτει από την αντιστροφή των ψηφίων του (π.χ. 271 σε 172). Η τιμή της συνάρτησης θα εμφανίζεται από το κυρίως πρόγραμμα.

**7.** Οι αριθμοί για τους οποίους το άθροισμα των κύβων των ψηφίων τους είναι ίσο με τον ίδιο τον αριθμό λέγονται αριθμοί Armstrong. (π.χ.  $153 = 1^3 + 5^3 + 3^3$ ). Να γραφεί πρόγραμμα που θα περιλαμβάνει τα παρακάτω:

- τη συνάρτηση Cube που θα επιστρέφει την τρίτη δύναμη ενός μονοψήφιου ακεραίου αριθμού.
- τη διαδικασία Separate που θα δέχεται έναν ακέραιο αριθμό και θα επιστρέφει τα ψηφία από τα οποία αποτελείται και
- τη συνάρτηση Sum\_Cube που θα δέχεται έναν ακέραιο αριθμό και θα επιστρέφει το άθροισμα των κύβων των ψηφίων του.

Το κυρίως πρόγραμμα θα βρίσκει και θα εμφανίζει ποιοι είναι αριθμοί Armstrong στο διάστημα (1-999).

**8.** Να γραφεί πρόγραμμα που θα περιλαμβάνει τα παρακάτω: μια συνάρτηση με όνομα Parag που θα δέχεται έναν ακέραιο αριθμό μικρότερο του 12 και θα επιστρέφει το παραγοντικό του και μια συνάρτηση με όνομα Force που θα βρίσκει τη n-οστή δύναμη ( $n$ =ακέραιος) ενός πραγματικού αριθμού  $x$ . Το πρόγραμμα θα βρίσκει την τιμή του  $\sinh x$  από τον τύπο:  $\sinh x = x - x^3/3! + x^5/5! - x^7/7! + \dots$

Ο υπολογισμός θα σταματά όταν υπολογιστεί ο όρος  $x^{11}/11!$  ή όταν η ακρίβεια του αποτελέσματος γίνει μικρότερη του 0.0001. Στη συνέχεια το κυρίως πρόγραμμα θα εμφανίζει το αποτέλεσμα, καθώς και τον εκθέτη του τελευταίου όρου που υπολογίστηκε (3, 5, 7 ή 11). Το  $x$  θα δίνεται σε μοίρες.

**9.** Να γραφεί υποπρόγραμμα για καθένα από τα παρακάτω:

- A) Να επιστρέφει το άθροισμα τριών ακεραίων αριθμών
- B) Να επιστρέφει το αποτέλεσμα της αναζήτησης, δηλαδή αν υπάρχει ή δεν υπάρχει και τη θέση ενός στοιχείου σε ένα μονοδιάστατο πίνακα.
- Γ) Να επιστρέφει τα ονόματα των δέκα καλύτερων βαθμολογικά μαθητών ενός σχολείου.

**10.** Η βιβλιοθήκη του δήμου, θέλει να οργανώσει τα διαθέσιμα βιβλία της ηλεκτρονικά. Διαθέτει 3000 βιβλία τα οποία θα αποθηκεύσει στον πίνακα Β[3000]. Επιπρόσθετα, στον πίνακα Σ[3000] θα αποθηκεύσει το όνομα του συγγραφέα. Η βιβλιοθήκη καταχωρεί σε έναν ακόμη πίνακα ΟΝ[1000] το ονοματεπώνυμο των δανειστών της. Ο δανεισμός των βιβλίων είναι μηνιαίος. Τέλος, σε έναν πίνακα Δ[3000, 12] καταχωρούνται οι δανεισμοί κάθε ενός από τα 3000 βιβλία για τους 12 μήνες του χρόνου. Αν κάποιο βιβλίο έχει δανειστεί για κάποιο μήνα τότε στην αντίστοιχη θέση του πίνακα καταχωρείται ο αύξων αριθμός του δανειστή από τον πίνακα ΟΝ, αλλιώς, αν το βιβλίο δεν είναι δανεισμένο μπαίνει ο αριθμός 0.

A. Να γίνει πρόγραμμα το οποίο:

1. Θα κάνει χρήση του υποπρογράμματος με όνομα ΕΙΣΟΔΟΣ το οποίο θα διαβάζει τους προαναφερθέντες πίνακες από το πληκτρολόγιο του χρήστη και θα τους επιστρέφει στο κύριο πρόγραμμα.
  2. Θα βρίσκει, και θα εμφανίζει για πιο βιβλίο (όνομα και συγγραφέας) έχουν γίνει οι περισσότεροι δανεισμοί. Αν είναι περισσότερα από ένα τότε να εμφανίζονται όλα.
  3. Θα εντοπίζει και θα εμφανίζει για κάθε μήνα, τον δανειστή που δανείστηκε τα περισσότερα βιβλία.
  4. Θα εντοπίζει και θα εμφανίζει τον πιο επιτυχημένο μήνα σε αριθμό δανεισμών.
  5. Θα διαβάζει το όνομα ενός συνδρομητή και να εμφανίζει, ποιον συγγραφέα συνήθως προτιμά ο συγκεκριμένος δανειστής.
- B. Να υλοποιήσετε το υποπρόγραμμα ΕΙΣΟΔΟΣ

**11.** Να γραφεί μια διαδικασία με όνομα  $swar(x, y)$  που θα εναλλάσσει τις τιμές 2 ακεραίων μεταβλητών  $x$  και  $y$ . Στη συνέχεια να γίνει ένα πρόγραμμα το οποίο θα διαβάζει τρεις ακέραιους αριθμούς και θα τους καταχωρεί στις μεταβλητές  $A, B, C$ , θα τους ταξινομεί κατά αύξουσα σειρά εναλλάσσοντας τις τιμές μεταξύ των μεταβλητών με τη βοήθεια της διαδικασίας  $swar(x,y)$ , έτσι ώστε  $A < B < C$ . Τέλος θα εμφανίζει τις ταξινομημένες τιμές των μεταβλητών  $A, B$  και  $C$ .

**12.** Γράψτε μια διαδικασία η οποία: θα στρογγυλοποιεί ένα δεδομένο ακέραιο αριθμό στην πλησιέστερη δεκάδα, εκατοντάδα, χιλιάδα κ.λπ. (ο χρήστης θα προσδιορίζει την επιθυμητή στρογγυλοποίηση) και θα βρίσκει το υπόλοιπο - το λεγόμενο "σφάλμα στρογγυλοποιήσεως" - που θα προκύπτει κάθε φορά. Αν για παράδειγμα ο αριθμός 17579 στρογγυλοποιηθεί στην πλησιέστερη εκατοντάδα (3ο ψηφίο από δεξιά), θα επιστρέφεται ο αριθμός 17600 και το υπόλοιπο -21, ενώ αν ο αριθμός 17579 στρογγυλοποιηθεί στην πλησιέστερη δεκάδα (2ο ψηφίο από δεξιά), θα επιστρέφεται ο αριθμός 17580 και το υπόλοιπο -1. Η κλήση της διαδικασίας θα γίνεται με τέσσερις παραμέτρους: τον ακέραιο αριθμό που δίνεται, το ψηφίο στο οποίο θα γίνεται η στρογγυλοποίηση, το νέο αριθμό που προκύπτει και το υπόλοιπο.

**13.** Δίνεται το παρακάτω πρόγραμμα και υποπρογράμματα:

**ΠΡΟΓΡΑΜΜΑ** Κύριο

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ :** A, B, Γ, I

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** A, B, Γ

I ← B

**ΟΣΟ** I ≤ B + 10 **ΕΠΑΝΑΛΑΒΕ**

**ΚΑΛΕΣΕ** Διαδ2(Γ, A, I)

**ΓΡΑΨΕ** I, A, Γ

I ← I + 4

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΔΙΑΔΙΚΑΣΙΑ** Διαδ2(Γ, A, B)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ :** A, B, Γ

**ΑΡΧΗ**

$$A \leftarrow A + 3$$

$$\Gamma \leftarrow A + B$$

ΓΡΑΨΕ Α, Β, Γ

### ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

Τι θα εμφανιστεί κατά την εκτέλεση του προγράμματος, αν ως τιμές εισόδου δοθούν οι αριθμοί 7, 10, 14;

**14.** Να γραφεί υποπρόγραμμα το οποίο δέχεται πίνακα Α 10\*10 ακεραίων και τον επιστρέφει μηδενίζοντας την κύρια διαγώνιό του.

**15.** Να γραφεί λογική συνάρτηση η οποία δέχεται τυχαίο θετικό ακέραιο και ελέγχει αν είναι παλινδρομικός ή όχι.

**16.** Να γραφεί συνάρτηση η οποία δέχεται τυχαίο θετικό ακέραιο και τον αντιστρέφει.

**17.** Να γραφεί υποπρόγραμμα που δέχεται τις καρτεσιανές συντεταγμένες x,y ενός σημείου και θα επιστρέφει τις πολικές του r,θ σύμφωνα με τους τύπους:  $x=r*\cos(\theta)$  και  $y=r*\sin(\theta)$ .

**18.** Να γραφεί υποπρόγραμμα το οποίο δέχεται ένα μονοδιάστατο πίνακα Ν ακεραίων και δημιουργεί ένα νέο πίνακα που περιέχει όλα τα στοιχεία του αρχικού, αλλά μόνο μία φορά το καθένα.

**19.** Η Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης καταγράφει σε πίνακες στοιχεία που αφορούν τις πενήνήμερες εκδρομές της Γ' τάξης των λυκείων του νομού. Αναλυτικότερα καταγράφει στον πίνακα ΣΧΟΛΕΙΟ[30] τα ονόματα των λυκείων, στον πίνακα ΜΑΘΗΤΕΣ[30] αντίστοιχα τον αριθμό των μαθητών που θα πάρουν μέρος στην εκδρομή και στον πίνακα ΠΡΟΟΡΙΣΜΟΣ[30] τον προορισμό της εκδρομής κάθε σχολείου. Για το συγκεκριμένο σχολικό έτος αποφασίστηκε οι πιθανοί προορισμοί να είναι μόνο τρεις: Η Θεσσαλονίκη, η Ρόδος και η Κύπρος. Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού ΓΛΩΣΣΑ το οποίο:

1. Να διαβάζει και να αποθηκεύει στους παραπάνω πίνακες τα δεδομένα ελέγχοντας την εγκυρότητα του προορισμού.
2. Να ζητάει το όνομα ενός σχολείου και να εμφανίζει το πλήθος των μαθητών του που συμμετέχουν στην εκδρομή καθώς και τον προορισμό της εκδρομής. Για το σκοπό αυτό να καλεί υποπρόγραμμα το οποίο να δέχεται τον πίνακα ΣΧΟΛΕΙΟ[30] και το όνομα του σχολείου και να επιστρέφει τη θέση (1-30) του σχολείου στον πίνακα.
3. Να εμφανίζει τα ονόματα των δέκα σχολείων με τους περισσότερους συμμετέχοντες.
4. Να βρίσκει και να εμφανίζει το δημοφιλέστερο προορισμό, δηλαδή αυτόν με τους περισσότερους συμμετέχοντες συνολικά.

