# ΛΙΓΟ ΠΡΙΝ ΤΗΝ ΕΞΕΤΑΣΗ

**ΣΗΜΕΙΩΣΗ ΠΡΟΣ ΤΟ ΜΑΘΗΤΗ**

*Υπογραμμίζουμε ότι τα επόμενα δεν αποτελούν (κατά το κοινώς λεγόμενο) SOS και ούτε ως τέτοια μπορεί να θεωρηθούν.*
Αγαπητέ μαθητή, στο φυλλάδιο αυτό θα βρεις ένα πλούσιο υλικό, που θα σε βοηθήσει ουσιαστικά, ώστε να κάνεις μια καλή επανάληψη και να καλύψεις τα τυχόν κενά σου. Όμως απαραίτητη προϋπόθεση είναι να το αξιοποιήσεις σωστά.
Πρώτα πρέπει να διαβάσεις πολύ καλά το σχολικό σου βιβλίο. Οι **βασικές έννοιες και ερωτήσεις στη θεωρία**, όπως δίνονται σ' αυτό το φυλλάδιο, θα σε βοηθήσουν να εντοπίσεις τυχόν κενά.
Στη συνέχεια να μελετήσεις τη μεθοδολογία των ασκήσεων, όπως καλά την έχουμε γράψει στο τετράδιο, μαζί με τις λυμένες ασκήσεις. Σε κάθε άσκηση, διάβασε προσεκτικά την εκφώνηση και μην κοιτάς αμέσως τη λύση. Πρώτα σκέψου. Γράψε τη δική σου λύση και μετά αντιπαράβαλέ την με τη λύση του τετραδίου. Αν δεν είναι ίδια, μην αποθαρρύνεσαι. Θα τα καταφέρεις στην επόμενη άσκηση.
Οι ασκήσεις που πρέπει να μελετήσεις προσεκτικά είναι ασκήσεις του τετραδίου μαθητή (όσες έχουμε στο φυλλάδιο).
Στη συνέχεια μελέτησε προσεκτικά στο παρόν φυλλάδιο τη μεθοδολογία των ασκήσεων και δούλεψε παρόμοια με τα προτεινόμενα θέματα.

**ΛΙΓΟ ΠΡΙΝ ΤΗΝ ΕΞΕΤΑΣΗ .....**

Όπως ήταν αναμενόμενο, τα θέματα των προηγούμενων χρόνων ήταν αισθητά πιο εύκολα. Και ήταν αναμενόμενο αφού το μάθημα ήταν εντελώς νέο, πρωτοδιδασκόμενο και δεν υπήρχε εμπειρία τόσο για τους μαθητές όσο και για τους διδάσκοντες.
Με την πάροδο των ετών το μάθημα δείχνει να έχει ... βρει το δρόμο του. Αυτό έχει αρχίσει να φαίνεται και στα θέματα τα οποία στα τελευταία χρόνια έχουν αρχίσει να παρουσιάζουν μία παρόμοια εικόνα αλλά και στις βαθμολογίες των μαθητών οι οποίες σημειώνουν αισθητή άνοδο με ταυτόχρονη αύξηση στη δυσκολία των θεμάτων που καλούνται να αντιμετωπίσουν.
Αρχική εκτίμηση είναι ότι τα θέματα θα κινηθούν στα περσινά επίπεδα με κάποιες ίσως μικρές διαφοροποιήσεις.
Το **1ο θέμα** είναι παραδοσιακά το θέμα της θεωρίας (με 40 μονάδες). Για την αξιολόγηση της κατανόησης της θεωρίας έχουν χρησιμοποιηθεί τόσο ανοικτού τύπου (ανάπτυξης) όσο και κλειστού τύπου ερωτήσεις. Οι ερωτήσεις κλειστού τύπου είχαν, κατά περίπτωση, τη μορφή Σωστού/Λάθους, πολλαπλής επιλογής ή και αντιστοίχισης.
Για θεωρία πρέπει να δώσουμε ιδιαίτερη προσοχή στο κεφάλαιο 10 (υποπρογράμματα), κεφάλαιο 3 (δομές δεδομένων), σειριακή αναζήτηση, πολλαπλασιασμός αλά Ρωσικά, στατικές και δυναμικές δομές, και φυσικά ερώτηση στο κεφάλαιο 6 (ιδιαίτερη προσοχή στην § 6.7). Δεν πρέπει να παραξενευτούμε αν δούμε ερωτήσεις και από άλλα κεφάλαια συνήθως σύντομης απάντησης.
Ισως στο 1ο θέμα να δούμε και μετατροπές σε δομές επιλογής (Αν … αλλιώς_αν, Επίλεξε) ή σε δομές επανάληψης (Όσο … Επανάλαβε, Αρχή_επανάληψης … μέχρις_ότου, Για … από … μέχρι), ποια κριτήρια παραβιάζουν δοσμένα τμήματα εντολών (συνήθως καθοριστικότητα, περατότητα), να βρούμε λάθη σε δοσμένο κώδικα, μετατροπή κώδικα σε διάγραμμα ροής ή το αντίστροφο, μικρές αυτόνομες ασκήσεις που έχουν να κάνουν με προτεραιότητα αριθμητικών τελεστών, λογικές συνθήκες, χρήση mod και div, μετατροπή μαθηματικών παραστάσεων σε πρόγραμμα ΓΛΩΣΣΑ.
Το **2ο θέμα** αν δεν διαφοροποιηθεί από τα καθιερωμένα (δηλ. να τρέχω κώδικα – πίνακας τιμών) μπορεί να περιλαμβάνει υποπρογράμματα. Δηλαδή να δίνεται έτοιμο το πρόγραμμα και το καλούμενο υποπρόγραμμα και να ζητάει να βρούμε τι θα τυπώσει (εκτελώντας το με πίνακα τιμών). Ενδεχόμενα να εμφανιστούν και διαγράμματα ροής για να γίνουν κώδικας και στη συνέχεια πίνακας τιμών. Αν πάντως αλλάξει η μορφή του 2ου θέματος τότε μπορεί να ζητείται να γράψουμε ένα υποπρόγραμμα (διαδικασία ή συνάρτηση) μπορεί να δίνεται ένα υποπρόγραμμα και να ζητείται να περιγράψουμε τη λειτουργία του υποπρογράμματος (τι κάνει), να γράψουμε ένα μικρό πρόγραμμα που να το καλεί ή να ζητείται να μετατρέψουμε μια διαδικασία σε συνάρτηση και το αντίστροφο. Παίζει πολύ και διόρθωση έτοιμου κώδικα ή ερωτήσεις πάνω σε έτοιμο κώδικα.
Το **3ο θέμα** μέχρι πρόπερσι ζητούσε από τον μαθητή απλά εφαρμογή των βασικών αλγοριθμικών δομών (επιλογή, επανάληψη) εκτός από το θέμα του 2005 που ξεφεύγει από τα συνηθισμένα πρότυπα (μικρό, συμπαγές και απαιτούσε ανάλυση του προβλήματος). Ενδεχόμενα και φέτος να είναι αυξημένης δυσκολίας και να απαιτεί χρήση πινάκων ή και υποπρογραμμάτων.
Ενώ στο **4ο θέμα** αναμένουμε πίνακες πιθανότατα δισδιάστατους και μονοδιάστατους, στο οποίο θα ζητείτε κάποια από τις τυπικές επεξεργασίες πινάκων. Αναμένεται πρόγραμμα και ενδεχόμενα κάποια από τα ερωτήματα να ζητηθεί να υλοποιηθούν με υποπρόγραμμα (Διαδικασία – Συνάρτηση).
Ο μαθητής πρέπει να διαβάσει τις εκφωνήσεις πολλές φορές και αφού τις κατανοήσει να προχωρήσει σε διαδικασία λύσης ενώ αν ζητηθεί να <τρέξουμε> κώδικα καλό θα είναι να επαληθεύσει την άσκηση δύο φορές σε διαφορετικές σελίδες πριν τη μεταφέρει στο καθαρό.

**ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΚΑΙ ΕΡΩΤΗΣΕΙΣ ΣΤΗ ΘΕΩΡΙΑ (με βάση το βιβλίο)**

**Κεφάλαιο 1**

Πρόβλημα, κατανόηση, σαφήνεια, δεδομένο, πληροφορία, επεξεργασία δεδομένων, ζητούμενο, δομή, διαγραμματική αναπαράσταση/ανάλυση, καθορισμός απαιτήσεων, στάδια αντιμετώπισης προβλήματος, κατανόηση, ανάλυση, επίλυση, κατηγορίες προβλημάτων, κριτήρια κατηγοριοποίησης, επιλυσιμότητα, επιλύσιμα, ανοικτά, άλυτα, βαθμός δόμησης, δομημένα, ημιδομημένα, αδόμητα, είδος επίλυσης, απόφασης, υπολογιστικά, βελτιστοποίησης, λόγοι που αναθέτουμε ένα πρόβλημα στον υπολογιστή, λειτουργίες υπολογιστή

1. Δώστε τον ορισμό του προβλήματος
2. Τι σημαίνει ο όρος "δεδομένο", "πληροφορία", και "επεξεργασία δεδομένων";'
3. Τι είναι η δομή ενός προβλήματος;
4. Από ποιους παράγοντες επηρεάζεται η κατανόηση ενός προβλήματος;
5. Περιγράψτε τα τρία στάδια αντιμετώπισης ενός προβλήματος.
6. Σύμφωνα με ποια κριτήρια κατηγοριοποιούμε τα προβλήματα;
7. Περιγράψτε τις κατηγορίες προβλημάτων, σύμφωνα με τη δυνατότητα επίλυσής τους και δώστε αντίστοιχα παραδείγματα.
8. Περιγράψτε τις κατηγορίες προβλημάτων, σύμφωνα με το βαθμό δόμησής τους και δώστε αντίστοιχα παραδείγματα.
9. Περιγράψτε τις κατηγορίες προβλημάτων, σύμφωνα με το είδος της επίλυσης που επιζητούμε και δώστε αντίστοιχα παραδείγματα
10. Για ποιους λόγους μπορεί να αναθέσουμε την επίλυση ενός προβλήματος σε υπολογιστή;
11. Ποιες είναι οι τρεις λειτουργίες που μπορεί να εκτελέσει ένας υπολογιστής;

**Κεφάλαιο 2**

Αλγόριθμος, είσοδος, έξοδος, καθοριστικότητα, περατότητα, αποτελεσματικότητα, σπουδαιότητα αλγορίθμων, υλικού, γλωσσών προγραμματισμού, θεωρητική, αναλυτική, περιγραφή αλγορίθμων, ελεύθερο κείμενο, διαγραμματικές τεχνικές, φυσική γλώσσα κατά βήματα, κωδικοποίηση, συμβολισμός διαγραμμάτων ροής, έλλειψη, πλάγιο παραλληλόγραμμο, ορθογώνιο παραλληλόγραμμο, ρόμβος, βέλη, ψευδογλώσσα, σταθερές, μεταβλητές, τύποι μεταβλητών, πραγματικός, ακέραιος, αλφαριθμητικός, λογικός, τελεστές, αριθμητικοί, συγκριτικοί, λογικοί, ιεραρχία τελεστών, πίνακας αληθείας, εκφράσεις, εντολή εκχώρησης, σύνταξη εντολής εκχώρησης/απόδοσης τιμής, σχόλια, δεσμευμένες λέξεις, διάβασε, εμφάνισε, εκτύπωσε, δεδομένα, αποτελέσματα, γενική μορφή αλγορίθμου, είδη δομών, δομή ακολουθίας, δομές επιλογής, δομές επανάληψης, απλή επιλογή, σύνθετη επιλογή, πολλαπλές επιλογές, εμφωλευμένες διαδικασίες, διαφορές 2 μορφών πολλαπλών επιλογών, βρόχος επανάληψης, είδη δομών επαναλήψεων, άγνωστο και γνωστό πλήθος επαναλήψεων, συνθήκη τέλους, συνθήκη συνέχειας, βήμα, αρχικοποίηση μεταβλητών, μετρητής, σύνταξη δομών, ονόματα δομών

1. Δώστε τον ορισμό του αλγόριθμου.
2. Περιγράψτε τα κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος.
3. Από ποιες σκοπιές μελετά τους αλγορίθμους η Πληροφορική;
4. Με ποιους τρόπους μπορεί να αναπαρασταθεί ένας αλγόριθμος; Ποια μειονεκτήματα έχει ο καθένας;

5. Ποια είναι τα σύμβολα που χρησιμοποιούμε στα διαγράμματα ροής; Σχεδιάστε και περιγράψτε τα.
6. Με ποιες εντολές ξεκινά και τερματίζεται ένας αλγόριθμος; Πώς μπορούμε να διαβάσουμε κάτι από το πληκτρολόγιο;
7. Πως μπορούμε να γράψουμε κάτι στην οθόνη και στον εκτυπωτή;
8. Τι είναι οι σταθερές; Τι είναι οι μεταβλητές;
9. Ποιες κατηγορίες τελεστών υπάρχουν; Αναφέρετε όλους τους τελεστές ανά κατηγορία.
10. Τι κάνει η εντολή εκχώρησης τιμής;
11. Τι εννοούμε με τον όρο αλγοριθμική δομή – συνιστώσα. Ποιες αλγοριθμικές δομές γνωρίζετε;
12. Γράψτε τις 4 διαφορετικές μορφές της εντολής ΑΝ.
13. Με ποιους δύο τρόπους μπορούμε να λάβουμε πολλές διαφορετικές αποφάσεις σε έναν αλγόριθμο;
14. Τι ονομάζουμε λογική συνθήκη;
15. Περιγράψτε τους λογικούς τελεστές ή, και, όχι.
16. Δώστε ένα παράδειγμα της εντολής ΟΣΟ … ΕΠΑΝΑΛΑΒΕ, ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ … ΜΕΧΡΙΣ_ΟΤΟΥ και ΓΙΑ … ΑΠΟ … ΜΕΧΡΙ … ΜΕ_ΒΗΜΑ. Κάντε και τα αντίστοιχα διαγράμματα ροής.
17. Τι είναι ολίσθηση;
18. Περιγράψτε τα βήματα της μεθόδου Πολ/σμός αλα-Ρωσικά.
19. Έστω οι ακέραιοι 45 και 37. Βρείτε το γινόμενό τους με τη μέθοδο του Πολ/σμού αλα-Ρωσικά.
20. Δώστε τον αλγόριθμο (πολ/σμός αλά_Ρωσικά) και εξηγήστε τι κάνει.

**Κεφάλαιο 3**

Δεδομένα, επεξεργασία, πληροφορία, σκοπιές που η Πληροφορική μελετά τα δεδομένα, υλικό, γλώσσες προγραμματισμού, ανάλυση δεδομένων, ορισμός δομής δεδομένων, λειτουργίες δομών δεδομένων, προσπέλαση, εισαγωγή, διαγραφή, αναζήτηση, ταξινόμηση, αντιγραφή, συγχώνευση, διαχωρισμός, στατικές, δυναμικές δομές, αλγόριθμοι+δομές δεδομένων = προγράμματα, ορισμός πίνακα, χρήση πίνακα, δείκτες, μονοδιάστατοι, δισδιάστατοι, θέση στοιχείου, τιμή στοιχείου, γραμμές, στήλες, τετραγωνικός πίνακας, στοίβα (LIFO), ώθηση, απώθηση, κορυφή, υπερχείλιση, υποχείλιση, ουρά (FIFO), εισαγωγή, εξαγωγή, σειριακή αναζήτηση, αλγόριθμος σειριακής αναζήτησης (πολύ καλά), περιπτώσεις χρήσης αλγορίθμου σειριακής, ταξινόμηση, μαθηματικός ορισμός ταξινόμησης κατά αύξουσα ή φθίνουσα διάταξη, αλγόριθμος ευθείας ανταλλαγής ή φυσαλίδας ή bubble sort (πολύ καλά), φθίνουσα ή αύξουσα διάταξη.

1. Περιγράψτε τις σκοπιές από τις οποίες μελετά τα δεδομένα η Πληροφορική.
2. Δώστε τον ορισμό της δομής δεδομένων. Ποιες είναι οι βασικές λειτουργίες επί των δομών δεδομένων;
3. Τι είναι οι δυναμικές δομές δεδομένων; Τι είναι οι στατικές δομές δεδομένων;
4. Περιγράψτε τη δομή της στοίβας, καθώς και τις κύριες λειτουργίες της. Τι σημαίνει δομή LIFO;
5. Πως μπορεί να υλοποιηθεί μια στοίβα με τη βοήθεια ενός μονοδιάστατου πίνακα;
6. Περιγράψτε τη δομή της ουράς, καθώς και τις κύριες λειτουργίες της. Τι σημαίνει δομή FIFO;
7. Πως μπορεί να υλοποιηθεί μια ουρά με τη βοήθεια ενός μονοδιάστατου πίνακα;
8. Τι είναι η σειριακή (γραμμική) μέθοδος αναζήτησης; Σε ποιες περιπτώσεις δικαιολογείται η χρήση της; Δώστε τον αντίστοιχο αλγόριθμο.
9. Δώστε τον ορισμό της ταξινόμησης (διάταξης).
10. Τι σημαίνει δομή δεδομένων δευτερεύουσας μνήμης;
11. Δώστε τον αλγόριθμο της ταξινόμησης ευθείας ανταλλαγής (ταξινόμηση φυσαλίδας)

**Κεφάλαιο 4**

Στόχος ανάλυσης προβλήματος, βήματα ανάλυσης σε υπολογιστικό περιβάλλον, τεχνικές σχεδίασης αλγορίθμων, χαρακτηριστικά τεχνικών, 'Διαίρει και Βασίλευε', Μέθοδος δυναμικού προγραμματισμού. Άπληστη μέθοδος

1. Ποια βήματα περιλαμβάνει η ανάλυση ενός προβλήματος σε ένα προγραμματιστικό περιβάλλον;
2. Ποια είναι τα πλεονεκτήματα των μεθόδων ανάλυσης και επίλυσης προβλημάτων;
3. Ποια χαρακτηριστικά πρέπει να έχει μία τεχνική επίλυσης προβλημάτων ώστε να αποτελεί ξεχωριστή κατηγορία;
4. Αναφέρατε τρεις τυποποιημένες κατηγορίες τεχνικών επίλυσης προβλημάτων.