**20.** Μία εταιρεία διοργάνωσε το 2005 δέκα εξειδικευμένες εκθέσεις (π.χ. Εκπαιδευτική, Διεθνής, Αυτοκινήτου, κ.λπ.). Τα ονόματα των δέκα εκθέσεων καταχωρούνται σε έναν πίνακα με το όνομα ektheseis. Σε ένα παράλληλο πίνακα με όνομα episkeptes καταχωρείται ο αριθμός των επισκεπτών της κάθε έκθεσης.

Να γράψετε πρόγραμμα με το όνομα askisi12, το οποίο να:

(α) ζητά από το χρήστη για κάθε έκθεση, το όνομά της και τον αριθμό των επισκεπτών και να τοποθετεί τις πληροφορίες αυτές στις κατάλληλες θέσεις των δύο πινάκων.

(β) χρησιμοποιεί μια διαδικασία με το όνομα report, η οποία να δέχεται από το κυρίως



πρόγραμμα τον πίνακα επισκερτες ως παράμετρο και να υπολογίζει και να επιστρέφει στο κυρίως πρόγραμμα τον συνολικό αριθμό επισκεπτών όλων των εκθέσεων και τον αριθμό των εκθέσεων που είχαν περισσότερους από 45000 επισκέπτες. Το κυρίως πρόγραμμα να τυπώνει τις τιμές αυτές.

(γ) τυπώνει τα ονόματα των εκθέσεων ταξινομημένα κατά φθίνουσα σειρά με βάση τον αριθμό επισκεπτών. Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων.

**21.** Μια αλυσίδα κινηματογράφων διαθέτει δέκα αίθουσες. Να γράψετε πρόγραμμα με όνομα askisi12, το οποίο:

(α) να διαβάζει τα ονόματα των αιθουσών και να τα καταχωρεί σε μονοδιάστατο πίνακα με το όνομα onomata.

(β) να διαβάζει τις μηνιαίες εισπράξεις κάθε αίθουσας για κάθε μήνα ενός χρόνου και να τις τοποθετεί σε δισδιάστατο πίνακα με το όνομα eispraxeis.

(γ) να υπολογίζει τη μέση μηνιαία τιμή των εισπράξεων για κάθε αίθουσα και να τις αποθηκεύει στο μονοδιάστατο πίνακα mesi.

(δ) να βρίσκει τη μικρότερη και τη μεγαλύτερη μέση μηνιαία τιμή χρησιμοποιώντας μια διαδικασία με το όνομα MinMax. Η διαδικασία αυτή να δέχεται από το κυρίως πρόγραμμα τον πίνακα mesi και να επιστρέφει πίσω στο κυρίως πρόγραμμα τη μικρότερη και τη μεγαλύτερη μέση μηνιαία τιμή, όπου και θα τυπώνονται.

(ε) να τυπώνει τα ονόματα των τεσσάρων (4) αιθουσών με τις ψηλότερες μέσες μηνιαίες εισπράξεις (για σκοπούς απλούστευσης της άσκησης θεωρείστε ότι όλες οι μέσες μηνιαίες τιμές των εισπράξεων είναι μοναδικές).

Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων.

**22.** Να γραφεί πρόγραμμα με χρήση των ακόλουθων υποπρογραμμάτων.

a) Υποπρόγραμμα που θα γεμίζει με τυχαίους αριθμούς έναν πίνακα a 100 θέσεων μνήμης.

b) Υποπρόγραμμα που θα εμφανίζει τα στοιχεία μονοδιάστατου πίνακα.

c) Υποπρόγραμμα που θα διαβάζει έναν ακέραιο αριθμό.

d) Υποπρόγραμμα που θα ελέγχει και θα καταχωρεί σε ένα νέο πίνακα b, όλους τους αριθμούς του πίνακα a που είναι μεγαλύτεροι από τον αριθμό ar. Το κυρίως πρόγραμμα θα καλεί κατάλληλα τα παραπάνω υποπρογράμματα

**23.** Μια εταιρεία κινητής τηλεφωνίας προσφέρει το πρόγραμμα Min240+15. Σύμφωνα με το πρόγραμμα αυτό, ο συνδρομητής έχει 240 λεπτά δωρεάν χρόνο ομιλίας και 15 δωρεάν μηνύματα. Το πάγιο του προγράμματος είναι 24€. Η χρέωση κάθε μηνύματος (πέραν του 15<sup>ου</sup>) είναι 0,09€/μήνυμα.

Για κάθε τηλεφώνημα υπάρχει ελάχιστη χρονοχρέωση 30sec, δηλ. αν ένα τηλεφώνημα διαρκεί λιγότερο των 30 sec χρεώνεται για 30 sec.

Για τα επιπλέον λεπτά ομιλίας η χρέωση είναι κλιμακωτή σύμφωνα με τον παρακάτω πίνακα.

Πρόγραμμα Min240+15	
Πάγιο 24 €	
Χρέωση Μηνυμάτων: 0.09 € /SMS πέραν του 15 <sup>ου</sup>	
Χρόνος τηλεφωνημάτων (λεπτά)	Χρονοχρέωση

	(€ / δευτερόλεπτο)
0 - 240	0
μέχρι και 300	0.0020 €
πάνω από 300	0.0018 €

A. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο :

α. περιλαμβάνει το τμήμα δηλώσεων.

β. διαβάζει το ονοματεπώνυμο ενός συνδρομητή που έχει σύνδεση προγράμματος min240+15 καθώς και τον αριθμό μηνυμάτων που έστειλε.

γ. για κάθε τηλεφώνημα του συνδρομητή διαβάζει τη διάρκεια του. Η εισαγωγή τερματίζεται όταν δοθεί διάρκεια=0.

δ. υπολογίζει και εμφανίζει τη συνολική διάρκεια των τηλεφωνημάτων για την οποία χρεώνεται ο συνδρομητής.

ε. υπολογίζει με χρήση υποπρογράμματος, τη χρέωση για το συνδρομητή, την οποία και στη συνέχεια εμφανίζει.

στ. υπολογίζει και εμφανίζει το Φ.Π.Α 19%, καθώς και το πληρωτέο ποσό.

B. Να αναπτύξετε το υποπρόγραμμα του ερωτήματος A.ε.

24. Να γίνει πρόγραμμα που θα διαβάζει 2 αριθμούς M, N στο κυρίως πρόγραμμα, και στη συνέχεια θα διαβάζει τα MxN στοιχεία ενός διδιάστατου πίνακα A (μέγιστη διάσταση 50x50) σε μια διαδικασία Read\_data. Στη συνέχεια, σε μια διαδικασία Change\_array να προσδιοριστεί το μέγιστο στοιχείο κάθε γραμμής του διδιάστατου πίνακα A κι όσα στοιχεία είναι αριστερά του μέγιστου να γίνουν ίσα με αυτό (δηλαδή όσα είναι στην ίδια γραμμή με αυτό, αλλά έχουν μικρότερο δείκτη στήλης να γίνουν ίσα με το μέγιστο). Η διαδικασία αυτή να επαναληφθεί σε εκείνα τα στοιχεία που είναι δεξιά του (δηλαδή είναι στην ίδια γραμμή με αυτό, αλλά έχουν μεγαλύτερο δείκτη στήλης), μέχρι να εξαντληθεί η γραμμή. Τέλος, σε μια διαδικασία Print\_array να εμφανιστούν τα στοιχεία του πίνακα κατά γραμμές με ένα κενό χαρακτήρα ανά στοιχείο.

25. Δίνεται το παρακάτω πρόγραμμα

**ΠΡΟΓΡΑΜΜΑ** Θ2

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, A[6], M1, M2

**ΑΡΧΗ**

! Τμήμα A - Εισαγωγή Δεδομένων

**Για** i **από** 1 **μέχρι** 6

**Αρχή\_επανάληψης**

**Διάβασε** A[i]

**Μέχρις\_ότου** A[i] > 0

**Τέλος\_επανάληψης**

! Τμήμα B - Υπολογισμοί

    M1 ← A[1]

    M2 ← 0

**Για** i **από** 2 **μέχρι** 6

**Αν** A[i] > M1 **τότε**

                M2 ← M1

                M1 ← A[i]

Αλλιώς  
**Αν**  $A[i] > M2$  **τότε**  
 $M2 \leftarrow A[i]$   
**Τέλος\_Αν**  
**Τέλος\_Αν**

**Τέλος\_επανάληψης**

! Τμήμα Γ - Εμφάνιση Αποτελεσμάτων

**Γράψε** M1, M2

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** Θ2

Για κάθε ένα από τα τμήματα Α, Β και Γ, να αναπτύξετε κατάλληλο υποπρόγραμμα που επιτελεί τις ίδιες εργασίες. Να ξαναγράψετε το πρόγραμμα, κάνοντας κλήση και των τριών παραπάνω υποπρογραμμάτων.

**26.** Η παρακάτω συνάρτηση υπολογίζει τον μικρότερο από δυο (2) ακέραιους αριθμούς. Χρησιμοποιώντας τη συνάρτηση που σας δόθηκε να κατασκευάσετε συνάρτηση η οποία θα υπολογίζει τον μικρότερο από τρεις (3) αριθμούς. Σημείωση: οι αριθμοί είναι διάφοροι μεταξύ τους.

**ΣΥΝΑΡΤΗΣΗ ΜΙΚΡΟΤΕΡΟΣ2 (Α,Β) : ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** Α,Β

**ΑΡΧΗ**

**ΑΝ**  $A > B$  **ΤΟΤΕ**

$ΜΙΚΡΟΤΕΡΟΣ2 \leftarrow A$

**ΑΛΛΙΩΣ**

$ΜΙΚΡΟΤΕΡΟΣ2 \leftarrow B$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

Να γράψετε κύριο πρόγραμμα το οποίο θα διαβάζει 3 αριθμούς και θα εκτυπώνει τον μικρότερο από αυτούς χρησιμοποιώντας τη συνάρτηση που σας δόθηκε και τη συνάρτηση που κατασκευάσατε στο προηγούμενο ερώτημα.

**27.** Το πανελλήνιο πρωτάθλημα ποδηλασίας χωρίζεται σε δυο ομίλους, το βόρειο και το νότιο όμιλο. Σε κάθε όμιλο συμμετέχουν 20 αθλητές σε 15 διαφορετικούς αγώνες. Τα ονόματα των ποδηλατών καταχωρίζονται σε πίνακες ΒΟΡ[20] και ΝΟΤ[20], ενώ η σειρά κατάταξης τους σε κάθε αγώνα σε πίνακες ΚΑΤΒ[20,15] και ΚΑΤΝ[20,15]. Να δημιουργήσετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Δημιουργεί τους παραπάνω πίνακες. Θεωρήστε ότι οι πίνακες κατάταξης περιέχουν αριθμούς από το 1 ως το 20 και δε χρειάζεται έλεγχος εγκυρότητας δεδομένων.
2. Εμφανίζει το όνομα του ποδηλάτη με τις περισσότερες πρωτιές σε κάθε όμιλο και τον αριθμό των νικών του, χρησιμοποιώντας το υποπρόγραμμα που περιγράφεται στο ερώτημα 4.
3. Υπολογίζει και εμφανίζει το όνομα του πρωταθλητή Ελλάδας, αυτόν δηλαδή που έχει τις περισσότερες νίκες και από τους δυο ομίλους. Σε περίπτωση ίσου αριθμού νικών εμφανίζει «ισοπαλία».

Να κατασκευάσετε κατάλληλο υποπρόγραμμα που θα δέχεται τον πίνακα με τα ονόματα και τον πίνακα με τη σειρά κατάταξης των ποδηλατών ενός ομίλου και θα επιστρέφει το όνομα και τον αριθμό των νικών του πρωταθλητή του ομίλου. Η εμφάνιση των

αποτελεσμάτων θα γίνεται στο κύριο πρόγραμμα.

**28.** Δίνονται το κύριο πρόγραμμα και μια διαδικασία.

<b>ΠΡΟΓΡΑΜΜΑ</b> Ασκ1	<b>ΔΙΑΔΙΚΑΣΙΑ</b> Minimum(χ,υ,ζ)
<b>ΜΕΤΑΒΛΗΤΕΣ</b>	<b>ΜΕΤΑΒΛΗΤΕΣ</b>
<b>ΑΚΕΡΑΙΕΣ:</b> Χ,Υ,Ζ	<b>ΑΚΕΡΑΙΕΣ:</b> Α,Β,Γ
<b>ΑΡΧΗ</b>	<b>ΑΡΧΗ</b>
<b>ΔΙΑΒΑΣΕ</b> Χ,Υ	Γ←Α
<b>ΚΑΛΕΣΕ</b> Minimum (Χ,Υ,Ζ)	<b>ΑΝ</b> Γ>Β <b>ΤΟΤΕ</b>
<b>ΓΡΑΨΕ</b> Ζ	Γ←Β
<b>ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ</b>	<b>ΤΕΛΟΣ_ΑΝ</b>
	<b>ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ</b>

Να ξαναγραφεί το κύριο πρόγραμμα και η αντίστοιχη συνάρτηση που θα αντικαταστήσει τη διαδικασία.

**29.** Να γραφεί ένα πρόγραμμα που θα διαβάζει έναν αριθμό (μεταβλητή number) και έναν αριθμό (μεταβλητή places) και θα περιλαμβάνει μια συνάρτηση με όνομα ceil\_dec(number, places), η οποία θα επιστρέφει την τιμή της μεταβλητής number στρογγυλοποιημένη με τον τρόπο που καθορίζει η συνάρτηση ceil() στον καθορισμένο αριθμό δεκαδικών ψηφίων places (π.χ. ceil\_dec(12.547,0), ceil\_dec(12.547,1), ceil\_dec(12.547,2) θα επιστρέφουν 13, 12.6 και 12.55 αντίστοιχα). Η εμφάνιση της στρογγυλοποιημένης τιμής της μεταβλητής number θα γίνεται από το κυρίως πρόγραμμα.

**30.** Να γραφεί πρόγραμμα που θα διαβάζει τις τιμές τριών μεταβλητών Κεφαλαίο, Επιτόκιο, (π.χ. για επιτόκιο 12% .. 0.12) και Ετι και στη συνέχεια θα υπολογίζει και θα εμφανίζει το συνολικό κεφάλαιο. Η εισαγωγή των στοιχείων θα γίνεται στο κυρίως πρόγραμμα και ο υπολογισμός του κεφαλαίου θα γίνεται με τη βοήθεια συνάρτησης. Για τον υπολογισμό του ετήσιου κεφαλαίου θα χρησιμοποιείται ο τύπος (Κεφάλαιο\*(1+Επιτόκιο)).

**31.** Γράψτε ένα πρόγραμμα καθοδηγούμενο από μενού επιλογών (menu-driven program) το οποίο μετατρέπει διάφορα μεγέθη από ένα είδος μονάδων σε άλλο, με τη βοήθεια συναρτήσεων:

- συνάρτηση που δέχεται λεπτά και επιστρέφει ώρες,
- συνάρτηση που δέχεται πόδια και επιστρέφει μέτρα (1 πόδι = 0.3048 μέτρα),
- συνάρτηση που δέχεται βαθμούς Celsius και επιστρέφει βαθμούς Fahrenheit ( $C=5/9*(F-32)$ ),
- συνάρτηση που δέχεται βαθμούς Fahrenheit και επιστρέφει βαθμούς Celsius.

Για την εμφάνιση του παρακάτω μενού επιλογών και το διάβασμα της επιλογής του χρήστη θα χρησιμοποιηθεί συνάρτηση:

AVAILABLE OPTIONS:

1. CONVERT MINUTES TO HOURS
2. CONVERT FEET TO METERS
3. CONVERT DEGREES CELSIUS TO DEGREES FAHRENHEIT
4. CONVERT DEGREES FAHRENHEIT TO DEGREES CELSIUS
5. QUIT

Όταν ο χρήστης επιλέξει μια από τις επιλογές του μενού στη συνέχεια θα δίνει την

αντίστοιχη τιμή για μετατροπή. Το αποτέλεσμα της μετατροπής θα εμφανίζεται στο κυρίως πρόγραμμα και όχι στη συνάρτηση. Πατώντας ENTER θα επαναλαμβάνεται το αρχικό μενού ώστε να συνεχίσει ο χρήστης να καλεί νέα διαδικασία.

**32.** Η ακολουθία Fibonacci ορίζεται ως εξής: ο πρώτος της όρος είναι 0, ο δεύτερος 1 και από εκεί και πέρα κάθε όρος προκύπτει από το άθροισμα των δύο τελευταίων όρων: 0 1 1 2 3 5 8 13 21 34 55...

Να γράψετε μια συνάρτηση Fibonacci, η οποία θα δέχεται ένα θετικό ακέραιο N και θα επιστρέφει το N-οστό όρο της ακολουθίας Fibonacci. Στη συνέχεια, να γράψετε ένα πρόγραμμα στο οποίο θα διαβάζεται το N και χρησιμοποιώντας την παραπάνω συνάρτηση θα υπολογίζονται και εμφανίζονται οι N πρώτοι όροι της ακολουθίας. Κάθε όρος της ακολουθίας θα εμφανίζεται σε διαφορετική γραμμή.

**33.** Να γράψετε συνάρτηση η οποία δέχεται έναν πραγματικό αριθμό και επιστρέφει το μήνυμα "πραγματικός" αν ήταν πραγματικός, αλλιώς το μήνυμα "άρτιος" ή "περιττός".

**34.** Να φτιάξετε συνάρτηση με όνομα ΣΤΡΟΓΓΥΛΟΠΟΙΗΣΗ που θα δέχεται σαν παράμετρο έναν πραγματικό αριθμό και θα τον επιστρέφει στρογγυλοποιημένο ως προς τη μονάδα. Δηλαδή, όταν καλέσουμε την συνάρτηση ΣΤΡΟΓΓΥΛΟΠΟΙΗΣΗ(14,219) θα πρέπει να μας επιστρέφει 14 και όταν καλούμε ΣΤΡΟΓΓΥΛΟΠΟΙΗΣΗ(-9,679) θα επιστρέφει -10.

**35.** Δίνονται οι απαραίτητοι ορισμοί:

Σκαληνό ονομάζεται το τρίγωνο που δεν έχει ίσες μεταξύ τους πλευρές.

Ισοσκελές ονομάζεται το τρίγωνο που έχει μόνο δύο ίσες μεταξύ τους πλευρές.

Ισόπλευρο ονομάζεται το τρίγωνο που όλες του οι πλευρές είναι μεταξύ τους ίσες.

Για τρία ευθύγραμμα τμήματα μήκους  $\alpha$ ,  $\beta$ ,  $\gamma$ , η τριγωνική ανισότητα ικανοποιείται όταν κάθε ένα από αυτά είναι μικρότερο από το άθροισμα των άλλων δύο μαζί. Να γράψετε πρόγραμμα το οποίο:

1. θα περιλαμβάνει τμήμα δήλωσης μεταβλητών
2. θα διαβάζει τα μήκη τριών ευθύγραμμων τμημάτων, ελέγχοντας ότι ικανοποιούν την τριγωνική ανισότητα
3. θα καλεί υποπρόγραμμα που θα κατασκευάσετε, το οποίο θα παίρνει ως είσοδο τα μήκη των τριών πλευρών, και θα επιστρέφει ως έξοδο το είδος του τριγώνου, δηλαδή τη λέξη «ΣΚΑΛΗΝΟ», «ΙΣΟΣΚΕΛΕΣ» ή «ΙΣΟΠΛΕΥΡΟ».
4. θα επαναλαμβάνει τα δύο προηγούμενα βήματα ώσπου να προκύψουν τρία συνεχόμενα τρίγωνα που να κατατάσσονται σε άλλο είδος το καθένα.

**36.** Ένα λεωφορείο που χωρά το πολύ 82 επιβάτες ξεκινά άδειο τη διαδρομή του, το πλήθος των σταθμών της οποίας είναι άγνωστο. Γράψτε πρόγραμμα με το οποίο θα προσομοιώνεται η διαδρομή του λεωφορείου ως εξής:

(a) Θα διαβάζεται κάθε φορά το όνομα του επόμενου σταθμού του λεωφορείου. Η διαδρομή του λεωφορείου (και το πρόγραμμα) θα ολοκληρώνεται όταν ως επόμενος σταθμός δοθεί η λέξη «τέρμα».

(b) Σε κάθε σταθμό εξαιρώντας τον πρώτο, τον τελευταίο, αλλά και όλους εκείνους στους οποίους το λεωφορείο είναι ήδη άδειο, και αξιοποιώντας κατάλληλα τη διαδικασία, θα διαβάζεται ο αριθμός των επιβατών που κατεβαίνουν, με τον περιορισμό να μην κατέβουν περισσότεροι απ' όσους είναι εκείνη τη στιγμή μέσα.

(c) Σε κάθε σταθμό εξαιρώντας τον τελευταίο αλλά και όλους εκείνους στους οποίους το λεωφορείο είναι ήδη γεμάτο, και αξιοποιώντας κατάλληλα τη διαδικασία, θα

διαβάζεται ο αριθμός των επιβατών που ανεβαίνουν, με τον περιορισμό να μην ανέβουν περισσότεροι απ' όσους χωράνε εκείνη τη στιγμή να ανέβουν.

(d) Στο τέλος θα εμφανίζεται το πλήθος των επιβατών που πήραν το λεωφορείο κατά μήκος ολόκληρης της διαδρομής του.

**37.** Γράψτε υποπρόγραμμα το οποίο ως παραμέτρους εισόδου θα δέχεται τους συντελεστές  $\alpha, \beta$  μιας πρωτοβάθμιας εξίσωσης και θα εξετάζει τις απαραίτητες περιπτώσεις για να εμφανίσει το κατάλληλο μήνυμα, ανάλογα με το αν η εξίσωση είναι αδύνατη, αόριστη, ή έχει μία ρίζα (δεν απαιτείται ο υπολογισμός της ρίζας).

**38.** Γράψτε υποπρόγραμμα το οποίο ως παραμέτρους εισόδου θα δέχεται τους συντελεστές  $\alpha, \beta, \gamma$  μιας δευτεροβάθμιας εξίσωσης και θα εξετάζει τις απαραίτητες περιπτώσεις για να εμφανίσει το κατάλληλο μήνυμα, ανάλογα με το αν η εξίσωση είναι αδύνατη, έχει μία διπλή ρίζα, ή έχει δύο ρίζες (δεν απαιτείται ο υπολογισμός των ριζών).