**Κεφάλαιο 6**

Πρόγραμμα, προγραμματισμός, δυαδικό σύστημα, πρώτοι υπολογιστές-σύγχρονοι υπολογιστές (μέγεθος, δυνατότητες, ταχύτητες), γλώσσες μηχανής, ορισμός-μειονεκτήματα γλωσσών μηχανής, συμβολικές γλώσσες/γλώσσες χαμηλού επιπέδου, ορισμός-πλεονεκτήματα και μειονεκτήματα συμβολικών γλωσσών, αρχιτεκτονική υπολογιστή, έλλειψη μεταφερσιμότητας, γλώσσες υψηλού επιπέδου/γλώσσες 3ης γενιάς, μεταγλωττιστής, ορισμός-πλεονεκτήματα γλωσσών 3ης γενιάς, βασικές γλώσσες υψηλού επιπέδου, FORTRAN-μαθηματικά και επιστημονικά προβλήματα, COBOL-εμπορικές εφαρμογές-διαχείριση αρχείων δεδομένων, ALGOL-στόχος-αποτυχία, PL/I-αποτυχία, LISP-χειρισμός και επεξεργασία λιστών, PROLOG-τεχνητή νοημοσύνη, BASIC-εκμάθηση και εκπαίδευση-μικροσκελή προγράμματα-μικροϋπολογιστές-αρίθμηση γραμμών, LOGO-μαθητές μικρής ηλικίας, PASCAL - γενικής χρήσης - εξελιγμένες εφαρμογές-δομημένα προγράμματα, C-UNIX-ομοιότητες με pascal-λειτουργικά συστήματα - C++ → αντικειμενοστραφής γλώσσα, JAVA-SUN - αντικειμενοστραφής γλώσσα - διαδίκτυο, visual programming (οπτικός προγραμματισμός), event-driven programming (οδηγούμενος από το γεγονός προγραμματισμός), visual C++, visual Basic, γλώσσες 4ης γενιάς (SQL), ταξινόμηση γλωσσών προγραμματισμού (σελ. 128 πολύ καλά), φυσικές-τεχνητές γλώσσες, αλφάβητο, λεξιλόγιο, γραμματική, τυπικό, συντακτικό, σημασιολογία, διαφορές φυσικών και τεχνητών γλωσσών, δυνατότητα εξέλιξης, τεχνικές σχεδίασης, ιεραρχική σχεδίαση/top-down σχεδίαση, τμηματικός προγραμματισμός-ενότητες-πλεονεκτήματα τμηματικού προγρ/σμού, δομημένος προγραμματισμός – GOTO - λόγος ανάπτυξης δομημένου-αρχές δομημένου - υποπρογράμματα – πλεονεκτήματα δομημένου-ιστορία GOTO, αντικειμενοστραφής προγραμματισμός - ιδέα αντικειμενοστραφούς – ενέργειες - αντικείμενα (objects)-ευέλικτα προγράμματα, παράλληλος προγραμματισμός-ιδέα-πολλοί επεξεργαστές-μεγαλύτερες ταχύτητες, προγραμματιστικά περιβάλλοντα, μεταγλωττιστές, διερμηνευτές, συντάκτης, πηγαίο (source) πρόγραμμα, αντικείμενο (object) πρόγραμμα, συνδέτης-φορτωτής, βιβλιοθήκες, εκτελέσιμο πρόγραμμα, μεταγλώττιση και σύνδεση, χαρακτηριστικά, πλεονεκτήματα και μειονεκτήματα μεταγλωττιστών και διερμηνευτών (διαφορές), άμεση εκτέλεση και διόρθωση, αργή εκτέλεση, συνδυασμός μεταγλωττιστή-διερμηνευτή

1. Ποια στάδια περιλαμβάνει η επίλυση ενός προβλήματος με τον υπολογιστή;
2. Τι είναι ένα πρόγραμμα σε γλώσσα μηχανής;
3. Τι είναι οι συμβολικές γλώσσες (γλώσσες χαμηλού επιπέδου) και τι ο συμβολομεταφραστής;
4. Ποια τα μειονεκτήματα των συμβολικών γλωσσών;
5. Αναφέρατε τρεις γλώσσες υψηλού επιπέδου.
6. Για τι είναι κατάλληλη η FORTRAN;
7. Σε τι προσανατολίζεται η COBOL;
8. Η ALGOL είναι γλώσσα γενικής φύσης ή συνδέεται με συγκεκριμένες εφαρμογές;
9. Η PL/1 είναι γλώσσα γενικής φύσης ή επιτρέπει παράλληλο προγραμματισμό;
10. Η LISP είναι αντικειμενοστραφής γλώσσα ή προσανατολίζεται σε χειρισμό λιστών από σύμβολα;
11. Η PROLOG προσανατολίζεται σε εμπορικές εφαρμογές ή σε προβλήματα τεχνητής νοημοσύνης;
12. Για τι είναι κατάλληλη η BASIC; Τα προγράμματά της μεταγλωττίζονται ή εκτελούνται με τη βοήθεια διερμηνευτή;
13. Για τι είναι κατάλληλη η PASCAL;
14. Ποιο είναι το σημαντικότερο πλεονέκτημα της JAVA; Χρησιμοποιείται για αντικειμενοστραφή ή δομημένο προγραμματισμό;
15. Η C είναι γλώσσα προγραμματισμού συστημάτων ή χειρισμού βάσεων δεδομένων;
16. Τι σημαίνουν οι όροι προγραμματισμός οδηγούμενος από τα γεγονότα και οπτικός προγραμματισμός;
17. Ποια είναι τα πλεονεκτήματα της χρήσης γλωσσών υψηλού επιπέδου;
18. Η SQL επιτρέπει οπτικό προγραμματισμό ή είναι γλώσσα χειρισμού βάσεων δεδομένων μέσω ερωταπαντήσεων;
19. Ποια είναι τα στοιχεία που προσδιορίζουν μία γλώσσα;
20. Τι είναι το αλφάβητο μιας γλώσσας; Τι είναι το λεξιλόγιο μιας γλώσσας;
21. Τι είναι το τυπικό και τι το συντακτικό της γραμματικής μιας γλώσσας; Τι σημαίνει σημασιολογία μίας γλώσσας;
22. Ποιες είναι οι διαφορές φυσικών και τεχνητών γλωσσών;
23. Είναι δυνατόν να εξελιχθεί μία γλώσσα προγραμματισμού;
24. Ποιος είναι ο σκοπός της από πάνω προς τα κάτω σχεδίασης προγραμμάτων;
25. Ποια τα πλεονεκτήματα του τμηματικού προγραμματισμού;

26. Η εντολή GOTO που αλλάζει τη ροή εκτέλεσης ενός προγράμματος χρειάζεται στο δομημένο προγραμματισμό ή είναι περιττή;
27. Δώστε τον ορισμό του δομημένου προγραμματισμού. Ποια είναι τα πλεονεκτήματά του;
28. Ο αντικειμενοστραφής προγραμματισμός στηρίζει τη φιλοσοφία του στα δεδομένα ή στις ενέργειες που γίνονται πάνω σε αυτά;
29. Σε ποιες βασικές αρχές στηρίζεται ο δομημένος προγραμματισμός;
30. Μπορεί οποιοδήποτε πρόγραμμα να εκτελεστεί σε πολλούς επεξεργαστές ή πρέπει να έχει προγραμματιστεί με αυτό το σκοπό;
31. Τι είναι ένας μεταγλωττιστής;
32. Περιγράψτε τα στάδια μεταγλώττισης και σύνδεσης ενός προγράμματος από το αρχικό πρόγραμμα ως το εκτελέσιμο.
33. Τι είναι ένας διερμηνευτής;
34. Ποιες οι διαφορές μεταγλωττιστή διερμηνευτή;
35. Περιγράψτε τη διαδικασία μετάφρασης και εκτέλεσης ενός προγράμματος σε έναν διερμηνευτή. Κάντε σχήμα.
36. Τι είναι ο συντάκτης και τι ο συνδέτης;
37. Ποια είναι τα πιθανά λάθη που μπορεί να εμφανιστούν κατά την υλοποίηση και την εκτέλεση ενός προγράμματος; Ποια είναι πιο δύσκολο να διορθωθούν;

## Κεφάλαια 7,8,9

Διαφορές ψευδογλώσσας και ΓΛΩΣΣΑΣ, ονόματα σταθερών, μεταβλητών, προγραμμάτων, δηλώσεις σταθερών, μεταβλητών, πινάκων, λόγοι χρήσης πινάκων, δήλωση μεγέθους πινάκων, ΧΑΡΑΚΤΗΡΕΣ, συναρτήσεις, ΗΜ(Χ), ΣΥΝ(Χ), ΕΦ(Χ), ΛΟΓ(Χ), Ε(Χ), Α_Μ(Χ), Α_Τ(Χ), Τ_Ρ(Χ), ΓΡΑΨΕ, μονά εισαγωγικά, τελεστές σύγκρισης, γενική μορφή προγράμματος σε ΓΛΩΣΣΑ, τιμή φρουρός.

1. Ποιους τύπους δεδομένων γνωρίζετε; Αναφέρατε δύο παραδείγματα για κάθε τύπο.
2. Ποιες κατηγορίες τελεστών γνωρίζετε; Ποιοι τελεστές ανήκουν ανά κατηγορία.
3. Ποια η προτεραιότητα των αριθμητικών πράξεων;
4. Τι είναι σταθερά και τι μεταβλητή;
5. Ποιους κανόνες ακολουθούμε για την ονοματολογία (μεταβλητών, σταθερών, τίτλος προγράμματος) σε ένα πρόγραμμα;
6. Ποια η διαφορά των εντολών ΔΙΑΒΑΣΕ και ΓΡΑΨΕ;
7. Περιγράψτε τη δομή ενός προγράμματος.
8. Να αναφέρετε τις συναρτήσεις που μπορούμε να χρησιμοποιήσουμε στη ΓΛΩΣΣΑ.
9. Τι γνωρίζετε για την εντολή εκχώρησης τιμής;
10. Πότε χρησιμοποιείται η εντολή ΕΠΙΛΕΞΕ;
11. Ποιες οι διαφορές της ΌΣΟ ... ΕΠΑΝΑΛΑΒΕ και της εντολής ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ_ΟΤΟΥ;
12. Ποια τα μειονεκτήματα της χρήσης πινάκων;
13. Τι γνωρίζετε για τους δείκτες ή διατακτικούς αριθμούς πινάκων;
14. Ποιες είναι οι τυπικές επεξεργασίες πινάκων;
15. Τι ονομάζεται συγχώνευση πινάκων;

## Κεφάλαιο 10

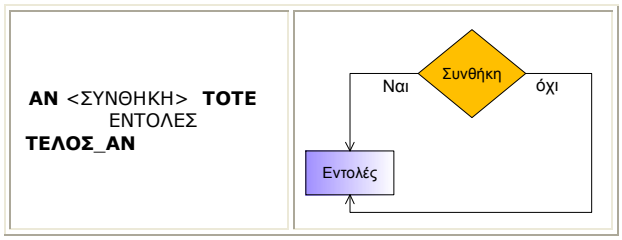
Τμηματικός προγραμματισμός, υποπρόγραμμα, χαρακτηριστικά υποπρογραμμάτων: α) μια είσοδο-μια έξοδο, β) ανεξαρτησία από τα άλλα, γ) όχι πολύ μεγάλο, πλεονεκτήματα τμηματικού προγραμματισμού: α) ευκολότερη ανάπτυξη αλγορίθμου προγράμματος, β) ευκολότερη κατανόηση-διόρθωση, γ) λιγότερος χρόνος δ) επέκταση δυνατοτήτων των γλωσσών, παράμετρος, συνάρτηση, διαδικασία, «ΚΑΛΕΣΕ», πραγματικές παράμετροι, τυπικές παράμετροι, κανόνες που πρέπει να ικανοποιούν οι παράμετροι

1. Τι είναι τμηματικός προγραμματισμός;
2. Τι είναι υποπρόγραμμα; Δώστε παραδείγματα.
3. Ποια είναι χαρακτηριστικά – ιδιότητες των υποπρογραμμάτων;
4. Ποια είναι τα πλεονεκτήματα του τμηματικού προγραμματισμού;
5. Τι είναι παράμετροι ενός υποπρογράμματος;
6. Τι είναι διαδικασία (υποπρόγραμμα). Δώστε παράδειγμα κλήσης διαδικασίας με χρήση παραμέτρων
7. Τι είναι συνάρτηση (υποπρόγραμμα). Δώστε παράδειγμα κλήσης συνάρτησης με χρήση παραμέτρων

## ΓΕΝΙΚΕΣ ΜΟΡΦΕΣ ΔΟΜΗΣ ΕΠΙΛΟΓΗΣ

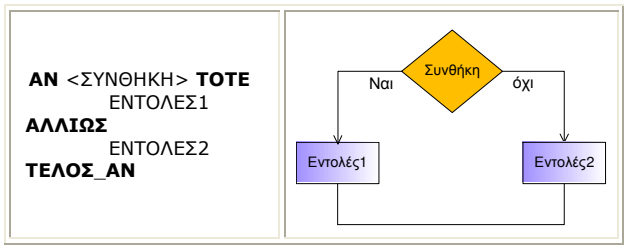
### Απλή επιλογή

Χρησιμοποιείται στην περίπτωση που έχουμε να εκτελέσουμε μια ομάδα εντολών όταν η συνθήκη είναι αληθής.



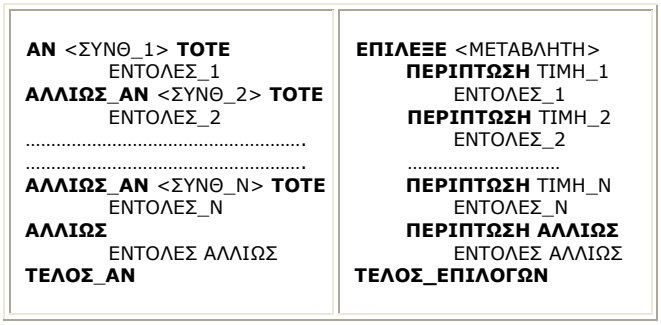

### Σύνθετη επιλογή

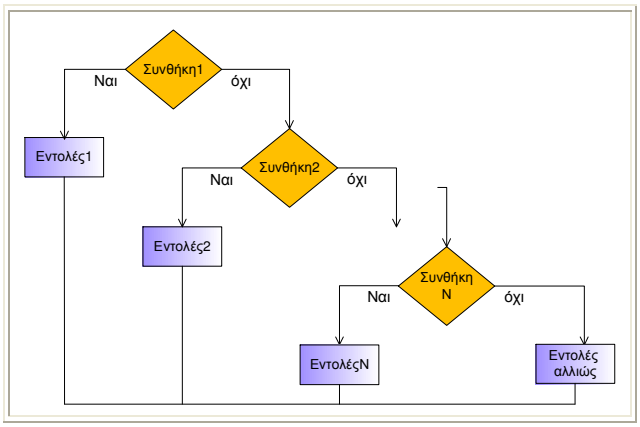
Χρησιμοποιείται στην περίπτωση που έχουμε να επιλέξουμε μεταξύ δύο περιπτώσεων που η μία αποκλείει την άλλη. Για παράδειγμα: Να γράψετε τμήμα εντολών το οποίο θα διαβάζει έναν αριθμό και θα εξετάζει αν είναι άρτιος ή περιττός.




### Πολλαπλή επιλογή

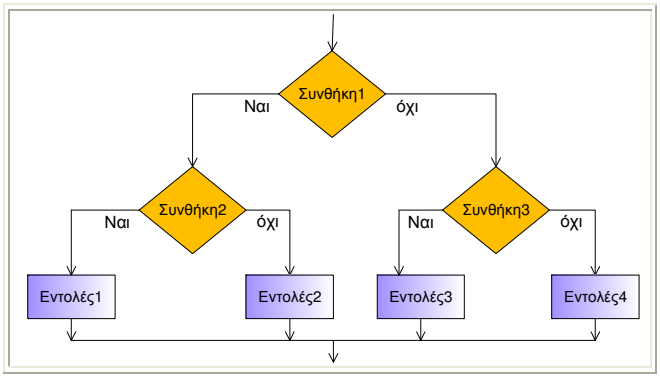
Χρησιμοποιείται στην περίπτωση που έχουμε να επιλέξουμε μεταξύ πολλών περιπτώσεων (περισσότερων των δύο) ή μεταξύ δύο περιπτώσεων που η μία δεν αποκλείει την άλλη. Για παράδειγμα: Να γράψετε τμήμα εντολών το οποίο να διαβάζει έναν αριθμό και αν αυτός είναι ίσος με 1 να τυπώνει ΄ΕΝΑ΄ ενώ αν είναι ίσος με 2 να τυπώνει ΄ΔΥΟ΄.




### Εμφωλευμένη επιλογή

```
ΑΝ <ΣΥΝΘΗΚΗ_1> ΤΟΤΕ
        ΑΝ <ΣΥΝΘΗΚΗ_2> ΤΟΤΕ
                ΕΝΤΟΛΕΣ_1
        ΑΛΛΙΩΣ
                ΕΝΤΟΛΕΣ_2
        ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
        ΑΝ <ΣΥΝΘΗΚΗ_3> ΤΟΤΕ
                ΕΝΤΟΛΕΣ_3
        ΑΛΛΙΩΣ
                ΕΝΤΟΛΕΣ_4
        ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
```



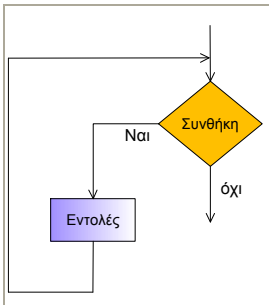

## ΓΕΝΙΚΕΣ ΜΟΡΦΕΣ ΔΟΜΗΣ ΕΠΑΝΑΛΗΨΗΣ

### Όσο ... επανάλαβε

Χρησιμοποιείται στις περιπτώσεις που πρέπει να ελεγχθεί πρώτα μία συνθήκη και μετά να εκτελεστεί η ομάδα εντολών. Προσοχή, σε αυτή τη δομή οι εντολές μπορεί να μην εκτελεστούν ποτέ. Αυτό οφείλεται στο ότι η συνθήκη βρίσκεται στην αρχή της δομής. Έτσι, αν η συνθήκη είναι ψευδής, θα εκτελεστεί η εντολή μετά το **Τέλος_επανάληψης**.