Γράψτε πρόγραμμα το οποίο:

1. θα ρωτά το χρήστη αν επιθυμεί τη διερεύνηση μιας δευτεροβάθμιας εξίσωσης.
2. θα εμφανίζει τη φράση «παρακαλώ απαντήστε με ένα ναι ή ένα όχι» κάθε φορά που ο χρήστης δίνει άλλη απάντηση.
3. κάθε φορά που ο χρήστης απαντά «ναι», θα του ζητά να εισάγει τους συντελεστές  $\alpha, \beta, \gamma$  της δευτεροβάθμιας εξίσωσης, και θα καλεί το κατάλληλο υποπρόγραμμα, ανάλογα με το αν οι δοθέντες συντελεστές συνιστούν όντως δευτεροβάθμια εξίσωση ή πρωτοβάθμια.
4. θα τερματίζεται όταν ο χρήστης απαντήσει «όχι».

**39.** Να δημιουργήσετε κατάλληλο υποπρόγραμμα το οποίο δέχεται παραμέτρους έναν αλφαριθμητικό πίνακα  $A[30,20]$ , δύο μεταβλητές  $x, y$  που αντιστοιχούν σε δύο γραμμές του ( $x < y$ ) και μία μεταβλητή χαρακτήρα  $On$ . Το υποπρόγραμμα ελέγχει αν η τιμή της  $On$  υπάρχει στον πίνακα μεταξύ των γραμμών  $x$  και  $y$  επιστρέφοντας την κατάλληλη λογική τιμή. Σε περίπτωση επιτυχούς αναζήτησης να σταματά.

**40.** Ένα ξενοδοχείο έχει 3 πτέρυγες και εξυπηρετεί συνολικά 100 πελάτες. Να δοθεί πρόγραμμα που να αποθηκεύει σε πίνακα  $ON[100]$  το ονοματεπώνυμο του κάθε πελάτη, σε πίνακα  $YΠ[100,3]$  να αποθηκεύει στην πρώτη στήλη αριθμό ημερών που έμεινε στο ξενοδοχείο, στη δεύτερη στήλη το κόστος ανά ημέρα και στην τρίτη στήλη το κόστος από τις έξτρα υπηρεσίες που είχε στο δωμάτιο και σε πίνακα  $ΠΤΕΡ[100]$  να αποθηκεύει την πτέρυγα που έμεινε ο κάθε πελάτης (όνομα πτέρυγας  $A, B, \Gamma, \dots$ ).

1. Να καλεί διαδικασία η οποία να ταξινομεί κατά αλφαβητική σειρά τις πτέρυγες κρατώντας σε αντιστοιχία όλους τους άλλους πίνακες.
2. Να υπολογίζει για κάθε πελάτη το ποσό που θα πληρώσει και να το αποθηκεύει σε πίνακα  $ΠΟΣΟ[100]$ .
3. Εμφανίζει το όνομα του καλύτερου πελάτη και το όνομα της πτέρυγας που έμεινε.
4. Εμφανίζει το όνομα του καλύτερου πελάτη ανά πτέρυγα μαζί με το ποσό που θα πληρώσει.
5. Να γραφεί η διαδικασία του ερωτήματος 1.

**41.** Να γράψετε συνάρτηση με το όνομα `multiple`, η οποία να δέχεται δύο ακέραιους αριθμούς  $A$  και  $B$  (να θεωρήσετε ότι ο  $A$  είναι μεγαλύτερος του  $B$  και δε χρειάζεται έλεγχος) και να επιστρέφει τη λογική τιμή  $ΑΛΗΘΗΣ$  αν ο αριθμός  $A$  είναι πολλαπλάσιος του  $B$ , διαφορετικά να επιστρέφει τη λογική τιμή  $ΨΕΥΔΗΣ$ .

**42.** Ένας ελαιοπαραγωγός διαθέτει δοχεία των πέντε λίτρων, των τριών λίτρων και του ενός λίτρου για αποθήκευση λαδιού. Να γράψετε συνάρτηση με το όνομα  $M(x)$  δοχεία, η οποία να δέχεται από το κυρίως πρόγραμμα την ποσότητα του λαδιού σε λίτρα (ως ακέραιο αριθμό) και να επιστρέφει το μικρότερο αριθμό δοχείων που μπορούν να χρησιμοποιηθούν από τον ελαιοπαραγωγό για αποθήκευση του λαδιού.

Παράδειγμα: Αν η συνάρτηση δεχτεί 504 λίτρα τότε θα επιστρέψει τον αριθμό 102 (δηλαδή 100 δοχεία των 5 λίτρων + 1 δοχείο των 3 λίτρων + 1 δοχείο του 1 λίτρου).

**43.** Να γραφεί διαδικασία με όνομα  $P(x)$ , η οποία δέχεται την ταχύτητα μιας μοτοσυκλέτας και υπολογίζει το πρόστιμο με βάση τον ακόλουθο πίνακα. Το υποπρόγραμμα επιστρέφει το πρόστιμο και κατάλληλο μήνυμα.

Ταχύτητα (Km/h)	Πρόστιμο	Μήνυμα
1-90	0	Καμία υπέρβαση
91-120	1,5 ευρώ ανά Km πέραν των 90	Τρεις βαθμοί ποινής
121 και άνω	0	Αφαίρεση πινακίδων & προσαγωγή σε δίκη

## ΠΑΡΑΡΤΗΜΑ 1 - Μερικές ενδεικτικές ερωτήσεις θεωρίας

1. Διατυπώστε τους κανόνες της διαγραμματικής αναπαράστασης της δομής ενός προβλήματος.
2. Σύγκριση ανάμεσα στον ανθρώπινο νου και στον υπολογιστή.
3. Τι είναι η λειτουργία της ολίσθησης.
4. Να περιγράψετε τα προβλήματα που είναι δυνατόν να παρουσιαστούν κατά την αναπαράσταση ενός αλγορίθμου χρησιμοποιώντας ελεύθερο κείμενο και φυσική γλώσσα κατά βήματα.
5. Δώστε τον αλγόριθμο του πολλαπλασιασμού αλά ρωσικά.
6. Διατυπώστε τον ορισμό της ταξινόμησης.
7. Τι γνωρίζετε για τις δομές δεδομένων δευτερεύουσας μνήμης;
8. Ποια στάδια περιλαμβάνει η επίλυση ενός προβλήματος με τη βοήθεια του υπολογιστή.
9. Τι είναι ένα πρόγραμμα σε γλώσσα μηχανής.
10. Τι είναι ο οδηγούμενος από το γεγονός προγραμματισμός.
11. Τι είναι ο οπτικός προγραμματισμός.
12. Τι γνωρίζετε για τις γλώσσες 4ης γενιάς.
13. Ποιες είναι οι διαφορές ανάμεσα στις φυσικές και στις τεχνητές γλώσσες.
14. Τι γνωρίζετε για την ιεραρχική σχεδίαση.
15. Τι γνωρίζετε για την εντολή GOTO.
16. Τι γνωρίζετε για το δομημένο προγραμματισμό.
17. Τι είναι ο συντάκτης.
18. Τι περιλαμβάνει ένα προγραμματιστικό περιβάλλον.
19. Ποια είναι η γενική μορφή του τμήματος δήλωσης συμβολικών σταθερών σε ένα πρόγραμμα.
20. Ποια είναι η γενική μορφή του τμήματος δήλωσης μεταβλητών σε ένα πρόγραμμα.
21. Τι είναι ένα υποπρόγραμμα.
22. Να αναφέρετε τα πλεονεκτήματα του τμηματικού προγραμματισμού.
23. Ποια είναι η γενική μορφή μιας συνάρτησης; και μιας διαδικασίας.
24. Τι είναι η διεύθυνση επιστροφής ενός υποπρογράμματος.
25. Τι γνωρίζετε για τη στοίβα χρόνου εκτέλεσης.





## ΠΑΡΑΡΤΗΜΑ 2 - ΘΕΜΑΤΑ ΠΑΝΕΛΛΑΔΙΚΩΝ 2013

### ΗΜΕΡΗΣΙΑ 2013

#### ΘΕΜΑ Α

**A1.** Να γράψετε στο τετράδιό σας τον αριθμό καθεμιάς από τις παρακάτω προτάσεις 1-6 και δίπλα τη λέξη ΣΩΣΤΟ, αν η πρόταση είναι σωστή, ή τη λέξη ΛΑΘΟΣ, αν η πρόταση είναι λανθασμένη.

1. Η τιμή μιας μεταβλητής και ο τύπος της μπορούν να αλλάζουν κατά την εκτέλεση ενός προγράμματος.
2. Όταν υπάρχουν δυο βρόχοι, ο ένας εμφωλευμένος μέσα στον άλλο, αυτός που ξεκινάει τελευταίος πρέπει να ολοκληρώνεται πρώτος.
3. Μια διαφορά της εντολής Όσο σε σχέση με την εντολή Μέχρις\_ότου οφείλεται στη θέση της λογικής συνθήκης στη ροή εκτέλεσης των εντολών.
4. Αν  $A=2$ ,  $B=3$ ,  $\Gamma=4$  και  $\Delta=ΑΛΗΘΗΣ$ , τότε η τιμή της έκφρασης  $(B < \Gamma > A+B)$  ΚΑΙ  $(ΟΧΙ(\Delta))$  είναι ΑΛΗΘΗΣ.
5. Κατά την εκτέλεση της εντολής ΔΙΑΒΑΣΕ, το πρόγραμμα διακόπτει την εκτέλεσή του και περιμένει την εισαγωγή τιμών από το πληκτρολόγιο.
6. Οι πίνακες δεν μπορούν να έχουν περισσότερες από δύο διαστάσεις. Μονάδες 6

**A2.** Δίνεται το παρακάτω ημιτελές τμήμα αλγορίθμου:

$k \leftarrow 1$

Για  $i$  ΑΠΟ 1 ΜΕΧΡΙ 4

    Για  $j$  ΑΠΟ 1 ΜΕΧΡΙ 5

        Αν ... ΤΟΤΕ

$A[k] \leftarrow i$

$A[\dots] \leftarrow \dots$

$A[\dots] \leftarrow \dots$

$k \leftarrow \dots$

        ΤΕΛΟΣ\_ΑΝ

    ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

Να ξαναγράψετε στο τετράδιό σας το παραπάνω τμήμα αλγορίθμου με τα κενά συμπληρωμένα, έτσι ώστε για τα μη μηδενικά στοιχεία ενός δισδιάστατου πίνακα ΠΙΝ[4,5] να τοποθετεί σε ένα μονοδιάστατο πίνακα  $A[60]$  τις ακόλουθες πληροφορίες: τη γραμμή, τη στήλη, και κατόπιν την τιμή του. Μονάδες 8

**A3.**

**α.** Να αναφέρετε ονομαστικά τους λόγους για τους οποίους αναθέτουμε την επίλυση ενός προβλήματος σε υπολογιστή (μονάδες 4).

**β.** Να γράψετε τις περιπτώσεις για τις οποίες δικαιολογείται η χρήση της σειριακής μεθόδου αναζήτησης σε έναν πίνακα (μονάδες 3).

**γ.** Να γράψετε τα πλεονεκτήματα των γλωσσών υψηλού επιπέδου (μονάδες 4).

**A4.**

**α.** Δίνεται τετραγωνικός πίνακας  $\Pi[100,100]$  και το παρακάτω τμήμα αλγορίθμου σε ψευδογλώσσα:

Για  $i$  από 1 μέχρι 100  
     Για  $j$  από 1 μέχρι 100  
         Αν  $i < j$  τότε  
             Διάβασε  $\Pi[i,j]$   
             Τέλος\_Αν  
     Τέλος\_επανάληψης  
 Τέλος\_επανάληψης

Να ξαναγράψετε στο τετράδιό σας το παραπάνω τμήμα αλγορίθμου χωρίς τη χρήση της δομής επιλογής, έτσι ώστε να επιτελεί την ίδια λειτουργία (μονάδες 4).

β. Να μεταφέρετε στο τετράδιό σας το παρακάτω τμήμα αλγορίθμου, έχοντας συμπληρώσει τις γραμμές εντολών 2, και 3 ώστε να εμφανίζει πάντα το μεγαλύτερο από τους δυο αριθμούς που διαβάστηκαν:

1. Διάβασε A, B  
 2. Αν A ... B τότε  
 3. ....  
 4. Τέλος\_Αν  
 5. Εμφάνισε A (μονάδες 4)

A5. Να γράψετε στο τετράδιό σας τους αριθμούς της στήλης A και δίπλα το γράμμα της στήλης B που αντιστοιχεί σωστά.

Στήλη A	Στήλη B
1. χαρακτήρες	α. λογική τιμή
2. ελεύθερο κείμενο	β. ουρά
3. ώθηση	γ. κριτήριο αλγορίθμου
4. αληθής	δ. επανάληψη
5. FIFO	ε. τύπος μεταβλητής
6. αποτελεσματικότητα	στ. στοίβα
7. βρόχος	ζ. τρόπος αναπαράστασης αλγορίθμου

## ΘΕΜΑ Β

B1. Δίνεται ο παρακάτω Αλγόριθμος:

**Αλγόριθμος Παράγοντες**

**Διάβασε** α

$k \leftarrow 2$

**Όσο**  $\alpha > 1$  **επανάλαβε**

**Αν**  $\alpha \bmod k = 0$  **τότε**

**Εμφάνισε** k

$\alpha \leftarrow \alpha \operatorname{div} k$

    αλλιώς

$k \leftarrow k+1$

**Τέλος\_Αν**

**Τέλος\_επανάληψης**

**Τέλος Παράγοντες**

Να σχεδιάσετε στο τετράδιό σας το αντίστοιχο διάγραμμα ροής.

Μονάδες 10

B2. Έστω μονοδιάστατος πίνακας  $\Pi[100]$ , του οποίου τα στοιχεία περιέχουν τις λογικές τιμές ΑΛΗΘΗΣ και ΨΕΥΔΗΣ. Να γραφεί τμήμα αλγορίθμου που χωρίς τη χρήση «αλγορίθμων ταξινόμησης» να τοποθετεί στις πρώτες θέσεις του πίνακα την τιμή

**ΘΕΜΑ Γ**

Η χρήση των κινητών τηλεφώνων, των φορητών υπολογιστών, των tablet υπολογιστών από τους νέους αυξάνεται ραγδαία. Ένας από τους στόχους των ερευνητών είναι να διερευνήσουν αν υπάρχουν επιπτώσεις στην υγεία των ανθρώπων από την αυξημένη έκθεση στα ηλεκτρομαγνητικά πεδία. Για τον σκοπό αυτό γίνονται μετρήσεις του ειδικού ρυθμού απορρόφησης (SAR) της ηλεκτρομαγνητικής ακτινοβολίας, πάνω στο ανθρώπινο σώμα. Ο δείκτης SAR μετράται σε Watt/Kgr και ο παγκόσμιος οργανισμός υγείας έχει θεσμοθετήσει ότι τα επιτρεπτά όρια για το κεφάλι και τον κορμό είναι μέχρι και 2 Watt/Kgr, ενώ για τα άκρα μέχρι και 4 Watt/Kgr. Θέλοντας να προσομοιάσουμε την έρευνα, θεωρούμε ότι σε 30 μαθητές έχουν τοποθετηθεί στον καθένα δυο μετρητές του δείκτη SAR, ο ένας στο κεφάλι και ο άλλος σε ένα από τα άνω άκρα, οι οποίοι καταγράφουν τις τιμές του αντίστοιχου δείκτη SAR κάθε 6 λεπτά.

Να αναπτύξετε αλγόριθμο σε ψευδογλώσσα, ο οποίος:

**Γ1.** Να διαβάζει τους πίνακες: ΚΩΔ[30], ο οποίος θα περιέχει τους κωδικούς των 30 μαθητών, τον πίνακα ΚΕΦ[30,10], του οποίου κάθε γραμμή θα αντιστοιχεί σε έναν μαθητή και θα έχει 10 τιμές που αντιστοιχούν στο SAR της κεφαλής για μια ώρα, καθώς και τον πίνακα ΑΚΡ[30,10] που κάθε γραμμή θα αντιστοιχεί σε έναν μαθητή και θα έχει 10 τιμές που αντιστοιχούν στο SAR του άκρου για μια ώρα. Μονάδες 2

**Γ2.** Για κάθε μαθητή να καταχωρεί σε διδιάστατο πίνακα ΜΟ[30,2] τις μέσες τιμές του SAR για το κεφάλι στην 1η στήλη και για το άκρο στη 2η στήλη. Μονάδες 4

**Γ3.** Να εμφανίζει για κάθε μαθητή τον κωδικό του και ένα από τα μηνύματα, «Χαμηλός SAR», «Κοντά στα όρια», «Εκτός ορίων», όταν η μέση τιμή του SAR της κεφαλής, καθώς και η μέση τιμή του SAR ενός εκ των άκρων του κυμαίνονται στις παρακάτω περιοχές:

Μ.Ο. SAR κεφαλής  $\leq 1,8$   $> 1,8$  και  $\leq 2$   $> 2$

Μ.Ο. SAR άκρου  $\leq 3,6$   $> 3,6$  και  $\leq 4$   $> 4$

Μήνυμα «Χαμηλός SAR» «Κοντά στα όρια» «Εκτός ορίων»

Το μήνυμα που θα εμφανίζεται θα πρέπει να είναι ένα μόνο για κάθε μαθητή και θα εξάγεται από τον συνδυασμό των τιμών των μέσων όρων των δυο SAR, όπου βαρύτητα θα έχει ο μέσος όρος, ο οποίος θα βρίσκεται σε μεγαλύτερη περιοχή τιμών. Για παράδειγμα, αν ο μέσος όρος SAR του άκρου έχει τιμή 3,8 και της κεφαλής έχει τιμή 1,5 τότε πρέπει να εμφανίζεται το μήνυμα «Κοντά στα όρια» και κανένα άλλο. Μονάδες 7

**Γ4.** Θεωρώντας ότι όλες οι τιμές του πίνακα ΜΟ[30,2] είναι διαφορετικές, να εμφανίζει τις τρεις μεγαλύτερες τιμές για τον μέσο όρο SAR της κεφαλής και τους κωδικούς των μαθητών που αντιστοιχούν σε αυτές. Μετά να εμφανίζει τις τρεις μεγαλύτερες τιμές για τον μέσο όρο SAR του άκρου και τους κωδικούς των μαθητών που αντιστοιχούν σε αυτές. Μονάδες 7

**ΘΕΜΑ Δ**

Σε ένα πρόγραμμα ανταλλαγής μαθητών Comenius συμμετέχουν μαθητές από δυο χώρες: Ελλάδα (EL) και Ισπανία (ES). Οι μαθητές αυτοί καλούνται να απαντήσουν σε μια ερώτηση όπου οι δυνατές απαντήσεις είναι:

**1.** Πολύ συχνά **2.** Συχνά **3.** Αρκετές φορές **4.** Σπάνια **5.** Ποτέ

Στην πρώτη φάση επεξεργασίας της ερώτησης πρέπει να καταγραφούν οι απαντήσεις από κάθε χώρα και να μετρήσουν για κάθε αριθμό απάντησης πόσες φορές υπάρχει, με σκοπό

να αναφέρουν για κάθε χώρα, ποια απάντηση είχε τα μεγαλύτερα ποσοστά.

Για να βοηθήσετε στην επεξεργασία να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

**Δ1. α.** Να περιέχει τμήμα δηλώσεων.

**β.** Να δημιουργεί δύο πίνακες EL[5] και ES[5] και να καταχωρίζει σε αυτούς την τιμή 0 σε όλα τα στοιχεία τους. Μονάδες 2

**Δ2.** Για κάθε μαθητή να διαβάζει το όνομα της χώρας του και τον αριθμό της απάντησής του. Οι δυνατές τιμές για τη χώρα είναι: EL, ES και για την απάντηση 1,2,3,4,5. Η κάθε απάντηση θα πρέπει να προσμετράται σε έναν από τους δύο πίνακες EL[5], ES[5] ανάλογα με τη χώρα και στο αντίστοιχο στοιχείο. Δηλαδή, αν δοθούν για τιμές οι ES και 4, τότε θα πρέπει στο 4ο στοιχείο του πίνακα ES[5] να προστεθεί μια ακόμα καταχώριση. (Δεν απαιτείται έλεγχος εγκυρότητας τιμών) Μονάδες 5

**Δ3.** Η προηγούμενη διαδικασία εισαγωγής δεδομένων και καταχώρισης απαντήσεων θα ελέγχεται από την ερώτηση «για Διακοπή της εισαγωγής πατήστε Δ ή δ», που θα εμφανίζεται και ο χρήστης θα πρέπει να δώσει το χαρακτήρα Δ ή δ για να σταματήσει την επαναληπτική διαδικασία. Μονάδες 3

**Δ4.** Στο τέλος για κάθε χώρα να εμφανίζει ποιος αριθμός απάντησης είχε το μεγαλύτερο ποσοστό, καθώς και το ποσοστό αυτό. Για την υλοποίηση αυτού του ερωτήματος θα χρησιμοποιήσετε δυο φορές το υποπρόγραμμα ΜΕΓ\_ΠΟΣ που θα κατασκευάσετε στο ερώτημα Δ5. Θεωρούμε ότι για κάθε χώρα τα ποσοστά των απαντήσεων είναι διαφορετικά μεταξύ τους και δεν υπάρχει περίπτωση ισοβαθμίας. Μονάδες 3

**Δ5.** Να αναπτύξετε το υποπρόγραμμα ΜΕΓ\_ΠΟΣ το οποίο:

1. Να δέχεται έναν πίνακα ακεραίων 5 θέσεων.

2. Να βρίσκει το μεγαλύτερο στοιχείο του πίνακα και σε ποια θέση βρίσκεται.

3. Να βρίσκει το ποσοστό που κατέχει το μεγαλύτερο στοιχείο σε σχέση με το άθροισμα όλων των στοιχείων του πίνακα.

4. Να επιστρέφει στο κυρίως πρόγραμμα το ποσοστό αυτό, καθώς και την θέση στην οποία βρίσκεται. Θεωρήστε ότι όλες οι τιμές των πινάκων είναι διαφορετικές και ότι για κάθε χώρα υπάρχει τουλάχιστον μια απάντηση στην ερώτηση.

## Λύσεις

### Θέμα Α

A1. 1. Λ 2. Σ 3. Σ 4. Λ 5. Σ 6.Λ

A2.

k ← 1

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 4

    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 5

        ΑΝ ΠΙΝ[i,j] <> 0 ΤΟΤΕ

            A[k] ← i

            A[k+1] ← j

            A[k+2] ← ΠΙΝ[i,j]

            k ← k +3

        ΤΕΛΟΣ\_ΑΝ

    ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

A3.

α. σελ 19

β. σελ 65

γ. σελ 127

A4.