Η δομή αυτή χρησιμοποιείται όταν είναι είτε γνωστός, είτε άγνωστος ο αριθμός των επαναλήψεων, αλλά προτείνεται να χρησιμοποιείται στη δεύτερη περίπτωση.

```
ΌΣΟ <ΣΥΝΘΗΚΗ> ΕΠΑΝΑΛΑΒΕ
        Εντολές
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

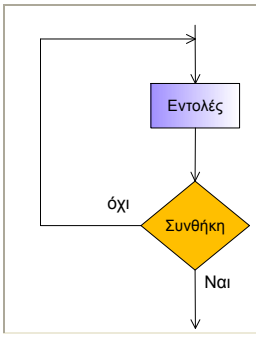



### Αρχή_επανάληψης ... μέχρις_ότου

Χρησιμοποιείται σε περιπτώσεις όπου οι εντολές στη δομή επανάληψης (**Βρόχος**) πρέπει να εκτελεστούν τουλάχιστον μία φορά. Αυτό οφείλεται στο ότι η συνθήκη βρίσκεται στο τέλος της δομής.

Η δομή αυτή, όπως και η προηγούμενη προτείνεται να χρησιμοποιείται όταν είναι άγνωστο το πόσες φορές χρειάζεται να επαναληφθεί η εκτέλεση της ομάδας εντολών.

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        Εντολές
ΜΕΧΡΙΣ_ΟΤΟΥ <ΣΥΝΘΗΚΗ>
```




### Για ... από ... μέχρι ... με_βήμα

Χρησιμοποιείται όταν είναι εκ των προτέρων γνωστό το πλήθος των επαναλήψεων της εκτέλεσης της ομάδας εντολών. Αν δεν είναι γνωστός ο αριθμός των επαναλήψεων, τότε δεν μπορεί να χρησιμοποιηθεί η δομή αυτή.

```
ΓΙΑ <μετ> ΑΠΟ <ατ> ΜΕΧΡΙ <ττ> ΜΕ_ΒΗΜΑ βμ
        Εντολές
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Αν το βήμα είναι 1 τότε παραλείπουμε την έκφραση **ΜΕ_ΒΗΜΑ** 1.

Αν το βήμα είναι **θετικό** τότε αυτή η δομή επανάληψης εκτελεί τα εξής βήματα:

1. Θέτει τον μετρητή ίσο με την αρχική τιμή.
2. Εξετάζει αν η τιμή του μετρητή είναι **μικρότερη ή ίση** από την τελική τιμή
3. **Αν είναι** τότε εκτελεί τις εντολές που βρίσκονται μέσα στην επανάληψη, αυξάνει τον μετρητή κατά όσο του υποδεικνύει το **βήμα** και πηγαίνει πάλι στο βήμα 2.
4. **Αν δεν είναι** τότε πηγαίνει στην πρώτη εντολή μετά το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**

Αν το βήμα είναι **αρνητικό** τότε αυτή η δομή επανάληψης εκτελεί τα εξής βήματα:

1. Θέτει τον μετρητή ίσο με την αρχική τιμή.
2. Εξετάζει αν η τιμή του μετρητή είναι **μεγαλύτερη ή ίση** από την τελική τιμή
3. **Αν είναι** τότε εκτελεί τις εντολές που βρίσκονται μέσα στην επανάληψη, αυξάνει τον μετρητή κατά όσο του υποδεικνύει το **βήμα** (το βήμα θα είναι αρνητικό άρα στην ουσία θα μειωθεί η τιμή του μετρητή) και πηγαίνει πάλι στο βήμα 2.
4. **Αν δεν είναι** τότε πηγαίνει στην πρώτη εντολή μετά το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**

Αν το βήμα είναι **μηδέν** τότε ο βρόχος εκτελείται επ' άπειρον (υπολογιστική διαδικασία ή **ατέρμονας βρόχος**).

## ΒΑΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΠΙΝΑΚΩΝ

**(*Όλα τα νούμερα και τα ονόματα των πινάκων είναι υποθετικά*)**

### Μονοδιάστατοι πίνακες

**Δημιουργία πίνακα Α[100]**

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
        ΔΙΑΒΑΣΕ Α[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**Εμφάνιση στοιχείων πίνακα Α[100]**

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100       ! Εμφανίζει τα στοιχεία του πίνακα
        ΓΡΑΨΕ Α[i]          ! με σειρά
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 100 ΜΕΧΡΙ 1 ΜΕ_ΒΗΜΑ -1  ! Εμφανίζει τα στοιχεία του
        ΓΡΑΨΕ Α[i]          ! πίνακα με ανάποδη σειρά
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ Α[5]          ! Εμφανίζει το 5ο στοιχείο του πίνακα
```

**Άθροισμα, γινόμενο, μέσος όρος στοιχείων πίνακα Α[100]**

```
ΑΘΡ ← 0
ΓΙΝ ← 1
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
        ΑΘΡ ← ΑΘΡ + Α[i]
        ΓΙΝ ← ΓΙΝ * Α[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ ← ΑΘΡ/100
```

**Μέγιστο/Θέση μεγίστου, Ελάχιστο/Θέση ελαχίστου πίνακα Α[100]**

```
ΜΑΧ ← Α[1]
ΘΕΣΗ_ΜΑΧ ← 1
ΜΙΝ ← Α[1]
ΘΕΣΗ_ΜΙΝ ← 1
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100
        ΑΝ Α[i] > ΜΑΧ ΤΟΤΕ
                ΜΑΧ ← Α[i]
                ΘΕΣΗ_ΜΑΧ ← i
        ΤΕΛΟΣ_ΑΝ
        ΑΝ Α[i] < ΜΙΝ ΤΟΤΕ
                ΜΙΝ ← Α[i]
                ΘΕΣΗ_ΜΙΝ ← i
        ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**Σειριακή ή γραμμική αναζήτηση του στοιχείου key στον πίνακα Α[100] (Αν τυχόν υπάρχει)**

```
ΔΙΑΒΑΣΕ key
ΘΕΣΗ ← 0
ΒΡΕΘΗΚΕ ← ψευδής
i ← 1
ΌΣΟ i <= 100 ΚΑΙ ΒΡΕΘΗΚΕ = ψευδής ΕΠΑΝΑΛΑΒΕ
        ΑΝ Α[i] = key ΤΟΤΕ
                ΒΡΕΘΗΚΕ ← αληθής
                ΘΕΣΗ ← i
        ΑΛΛΙΩΣ
                i ← i + 1
        ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ ΒΡΕΘΗΚΕ = ψευδής ΤΟΤΕ
        ΓΡΑΨΕ ΄Δεν βρέθηκε΄
ΑΛΛΙΩΣ
        ΓΡΑΨΕ ΄Βρέθηκε στη θέση ΄, ΘΕΣΗ
ΤΕΛΟΣ_ΑΝ
```

Η παραπάνω αναζήτηση απαντά σε <u>ερωτήματα του τύπου:</u> Αναζητήστε αν τυχόν υπάρχει το στοιχείο key στον πίνακα Α, εμφανίζοντας σχετικό μήνυμα.

Αν τυχόν βρεθεί το στοιχείο, σταματά την αναζήτηση και εμφανίζει τη θέση που το εντόπισε για πρώτη φορά.

**Απλή αναζήτηση (Πόσες φορές και σε ποιες θέσεις)**

```
κ ← 0
ΔΙΑΒΑΣΕ key
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
    ΑΝ Α[i] = key ΤΟΤΕ
        κ ← κ + 1
        ΓΡΑΨΕ ΄Βρέθηκε στη θέση ΄, i
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ κ
```

**Ταξινόμηση με μέθοδο φυσαλίδας ή ευθείας ανταλλαγής στοιχείων πίνακα Α[100]**

```
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100
    ΓΙΑ j ΑΠΟ 100 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
        ΑΝ Α[j-1] > Α[j] ΤΟΤΕ
                Temp ← Α[j-1]
                Α[j-1] ← Α[j]
                Α[j] ← Temp
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Η παραπάνω ταξινόμηση ταξινομεί ένα μονοδιάστατο πίνακα 100 θέσεων σε **αύξουσα** διάταξη. Αν θελήσω να τον ταξινομήσω κατά φθίνουσα τότε αλλάζω μόνο το συγκριτικό τελεστή του ΑΝ

ΑΝ Α[j-1] **<** Α[j] ΤΟΤΕ

Η μεταβλητή i κάνει τα Ν-1 = 99 περάσματα της φυσαλίδας, ενώ η μεταβλητή j κάνει τις συγκρίσεις γειτονικών στοιχείων μέσα από κάθε πέρασμα.

Αντί για τις τρείς εντολές με το Temp (που κάνουν την αντιμετάθεση των στοιχείων) μπορώ να χρησιμοποιήσω την εντολή

Αντιμετάθεσε Α[j-1], Α[j]

αλλά **μόνο** σε αλγόριθμο.

**Ταξινόμηση παράλληλων πινάκων**

```
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100
    ΓΙΑ j ΑΠΟ 100 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
        ΑΝ Α[j-1] > Α[j] ΤΟΤΕ
                Temp1 ← Α[j-1]
                Α[j-1] ← Α[j]
                Α[j] ← Temp1
                Temp2 ← Β[j-1]
                Β[j-1] ← Β[j]
                Β[j] ← Temp2
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Οι παραπάνω εντολές χρησιμοποιούνται όταν έχουμε <u>πίνακες με αντιστοιχία θέσεων</u> (<u>παράλληλοι</u> πίνακες) και για τις ανάγκες κάποιου ερωτήματος χρειάζεται να ταξινομήσουμε τον έναν από αυτούς (εδώ τον Α). Τότε, περνάμε την ίδια αντιμετάθεση θέσεων και στον άλλο (Β).

**Έξυπνη ταξινόμηση (Για Προχωρημένους)**

```
i ← 2
ΑΝΤΙΜΕΤΑΘΕΣΗ ← αληθής
ΌΣΟ i <= 100 ΚΑΙ ΑΝΤΙΜΕΤΑΘΕΣΗ = αληθής ΕΠΑΝΑΛΑΒΕ
        ΑΝΤΙΜΕΤΑΘΕΣΗ ← ψευδής
        ΓΙΑ j ΑΠΟ 100 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
                ΑΝ Α[j-1] > Α[j] ΤΟΤΕ
                        Temp ← Α[j-1]
                        Α[j-1] ← Α[j]
                        Α[j] ← Temp
                        ΑΝΤΙΜΕΤΑΘΕΣΗ ← αληθής
                ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
        i ← i + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Οι παραπάνω εντολές ταξινομούν τον πίνακά μας κατά αύξουσα σειρά με όσο το δυνατόν λιγότερο αριθμό περασμάτων. Μόλις ο πίνακας έχει ταξινομηθεί σταματά και η εκτέλεση των εντολών.

**Δισδιάστατοι πίνακες**

**Δημιουργία πίνακα Α[5, 10]**

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
        ΔΙΑΒΑΣΕ Α[i, j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**Εμφάνιση στοιχείων πίνακα Α[5, 10]**

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5    ! Σε πρόγραμμα χρησιμοποιείται
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10    ! η εντολή ΓΡΑΨΕ ενώ σε
        ΓΡΑΨΕ Α[i, j]       ! αλγόριθμο η εντολή
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ   ! Εμφάνισε ή Εκτύπωσε
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10    ! εμφανίζει τα στοιχεία μόνο
        ΓΡΑΨΕ Α[2, j]   ! της 2ης γραμμής του πίνακα
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5     ! εμφανίζει τα στοιχεία μόνο
        ΓΡΑΨΕ Α[i, 3]   ! της 3ης στήλης του πίνακα
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**Αθροισμα, μέσος όρος στοιχείων πίνακα Α[5, 10]**

```
ΑΘΡ ← 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
        ΑΘΡ ← ΑΘΡ + Α[i, j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ ← ΑΘΡ/50
```

**Μέγιστο/Θέση μεγίστου πίνακα Α[5, 10]**

```
max ← Α[1, 1]
Γραμμή_max ← 1
Στήλη_max ← 1
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
        ΑΝ Α[i, j] > max ΤΟΤΕ
            max ← Α[i, j]
            Γραμμή_max ← i
            Στήλη_max ← j
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΄Το μέγιστο είναι το ΄, max, ΄ και βρίσκεται
& στην γραμμή ΄, γραμμή_max, ΄ και στη
& στήλη ΄, στήλη_max
```

Με παρόμοιο τρόπο βρίσκουμε το ελάχιστο/θέση ελαχίστου.

**Άθροισμα στοιχείων 2ης γραμμής – Γινόμενο στοιχείων 3ης στήλης πίνακα Α[5, 10]**

```
ΑΘΡ ← 0
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
        ΑΘΡ ← ΑΘΡ + Α[2, j]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΝ ← 1
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
        ΓΙΝ ← ΓΙΝ * Α[i, 3]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**Μέγιστο στοιχείων 2ης γραμμής – Ελάχιστο στοιχείων 3ης στήλης πίνακα Α[5, 10]**

```
Max_2ΓΡ ← Α[2, 1]
ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 10
    ΑΝ Α[2, j] > Max_2ΓΡ ΤΟΤΕ
        Max_2ΓΡ ← Α[2, j]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Min_3ΣΤ ← Α[1, 3]
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 5
    ΑΝ Α[i, 3] < Min_3ΣΤ ΤΟΤΕ
        Min_3ΣΤ ← Α[i, 3]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**Άθροισμα ανά γραμμές/μέσος όρος ανά γραμμές πίνακα Α[5, 10]**

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    Sum[i] ← 0
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
        Sum[i] ← sum[i] + Α[i, j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟ[i] ← Sum[i] / 10
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**Άθροισμα ανά στήλες/μέσος όρος ανά στήλες πίνακα Α[5, 10]**

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    Sum[j] ← 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
        Sum[j] ← sum[j] + Α[i, j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟ[j] ← Sum[j] / 5
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**Μέγιστο ανά γραμμές πίνακα Α[5, 10]**

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    Max[i] ← Α[i, 1]
    ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 10
        ΑΝ Α[i, j] > Max[i] ΤΟΤΕ
            Max[i] ← Α[i, j]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Με παρόμοιο τρόπο βρίσκουμε το ελάχιστο ανά γραμμές.

**Ελάχιστο ανά στήλες πίνακα Α[5, 10]**

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    Min[j] ← Α[1, j]
    ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 5
        ΑΝ Α[i, j] < Min[j] ΤΟΤΕ
            Min[j] ← Α[i, j]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Με παρόμοιο τρόπο βρίσκουμε το μέγιστο ανά στήλες.

**Ταξινόμηση δισδιάστατου πίνακα Α[5, 10] ανά γραμμή**

```
ΓΙΑ k ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 10
        ΓΙΑ j ΑΠΟ 10 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
            ΑΝ Α[k, j-1] > Α[k, j] ΤΟΤΕ
                Temp ← Α[k, j-1]
                Α[k, j-1] ← Α[k, j]
                Α[k, j] ← Temp
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Οι παραπάνω εντολές ταξινομούν κάθε γραμμή ενός δισδιάστατου πίνακα Α[5, 10] σε **αύξουσα** διάταξη.

## ΠΡΟΤΕΙΝΟΜΕΝΑ ΘΕΜΑΤΑ

**1ο ΘΕΜΑ**

Δίνεται ότι Χ=10. Επίσης δίνεται ότι οι μεταβλητές Κ, Λ είναι πραγματικές και οι μεταβλητές Λ1, Λ2 λογικές οι οποίες έχουν πάρει τιμή. Να υπολογίσετε τις παρακάτω λογικές προτάσεις και να τις χαρακτηρίσετε χρησιμοποιώντας μια από τις τιμές Αληθής ή Ψευδής.

1. Πρόταση Α: Αληθής και όχι (Ψευδής = Αληθής)
2. Πρόταση Β: Αληθής και όχι ('Ψευδής' > 'Αληθής')
3. Πρόταση Γ: 'Μανόλης' > 'Μαρία' ή Χ ≠ 10
4. Πρόταση Δ: Χ <= 9 και (Κ*Λ > 100 ή Κ^2 + Λ^2 > 100)
5. Πρόταση Ε: (Λ1 ή όχι Λ1) ή (Λ2 ή όχι Λ1)
6. Πρόταση ΣΤ: (Λ1 ή όχι Λ2) και (Λ2 και όχι Λ2)

**ΛΥΣΗ**

1-Α, 2-Ψ, 3-Ψ, 4-Ψ, 5-Α, 6-Ψ

**1ο ΘΕΜΑ**

Δίνεται το παρακάτω τμήμα αλγορίθμου. Να βρείτε τι τιμές θα μπορούσε τυπικά να έχει η μεταβλητή z, αν γνωρίζουμε ότι μετά την εκτέλεση του η τιμή της μεταβλητής a είναι 3. (Εξηγήστε αναλυτικά την απάντησή σας)