α. Για  $i$  από 1 μέχρι 99  
    Για  $j$  από  $i+1$  μέχρι 100  
        Διάβασε  $\Pi[i,j]$   
        Τέλος\_επανάληψης  
    Τέλος\_επανάληψης

β. Διάβασε  $A, B$   
    Αν  $A < B$  τότε  
         $A \leftarrow B$   
    Τέλος\_Αν  
    Εμφάνισε  $A$

A5.

1.ε

2.ζ

3.στ

4.α

5.β

6.γ

7.δ

## ΘΕΜΑ Β

B2.

(A' τρόπος)

$j \leftarrow 0$

Για  $i$  από 1 μέχρι 100

    Αν  $\Pi[i] = \text{ΑΛΗΘΗΣ}$  τότε

$j \leftarrow j + 1$

    Τέλος\_Αν

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι  $j$

$\Pi[i] \leftarrow \text{ΑΛΗΘΗΣ}$

Τέλος\_επανάληψης

Για  $i$  από  $j+1$  μέχρι 100

$\Pi[i] \leftarrow \text{ΨΕΥΔΗΣ}$

Τέλος\_επανάληψης

(B' τρόπος (με αντιμεταθέσεις) )

$\kappa \leftarrow 1$

$\lambda \leftarrow 100$

Όσο  $k < \lambda$  επανάλαβε

Όσο  $\Pi[k] = \text{Αληθής}$  επανάλαβε

$k \leftarrow k + 1$

Τέλος\_επανάληψης

Όσο  $\Pi[\lambda] = \text{Ψευδής}$  επανάλαβε

$\lambda \leftarrow \lambda - 1$

Τέλος\_επανάληψης

Αν  $k < \lambda$  και  $\Pi[k] = \text{Ψευδής}$  και  $\Pi[\lambda] = \text{Αληθής}$  τότε

Αντιμετάθεσε  $\Pi[k], \Pi[\lambda]$

$k \leftarrow k + 1$

$\lambda \leftarrow \lambda - 1$

Τέλος\_Αν

Τέλος\_επανάληψης

**ΘΕΜΑ Γ**

Αλγόριθμος Ακτινοβολία

*!Ερώτημα Γ1*

Για  $i$  από 1 μέχρι 30

Διάβασε  $\text{ΚΩΔ}[i]$

Για  $j$  από 1 μέχρι 10

Διάβασε  $\text{ΚΕΦ}[i, j]$

Διάβασε  $\text{ΑΚΡ}[i, j]$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

*!Ερώτημα Γ2*

Για  $i$  από 1 μέχρι 30

$\text{sumk} \leftarrow 0$

$\text{suma} \leftarrow 0$

Για  $j$  από 1 μέχρι 10

$\text{sumk} \leftarrow \text{sumk} + \text{ΚΕΦ}[i, j]$

$\text{suma} \leftarrow \text{suma} + \text{ΑΚΡ}[i, j]$

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι 30

$\text{ΜΟ}[i, 1] \leftarrow \text{sumk}/10$

$\text{ΜΟ}[i, 2] \leftarrow \text{suma}/10$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

*!Ερώτημα Γ3*

Για  $i$  από 1 μέχρι 30

Αν  $\text{ΜΟ}[i, 1] \leq 1.8$  τότε

$k \leftarrow 1$

αλλιώς\_Αν  $\text{ΜΟ}[i, 1] \leq 2$  τότε

$k \leftarrow 2$

αλλιώς

$k \leftarrow 3$

Τέλος\_Αν

Αν  $MO[i, 2] \leq 3.6$  τότε

α ← 1

αλλιώς\_Αν  $MO[i, 2] \leq 4$  τότε

α ← 2

αλλιώς

α ← 3

Τέλος\_Αν

Αν  $\kappa > \alpha$  τότε

max ← κ

αλλιώς

max ← α

Τέλος\_Αν

Αν max = 1 τότε

Εμφάνισε ΚΩΔ[i], " Χαμηλός SAR"

αλλιώς\_Αν max = 2 τότε

Εμφάνισε ΚΩΔ[i], " Κοντά στα όρια"

αλλιώς

Εμφάνισε ΚΩΔ[i], " Εκτός ορίων"

Τέλος\_Αν

Τέλος\_επανάληψης

*!Ερώτημα Γ4*

Για i από 2 μέχρι 30

Για j από 30 μέχρι i με\_βήμα -1

Αν  $MO[j - 1, 1] < MO[j, 1]$  τότε

Αντιμετάθεσε  $MO[j - 1, 1], MO[j, 1]$

Αντιμετάθεσε  $MO[j - 1, 2], MO[j, 2]$

Αντιμετάθεσε  $KΩΔ[j - 1], KΩΔ[j]$

Τέλος\_Αν

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Εμφάνισε  $KΩΔ[1], MO[1, 1], KΩΔ[2], MO[2, 1], KΩΔ[3], MO[3, 1]$

Για i από 2 μέχρι 30

Για j από 30 μέχρι i με\_βήμα -1

Αν  $MO[j - 1, 2] < MO[j, 2]$  τότε

Αντιμετάθεσε  $MO[j - 1, 1], MO[j, 1]$

Αντιμετάθεσε  $MO[j - 1, 2], MO[j, 2]$

Αντιμετάθεσε  $KΩΔ[j - 1], KΩΔ[j]$

Τέλος\_Αν

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Εμφάνισε  $KΩΔ[1], MO[1, 2], KΩΔ[2], MO[2, 2], KΩΔ[3], MO[3, 2]$

Τέλος Ακτινοβολία

ΘΕΜΑ Δ



**ΠΡΟΓΡΑΜΜΑ** Comenius

*!Ερώτημα Δ1*

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** EL[5], ES[5], ΑΠΑΝΤΗΣΗ, i, COUNTER, ΘΕΣΗ

**ΧΑΡΑΚΤΗΡΕΣ:** ΔΙΑΚΟΠΗ, ΧΩΡΑ

**ΠΡΑΓΜΑΤΙΚΕΣ:** ΠΟΣΟΣΤΟ

**ΑΡΧΗ**

*!Ερώτημα Δ1*

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 5

EL[i] ← 0

ES[i] ← 0

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*!Ερώτημα Δ3*

COUNTER ← 0

**ΓΡΑΨΕ** 'Για Διακοπή της εισαγωγής πατήστε Δ ή δ'

**ΔΙΑΒΑΣΕ** ΔΙΑΚΟΠΗ

**ΟΣΟ** ΔΙΑΚΟΠΗ <> 'Δ' **ΚΑΙ** ΔΙΑΚΟΠΗ <> 'δ' **ΕΠΑΝΑΛΑΒΕ**

*!Ερώτημα Δ2*

**ΔΙΑΒΑΣΕ** ΧΩΡΑ

**ΔΙΑΒΑΣΕ** ΑΠΑΝΤΗΣΗ

COUNTER ← COUNTER + 1

**ΑΝ** ΧΩΡΑ = 'ΕΛ' **ΤΟΤΕ**

EL[ΑΠΑΝΤΗΣΗ] ← EL[ΑΠΑΝΤΗΣΗ] + 1

**ΑΛΛΙΩΣ**

ES[ΑΠΑΝΤΗΣΗ] ← ES[ΑΠΑΝΤΗΣΗ] + 1

**ΤΕΛΟΣ\_ΑΝ**

**ΓΡΑΨΕ** 'Για Διακοπή της εισαγωγής πατήστε Δ ή δ'

**ΔΙΑΒΑΣΕ** ΔΙΑΚΟΠΗ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*!Ερώτημα Δ4*

**ΑΝ** COUNTER > 0 **ΤΟΤΕ**

**ΚΑΛΕΣΕ** ΜΕΓ\_ΠΟΣ(EL, ΠΟΣΟΣΤΟ, ΘΕΣΗ)

**ΓΡΑΨΕ** 'ΕΛΛΑΔΑ: Αριθμός απάντησης=', ΘΕΣΗ, ' Ποσοστό=', ΠΟΣΟΣΤΟ, '%'

**ΚΑΛΕΣΕ** ΜΕΓ\_ΠΟΣ(ES, ΠΟΣΟΣΤΟ, ΘΕΣΗ)

**ΓΡΑΨΕ** 'ΙΣΠΑΝΙΑ: Αριθμός απάντησης=', ΘΕΣΗ, ' Ποσοστό=', ΠΟΣΟΣΤΟ, '%'

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'Δεν υπάρχουν στοιχεία Για επεξεργασία'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

*!Ερώτημα Δ5*

**ΔΙΑΔΙΚΑΣΙΑ** ΜΕΓ\_ΠΟΣ(ΠΙΝΑΚΑΣ, ΠΟΣΟΣΤΟ, ΤΕΛΙΚΗ\_ΘΕΣΗ)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** ΠΙΝΑΚΑΣ[5], i, SUM, MAX, ΘΕΣΗ, ΤΕΛΙΚΗ\_ΘΕΣΗ