```
a ← 0
c ← 0
Για i από z μέχρι 4 με_βήμα -2
        c ← c + 1
        Αν c mod 2 = 1 τότε
                a ← a + 1
        τέλος_αν
Τέλος_επανάληψης
```

**ΛΥΣΗ**

Για να είναι 3 η τιμή της μεταβλητής α, μετά την εκτέλεση των εντολών, πρέπει ο βρόχος να εκτελεστεί 5 φορές. Με βήμα -2 οι τιμές που θα πάρει ο μετρητής είναι z, z-2, z-4, z-6, z-8. Θα πρέπει λοιπόν να ισχύει: 4<= z-8 < 6. Οπότε προκύπτει ότι 12<= z < 14.

Με δεδομένο όμως ότι και 6η φορά να εκτελεστεί ο βρόχος το c παραμένει άρτιο (η τιμή του α δεν αλλάζει) μπορούμε να πούμε ότι ο z μπορεί να έχει **οποιαδήποτε τιμή στο [12, 16)**.

**1ο ΘΕΜΑ**

Να μετατρέψετε τα παρακάτω τμήματα εντολών σε ισοδύναμα χρησιμοποιώντας κάθε μία από τις δύο άλλες δομές επανάληψης:

| **ΌΣΟ ... ΕΠΑΝΑΛΑΒΕ** | **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ_ΟΤΟΥ** | **ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ ... ΜΕ_ΒΗΜΑ** |
|---|---|---|
| Α ← 5<br>Όσο Α > 0 επανάλαβε<br>Γράψε Α<br>Α ← Α – 1<br>Τέλος_επανάληψης | | |
| | Ι ← -4<br>Α ← 1<br>Αρχή_επανάληψης<br>Α ← Α * Ι<br>Ι ← Ι + 1<br>Μέχρις_ότου Ι > 0 | |
| | | Α ← 2<br>Για Ι από -4 μέχρι 4 με_βήμα 2<br>Α ← Α + 2<br>Τέλος_επανάληψης<br>Γράψε Α |

**ΛΥΣΗ**

| **ΌΣΟ ... ΕΠΑΝΑΛΑΒΕ** | **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ_ΟΤΟΥ** |
|---|---|
| Α ← 5<br>Όσο Α > 0 επανάλαβε<br>Γράψε Α<br>Α ← Α – 1<br>Τέλος_επανάληψης | Α ← 5<br>Αρχή_επανάληψης<br>Γράψε Α<br>Α ← Α – 1<br>Μέχρις_ότου Α<=0 |
| Ι ← -4<br>Α ← 1<br>Όσο Ι <= 0 επανάλαβε<br>Α ← Α * Ι<br>Ι ← Ι + 1<br>Τέλος_επανάληψης | Ι ← -4<br>Α ← 1<br>Αρχή_επανάληψης<br>Α ← Α * Ι<br>Ι ← Ι + 1<br>Μέχρις_ότου Ι > 0 |
| Α ← 2<br>Ι ← -4<br>Όσο Ι <= 4 επανάλαβε<br>Α ← Α + 2<br>Ι ← Ι + 2<br>Τέλος_επανάληψης<br>Γράψε Α | Α ← 2<br>Ι ← -4<br>Αρχή_επανάληψης<br>Α ← Α + 2<br>Ι ← Ι + 2<br>Μέχρις_ότου Ι > 4<br>Γράψε Α |

| **ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ ... ΜΕ_ΒΗΜΑ** |
|---|
| Για Α από 5 μέχρι 1 με_βήμα -1<br>Γράψε Α<br>Τέλος_επανάληψης |
| Για Ι από -4 μέχρι 0<br>Α ← Α * Ι<br>Τέλος_επανάληψης |
| Α ← 2<br>Για Ι από -4 μέχρι 4 με_βήμα 2<br>Α ← Α + 2<br>Τέλος_επανάληψης<br>Γράψε Α |

Δοκιμάστε να κάνετε το ίδιο για τα παρακάτω τμήματα εντολών:

| **ΌΣΟ ... ΕΠΑΝΑΛΑΒΕ** | **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ_ΟΤΟΥ** |
|---|---|
| i ← τιμή1<br>Όσο i <= τιμή2 επανάλαβε<br>Εντολές<br>i ← i + β<br>Τέλος_επανάληψης | |

| | |
|---|---|
| | Α ← 2<br>Β ← 3<br>Αρχή_επανάληψης<br>Εκτύπωσε Β<br>Β ← Β + 2<br>Μέχρις_ότου Β > 11 |
| | |

| **ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ ... ΜΕ_ΒΗΜΑ** |
|---|
| |
| |
| κ ← 0<br>Για i από 0 μέχρι 100 με_βήμα 5<br>α ← i^3<br>κ ← κ + α<br>Εκτύπωσε i, α<br>Τέλος_επανάληψης<br>Εκτύπωσε κ |

**1ο ΘΕΜΑ**

Να γράψετε τμήμα αλγορίθμου, που θα έχει το ίδιο αποτέλεσμα με το παρακάτω τμήμα:

```
δ ← α mod 10
Όσο δ > 0 επανάλαβε
    δ ← δ - 1
    γ ← γ + β
Τέλος_επανάληψης
```

χρησιμοποιώντας αντί της εντολής Όσο την εντολή Για. Στο νέο τμήμα αλγορίθμου να χρησιμοποιήσετε μόνο τις μεταβλητές α, β, γ, δ που χρησιμοποιεί το αρχικό τμήμα.

**ΛΥΣΗ**

```
Για δ από (α mod 10) - 1 μέχρι 0 με_βήμα -1
    γ ← γ + β
Τέλος_επανάληψης
Εμφάνισε Σ
```

**1ο ΘΕΜΑ**

Το παρακάτω τμήμα αλγορίθμου να μετατραπεί σε ισοδύναμο, χρησιμοποιώντας αποκλειστικά τη δομή επανάληψης Όσο ... Επανάλαβε

```
S ← 0
Για Κ από 1 μέχρι 5
    Για L από 1 μέχρι 7
        S ← S + 1
    Τέλος_επανάληψης
Τέλος_επανάληψης
Εμφάνισε S
```

**ΛΥΣΗ**

```
S ← 0
K ← 1
Όσο K <= 5 επανάλαβε
    L ← 1
    Όσο L <= 7 επανάλαβε
        S ← S + 1
        L ← L + 1
    Τέλος_επανάληψης
    K ← K + 1
Τέλος_επανάληψης
Εμφάνισε S
```

**1ο ΘΕΜΑ**

Δίνεται το τμήμα αλγορίθμου:

```
Διάβασε κ
Σ← 0
Αρχή_επανάληψης
    Διάβασε α
    Σ ← Σ+α
Μέχρις_ότου Σ>κ
Εμφάνισε Σ
```

Μετατρέψτε το παραπάνω τμήμα αλγορίθμου χρησιμοποιώντας την εντολή Όσο αντί της εντολής Αρχή_επανάληψης… Μέχρις_ότου, έτσι ώστε τα δύο τμήματα αλγορίθμου ( Όσο και Αρχή_επανάληψης… Μέχρις_ότου) να δίνουν ακριβώς την ίδια έξοδο όταν έχουν την ίδια είσοδο.

**ΛΥΣΗ**

```
Διάβασε κ
Διάβασε α
Σ ← α
Όσο Σ <= κ επανάλαβε
    Σ ← Σ + α
    Διάβασε α
Τέλος_επανάληψης
Εμφάνισε Σ
```

**1ο ΘΕΜΑ**

Δίνεται ο παρακάτω αλγόριθμος ο οποίος δημιουργήθηκε με σκοπό να υπολογίζει την τιμή της παράστασης F(X)=(X^2+1)/(X-1) για τις ακέραιες τιμές του Χ στο διάστημα [-5,7], να εκτυπώνει την τιμή που υπολογίζεται και να αθροίζει τις τιμές της F(X).

Α) Περιγράψτε το πρόβλημα που θα έχει ο αλγόριθμος όταν εκτελεστεί.

Β) Διορθώστε τον αλγόριθμο ώστε να τρέχει χωρίς πρόβλημα.

```
Αλγόριθμος  Παράσταση
Χ ← -5
S ← 0
Όσο  X<=7  επανάλαβε
        F ← (X^2+1)/(X-1)
        Εκτύπωσε  F
        S ← S+ F
        X ← X+1
Τέλος_επανάληψης
Εκτύπωσε  X, S
Τέλος  Παράσταση
```

**ΛΥΣΗ**

Α) Παραβίαση καθοριστικότητας όταν το Χ λάβει την τιμή 1

Β)

```
Αλγόριθμος  Παράσταση
Χ ← -5
S ← 0
Όσο  X<=7  επανάλαβε
    Αν X–1 <> 0 τότε
        F ← (X^2+1)/(X-1)
        Εκτύπωσε  F
        S ← S + F
    Τέλος_αν
    X ← X+1 !προσοχή, περατότητα, όχι εντός του Αν
Τέλος_επανάληψης
Εκτύπωσε  X, S
Τέλος  Παράσταση
```

**1ο ΘΕΜΑ**

Ο μονοδιάστατος αριθμητικός πίνακας Α έχει τα ακόλουθα στοιχεία:

| 1η θέση | 2η θέση | 3η θέση | 4η θέση | 5η θέση |
|---|---|---|---|---|
| 36 | 22 | 29 | 18 | 31 |

Στον πίνακα Α εφαρμόζεται φθίνουσα ταξινόμηση σύμφωνα με τον αλγόριθμο φυσαλίδας. Στον αλγόριθμο αυτό θεωρούμε ότι ο εξωτερικός βρόχος χρησιμοποιεί τη μεταβλητή Ι και ο εσωτερικός τη μεταβλητή J.

Να συμπληρώσετε τον ακόλουθο πίνακα τιμών για τις μεταβλητές και τα στοιχεία του πίνακα.

| | | Πίνακας Α | | | | |
|---|---|---|---|---|---|---|
| **I** | **J** | **1η** | **2η** | **3η** | **4η** | **5η** |
| | | 36 | 22 | 29 | 18 | 31 |
| 2 | 5 | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| 3 | | | | | | |
| | | | | | | |
| | | | | | | |

**ΛΥΣΗ**

| | | Πίνακας Α | | | | |
|---|---|---|---|---|---|---|
| **I** | **J** | **1η** | **2η** | **3η** | **4η** | **5η** |
| | | 36 | 22 | 29 | **18** | **31** |
| 2 | 5 | 36 | 22 | **29** | **31** | 18 |
| 2 | 4 | 36 | **22** | **31** | 29 | 18 |
| 2 | 3 | **36** | **31** | 22 | 29 | 18 |
| 2 | 2 | 36 | 31 | 22 | **29** | **18** |
| 3 | 5 | 36 | 31 | **22** | **29** | 18 |
| 3 | 4 | 36 | **31** | **29** | 22 | 18 |
| 3 | 3 | ***36*** | ***31*** | ***29*** | ***22*** | ***18*** |

**1ο ΘΕΜΑ**

Να μετατρέψετε το παρακάτω τμήμα εντολών από εμφωλευμένη επιλογή σε δομή πολλαπλής επιλογής Αν … αλλιώς_αν και Επίλεξε

```
Αν ποσότητα<=50 τότε
    κόστος ← ποσότητα*200
αλλιώς
    αν ποσότητα<=100 τότε
        κόστος ← ποσότητα *180
    αλλιώς
        αν ποσότητα <=200 τότε
            κόστος ← ποσότητα*160
        αλλιώς
            κόστος ← ποσότητα*150
        τέλος_αν
    τέλος_αν
τέλος_αν
```

**ΛΥΣΗ**

```
Αν ποσότητα<=50 τότε
    κόστος ← ποσότητα*200
αλλιώς_αν ποσότητα<=100 τότε
    κόστος ← ποσότητα *180
αλλιώς_αν ποσότητα <=200 τότε
    κόστος ← ποσότητα*160
αλλιώς
    κόστος ← ποσότητα*150
τέλος_αν
```

```
Επίλεξε ποσότητα
    Περίπτωση <= 50
        κόστος ← ποσότητα*200
    Περίπτωση <= 100
        κόστος ← ποσότητα*180
    Περίπτωση <= 200
        κόστος ← ποσότητα*160
    Περίπτωση αλλιώς
        κόστος ← ποσότητα*150
Τέλος_επιλογών
```

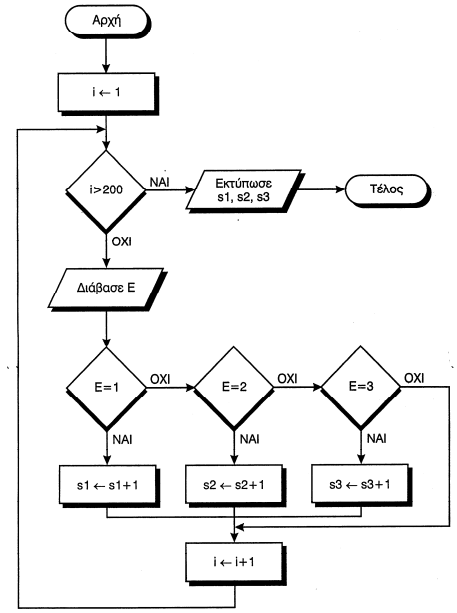
Δοκιμάστε να μετατρέψετε το παρακάτω τμήμα εντολών σε δομή πολλαπλής επιλογής Αν … αλλιώς_αν και εμφωλευμένη δομή επιλογής

```
ΔΙΑΒΑΣΕ α
ΕΠΙΛΕΞΕ α
    ΠΕΡΙΠΤΩΣΗ <= 0
        β ← 0
    ΠΕΡΙΠΤΩΣΗ <= 5
        β ← 5
    ΠΕΡΙΠΤΩΣΗ <= 10
        β ← 10
    ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
        β ← 100
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
ΓΡΑΨΕ β
```

**1ο ΘΕΜΑ**

Για το παρακάτω διάγραμμα ροής να γίνουν τα εξής:

- ✓ Να εξηγηθεί τι επιτυγχάνει
- ✓ Σε περίπτωση που περιέχει λάθη, να γίνουν οι απαραίτητες διορθώσεις ή προσθήκες εντολών
- ✓ Να μετατραπεί στον ισοδύναμο αλγόριθμο υπό μορφή ψευδοκώδικα




**ΛΥΣΗ**

- ✓ Ο αλγόριθμος διαβάζει 200 αριθμούς και υπολογίζει και εκτυπώνει τα πλήθη εκείνων που είναι ίσοι με 1, ίσοι με 2 και ίσοι με 3. Δηλαδή τυπώνει πόσοι άσσοι, πόσα δυάρια και πόσα τριάρια διαβάστηκαν
- ✓ Λάθος, έχει ξεχαστεί η αρχικοποίηση των τριων μετρητών (s1, s2 και s3) στην αρχή του αλγορίθμου. Δηλαδή πριν την έναρξη της επανάληψης πρέπει να βάλουμε τις εντολές:

```
S1 ← 0
S2 ← 0
S3 ← 0
```

- ✓

```
Αλγόριθμος Αριθμοί
i ← 1
s1 ← 0    s2 ← 0    s3 ← 0
Όσο ΟΧΙ (i > 200) επανάλαβε
        Διάβασε Ε
        Αν Ε = 1 τότε
            s1 ← s1 + 1
        Αλλιώς_αν  Ε = 2 τότε
            s2 ← s2 + 1
        Αλλιώς_αν  Ε = 3 τότε
            s3 ← s3 + 1
        Τέλος_αν
        i ← i + 1
Τέλος_επανάληψης
Εκτύπωσε s1,s2,s3
Τέλος Αριθμοί
```

**1ο ΘΕΜΑ**

Για τα παρακάτω τμήματα αλγορίθμων να αναφερθεί ποιο κριτήριο πληρότητας αλγορίθμου δεν ικανοποιείται;

| (1) | (2) |
|---|---|
| Αρχή_επανάληψης<br>α ← 10<br>Για i από 1 μέχρι 3<br>α ← α – 10<br>Τέλος_επανάληψης<br>Μέχρις_ότου α = 0 | Διάβασε α,β,γ<br>Αν β > γ τότε<br>X ← β/(2*α)<br>Τέλος_αν |

**ΛΥΣΗ**

(1) Πρόβλημα Περατότητας:
Η μεταβλητή α δεν θα γίνει ποτέ ίση με 0 μετά το τέλος της εντολής Για. Αρα η συνθήκη της Μέχρις_ότου θα είναι πάντα ψευδής (άπειρες επαναλήψεις)

(2) Πρόβλημα Καθοριστικότητας:
Αν διαβαστεί η τιμή 0 για την μεταβλητή α τότε επιχειρείται διαίρεση με το 0 στην εντολή εκχώρησης. Η συνθήκη της Αν πρέπει μετατραπεί σε: Αν (β>γ) ΚΑΙ (α≠0) τότε

**1ο ΘΕΜΑ**

Πόσες φορές θα εκτελεστούν οι παρακάτω επαναληπτικές δομές;

| (1) | (2) |
|---|---|
| κ ← 6<br>Αρχή_Επανάληψης<br>κ ← κ – 2<br>Μέχρις_ότου (κ > 0) | Αρχή_επανάληψης<br>κ ← 1<br>κ ← κ - 1<br>Μέχρις_ότου (κ = 0) |
| **(3)** | **(4)** |
| κ ← 4<br>Όσο κ ≠ 4 επανάλαβε<br>κ ← κ -1<br>Τέλος_επανάληψης | κ ← 0<br>Όσο κ ≠ 4 επανάλαβε<br>κ ← κ – 1<br>Τέλος_επανάληψης |

**ΛΥΣΗ**

(1) 1 φορά (2) 1 φορά

(3) 0 φορές (4) άπειρες φορές

**1ο ΘΕΜΑ**

Να μετατραπεί το παρακάτω τμήμα εντολών σε ένα ισοδύναμο που να χρησιμοποιεί μονο μία δομή πολλαπλής επιλογής (ΑΛΛΙΩΣ_ΑΝ). Δηλαδή να μετατραπούν οι συνθήκες έτσι ώστε οι εντολές να γραφούν ισοδύναμα με ένα ΤΕΛΟΣ_ ΑΝ

```
ΔΙΑΒΑΣΕ X,Y
ΑΝ X = 5 ΤΟΤΕ
    ΑΝ Y < 10 ΤΟΤΕ
        ΓΡΑΨΕ 'Summary'
        S ← X + Y
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'Subtract'
        S ← X - Y
    ΤΕΛΟΣ_ΑΝ
        Y ← Y + 1
        X ← 3
ΑΛΛΙΩΣ
        Y ← Y + 5
ΤΕΛΟΣ_ΑΝ
```

**ΛΥΣΗ**

```
ΔΙΑΒΑΣΕ X,Y
ΑΝ X = 5 ΚΑΙ Y < 10 ΤΟΤΕ
    ΓΡΑΨΕ 'Summary'
    S ← X + Y
    Y ← Y + 1
    X ← 3
ΑΛΛΙΩΣ_ΑΝ X=5 ΚΑΙ Y >=10 ΤΟΤΕ
    ΓΡΑΨΕ 'Subtract'
    S ← X – Y
    Y ← Y + 1
    X ← 3
ΑΛΛΙΩΣ
    Y ← Y + 5
ΤΕΛΟΣ_ΑΝ
```

**1ο ΘΕΜΑ**

Να αναφέρετε τα βήματα της μεθόδου πολλαπλασιασμού δύο ακέραιων αριθμών αλά Ρωσικά για δύο φυσικούς αριθμούς Μ1 και Μ2. Στη συνέχεια να κάνετε τη διαίρεση των αριθμών 45 και 37 σύμφωνα με τη μέθοδο αυτή.
Κατόπιν να γράψετε πρόγραμμα που να παίρνει σαν είσοδο τους θετικούς ακέραιους Μ1, Μ2 και εμφανίζει το γινόμενό τους σύμφωνα με τη παραπάνω μέθοδο.

**ΛΥΣΗ**

ΒΗΜΑ 1ο: Αντικαθιστώ τον ακέραιο Μ2 με το ακέραιο πηλίκο της διαίρεσης του με το 2. Η διαδικασία συνεχίζεται μέχρι να φτάσω στη μονάδα.

ΒΗΜΑ 2ο: Αντίστοιχες φορές πολλαπλασιάζω τον ακέραιο Μ1 με το 2.

ΒΗΜΑ 3ο: Όπου ο Μ2 είναι περιττός, προσθέτω τους αντίστοιχους Μ1.

| M1 | M2 | |
|---|---|---|
| 45 | 37 | 45 |
| 90 | 18 | |
| 180 | 9 | 180 |
| 360 | 4 | |
| 720 | 2 | |
| 1440 | 1 | +1440 |
| | | **1665** |

```
ΠΡΟΓΡΑΜΜΑ αλα_ρωσικά
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: Μ1, Μ2, SUM
ΑΡΧΗ
SUM ← 0
ΔΙΑΒΑΣΕ Μ1, Μ2
ΌΣΟ Μ2>0 ΕΠΑΝΑΛΑΒΕ
        ΑΝ Μ2 MOD 2=1 ΤΟΤΕ
                SUM ← SUM + Μ1
        ΤΕΛΟΣ_ΑΝ
        Μ1 ← 2*Μ1
        Μ2 ← Μ2 DIV 2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ SUM
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ αλα_ρωσικά
```

**1ο ΘΕΜΑ**

Να γράψετε συνάρτηση η οποία να υπολογίζει το γινόμενο δύο φυσικών αριθμών Μ1, Μ2 με τη μέθοδο πολ/σμού αλα Ρωσικά. Πως θα γράψετε την αντίστοιχη διαδικασία;

**ΛΥΣΗ**

```
ΣΥΝΑΡΤΗΣΗ ΠΟΛ_ΡΩΣ(Μ1, Μ2): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: Μ1, Μ2, SUM
ΑΡΧΗ
SUM ← 0
ΌΣΟ Μ2>0 ΕΠΑΝΑΛΑΒΕ
        ΑΝ Μ2 MOD 2=1 ΤΟΤΕ
                SUM ← SUM + Μ1
        ΤΕΛΟΣ_ΑΝ
        Μ1 ← 2*Μ1
        Μ2 ← Μ2 DIV 2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΟΛ_ΡΩΣ ← SUM
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΔΙΑΔΙΚΑΣΙΑ ΠΟΛ_ΡΩΣ(Μ1, Μ2, SUM)
        ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: Μ1, Μ2, SUM
ΑΡΧΗ
SUM ← 0
ΌΣΟ Μ2>0 ΕΠΑΝΑΛΑΒΕ
        ΑΝ Μ2 MOD 2=1 ΤΟΤΕ
                SUM ← SUM + Μ1
        ΤΕΛΟΣ_ΑΝ
        Μ1 ← 2*Μ1
        Μ2 ← Μ2 DIV 2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

**1ο ΘΕΜΑ**

Να γράψετε πρόγραμμα το οποίο χρησιμοποιώντας την παραπάνω συνάρτηση να υπολογίζει το γινόμενο οποιοδήποτε αριθμών (είτε αρνητικοί είναι αυτοί, είτε θετικοί, είτε μηδέν)

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ Γενίκευση_Πολλαπλασιασμού
ΜΕΤΑΒΛΗΤΕΣ
   ΑΚΕΡΑΙΕΣ: Α1, Α2, Αποτ
ΑΡΧΗ
  ΓΡΑΨΕ 'Εισάγετε 2 ακέραιους αριθμούς'
  ΔΙΑΒΑΣΕ Α1, Α2
  ΑΝ (Α1 = 0) Ή (Α2 = 0) ΤΟΤΕ
     Αποτ ← 0
  ΑΛΛΙΩΣ_ΑΝ (Α1 > 0 ΚΑΙ Α2 > 0) Ή (Α1 < 0 ΚΑΙ Α2 < 0) ΤΟΤΕ
     Α1 ← Α_Τ (Α1)
     Α2 ← Α_Τ (Α2)
     Αποτ ← ΠΟΛ_ΡΩΣ(Α1, Α2)
  ΑΛΛΙΩΣ
     Α1 ← Α_Τ (Α1)
     Α2 ← Α_Τ (Α2)
     Αποτ ← (-1)*ΠΟΛ_ΡΩΣ(Α1, Α2)
  ΤΕΛΟΣ_ΑΝ
  ΓΡΑΨΕ 'Το αποτέλεσμα είναι: ', Αποτ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Πως θα γράψετε το πρόγραμμα ώστε να χρησιμοποιεί τη διαδικασία;

**2ο ΘΕΜΑ**

Να γίνει ο πίνακας τιμών για όλες τις επαναλήψεις, για το παρακάτω τμήμα αλγορίθμου:

```
Χ ← 16
i ← 12
Όσο (i >= 10) ή (Χ > i) επανάλαβε
        Χ ← Χ/2 + i mod 2
        i ← i div 2
        Υ ← i + Χ div i
        Για j από 12 μέχρι i με βήμα –3
                Υ ← Υ + 1
                Ζ ← Χ - Υ
                Χ ← Ζ div Υ
        Τέλος_Επανάληψης
Τέλος_Επανάληψης
```

**ΛΥΣΗ**

| X | I | Y | Z | J |
|---|---|---|---|---|
| 16 | 12 | - | - | - |
| 8 | 6 | 7 | - | - |
| 0 | 6 | 8 | 0 | 12 |
| -1 | 6 | 9 | -9 | 9 |
| -1 | 6 | 10 | -11 | 6 |

**Λάθη που παρατηρήθηκαν:**

- η εντολή Τέλος_επανάληψης σημαίνει ότι ολοκληρώθηκε ΑΥΤΗ η επανάληψη και πρέπει να ελέγξουμε ΠΑΛΙ τη συνθήκη συνέχειας (ή τέλους) στην κορυφή του βρόχου (ΔΕ σημαίνει ότι τελείωσε η επανάληψη και πρέπει να συνεχίσουμε με την επόμενη εντολή!!).
- Όταν αφορά δομή επανάληψης ΓΙΑ θα πρέπει να γίνει μεταβολή της τιμής της μεταβλητής του ΓΙΑ κατά ... βήμα (αύξηση όταν το βήμα είναι θετικό, μείωση όταν το βήμα είναι αρνητικό)
- ΠΟΛΛΑ λάθη απλών αριθμητικών πράξεων!! ΜΗΝ κάνετε τις πράξεις με το μυαλό σας. Χαρτί και Μολύβι. Ειδικά με το mod και το div. Κάντε τη διαιρεσούλα! Το div είναι το πηλίκο, το mod το υπόλοιπο! Και αν δεν το θυμάστε, στο νου σας ... div από το divide.
- Επίσης ΠΡΟΣΟΧΗ στη προτεραιότητα των πράξεων: 1) ύψωση σε δύναμη, 2) πολ/σμοι και διαιρέσεις (mod, div), 3) προσθέσεις και αφαιρέσεις.

**2ο ΘΕΜΑ**

Να εξηγήσετε τι θα εμφανίσει ο παρακάτω αλγόριθμος:

```
Αλγόριθμος Καλή_επιτυχία
Α ← 0
Β ← 1
Γ ← 2
Για i από 2 μέχρι 9 με_βήμα 3
        Α ← Α + i
        Αν Α ≤ 8 τότε
                Για j από 1 μέχρι Α
                        Β ← Β + 1
                Τέλος_επανάληψης
        αλλιώς
                Γ ← Γ * j
        Τέλος_αν
Τέλος_επανάληψης
Εμφάνισε Α, Β, Γ
Τέλος Καλή_επιτυχία
```

**ΛΥΣΗ**

| i | A | B | Γ | J |
|---|---|---|---|---|
| - | 0 | 1 | 2 | - |
| 2 | 2 | 1 | 2 | - |
| 2 | 2 | 2 | 2 | 1 |
| 2 | 2 | 3 | 2 | 2 |

| | | | | |
|---|---|---|---|---|
| 5 | 7 | 3 | 2 | 2 |
| 5 | 7 | 4 | 2 | 1 |
| 5 | 7 | 5 | 2 | 2 |
| 5 | 7 | 6 | 2 | 3 |
| 5 | 7 | 7 | 2 | 4 |
| 5 | 7 | 8 | 2 | 5 |
| 5 | 7 | 9 | 2 | 6 |
| 5 | 7 | 10 | 2 | 7 |
| 8 | 15 | 10 | 14 | 7 |

Εμφανίζει τις τιμές: 15, 10 και 14

**Προσοχή:** Δεν υπάρχει αντικειμενικά τρόπος να κάνει κανείς λάθος σε αυτό το θέμα! Όμως λάθη έχουν γίνει! Και είναι κρίμα να χάνονται **εύκολες μονάδες**!
Όσα λάθη έχουν γίνει σε αυτό το θέμα οφείλονται μόνο σε απροσεξία, βιασύνη, άγχος.
Εδώ καλείστε να παίξετε το ρόλο του Η/Υ. Ο Η/Υ δεν έχει τίποτα από αυτά. Επιπλέον έχει το πλεονέκτημα να ... μην είναι έξυπνος! Δεν σκέφτεται, δεν υποθέτει, δεν προτρέχει. Και γι αυτό το λόγο δεν κάνει ποτέ του λάθος.
Ο Η/Υ κάνει ακριβώς ότι ζητάει η εντολή. 'Βλέπει' μόνο την εντολή που έχει μπροστά του και ... τις τιμές των μεταβλητών που έχει στη μνήμη. Κάντε ΤΟ ΙΔΙΟ!!!
Δουλέψτε το θέμα πειθαρχημένα και μεθοδικά και δεν υπάρχει περιθώριο λάθους.

- Ακολουθήστε βήμα-προς-βήμα τις εντολές που σας δίνονται. Με το αριστερό χέρι να δείχνει **την εντολή που εκτελείται,** ξεχάστε όλες τις υπόλοιπες και με το δεξί 'δουλέψτε' **τον πίνακα τιμών** στη 'μνήμη'.
- **ΜΗΝ** κοιτάτε την επόμενη εντολή, ούτε την προηγούμενη.
- **ΜΗΝ** υποθέτετε ΤΙ θα γίνει, κάντε ακριβώς ότι λέει η εντολή.
- Δουλέψτε το θέμα στο πρόχειρο κάνοντας τον πίνακα τιμών και σημειώστε τι ... **τυπώνεται** ... **μόλις** συναντήσετε την εντολή εκτύπωσε ή εμφάνισε με βάση τις τιμές που έχει ο πίνακας τιμών εκείνη τη στιγμή.
- Απαντήστε στο θέμα αυτό μονορούφι! Μη διακόψετε και προσπαθήσετε να το συνεχίσετε αργότερα. Υπάρχει ο κίνδυνος να ξεκινήσετε από λάθος σημείο. Αν συμβεί, αν δεν είστε απόλυτα σίγουροι σε ποιο σημείο βρίσκεστε ξεκινήστε ΞΑΝΑ το θέμα από την αρχή.
- Αντί για επαλήθευση (έλεγχο), αν έχετε χρόνο ΞΑΝΑ λύστε το θέμα απ' αρχής. ΜΗΝ αρκεστείτε στο να το ελέγξετε.

## 2ο ΘΕΜΑ

Δίνεται το παρακάτω πρόγραμμα:

```
ΠΡΟΓΡΑΜΜΑ Σύγκλιση
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: Α, Β, βήμα
ΑΡΧΗ
  Α ← 8
  Β ← - 4
  βήμα ← 2
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΚΑΛΕΣΕ Σύγκλιση_με_βήμα (Α, Β, βήμα)
        Α ← 2*Α – 5
        Β ← Β + 1
        ΓΡΑΨΕ Α, Β
  ΜΕΧΡΙΣ_ΟΤΟΥ Α <= Β
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Σύγκλιση

ΔΙΑΔΙΚΑΣΙΑ Σύγκλιση_με_βήμα (Χ, Υ, Ζ)
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: Χ, Υ, Ζ
ΑΡΧΗ
  Χ ← Χ – Ζ
  Υ ← Υ + Ζ
  ΓΡΑΨΕ Χ, Υ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Να γράψετε τι θα τυπώσει το παραπάνω πρόγραμμα.

**ΛΥΣΗ**

| | **Κυρίως Πρόγραμμα** | | | **Υποπρόγραμμα** | | |
|---|---|---|---|---|---|---|
| | **Α** | **Β** | **βήμα** | **Χ** | **Υ** | **Ζ** |
| | 8 | - 4 | 2 | | | |
| 1η κλήση διαδικασίας : | | | | 8 | - 4 | 2 |
| 1η εκτέλεση διαδικασίας : | | | | **6** | **- 2** | 2 |
| Επιστροφή στο κυρίως πρόγραμμα : | 6 | - 2 | 2 | | | |
| | **7** | **- 1** | 2 | | | |
| 2η κλήση διαδικασίας : | | | | 7 | - 1 | 2 |
| 2η εκτέλεση διαδικασίας : | | | | **5** | **1** | 2 |
| Επιστροφή στο κυρίως πρόγραμμα : | 5 | 1 | 2 | | | |
| | **5** | **2** | 2 | | | |
| 3η κλήση διαδικασίας : | | | | 5 | 2 | 2 |

## 2ο ΘΕΜΑ

Τι θα εκτυπώσει το παρακάτω πρόγραμμα:

```
ΠΡΟΓΡΑΜΜΑ Ασκ_1
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: α,β,γ
  ΠΡΑΓΜΑΤΙΚΕΣ: δ
ΑΡΧΗ
  α ← 5
  β ← 1
  α ← συναρ(α,β)
  ΓΡΑΨΕ α,β
  ΚΑΛΕΣΕ διαδ(α,β,γ,δ)
  ΓΡΑΨΕ γ,δ
  β ← συναρ(β,α)
  ΓΡΑΨΕ α,β
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ διαδ (x,y,z,p)
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: x,y,z
  ΠΡΑΓΜΑΤΙΚΕΣ: p
ΑΡΧΗ
  z ← Α_Τ(x * y)
  p ← Τ_Ρ(z+8)*2
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΣΥΝΑΡΤΗΣΗ συναρ (x,y):ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: x,y
ΑΡΧΗ
  ΌΣΟ x > y ΕΠΑΝΑΛΑΒΕ
      x ← x – 3
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  συναρ ← x
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

## 2ο ΘΕΜΑ

Α) Δίνονται δύο υποπρογράμματα τα οποία θα πρέπει να χρησιμοποιήσετε.

```
ΔΙΑΔΙΚΑΣΙΑ Τ(Α, Ν)
ΜΕΤΑΒΛΗΤΕΣ
      ΠΡΑΓΜΑΤΙΚΕΣ: Α[50], ΤΕΜΡ
      ΑΚΕΡΑΙΕΣ: Ι, Κ, Ν
ΑΡΧΗ
  ΓΙΑ Κ ΑΠΟ 2 ΜΕΧΡΙ Ν
      ΓΙΑ Ι ΑΠΟ Ν ΜΕΧΡΙ Κ ΜΕ_ΒΗΜΑ -1
          ΑΝ Α[Ι]<Α[Ι-1] ΤΟΤΕ
              ΤΕΜΡ ← Α[Ι]
              Α[Ι] ← Α[Ι-1]
              Α[Ι-1] ← ΤΕΜΡ
          ΤΕΛΟΣ_ΑΝ
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΣΥΝΑΡΤΗΣΗ Πλήθος(Ν, Α, Χ): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
      ΠΡΑΓΜΑΤΙΚΕΣ: Α[50], Χ
      ΑΚΕΡΑΙΕΣ: Ι, Ν, Σ
ΑΡΧΗ
  Σ ← 0
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
      ΑΝ Α[Ι] = Χ ΤΟΤΕ
          Σ ← Σ+1
```

```
      ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  Πλήθος ← Σ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

1) Περιγράψτε τη λειτουργία που εκτελούν τα δύο υποπρογράμματα.

2) Υλοποιήσετε ένα πρόγραμμα το οποίο :

α) Διαβάζει τους βαθμούς που πήραν οι μαθητές δυο τμημάτων Γ΄ τάξης, τους οποίους καταχωρεί σε δύο πίνακες 50 θέσεων ο καθένας. Το ένα τμήμα έχει 15 μαθητές και το άλλο 22. Πρέπει να γίνεται έλεγχος έτσι ώστε οι βαθμοί που διαβάζονται να είναι μεγαλύτεροι του 0 και μικρότεροι ή ίσοι του 20.

β) Ταξινομεί τους πίνακες κατά αύξουσα σειρά.

γ) Διαβάζει από τον χρήστη βαθμούς και θα εμφανίζει πόσες φορές εμφανίζονται οι βαθμοί αυτοί στο πρώτο τμήμα και πόσες στο δεύτερο. Το πρόγραμμα θα σταματάει να διαβάζει βαθμούς όταν ο χρήστης δώσει βαθμό -1.

**ΛΥΣΗ**

1) Η διαδικασία Τ δέχεται έναν πίνακα πραγματικών αριθμών το πολύ 50 θέσεων καθώς και το πλήθος των θέσεων του πίνακα που χρησιμοποιούνται (το πλήθος των αριθμών που επεξεργαζόμαστε), τον ταξινομεί με αύξουσα διάταξη και τον επιστρέφει ταξινομημένο στο τμήμα προγράμματος που την κάλεσε.
Η συνάρτηση Πλήθος δέχεται έναν πίνακα πραγματικών αριθμών το πολύ 50 θέσεων καθώς το πλήθος των θέσεων του πίνακα που χρησιμοποιούνται (με αντίθετη σειρά στις παραμέτρους), καθώς και έναν πραγματικό αριθμό Χ και μετράει πόσες φορές υπάρχει η τιμή Χ στον πίνακα. Το αποτέλεσμα της καταμέτρησης επιστρέφεται στο τμήμα προγράμματος που καλεί τη συνάρτηση.