**ΠΡΑΓΜΑΤΙΚΕΣ:** ΠΟΣΟΣΤΟ

**ΑΡΧΗ**

```

MAX ← ΠΙΝΑΚΑΣ[1]
ΘΕΣΗ ← 1
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 5
  ΑΝ ΠΙΝΑΚΑΣ[i] > MAX ΤΟΤΕ
    MAX ← ΠΙΝΑΚΑΣ[i]
    ΘΕΣΗ ← i
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

SUM ← ΠΙΝΑΚΑΣ[1] + ΠΙΝΑΚΑΣ[2] + ΠΙΝΑΚΑΣ[3] + ΠΙΝΑΚΑΣ[4] + ΠΙΝΑΚΑΣ[5]
ΠΟΣΟΣΤΟ ← 100*ΠΙΝΑΚΑΣ[ΘΕΣΗ]/SUM
ΤΕΛΙΚΗ_ΘΕΣΗ ← ΘΕΣΗ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

## ΕΣΠΕΡΙΝΑ 2013

### ΘΕΜΑ Γ

Ανατέθηκε σε μια περιβαλλοντική ομάδα να φτιάξει έναν χάρτη επικινδυνότητας πυρκαγιών για την οροσειρά του Ταυγέτου. Ο χάρτης αυτός θα δείχνει σε ποιες περιοχές υπάρχει μεγάλη πιθανότητα πυρκαγιάς, σε ποιες μέτρια και σε ποιες χαμηλή. Για να μπορέσουν να κατασκευάσουν το χάρτη, θα πρέπει σε κάθε περιοχή να μετρήσουν τη μέση ταχύτητα του αέρα και την υγρασία. Για να χαρακτηριστεί μια περιοχή ως υψηλής επικινδυνότητας θα πρέπει η μέση ταχύτητα του αέρα να ξεπερνά τα 10 m/s και η υγρασία να είναι σε «χαμηλά επίπεδα», για να χαρακτηριστεί ως μέτριας επικινδυνότητας θα πρέπει η μέση ταχύτητα του αέρα να ξεπερνά τα 10 m/s και η υγρασία να είναι σε «υψηλά επίπεδα». Τέλος, για να χαρακτηριστεί ως χαμηλής επικινδυνότητας θα πρέπει η μέση ταχύτητα του αέρα να είναι μικρότερη ή ίση των 10 m/s ανεξάρτητα από τα επίπεδα της υγρασίας.

Να αναπτύξετε αλγόριθμο, ο οποίος:

- Γ1. Να διαβάζει για 10 περιοχές την υγρασία και τη μέση ταχύτητα του ανέμου. Μονάδες 4
- Γ2. Για κάθε περιοχή να εμφανίζει τα μηνύματα «Υψηλή επικινδυνότητα», «Μεσαία επικινδυνότητα» και «Χαμηλή επικινδυνότητα» ανάλογα με τους συνδυασμούς μέσης ταχύτητας και υγρασίας. Μονάδες 10
- Γ3. Να εμφανίζει το πλήθος των περιοχών με υψηλή επικινδυνότητα. Μονάδες 6

### ΘΕΜΑ Δ

Μια εικόνα 128 x 128 εικονοστοιχείων (pixels) αποθηκεύεται σε ένα δισδιάστατο πίνακα A[128,128]. Ένα pixel με μαύρο χρώμα αντιστοιχεί στην τιμή 0, ενώ ένα pixel με άσπρο χρώμα αντιστοιχεί στην τιμή 255. Συνεπώς, ο πίνακας χρησιμεύει στην αποθήκευση των 256 αποχρώσεων του γκρι, δηλαδή στα κελιά αποθηκεύει τους ακέραιους αριθμούς από 0 έως 255. Ορίζουμε ως «αρνητική» της αρχικής εικόνας, εκείνη που έχει τιμή 0 (μαύρο χρώμα) εκεί όπου η αρχική έχει τιμή 255 (άσπρο χρώμα) και έχει τιμή 1 εκεί όπου η αρχική εικόνα έχει τιμή 254, κ.ο.κ. Επίσης, μια συνήθης διαδικασία επεξεργασίας εικόνων είναι η λεύκανση, κατά την οποία η τιμή ενός χρώματος πολλαπλασιάζεται με συντελεστή μεγαλύτερο ή ίσο του 1. Να αναπτύξετε αλγόριθμο, ο οποίος:

- Δ1.** Να καταχωρίζει στο δισδιάστατο πίνακα  $A[128,128]$  την αρχική εικόνα. Μονάδες 2
- Δ2.** Να υπολογίζει το «αρνητικό» της εικόνας σε νέο πίνακα  $B[128,128]$  και να τον εμφανίζει στην οθόνη. Μονάδες 5
- Δ3.** Να εκτελεί λεύκανση της αρχικής εικόνας με συντελεστή 1,3 σε πίνακα  $\Gamma[128,128]$  και να τον εμφανίζει στην οθόνη. Σημειώνεται ότι, εάν η νέα τιμή είναι μεγαλύτερη του 255, τότε ως νέα τιμή εκχωρείται το 255. Στην περίπτωση, που η προκύπτουσα τιμή δεν είναι ακέραια, πραγματοποιείται αποκοπή των δεκαδικών ψηφίων. Μονάδες 6
- Δ4.** Να εμφανίζει στην οθόνη τις συντεταγμένες  $i, j$  των θέσεων (κελιών του πίνακα  $A[128,128]$ ), όπου η χρωματική τιμή (ταυτίζεται με την αριθμητική τιμή) είναι μέγιστη.

## Λύσεις

### ΘΕΜΑ Γ

#### Αλγόριθμος ΘΓ

πλυε ← 0

Για  $i$  από 1 μέχρι 10

Διάβασε ΥΓΡ, ΜΤΑ

Αν  $ΜΤΑ > 10$  και  $ΥΓΡ = \text{"Χαμηλά Επίπεδα"}$  τότε

Εμφάνισε "Υψηλή επικινδυνότητα"

πλυε ← πλυε + 1

αλλιώς\_Αν  $ΜΤΑ > 10$  και  $ΥΓΡ = \text{"Υψηλά Επίπεδα"}$  τότε

Εμφάνισε "Μεσαία επικινδυνότητα"

αλλιώς

Εμφάνισε "Χαμηλή επικινδυνότητα"

Τέλος\_Αν

Τέλος\_επανάληψης

Εμφάνισε πλυε

Τέλος ΘΓ

### ΘΕΜΑ Δ

#### Αλγόριθμος ΘΔ

Για  $i$  από 1 μέχρι 128

Για  $j$  από 1 μέχρι 128

Διάβασε  $A[i, j]$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι 128

Για  $j$  από 1 μέχρι 128

$B[i, j] \leftarrow 255 - A[i, j]$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι 128

Για  $j$  από 1 μέχρι 128

Εμφάνισε  $B[i, j]$

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι 128

Για  $j$  από 1 μέχρι 128

$\Gamma[i, j] \leftarrow A\_M(1.3 * A[i, j])$

```

        Αν  $\Gamma[i, j] > 255$  τότε  $\Gamma[i, j] \leftarrow 255$ 
    Τέλος_επανάληψης
Τέλος_επανάληψης
Για  $i$  από 1 μέχρι 128
    Για  $j$  από 1 μέχρι 128
        Εμφάνισε  $\Gamma[i, j]$ 
    Τέλος_επανάληψης
Τέλος_επανάληψης
max  $\leftarrow A[1, 1]$ 
Για  $i$  από 1 μέχρι 128
    Για  $j$  από 1 μέχρι 128
        Αν  $A[i, j] > \text{max}$  τότε max  $\leftarrow A[i, j]$ 
    Τέλος_επανάληψης
Τέλος_επανάληψης
Για  $i$  από 1 μέχρι 128
    Για  $j$  από 1 μέχρι 128
        Αν  $A[i, j] = \text{max}$  τότε Εμφάνισε  $i, j$ 
    Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος Θδ

```

### Επαναληπτικές Ημερησίων 2013

#### ΘΕΜΑ Α

**A1.α.** Να γράψετε στο τετράδιό σας τον αριθμό καθεμιάς από τις παρακάτω προτάσεις 1-4 και δίπλα τη λέξη ΣΩΣΤΟ, αν η πρόταση είναι σωστή, ή τη λέξη ΛΑΘΟΣ, αν η πρόταση είναι λανθασμένη.

1. Η εύρεση του μικρότερου από πέντε αριθμούς είναι πρόβλημα βελτιστοποίησης.
2. Ο δείκτης εμπρός (front) μιας ουράς μας δίνει τη θέση του στοιχείου, το οποίο σε πρώτη ευκαιρία θα εξαχθεί.
3. Ο διαχωρισμός αποτελεί την αντίστροφη πράξη της συγχώνευσης.
4. Στη ΓΛΩΣΣΑ, ο μέσος όρος ενός συνόλου ακεραίων μεταβλητών πρέπει να αποθηκευτεί σε μεταβλητή πραγματικού τύπου. (μονάδες 4)

**A2.** Να περιγράψετε τα στάδια αντιμετώπισης ενός προβλήματος. Μονάδες 6

**A3.** Να γράψετε συμπληρωμένο στο τετράδιό σας το ακόλουθο τμήμα αλγορίθμου, το οποίο πραγματοποιεί αναζήτηση όλων των στοιχείων του πίνακα  $W[10]$  στον πίνακα  $S[1000]$ , έτσι ώστε τα στοιχεία του πίνακα  $W[10]$  να καταλαμβάνουν συνεχόμενες θέσεις στον πίνακα  $S[1000]$ . Ο αλγόριθμος βρίσκει τη θέση  $i$  του  $S$ , απ' όπου αρχίζει η πρώτη εμφάνιση των στοιχείων του  $W[10]$ .

$F \leftarrow \Psi\text{ΕΥ}\Delta\text{Η}\Sigma$

$i \leftarrow 1$

Όσο ..... ΚΑΙ .....ΕΠΑΝΑΛΑΒΕ

$j \leftarrow 0$

    Όσο ..... ΚΑΙ .....ΕΠΑΝΑΛΑΒΕ

$j \leftarrow j + 1$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Αν .....ΤΟΤΕ**

**F ← ΑΛΗΘΗΣ**

**ΑΛΛΙΩΣ**

**i ← i + 1**

**ΤΕΛΟΣ\_Αν**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Αν F = ΑΛΗΘΗΣ ΤΟΤΕ**

**ΓΡΑΨΕ i**

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ 'ΔΕ ΒΡΕΘΗΚΕ'**

**ΤΕΛΟΣ\_ΑΝ**

Μονάδες 10

**A4.** Πρώτος ονομάζεται ένας φυσικός αριθμός, όταν έχει ακριβώς δύο διαιρέτες: τον εαυτό του και τη μονάδα. Ο παρακάτω αλγόριθμος γράφτηκε, έτσι ώστε να εμφανίζει τους πρώτους αριθμούς από το 2 μέχρι και το 100.

**ΑΛΓΟΡΙΘΜΟΣ** πρώτοι

**Για i ΑΠΟ 1 ΜΕΧΡΙ 100**

**M ← i**

**Για j ΑΠΟ 0 ΜΕΧΡΙ i**

**Αν i / j = 0 ΤΟΤΕ M ← M + 1**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Αν M < 3 ΤΟΤΕ ΕΜΦΑΝΙΣΕ i**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ** πρώτοι

Ο παραπάνω Αλγόριθμος έχει λάθη. Να τον γράψετε στο τετράδιό σας, κάνοντας τις απαραίτητες διορθώσεις, ώστε να λειτουργεί σωστά, χωρίς την προσθήκη νέων εντολών. Μονάδες 10

### **ΘΕΜΑ Γ**

Ο σύλλογος γονέων και κηδεμόνων μιας περιοχής θέλει να διοργανώσει μια πολιτιστική εκδήλωση. Για το σκοπό αυτό, ζητά από κάθε σχολείο της περιοχής να προσφέρει κάποιο χρηματικό ποσό για την πραγματοποίησή της. Κάθε σχολείο έχει τη δυνατότητα να επικοινωνεί περισσότερες από μία φορές με το σύλλογο και να τροποποιεί την προσφορά του. Να αναπτύξετε αλγόριθμο σε ψευδογλώσσα, ο οποίος:

**Γ1.** Να θεωρεί δεδομένο ένα πίνακα Σ[100] που περιέχει τα ονόματα των 100 σχολείων της περιοχής και να δημιουργεί πίνακα Π[100] που θα περιέχει τις αντίστοιχες χρηματικές προσφορές από κάθε σχολείο. Αρχικά να τοποθετηθεί σε κάθε στοιχείο του πίνακα Π[100] την τιμή -1. Μονάδες 3

**Γ2. α)** Να διαβάσει το όνομα ενός σχολείου και να το αναζητά στον πίνακα Σ. (μονάδες 4)  
**β)** Να εμφανίζει το μήνυμα «Άγνωστο», όταν το σχολείο δε βρεθεί. Όταν το σχολείο βρεθεί, να σταματά την αναζήτηση, να διαβάσει τη χρηματική προσφορά του σχολείου και να την τοποθετεί στην αντίστοιχη θέση του πίνακα Π. (Όταν δοθεί η τιμή 0, σημαίνει ότι το σχολείο δεν μπορεί να προσφέρει χρήματα, δηλαδή έδωσε μηδενική προσφορά). Όταν δεν είναι η πρώτη φορά που δίνει προσφορά **τότε** να εμφανίζει το μήνυμα «ΤΡΟΠΟΠΟΙΗΣΗ <\ΠΡΟΣΦΟΡΑΣ>» και να αντικαθιστά την προηγούμενη προσφορά του με τη νέα. (μονάδες 6)

Μονάδες 10

**Γ3.** Να επαναλαμβάνει τις ενέργειες που περιγράφονται στο ερώτημα Γ2, μέχρις ότου όλα τα σχολεία να δώσουν τουλάχιστον μία προσφορά. Μονάδες 3

**Γ4.** Να εμφανίζει: α) το συνολικό χρηματικό ποσό που έχει συγκεντρωθεί, β) το πλήθος των σχολείων που έδωσαν μηδενική προσφορά, γ) το πλήθος των τροποποιήσεων που έγιναν στις προσφορές. Μονάδες 4

#### **ΘΕΜΑ Δ**

Τα δεδομένα (κείμενο, εικόνα, ήχος, κλπ), κατά τη μετάδοσή τους μέσω ενσύρματων ή ασύρματων καναλιών επικοινωνίας, αλλοιώνονται λόγω του θορύβου που χαρακτηρίζει κάθε κανάλι. Ο τρόπος προστασίας των δεδομένων μετάδοσης είναι ο ακόλουθος: Για κάθε bit (ακεραίος με τιμή 0 ή 1), που ο πομπός θέλει να στείλει, μεταδίδει μια λέξη, που αντιστοιχεί σε πίνακα ΜΕΤΑΔΟΣΗ[31] με όλες τις τιμές του ταυτόσημες με το προς μετάδοση bit, δηλαδή, Αν πρόκειται να σταλεί το bit 1, τότε η λέξη που μεταδίδεται είναι η 11...1 μήκους 31 bits, ενώ αν πρόκειται να σταλεί το bit 0, τότε η λέξη που μεταδίδεται είναι η 00...0, μήκους 31 bits. Ο δέκτης λαμβάνει λέξη μήκους 31 bits, τα οποία τοποθετούνται σε πίνακα ΛΗΨΗ[31]. Έχουμε «ΛΑΝΘΑΣΜΕΝΗ ΛΗΨΗ», εάν υπάρχει τουλάχιστον ένα στοιχείο του πίνακα ΛΗΨΗ[31] με διαφορετική τιμή από αυτήν του αντίστοιχου στοιχείου του πίνακα ΜΕΤΑΔΟΣΗ[31]. Εάν το πλήθος των 1 του πίνακα ΛΗΨΗ[31] είναι μεγαλύτερο από το πλήθος των 0, τότε ο δέκτης αποφασίζει ότι ο πομπός έστειλε 1, ενώ σε αντίθετη περίπτωση ο δέκτης αποφασίζει ότι ο πομπός έστειλε 0. Σε κάθε περίπτωση, Αν περισσότερα από τα μισά των 31 bits της λέξης μετάδοσης έχουν αλλοιωθεί, τότε ο δέκτης θα έχει πάρει «ΛΑΝΘΑΣΜΕΝΗ ΑΠΟΦΑΣΗ».

Να γραφεί πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο να κάνει τα εξής:

**Δ1.** Να περιλαμβάνει κατάλληλο τμήμα δηλώσεων. Μονάδες 3

**Δ2.** Για κάθε τιμή ποιότητας του καναλιού, που χαρακτηρίζεται από ακεραίους από 1 έως και 10, να πραγματοποιούνται το πολύ 100.000 διαφορετικές προσπάθειες μετάδοσης-λήψης και διόρθωσης λαθών. Εάν όμως ληφθούν 100 λανθασμένες αποφάσεις, τότε να διακόπτεται η διαδικασία για τη συγκεκριμένη τιμή ποιότητας του καναλιού. Μονάδες 4

**Δ3.** Σε κάθε προσπάθεια μετάδοσης-λήψης και διόρθωσης λαθών να πραγματοποιούνται οι ακόλουθες ενέργειες:

**α.** Να διαβάσει (χωρίς έλεγχο εγκυρότητας των τιμών τους) τη μεταδοθείσα λέξη, καθώς και τη ληφθείσα λέξη και να ελέγχει, εάν αυτές ταυτίζονται.

**β.** Να διορθώνει τη ληφθείσα λέξη στο δέκτη, βάσει της παραπάνω περιγραφής του αλγορίθμου.

Μονάδες 9

**Δ4. α.** Να αποθηκεύει, για κάθε τιμή ποιότητας καναλιού, σε πίνακα ΛΑΘΗΑΠΟΦ[10] το ποσοστό των λανθασμένων αποφάσεων και σε πίνακα ΛΑΘΗΛΗΨ[10] το ποσοστό των λανθασμένων λήψεων.

**β.** Να εμφανίζει συγκεντρωτικά τα ποσοστά των λανθασμένων αποφάσεων και λανθασμένων λήψεων στο δέκτη. Μονάδες 4

#### **Λύσεις**

## ΘΕΜΑ Α

A1. α.

- 1) Λάθος
- 2) Σωστό
- 3) Σωστό
- 4) Σωστό

A1. β.

1 - β, 2 - α, 3 - δ, 4 - γ.

A2.

Σχολικό βιβλίο σελ. 16

A3.

F ← ΨΕΥΔΗΣ

i ← 1

Όσο F = ΨΕΥΔΗΣ ΚΑΙ i ≤ 1000 ΕΠΑΝΑΛΑΒΕ

j ← 0

    Όσο j ≤ 9 ΚΑΙ W[j + 1] = S[i + j] ΕΠΑΝΑΛΑΒΕ

    j ← j + 1

    ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

    Αν j = 10 ΤΟΤΕ

    F ← ΑΛΗΘΗΣ

    ΑΛΛΙΩΣ

    i ← i + 1

    ΤΕΛΟΣ\_Αν

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

Αν F = ΑΛΗΘΗΣ ΤΟΤΕ

ΓΡΑΨΕ i

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'ΔΕ ΒΡΕΘΗΚΕ'

ΤΕΛΟΣ\_Αν

A4.

ΑΛΓΟΡΙΘΜΟΣ πρώτοι

Για i ΑΠΟ 2 ΜΕΧΡΙ 100

M ← 0

Για j ΑΠΟ 1 ΜΕΧΡΙ i

    Αν i MOD j = 0 ΤΟΤΕ M ← M + 1

    ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

    Αν M < 3 ΤΟΤΕ ΕΜΦΑΝΙΣΕ i

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ πρώτοι

A5.

α.

5 2 8 7 3

β. ΨΕΥΔΗΣ

## ΘΕΜΑ Γ

Αλγόριθμος θέμα\_Γ

Δεδομένα // Σ //