```
2)  ΠΡΟΓΡΑΜΜΑ άσκηση
        ΜΕΤΑΒΛΗΤΕΣ
        ΠΡΑΓΜΑΤΙΚΕΣ: Γ1[50], Γ2[50], βαθ
        ΑΚΕΡΑΙΕΣ: Ι, π1, π2, φ1, φ2
    ΑΡΧΗ
      π1 ← 15
      ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ π1
          ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
              ΔΙΑΒΑΣΕ Γ1[Ι]
          ΜΕΧΡΙΣ_ΟΤΟΥ Γ1[Ι]>0 ΚΑΙ Γ1[Ι]<=20
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
      π2 ← 22
      ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ π2
          ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
              ΔΙΑΒΑΣΕ Γ2[Ι]
          ΜΕΧΡΙΣ_ΟΤΟΥ Γ2[Ι]>0 ΚΑΙ Γ2[Ι]<=20
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
      ΚΑΛΕΣΕ Τ(Γ1, π1)
      ΚΑΛΕΣΕ Τ(Γ2, π2)
      ΔΙΑΒΑΣΕ βαθ
      ΌΣΟ βαθ <> -1 ΕΠΑΝΑΛΑΒΕ
          φ1 ← Πλήθος(π1, Γ1, βαθ)
          φ2 ← Πλήθος(π2, Γ2, βαθ)
          ΓΡΑΨΕ φ1, φ2
          ΔΙΑΒΑΣΕ βαθ
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

**2ο ΘΕΜΑ**

Δίνεται το παρακάτω πρόγραμμα και υποπρόγραμμα:

```
ΠΡΟΓΡΑΜΜΑ Άσκηση
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: κ, λ
ΑΡΧΗ
  ΔΙΑΒΑΣΕ κ, λ
  ΑΝ Αξιολόγηση(κ, λ) >= κ ΤΟΤΕ
      ΓΡΑΨΕ κ
  ΑΛΛΙΩΣ
      ΓΡΑΨΕ λ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Άσκηση

ΣΥΝΑΡΤΗΣΗ Αξιολόγηση(α, β): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: α, β, ω
ΑΡΧΗ
  α ← α + 2
  β ← β – 3
  ω ← α * β – 2
  Αξιολόγηση ← ω ^ 2 – (α + β)
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

1. Να γράψετε διαδικασία με όνομα Αξιολόγ_διαδ που να υλοποιεί τις ίδιες λειτουργίες με τη συνάρτηση Αξιολόγηση.
2. Να παρουσίασετε τη νέα μορφή του προγράμματος ώστε να επιτελεί τις ίδιες λειτουργίες με τη βοήθεια της διαδικασίας Αξιολόγ_διαδ.
3. Να μετατρέψετε το πρόγραμμα Άσκηση σε ψευδογλώσσα.

**ΛΥΣΗ**

```
1.  ΔΙΑΔΙΚΑΣΙΑ Αξιολογ_Διαδ (α, β, αξ)
    ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: α, β, αξ, τα, τβ, ω
    ΑΡΧΗ
      τα ← α + 2
      τβ ← β – 3
      ω ← τα * τβ – 2
      αξ ← ω ^ 2 – (τα + τβ)
    ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

2. ΠΡΟΓΡΑΜΜΑ Άσκηση_ΙΙ
    ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: κ, λ, θ
    ΑΡΧΗ
     ΔΙΑΒΑΣΕ κ, λ
     ΚΑΛΕΣΕ Αξιολογ_Διαδ (κ, λ, θ)
     ΑΝ θ >= κ ΤΟΤΕ
         ΓΡΑΨΕ κ
     ΑΛΛΙΩΣ
         ΓΡΑΨΕ λ
     ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Άσκηση_ΙΙ

3. Αλγόριθμος Άσκηση_ΙΙ
    Διάβασε κ, λ
    τκ ← κ + 2
    τλ ← λ – 3
    ω ← τκ * τλ – 2
    θ ← ω ^ 2 – (τκ + τλ)
    Αν θ >= κ τότε
        Εμφάνισε κ
    Αλλιώς
        Εμφάνισε λ
    Τέλος_αν
  Τέλος Άσκηση_ΙΙ
```

**2ο ΘΕΜΑ**

Να σχηματίσετε τους πίνακες τιμών για τα παρακάτω:

**(i)**
```
ΠΡΟΓΡΑΜΜΑ Σύγκλιση
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Α, Β, βήμα
ΑΡΧΗ
  Α ← 8
  Β ← - 4
  βήμα ← 2
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
     ΚΑΛΕΣΕ Σύγκλιση_με_βήμα (Α, Β, βήμα)
     Α ← 2*Α – 5
     Β ← Β + 1
     ΓΡΑΨΕ Α, Β
  ΜΕΧΡΙΣ_ΟΤΟΥ Α <= Β
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Σύγκλιση

ΔΙΑΔΙΚΑΣΙΑ Σύγκλιση_με_βήμα (Χ, Υ, Ζ)
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Χ, Υ, Ζ
ΑΡΧΗ
  Χ ← Χ – Ζ
  Υ ← Υ + Ζ
  ΓΡΑΨΕ Χ, Υ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

**(ii)**
```
ΠΡΟΓΡΑΜΜΑ Πίνακας_Τιμών2
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Α, Β, Γ
ΑΡΧΗ
    Α ← 3
```

```
        Β ← 13
        Γ ← 2
        ΓΡΑΨΕ Α, Β, Γ
        ΚΑΛΕΣΕ Επεξεργασία_Τιμών2 (Β, Γ)
        ΓΡΑΨΕ Α, Β, Γ
        ΚΑΛΕΣΕ Επεξεργασία_Τιμών2 (Γ, Α)
        ΓΡΑΨΕ Α, Β, Γ
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Πίνακας_Τιμών2

    ΔΙΑΔΙΚΑΣΙΑ Επεξεργασία_Τιμών2 (αρ1, αρ2)
    ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: αρ1, αρ2
    ΑΡΧΗ
        αρ1 ← αρ1 DIV 2
        αρ2 ← αρ2 ^ 3
    ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

**(iii)**

```
    ΠΡΟΓΡΑΜΜΑ Πίνακας_Τιμών3
    ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: Α, Β
    ΑΡΧΗ
      Α ← 1
      Β ← 16
      ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
            Γ ← Επεξεργασία (Α, Β)
            Α ← Α – Γ
            Β ← Β + Γ
      ΓΡΑΨΕ Α, Β
      ΜΕΧΡΙΣ_ΟΤΟΥ Γ < 3
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Πίνακας_Τιμών3

    ΣΥΝΑΡΤΗΣΗ Επεξεργασία (α1, α2) : ΑΚΕΡΑΙΑ
    ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: α1, α2, εκτίμηση
    ΑΡΧΗ
      α1 ← Α_Τ (α1)
      α2 ← Α_Τ (α2)
      εκτίμηση ← α2 DIV α1
      Επεξεργασία ← εκτίμηση
    ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

**2ο ΘΕΜΑ**

Ο αλγόριθμος στην αριστερή στήλη έχει γραφτεί με μη δομημένο προγραμματισμό. Να γράψετε τον αλγόριθμο αυτό με τις αρχές του δομημένου προγραμματισμού.

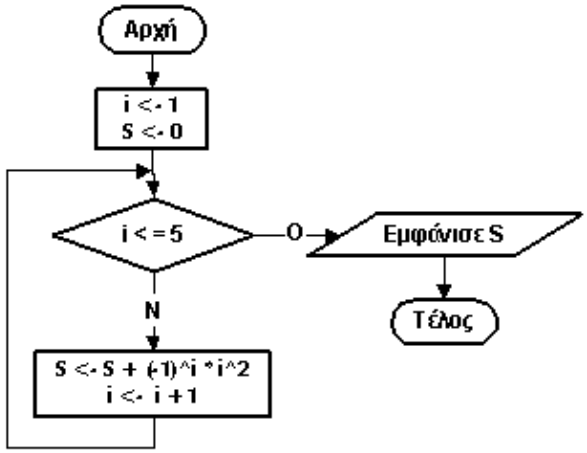
**ΛΥΣΗ**

| Αλγόριθμος Αδόμητο | Αλγόριθμος Δομημένο |
|---|---|
| Αν <συνθήκη1> τότε | Όσο <συνθήκη1> επανάλαβε |
| Εντολή1 | Εντολή1 |
| Αν <συνθήκη2> τότε | Αν <συνθήκη2> τότε |
| Εντολή2 | Εντολή2 |
| Εντολή3 | Εντολή3 |
| Πήγαινε στην εντολή 5 | Αλλιώς |
| Τέλος_αν | Εντολή4 |
| Εντολή4 | Τέλος_αν |
| Εντολή5 | Εντολή5 |
| Πήγαινε στην αρχή | Τέλος_επανάληψης |
| Τέλος_αν | Τέλος Δομημένο |
| Τέλος Αδόμητο | |

Παρατήρηση: Αν σας ζητηθεί να μετατρέψετε από αδόμητο σε δομημένο καλό είναι πρώτα να κάνετε το **διάγραμμα ροής** που αντιστοιχεί σ' αυτό και ύστερα να γράψετε το δομημένο.
Εδώ με την εντολή Πήγαινε η ροή του αλγορίθμου πηγαίνει στην Εντολή5 και στη συνέχεια στην αρχή του αλγορίθμου.
Για να επιτευχθεί το ίδιο στο δομημένο προγραμματισμό πρέπει να ισχύουν τα ακόλουθα: Αν αληθεύει η <συνθήκη2> εκτελούνται οι Εντολές2 και 3, ολοκληρώνεται το Αν, εκτελείται η Εντολή5 και επαναλαμβάνεται ο βρόχος. Η εκτέλεση της Εντολής4 γίνεται μόνο στην περίπτωση που η <συνθήκη2> δεν είναι αληθής και αυτό επιτυγχάνεται με τη χρήση του αλλιώς.

**2ο ΘΕΜΑ**

Να αναπαραστήσετε τον αλγόριθμο που αντιστοιχεί στο παρακάτω διάγραμμα ροής και να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τί θα εκτυπωθεί τελικά;




**ΛΥΣΗ**

```
Αλγόριθμος Διάγραμμα_Ροής
    i ← 1
    S ← 0
    Όσο (i <= 5) επανάλαβε
          S ← S + (-1) ^ i *  i ^2
          i ← i + 1
    Τέλος_Επανάληψης
    Εμφάνισε S
 Τέλος Διάγραμμα_Ροής
```

Ή εναλακτικά μπορώ να κάνω τον παρακάτω τρόπο (Η Όσο ... επανάλαβε έχει το ίδιο διάγραμμα ροής με την Για .. από ... μέχρι)

```
 Αλγόριθμος Διάγραμμα_Εναλ
    S ← 0
    Για i από 1 μέχρι 5
          S ← S + (-1) ^ i * i ^ 2
    Τέλος_Επανάληψης
    Εμφάνισε S
 Τέλος Διάγραμμα_Εναλ
```

Η άσκηση θα εκτυπώσει το αποτέλεσμα S=-15. Η δομή επανάληψης θα εκτελεστεί για 5 επαναλήψεις

| Επανάληψη: | | 1η | 2η | 3η | 4η | 5η |
|---|---|---|---|---|---|---|
| S : | 0 | -1 | 3 | -6 | 10 | -15 |
| i : | 1 | 2 | 3 | 4 | 5 | 6 |

**2ο ΘΕΜΑ**

Να αναπτυχθεί υποπρόγραμμα που δέχεται το βασικό μισθό ενός υπαλλήλου και τον αριθμό των παιδιών του και επιστρέφει το επίδομα που αυτός δικαιούται για τα παιδιά του. Το επίδομα υπολογίζεται ως εξής: Για τα δύο πρώτα παιδιά είναι 3% για το κάθε παιδί επί του μισθού. Για το τρίτο παιδί είναι 6% επί του μισθού. Για κάθε επιπλέον παιδί είναι 10% επί του μισθού. Με ποιο τρόπο θα κληθεί το παραπάνω υποπρόγραμμα μέσα από το κυρίως πρόγραμμα.

**ΛΥΣΗ**

```
ΣΥΝΑΡΤΗΣΗ ΕΠΙΔΟΜΑ(Μ, Π):ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
      ΑΚΕΡΑΙΕΣ:Π
      ΠΡΑΓΜΑΤΙΚΕΣ:Μ
ΑΡΧΗ
  ΑΝ Π <= 2 ΤΟΤΕ
        ΕΠΙΔΟΜΑ ← Π * Μ * 0.03
  ΑΛΛΙΩΣ_ΑΝ Π = 3 ΤΟΤΕ
        ΕΠΙΔΟΜΑ ← 2 * Μ * 0.03 + Μ * 0.06
  ΑΛΛΙΩΣ
        ΕΠΙΔΟΜΑ ← 2 * Μ * 0.03 + Μ * 0.06 + (Π-3) * Μ * 0.1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

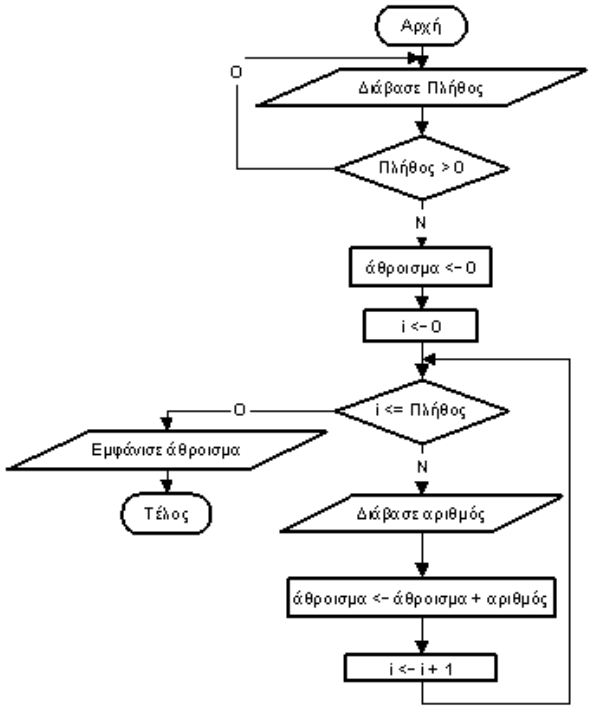
**Κλήση υποπρογράμματος από το κυρίως πρόγραμμα**

```
επ ← ΕΠΙΔΟΜΑ(μισθός,παιδιά)
```

Πως γράφεται το συγκεκριμένο με διαδικασία;

## 2ο ΘΕΜΑ

Να μεταφέρετε το παρακάτω διάγραμμα ροής σε ψευδοκώδικα:




## 2ο ΘΕΜΑ

Δίνεται μονοδιάστατος πίνακας Α, 10 θέσεων, ο οποίος στις θέσεις 1 έως 10 περιέχει αντίστοιχα τους αριθμούς: 15, 3, 0, 5, 16, 2, 17, 8, 19, 1 και τμήμα αλγορίθμου:

```
Για i από 1 μέχρι 9 με_βήμα 2
    k ← ((i + 10) mod 10) + 1
    A[i] ← A[k]
    Εκτύπωσε i, k, A[i], A[k]
Τέλος_επανάληψης
```

Ποιες τιμές τυπώνονται με την εντολή

Εκτύπωσε i, k, A[i], A[k]

καθώς εκτελείται το παραπάνω τμήμα αλγορίθμου;

**ΛΥΣΗ**

| | i | k | Πίνακας Α | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 15 | 3 | 0 | 5 | 16 | 2 | 17 | 8 | 19 | 1 |
| 1η επαν. | 1 | 2 | 3 | 3 | 0 | 5 | 16 | 2 | 17 | 8 | 19 | 1 |
| 2η επαν. | 3 | 4 | 3 | 3 | 5 | 5 | 16 | 2 | 17 | 8 | 19 | 1 |
| 3η επαν. | 5 | 6 | 3 | 3 | 5 | 5 | 2 | 2 | 17 | 8 | 19 | 1 |
| 4η επαν. | 7 | 8 | 3 | 3 | 5 | 5 | 2 | 2 | 8 | 8 | 19 | 1 |
| 5η επαν. | 9 | 10 | 3 | 3 | 5 | 5 | 2 | 2 | 8 | 8 | 1 | 1 |

Θα εκτυπωθούν διαδοχικά οι τιμές:

1, 2, 3, 3
3, 4, 5, 5
5, 6, 2, 2
7, 8, 8, 8
9, 10, 1, 1

## 2ο ΘΕΜΑ

Δίδεται πίνακας Α:

| 3 | 8 | 11 | 18 | 22 | 44 | 57 |
|---|---|---|---|---|---|---|

Ο οποίος είναι ταξινομημένος με αύξουσα διάταξη.

Α) Αν αναζητούμε στον πίνακα αυτό π.χ. τον αριθμό 16, ποιο είναι το ελάχιστο πλήθος επαναλήψεων που απαιτείται ώστε να απαντήσουμε ότι δεν υπάρχει στον πίνακα. Περιγράψτε τη σκέψη σας.

Β) Να παρουσιάσετε τον αλγόριθμο σειριακής αναζήτησης τροποποιημένο, ώστε να λειτουργεί βέλτιστα σε ταξινομημένο πίνακα αποφεύγοντας τις περιττές επαναλήψεις, κατά την αναζήτηση κάποιου στοιχείου.

**ΛΥΣΗ**

Α) Αρκούν 4 επαναλήψεις γιατί δεν πρόκειται ο αριθμός 16 να βρίσκεται μετά τον 18 (αύξουσα ταξινόμηση). Γενικά, όταν συναντήσω κάποιο στοιχείο μεγαλύτερο από το αναζητούμενο σταματάω την αναζήτηση.

Β)

```
Αλγόριθμος Σειριακή_Αναζήτηση_Ταξινομημένος
Δεδομένα // N, table, key //
done ← ψευδής
position ← 0
i ← 1
Όσο (done = ψευδής) και (i <= N) επανάλαβε
    Αν (table[i] = key) τότε
        done ← αληθής
        position ← i
    Αλλιώς_αν (table[i] > key) τότε
        done ← αληθής
    Αλλιώς
        i ← i + 1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // done, position //
Τέλος Σειριακή_Αναζήτηση_Ταξινομημένος
```

## 2ο ΘΕΜΑ

Να γραφεί αλγόριθμος που να εφαρμόζει τη σειριακή αναζήτηση σε έναν δισδιάστατο πίνακα μ γραμμών και ν στηλών και να εμφανίζει τη θέση μιας τιμής που δίνεται από τον χρήστη. Θεωρείστε ότι όλα τα στοιχεία του πίνακα είναι διαφορετικά. Η διαδικασία αναζήτησης θα σταματά μόλις γίνει ο εντοπισμός του στοιχείου που δόθηκε από τον χρήστη.

**ΛΥΣΗ**

```
Αλγόριθμος  Σειριακή_δισδιάστατος_1
Δεδομένα // A, ν, μ, key //
Done ← Ψευδής
γραμμή ← 0
στήλη ← 0
i ←  1
Όσο (i<=μ) και (done=Ψευδής) επανάλαβε
    j ← 1
    Όσο (j<=ν) και (done=Ψευδής) επανάλαβε
        Αν A[i,j]=key τότε
            Done ← Αληθής
            γραμμή ← i
            στήλη ← j
        Αλλιώς
            j ← j+1
        Τέλος_αν
    Τέλος_επανάληψης
    i ← i+1
Τέλος_επανάληψης
Αν  done=Ψευδής  τότε
    Εμφάνισε  "Δεν βρέθηκε η τιμή", key
αλλιώς
    Εμφάνισε  "βρέθηκε στη γραμμή", γραμμή
    Εμφάνισε  "βρέθηκε στη στήλη", στήλη
Τέλος_αν
Τέλος  Σειριακή_δισδιάστατος_1
```

## 3ο ΘΕΜΑ

Ο υπολογισμός των ταχυδρομικών τελών για αποστολή μικροδέματος στο εξωτερικό με χρονοεπίδοση πραγματοποιείται με τον εξής τρόπο:

Πάγιο τέλος 0,4 € (επιβάρυνση όλων των μικροδεμάτων ανεξαρτήτως λοιπών τελών)
Επιπλέον χρέωση (ανεξάρτητα χώρας προορισμού):

- ✓ Για δέματα από 1 έως και 20 γραμμάρια 0,05 € ανά γραμμάριο
- ✓ Για δέματα από 20 έως και 100 γραμμάρια χρέωση 0,07 € ανά γραμμάριο για κάθε γραμμάριο επιπλέον των 20
- ✓ Για δέματα άνω 100 γραμμαρίων χρέωση 0,1 € ανά γραμμάριο για κάθε γραμμάριο επιπλέον των 100

Τα μικροδέματα εξωτερικού επιβαρύνονται με προσαύξηση 12% αν αποστέλλονται προς χώρα της ΕΕ (Ευρωπαϊκής `Ενωσης) ή με 15% προς τις λοιπές χώρες. Η επιβάρυνση για κάθε μικροδέμα υπολογίζεται σύμφωνα με τον τύπο:
Επιβάρυνση = (πάγιο + επιπλέον χρέωση) * προσαύξηση
Να γραφεί αλγόριθμος ο οποίος:
(Α) να παίρνει σαν είσοδο το βάρος μικροδέματος σε γραμμάρια και τη χώρα αποστολής ως εξής: 0 για αποστολή προς χώρα της ΕΕ, 1 για αποστολή προς χώρα εκτός της ΕΕ.
(Β) να εμφανίζει το βάρος του μικροδέματος, την επιπλέον χρέωση και την επιβάρυνση του μικροδέματος ανάλογα με τη χώρα προορισμού.

**ΛΥΣΗ**

```
Αλγόριθμος Ταχυδρομικά_τέλη
Εμφάνισε "Δώσε το βάρος του μικροδέματος σε γραμμάρια"
Διάβασε Β
Εμφάνισε "Δώσε 0 για χώρα της ΕΕ ή 1 για χώρα εκτός ΕΕ"
Διάβασε Χ
Αν Β <= 20 τότε
        ΕΧ ← Β * 0.05
αλλιώς_αν Β <= 100 τότε
        ΕΧ ← 20 * 0.05 + (Β – 20) * 0.07
αλλιώς
        ΕΧ ← 20 * 0.05 + 80 * 0.07 + (Β – 100) * 0.1
Τέλος_αν
Αν Χ = 0 τότε
        ΕΠ ← (0.4 + ΕΧ) * 12 / 100
αλλιώς_αν Χ = 1 τότε
        ΕΠ ← (0.4 + ΕΧ) * 15 / 100
Τέλος_αν
Εμφάνισε "Βάρος μικροδέματος = ", Β,"γραμμάρια"
Εμφάνισε "Επιπλέον χρέωση = ", ΕΧ, "Ευρώ"
Εμφάνισε "Επιβάρυνση = ", ΕΠ,"Ευρώ"
Τέλος Ταχυδρομικά_τέλη
```

**3ο ΘΕΜΑ**

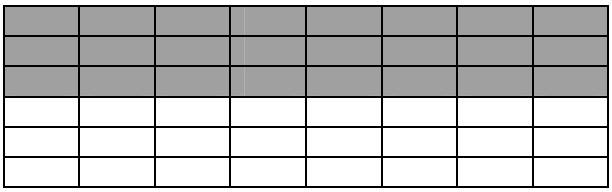
Μια σχολή οδηγών προσφέρει στους πελάτες της τα ακόλουθα: Οι δύο πρώτες ώρες μαθήματος είναι δωρεάν, ενώ από τις τρεις μέχρι και τις είκοσι ώρες κοστίζει € 25 η ώρα. Αν όμως ο πελάτης κάνει πάνω από είκοσι ώρες, τότε για κάθε επιπλέον ώρα άνω των είκοσι πληρώνει € 20. Τέλος, αν το συνολικό κόστος για τον πελάτη υπερβαίνει τα € 800, τότε η σχολή κάνει 3% έκπτωση στον πελάτη.
Να γράψετε αλγόριθμο ο οποίος:
(Α) Θα διαβάζει τις ώρες των μαθημάτων ενός πελάτη.
(Β) Θα υπολογίζει το συνολικό κόστος των μαθημάτων.
(Γ) Θα εκτυπώνει το μήνυμα ”ΚΟΣΤΟΣ ΜΑΘΗΜΑΤΩΝ” και το συνολικό κόστος.

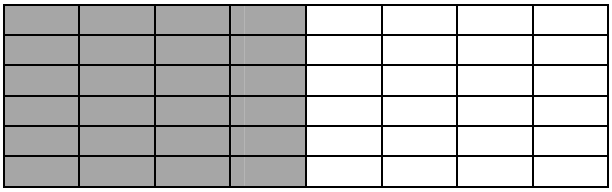
**ΛΥΣΗ**

```
Αλγόριθμος Σχολή_οδηγών
Αρχή_επανάληψης
        Διάβασε ΩΡΕΣ
Μέχρις_ότου ΩΡΕΣ>=0
Αν ΩΡΕΣ<=2 τότε
        ΚΟΣΤΟΣ ← 0
αλλιώς_αν ΩΡΕΣ<=20 τότε
        ΚΟΣΤΟΣ ← (ΩΡΕΣ–18)*25
αλλιώς
        ΚΟΣΤΟΣ ← 18*25 + (ΩΡΕΣ-20)*20
Τέλος_αν
Αν ΚΟΣΤΟΣ>800 τότε
        ΚΟΣΤΟΣ ← ΚΟΣΤΟΣ*0.97
Τέλος_αν
Εκτύπωσε "ΚΟΣΤΟΣ ΜΑΘΗΜΑΤΩΝ ", ΚΟΣΤΟΣ
Τέλος Σχολή_οδηγών
```

**3ο ΘΕΜΑ**

Να γραφεί πρόγραμμα που διαχωρίζει 1 πίνακα διαστάσεων 6x8 α) σε 2 πίνακες 3x8, και β) σε 2 πίνακες 6x4





**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ ΔΙΑΧΩΡΙΣΜΟΣ
ΜΕΤΑΒΛΗΤΕΣ
        ΠΡΑΓΜΑΤΙΚΕΣ: Α[6,8], Β[3,8], Γ[3,8], Δ[6,4], Ο[6,4]
        ΑΚΕΡΑΙΕΣ:Κ, Λ
ΑΡΧΗ
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 3
        ΓΙΑ Λ ΑΠΟ 1 ΜΕΧΡΙ 8
                Β[Κ,Λ] ← Α[Κ,Λ]
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 3
        ΓΙΑ Λ ΑΠΟ 1 ΜΕΧΡΙ 8
                Γ[Κ,Λ] ← Α[Κ+3,Λ]
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 6
        ΓΙΑ Λ ΑΠΟ 1 ΜΕΧΡΙ 4
                Δ[Κ,Λ] ← Α[Κ,Λ]
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 6
        ΓΙΑ Λ ΑΠΟ 1 ΜΕΧΡΙ 4
                Ο[Κ,Λ] ← Α[Κ,Λ+4]
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΔΙΑΧΩΡΙΣΜΟΣ
```

**3ο ΘΕΜΑ**

Ένα συγκρότημα κινηματογραφικών αιθουσών διαθέτει 8 αίθουσες, και έχουν καταχωρηθεί οι εισπράξεις κάθε αίθουσας για κάθε μέρα μιας εβδομάδας. Να αναπτύξετε πρόγραμμα το οποίο να υπολογίζει και να εμφανίζει:
Α) την ημέρα με τις λιγότερες εισπράξεις για κάθε αίθουσα
Β) την ημέρα που σημειώθηκαν οι περισσότερες συνολικές εισπράξεις (π.χ. την 3η ημέρα της εβδομάδας)

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ κινηματογράφοι
ΜΕΤΑΒΛΗΤΕΣ
        ΠΡΑΓΜΑΤΙΚΕΣ: Α[8,7], MIN, MAX, SUM[7]
        ΑΚΕΡΑΙΕΣ: i, j, ΜΕΡΑ_MIN, ΜΕΡΑ_MAX
ΑΡΧΗ
!Οι εισπράξεις είναι προφανώς καταχωρημένες σε ένα πίνακα Α
!8x7, όπου οι γραμμές αναπαριστούν τις αίθουσες, και οι στήλες
!τις ημέρες
!Για την εύρεση της ημέρας με τις λιγότερες εισπράξεις για κάθε
!αίθουσα, πρέπει να βρούμε την ελάχιστη είσπραξη κάθε
!αίθουσας, καθώς και την ημέρα στην οποία αυτή αντιστοιχεί.
!Δηλαδή πρέπει να βρούμε το MIN ανά γραμμή, καθώς και τη
!στήλη (ΜΕΡΑ_MIN) στην οποία αυτό αντιστοιχεί.
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8
    MIN ← Α[i,1]
    ΜΕΡΑ_MIN ← 1
    ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 7
        ΑΝ Α[i,j]<MIN ΤΟΤΕ
                MIN ← Α[i,j]
                ΜΕΡΑ_MIN ← j
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ ΜΕΡΑ_MIN
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
!Για την εύρεση της ημέρας με τις περισσότερες εισπράξεις,
!πρέπει αρχικά να βρεθούν οι συνολικές εισπράξεις ανά ημέρα,
!δηλαδή το άθροισμα των εισπράξεων (τιμών του πίνακα Α) ανά
```

```
!στήλη. Τα αθροίσματα αυτά θα αποθηκευτούν σε μονοδιάστατο
!πίνακα SUM, με τόσα στοιχεία όσες είναι και οι ημέρες της
!εβδομάδας, δηλαδή 7.
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 7
    SUM[j] ← 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8
        SUM[j] ← SUM[j]+A[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
!Μετά την αποθήκευση των συνολικών εισπράξεων ανά ημέρα σε
!πίνακα, απομένει να βρεθεί η ημέρα με τη μεγαλύτερη συνολική
!είσπραξη, δηλαδή να βρεθεί το MAX του πίνακα SUM, καθώς και
!η θέση (ΜΕΡΑ_ΜΑΧ) στην οποία αντιστοιχεί αυτό το MAX. Έτσι,
!κάθε φορά που –με το γνωστό τρόπο- θα βρίσκεται το εκάστοτε
!MAX, θα 'κρατιέται' στη μεταβλητή ΜΕΡΑ_ΜΑΧ και η αντίστοιχη
!θέση.
MAX ← SUM[1]
ΜΕΡΑ_ΜΑΧ ← 1
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 7
    ΑΝ SUM[i]>MAX ΤΟΤΕ
        MAX ← SUM[i]
        ΜΕΡΑ_ΜΑΧ ← i
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΜΕΡΑ_ΜΑΧ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ κινηματογράφοι
```

**3ο ΘΕΜΑ**

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένα τα στοιχεία ενός τετραγωνικού πίνακα διαστάσεων NxN, θα ελέγχει αν ο πίνακας είναι άνω τριγωνικός, κάτω τριγωνικός ή διαγώνιος.
Παρατήρηση: (α) Ένας πίνακας χαρακτηρίζεται ως άνω τριγωνικός όταν όλα τα στοιχεία που βρίσκονται κάτω της κύριας διαγωνίου είναι μηδέν. Τα στοιχεία αυτά είναι τα περιεχόμενα των κελιών A[i, j], όπου i > j.
(β) Ένας πίνακας χαρακτηρίζεται ως κάτω τριγωνικός όταν όλα τα στοιχεία που βρίσκονται άνω της κύριας διαγωνίου είναι μηδέν. Τα στοιχεία αυτά είναι τα περιεχόμενα των κελιών A[i, j], όπου i < j.
(γ) Ένας πίνακας χαρακτηρίζεται ως διαγώνιος αν είναι ταυτόχρονα άνω και κάτω τριγωνικός.

**ΛΥΣΗ**

```
Αλγόριθμος Τριγωνικός_πίνακας
 Δεδομένα // A, N //
 ! υποθέτουμε ότι ο πίνακας είναι άνω τριγωνικός
 άνω ← Αληθής
 Για i από 1 μέχρι N
     Για j από 1 μέχρι N   ! θα μπορούσε:  Για j από 1 μέχρι i – 1
     ! αν έστω και ένα στοιχείο <> 0, τον αποχαρακτηρίζουμε
         Αν (A[i, j] <> 0) και (i > j) τότε
             άνω ← Ψευδής
         Τέλος_αν
     Τέλος_επανάληψης
Τέλος_επανάληψης
! υποθέτουμε ότι ο πίνακας είναι κάτω τριγωνικός
κάτω ← Αληθής
Για i από 1 μέχρι N
    Για j από 1 μέχρι N   ! θα μπορούσε:  Για j από i + 1 μέχρι N
    ! αν έστω και ένα στοιχείο <> 0, τον αποχαρακτηρίζουμε
        Αν (A[i, j] <> 0) και (i < j) τότε
            κάτω ← Ψευδής
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Αν (άνω = Αληθής) και (κάτω = Αληθής) τότε
    Εκτύπωσε "Ο πίνακας είναι διαγώνιος"
Αλλιώς_αν (άνω = Αληθής) τότε
    Εκτύπωσε "Ο πίνακας είναι άνω τριγωνικός"
Αλλιώς_αν (κάτω = Αληθής) τότε
    Εκτύπωσε "Ο πίνακας είναι κάτω τριγωνικός"
Αλλιώς
    Εκτύπωσε "Ο πίνακας δεν έχει καμία από τις ιδιότητες"
Τέλος_αν
Τέλος Τριγωνικός_πίνακας
```

**3ο ΘΕΜΑ**

Σε ένα σχολείο κατά τη διαδικασία της εκλογής του 15μελούς μαθητικού συμβουλίου πήραν μέρος 30 υποψήφιοι. Για την καλύτερη οργάνωση η εφορευτική επιτροπή δημιούργησε δύο μονοδιάστατους πίνακες 30 θέσεων, τον ένα με τα ονόματα των υποψηφίων και τον άλλο με τις αντίστοιχες συνολικές τους ψήφους και με τα αντίστοιχα ονόματα ΥΠΟΨΗΦΙΟΣ και ΨΗΦΟΙ. Να γραφεί πρόγραμμα που:
(Α) να διαβάζει τα στοιχεία των δύο πινάκων
(Β) να υπολογίζει τα αποτελέσματα εμφανίζοντας τα ονόματα των 15 μαθητών που εκλέχθηκαν κατά φθίνουσα σειρά ψήφων.

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ εκλογές
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: ΨΗΦΟΙ[30], temp1
    ΑΚΕΡΑΙΕΣ: i, j
    ΧΑΡΑΚΤΗΡΕΣ: ΥΠΟΨΗΦΙΟΣ[30], temp2
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 30
    ΔΙΑΒΑΣΕ ΥΠΟΨΗΦΙΟΣ[i], ΨΗΦΟΙ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
!Ταξινόμηση του πίνακα ΨΗΦΟΙ από το μεγαλύτερο στο
!μικρότερο
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 30
    ΓΙΑ j ΑΠΟ 30 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ –1
        ΑΝ ΨΗΦΟΙ[j-1]<ΨΗΦΟΙ[j] ΤΟΤΕ
            temp1 ← ΨΗΦΟΙ[j-1]
            ΨΗΦΟΙ[j-1] ← ΨΗΦΟΙ[j]
            ΨΗΦΟΙ[j] ← temp1
            temp2 ← ΥΠΟΨΗΦΙΟΣ[j-1]
            ΥΠΟΨΗΦΙΟΣ[j-1] ← ΥΠΟΨΗΦΙΟΣ[j]
            ΥΠΟΨΗΦΙΟΣ[j] ← temp2
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 15
    ΓΡΑΨΕ ΥΠΟΨΗΦΙΟΣ[j]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ εκλογές
```

**3ο ΘΕΜΑ**

Να γραφεί πρόγραμμα το οποίο για ένα ξενοδοχείο που έχει 4 ορόφους και σε κάθε όροφο έχει 15 δωμάτια,
(Α) Εισάγει σε ένα δισδιάστατο πίνακα 4*15 το είδος του δωματίου το οποίο και αντιπροσωπεύει τον αριθμό κλινών του δωματίου ('Μ' για μονόκλινο, 'Δ' για δίκλινο και 'Τ' για τρίκλινο), κάνοντας και έλεγχο εγκυρότητας
Β) Με τη βοήθεια της διαδικασίας ΣΥΝΟΛΟ υπολογίζει και εμφανίζει πόσες κλίνες έχει συνολικά το ξενοδοχείο
Γ) Με τη βοήθεια της συνάρτησης ΟΡΟΦΟΣ εμφανίζει ποιος όροφος έχει τα περισσότερα δίκλινα.