```

Σποσό ← 0
Πλημεδενική ← 0
N ← 0
Για i από 1 μέχρι 100
  Π[i] ← -1
Τέλος_επανάληψης
Αρχή_επανάληψης
done ← αληθής
Διάβασε on
θέση ← 0
i ← 1
βρέθηκε ← ψευδής
Όσο βρέθηκε = ψευδής και i <= 100 επανάλαβε
  Αν Σ[i] = on τότε
    βρέθηκε ← αληθής
    θέση ← i
    αλλιώς
    i ← i + 1
  Τέλος_Αν
Τέλος_επανάληψης
Αν βρέθηκε = ψευδής τότε
Εμφάνισε "Άγνωστο"
αλλιώς
Διάβασε χρημ
  Αν Π[θέση] < > -1 τότε
    Εμφάνισε "ΤΡΟΠΟΠΟΙΗΣΗ ΠΡΟΣΦΟΡΑΣ"
    N ← N + 1
  Τέλος_Αν
Π[θέση] ← χρημ
Τέλος_Αν
Για i από 1 μέχρι 100
  Αν Π[i] = -1 τότε
    done ← ψευδής
  Τέλος_Αν
Τέλος_επανάληψης
Μέχρις_ότου done = αληθής
Για i από 1 μέχρι 100
Σποσό ← Σποσό + Π[i]
Αν Π[i] = 0 τότε
  Πλημεδενική ← Πλημεδενική + 1
Τέλος_Αν
Τέλος_επανάληψης
Εμφάνισε Σποσό, Πλημεδενική, N
Τέλος θέμα_Γ

```

**ΘΕΜΑ Δ**

**ΠΡΟΓΡΑΜΜΑ** θέμα\_Δ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** Πλήθος, Πλήθος\_λήψεων, Πλήθος\_αποφ, ΜΕΤΑΔΟΣΗ[31]



**ΑΚΕΡΑΙΕΣ:** ΛΗΨΗ[31], Κ, Πλ1, Πλ0, δέκτη, i, j  
**ΠΡΑΓΜΑΤΙΚΕΣ:** ΛΑΘΗΑΠΟΦ[10], ΛΑΘΗΛΗΨ[10]  
 ΑΡΧΗ  
**Για i ΑΠΟ 1 ΜΕΧΡΙ 10**  
 Πλήθος  $\leftarrow$  0  
 Πλήθος\_λήψεων  $\leftarrow$  0  
 Πλήθος\_αποφ  $\leftarrow$  0  
     **Όσο Π < 100 ΚΑΙ Ν < 100000 ΕΠΑΝΑΛΑΒΕ**  
         Πλήθος  $\leftarrow$  Πλήθος + 1  
             **Για j ΑΠΟ 1 ΜΕΧΡΙ 31**  
                 **ΔΙΑΒΑΣΕ** ΜΕΤΑΔΟΣΗ[j], ΛΗΨΗ[j]  
                 **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  
                 Κ  $\leftarrow$  0  
             **Για j ΑΠΟ 1 ΜΕΧΡΙ 31**  
                 **ΑΝ** ΜΕΤΑΔΟΣΗ[j] = ΛΗΨΗ[j] **ΤΟΤΕ**  
                     Κ  $\leftarrow$  Κ + 1  
                     **ΤΕΛΟΣ\_Αν**  
             **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  
                 **ΑΝ** Κ < 31 **ΤΟΤΕ**  
                     Πλήθος\_λήψεων  $\leftarrow$  Πλήθος\_λήψεων + 1  
                     **ΤΕΛΟΣ\_Αν**  
         Πλ1  $\leftarrow$  0  
         Πλ0  $\leftarrow$  0  
         **Για j ΑΠΟ 1 ΜΕΧΡΙ 31**  
             **ΑΝ** ΛΗΨΗ[j] = 1 **ΤΟΤΕ**  
                 Πλ1  $\leftarrow$  Πλ1 + 1  
                 **ΑΛΛΙΩΣ**  
                 Πλ0  $\leftarrow$  Πλ0 + 1  
                 **ΤΕΛΟΣ\_Αν**  
         **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  
         **ΑΝ** Πλ1 > Πλ0 **ΤΟΤΕ**  
             δέκτη  $\leftarrow$  1  
             **ΑΛΛΙΩΣ**  
             δέκτη  $\leftarrow$  0  
             **ΤΕΛΟΣ\_Αν**  
         **Αν** δέκτη < > ΜΕΤΑΔΟΣΗ[1] **ΤΟΤΕ**  
             Πλήθος\_αποφ  $\leftarrow$  Πλήθος\_αποφ + 1  
             **ΤΕΛΟΣ\_Αν**  
             **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  
             ΛΑΘΗΑΠΟΦ[i]  $\leftarrow$  Πλήθος\_αποφ \* 100 / Πλήθος  
             ΛΑΘΗΛΗΨ[i]  $\leftarrow$  Πλήθος\_λήψεων \* 100 / Πλήθος  
         **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  
         **Για i ΑΠΟ 1 ΜΕΧΡΙ 10**  
             **ΓΡΑΨΕ** ΛΑΘΗΑΠΟΦ[i], ΛΑΘΗΛΗΨ[i]  
         **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  
         **ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## Πηγές-Βιβλιογραφία

1. Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, Βιβλίο & Τετράδιο μαθητή
2. Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ
3. <http://alkisg.mysch.gr/steki>
4. <http://www.pdp.gr>