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ Ξενοδοχείο
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: ι, j, Χ
        ΧΑΡΑΚΤΗΡΕΣ: ΞΕΝ[4, 15]
ΑΡΧΗ
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 4
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 15
        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
                ΔΙΑΒΑΣΕ ΞΕΝ[ι,j]
        ΜΕΧΡΙΣ_ΟΤΟΥ  ΞΕΝ[ι,j]  =  'Μ'  Ή  ΞΕΝ[ι,j]  =  'Δ'  Ή
ΞΕΝ[ι,j] = 'Τ'
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΚΑΛΕΣΕ ΣΥΝΟΛΟ(ΞΕΝ)
Χ ← ΟΡΟΦΟΣ(ΞΕΝ)
ΓΡΑΨΕ 'Περισσότερα δίκλινα στον', Χ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ξενοδοχείο

ΔΙΑΔΙΚΑΣΙΑ ΣΥΝΟΛΟ(Α)
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: ΣΥΝ, ι, j
        ΧΑΡΑΚΤΗΡΕΣ: Α[4, 15]
ΑΡΧΗ
ΣΥΝ ← 0
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 4
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 15
        ΑΝ Α[ι,j] = 'Μ' ΤΟΤΕ
                ΣΥΝ ← ΣΥΝ + 1
        ΑΛΛΙΩΣ_ΑΝ Α[ι,j] = 'Δ' ΤΟΤΕ
                ΣΥΝ ← ΣΥΝ + 2
        ΑΛΛΙΩΣ
                ΣΥΝ ← ΣΥΝ + 3
        ΤΕΛΟΣ_ΑΝ
```

```
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Συνολικός αριθμός κλινών', ΣΥΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΣΥΝΑΡΤΗΣΗ ΟΡΟΦΟΣ(Β): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: πλ, ι, j, Δικλ[4], ΜΑΧ, ΟΡΟΦ
        ΧΑΡΑΚΤΗΡΕΣ: Β[4, 15]
ΑΡΧΗ
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 4
    πλ ← 0
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 15
        ΑΝ Β[ι,j] = 'Δ' ΤΟΤΕ
                πλ ← πλ + 1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    Δικλ[ι] ← πλ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ ← Δικλ[1]
ΟΡ ← 1
ΓΙΑ ι ΑΠΟ 2 ΜΕΧΡΙ 4
    ΑΝ Δικλ[ι] > ΜΑΧ ΤΟΤΕ
        ΜΑΧ ← Δικλ[ι]
        ΟΡ ← ι
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
!Στο όνομα της συνάρτησης καταχωρούμε το αποτέλεσμα
ΟΡΟΦΟΣ ← ΟΡ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

**3ο ΘΕΜΑ**

Το ταξιδιωτικό γραφείο διοργανώνει εκδρομή σε ένα καλοκαιρινό θέρετρο της Μεσογείου και διαθέτει 100 αεροπορικά εισιτήρια Α θέσης και 200 Β θέσης. Να γράψετε αλγόριθμο ο οποίος:
(Α) θα διαβάζει επαναληπτικά το πλήθος των εισιτηρίων που ζητά κάθε πελάτης από κάθε θέση (Α και Β). Η επανάληψη θα τερματίζεται όταν εξαντληθούν τα εισιτήρια μίας από τις δύο θέσεις.
(Β) θα υπολογίζει και θα εμφανίζει την χρέωση για κάθε πελάτη **κλιμακωτά** σύμφωνα με τον ακόλουθο πίνακα:

| Πλήθος εισιτηρίων | (Χρέωση/Εισιτήριο) | |
|---|---|---|
| | Α θέση | Β θέση |
| 1 – 3 | 115 € | 100 € |
| 4 και άνω | 90 € | 80 € |

Αν κάποιος πελάτης έχει ζητήσει συνολικά (και από τις δύο θέσεις) περισσότερα από 6 εισιτήρια τότε η χρέωση δεν ακολουθεί την παραπάνω πολιτική αλλά είναι 75 € για κάθε εισιτήριο.
(Γ) θα εμφανίζει τις συνολικές εισπράξεις του γραφείου και το πλήθος των εισιτηρίων που δεν διατέθηκαν από κάθε θέση.

**ΛΥΣΗ**

```
Αλγόριθμος Θέμα3
Σ_εισΑ ← 0
Σ_εισΒ ← 0
S ← 0
Διάβασε εισΑ, εισΒ
Όσο (Σ_εισΑ+εισΑ<=100) και (Σ_εισΒ+εισΒ<=200) επανάλαβε
    Σ_εισΑ ← Σ_εισΑ + εισΑ
    Σ_εισΒ ← Σ_εισΒ + εισΒ
    Αν εισΑ + εισΒ > 6 τότε
        Χρέωση ← (εισΑ + εισΒ) * 75
    αλλιώς
        Αν  εισΑ <= 3 τότε
            Χρέωση1 ← 115 * εισΑ
        αλλιώς
            Χρέωση1 ← 115 * 3 + (εισΑ - 3) * 90
        Τέλος_αν
        Αν  εισΒ <= 3 τότε
            Χρέωση2 ← 100 * εισΒ
        αλλιώς
            Χρέωση2 ← 100 * 3 + (εισΒ - 3) * 80
        Τέλος_αν
        Χρέωση ← Χρέωση1 + Χρέωση2
    Τέλος_αν
    Εμφάνισε "Το κόστος των εισιτηρίων είναι:", Χρέωση
    S ← S + Χρέωση
    Διάβασε εισΑ, εισΒ
Τέλος_επανάληψης
Εμφάνισε "Τα συνολικά έσοδα του γραφείου είναι:", S
Κενές_Α ← 100 - Σ_εισΑ
Κενές_Β ← 200 - Σ_εισΒ
Εμφάνισε "Το υπόλοιπο των εισιτηρίων Α θέσης είναι:", Κενές_Α
Εμφάνισε "Το υπόλοιπο των εισιτηρίων Β θέσης είναι:", Κενές_Β
Τέλος Θέμα3
```

**4ο ΘΕΜΑ**

Εργάζεστε σε μια εταιρεία λογισμικού και έχετε αναλάβει την κατασκευή του λογισμικού που θα χρησιμοποιηθεί στους Ολυμπιακούς αγώνες για να καταχωρεί και να αξιολογεί τις επιδόσεις των αθλητών στο αγώνισμα της κατάδυσης.
Στο αγώνισμα λαμβάνουν μέρος 8 αθλητές. Ο κάθε αθλητής έχει μια μόνο (1) προσπάθεια η οποία βαθμολογείται από μια επιτροπή 10 κριτών. Οι κριτές βαθμολογούν με ένα ακέραιο αριθμό από το 0 έως και το 10.
Η βαθμολογία κάθε κριτή καταχωρείται και η βαθμολογία για την προσπάθεια του αθλητή εξάγεται με τον εξής τρόπο:
Η μέγιστη και η ελάχιστη βαθμολογία, δεν λαμβάνονται υπόψη στη βαθμολογία της προσπάθειας. Ο μέσος όρος από τις υπόλοιπες οκτώ (8) βαθμολογίες είναι ο τελικός βαθμός του αθλητή.
Ο αθλητής με τη καλύτερη βαθμολογία ανακηρύσσεται νικητής. Σε περίπτωση ισοβαθμίας μεταξύ δύο ή περισσότερων αθλητών, νικητής ανακηρύσσεται ο αθλητής που έχει την μικρότερη διαφορά μέγιστης και ελάχιστης βαθμολογίας του, από τις (10) αποδεκτές βαθμολογίες των κριτών.
Να γράψετε πρόγραμμα που θα εμφανίζει τον πρώτο νικητή του αγωνίσματος κατάδυσης.

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ ATHENS_2004
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, J, Β[8,10], MAX[8], MIN[8], SUM[8], ΔΙΑΦ[8],
              ΤΕΜΡ2
    ΠΡΑΓΜΑΤΙΚΕΣ: ΤΒ[8], ΤΕΜΡ1
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[8], ΤΕΜΡ3
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 8
    ΔΙΑΒΑΣΕ ΟΝ[Ι]
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10
        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
            ΔΙΑΒΑΣΕ Β[Ι,J]
        ΜΕΧΡΙΣ_ΟΤΟΥ Β[Ι,J] >= 0 ΚΑΙ Β[Ι,J] <= 10
   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 8
    MAX[I] ← Β[Ι,1] !Η μέγιστη βαθμολογία κάθε αθλητή
    MIN[I] ← Β[Ι,1] !Η ελάχιστη βαθμολογία κάθε αθλητή
    ΓΙΑ J ΑΠΟ 2 ΜΕΧΡΙ 10
        ΑΝ Β[Ι,J] > MAX[I] ΤΟΤΕ
            MAX[I] ← Β[Ι,J]
        ΤΕΛΟΣ_ΑΝ
        ΑΝ Β[Ι,J] < MIN[I] ΤΟΤΕ
            MIN[I] ← Β[Ι,J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΦ[I] ← MAX[I] – MIN[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 8
    SUM[I] ← 0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10
        SUM[I] ← SUM[I] + Β[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΒ[I] ← (SUM[I] – MAX[I] – MIN[I]) / 8
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 8
    ΓΙΑ J ΑΠΟ 8 ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
        ΑΝ ΤΒ[j-1] < ΤΒ[j] ΤΟΤΕ
            ΤΕΜΡ1 ← ΤΒ[j-1]
            ΤΒ[j-1] ← ΤΒ[j]
            ΤΒ[j] ← ΤΕΜΡ1
            ΤΕΜΡ2 ← ΔΙΑΦ[j-1]
            ΔΙΑΦ[j-1] ← ΔΙΑΦ[j]
            ΔΙΑΦ[j] ← ΤΕΜΡ2
            ΤΕΜΡ3 ← ΟΝ[j-1]
            ΟΝ[j-1] ← ΟΝ[j]
            ΟΝ[j] ← ΤΕΜΡ3
        ΑΛΛΙΩΣ_ΑΝ ΤΒ[j-1] = ΤΒ[j] ΤΟΤΕ
            ΑΝ ΔΙΑΦ[j-1] > ΔΙΑΦ[j] ΤΟΤΕ
                ΤΕΜΡ2 ← ΔΙΑΦ[j-1]
```

```
                    ΔΙΑΦ[j-1] ← ΔΙΑΦ[j]
                    ΔΙΑΦ[j] ← ΤΕΜΡ2
                    ΤΕΜΡ3 ← ΟΝ[j-1]
                    ΟΝ[j-1] ← ΟΝ[j]
                    ΟΝ[j] ← ΤΕΜΡ3
                ΤΕΛΟΣ_ΑΝ
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΄Νικητής ο/η΄, ΟΝ[1], ΄ με επίδοση: ΄, ΤΒ[1]
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ATHENS_2004
```

**4ο ΘΕΜΑ**

Για την τελική βαθμολογία των χωρών κατά τους Ολυμπιακούς αγώνες της Αθήνας, η στατιστική υπηρεσία χρησιμοποιεί τον μονοδιάστατο πίνακα ΧΩΡΕΣ[200] όπου είναι αποθηκευμένες οι 200 χώρες που λαμβάνουν μέρος και τον δισδιάστατο πίνακα ΜΕΤ[200,3] όπου αποθηκεύει τον συνολικό αριθμό των μεταλλίων που έλαβε κάθε χώρα από όλα τα αγωνίσματα. Για παράδειγμα η χώρα ΕΛΛΑΔΑ μπορεί να έχει πάρει 12 χρυσά, 6 αργυρά και 3 χάλκινα. Ο πίνακας ΧΩΡΕΣ και ο πίνακας ΜΕΤ είναι σε αντιστοιχία θέσεων δηλαδή, η χώρα που βρίσκεται στη θέση i στον πίνακα ΧΩΡΕΣ θα έχει πάρει τόσα μετάλλια όσα βρίσκονται στην i γραμμή του πίνακα ΜΕΤ.
Να γράψετε πρόγραμμα όπου:
(Α) Να διαβάζετε τα στοιχεία των 2 πινάκων.
(Β) Να εμφανίζετε τη χώρα με τα περισσότερα χρυσά μετάλλια.
(Γ) Αν οι χώρες παίρνουν 3 βαθμούς για κάθε χρυσό μετάλλιο, 2 για κάθε αργυρό και 1 για κάθε χάλκινο, να εμφανίζετε τις τρεις πρώτες χώρες μαζί με τη συνολική βαθμολογία τους.

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ Αθήνα
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: i, j, ΜΕΤ[200,3], ΜΑΧ, ΘΕΣΗ_ΜΑΧ,
              ΒΑΘ[200], ΤΕΜΡ
    ΧΑΡΑΚΤΗΡΕΣ: ΧΩΡΕΣ[200], ΤΕΜΡ1
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 200
    ΔΙΑΒΑΣΕ ΧΩΡΕΣ[i]
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 3
        ΔΙΑΒΑΣΕ ΜΕΤ[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ ← ΜΕΤ[1,1]
ΘΕΣΗ_ΜΑΧ ← 1
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 200
    ΑΝ ΜΕΤ[i,1] > ΜΑΧ ΤΟΤΕ
        ΜΑΧ ← ΜΕΤ[i,1]
        ΘΕΣΗ_ΜΑΧ ← i
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Η χώρα ', ΧΩΡΕΣ[ΘΕΣΗ_ΜΑΧ], ' με', ΜΑΧ, ' χρυσά'
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 200
    ΒΑΘ[i] ← ΜΕΤ[i,1] * 3 + ΜΕΤ[i,2] * 2 +  ΜΕΤ[i,3]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 200
    ΓΙΑ j ΑΠΟ 200 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
        ΑΝ ΒΑΘ[j-1] < ΒΑΘ[j] ΤΟΤΕ
                ΤΕΜΡ ← ΒΑΘ[j-1]
                ΒΑΘ[j-1] ← ΒΑΘ[j]
                ΒΑΘ[j] ← ΤΕΜΡ
                ΤΕΜΡ1 ← ΧΩΡΕΣ[j-1]
                ΧΩΡΕΣ[j-1] ← ΧΩΡΕΣ[j]
                ΧΩΡΕΣ[j] ← ΤΕΜΡ1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 3
    ΓΡΑΨΕ ΧΩΡΕΣ[i], ΒΑΘ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αθήνα
```

**4ο ΘΕΜΑ**

Στον πίνακα ΟΝΟΜΑ[50] καταχωρούνται τα ονόματα των ισάριθμων αθλητών που συμμετείχαν στον τελικό της δισκοβολίας και στον πίνακα ΕΠΙΔΟΣΗ[50, 6] οι αντίστοιχες βολές τους.
Να αναπτύξετε αλγόριθμο που θα εκτυπώνει τα ονόματα των αθλητών που πήραν μετάλλια (χρυσό, αργυρό και χάλκινο). Πρέπει να σημειωθεί ότι αν υπάρχουν αθλητές που έχουν την ίδια επίδοση, το μετάλλιο παίρνει ο αθλητής που πέτυχε την επίδοση νωρίτερα. Ενώ αν υπάρχουν αθλητές που έχουν την ίδια επίδοση, στην ίδια προσπάθεια, τότε θα μοιραστούν το μετάλλιο.

**ΛΥΣΗ**

```
Αλγόριθμος Δισκοβολία
Για i από 1 μέχρι 50
    Διάβασε ΟΝΟΜΑ[i]
    Για j από 1 μέχρι 6
        Διάβασε ΕΠΙΔΟΣΗ[i, j]
    Τέλος_επανάληψης
Τέλος_επανάληψης
Για i από 1 μέχρι 50
    ΜΑΧ[i] ← ΕΠΙΔΟΣΗ[i, 1]
    ΘΕΣΗ[i] ← 1
    Για j από 2 μέχρι 6
        Αν ΕΠΙΔΟΣΗ[i, j] > ΜΑΧ[i] ΤΟΤΕ
            ΜΑΧ[i] ← ΕΠΙΔΟΣΗ[i, j]
            ΘΕΣΗ[i] ← j
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Για i από 2 μέχρι 50
    Για j από 50 μέχρι i με_βήμα -1
        Αν ΜΑΧ[j-1] < ΜΑΧ[j] τότε
            Αντιμετάθεσε ΜΑΧ[j-1] , ΜΑΧ[j]
            Αντιμετάθεσε ΘΕΣΗ[j-1] , ΘΕΣΗ[j]
            Αντιμετάθεσε ΟΝΟΜΑ[j-1] , ΟΝΟΜΑ[j]
        Αλλιώς_αν ΜΑΧ[j-1] = ΜΑΧ[j] τότε
            Αν ΘΕΣΗ[j-1] > ΘΕΣΗ[j] τότε
                Αντιμετάθεσε ΘΕΣΗ[j-1] , ΘΕΣΗ[j]
                Αντιμετάθεσε ΟΝΟΜΑ[j-1] , ΟΝΟΜΑ[j]
            Τέλος_αν
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
i ← 1
Όσο ΜΑΧ[i]=ΜΑΧ[1] και ΘΕΣΗ[i]=ΘΕΣΗ[1] και i≤50 επανάλαβε
        Εκτύπωσε "Χρυσό: ", ΟΝΟΜΑ[i]
        i ← i + 1
Τέλος_επανάληψης
k ← i
Όσο ΜΑΧ[k]=ΜΑΧ[i] και ΘΕΣΗ[k]=ΘΕΣΗ[i] και k≤50 επανάλαβε
        Εκτύπωσε "Αργυρό: ", ΟΝΟΜΑ[k]
        k ← k + 1
Τέλος_επανάληψης
m ← k
Όσο ΜΑΧ[m]=ΜΑΧ[k] και ΘΕΣΗ[m]=ΘΕΣΗ[k] και m≤50 επανάλαβε
        Εκτύπωσε "Χάλκινο: ", ΟΝΟΜΑ[m]
        m ← m + 1
Τέλος_επανάληψης
Τέλος Δισκοβολία
```

**4ο ΘΕΜΑ**

Δυο βιβλιοπωλεία διαθέτουν Πλήθος_Α και Πλήθος_Β βιβλία αντίστοιχα. Να αναπτύξετε αλγόριθμο που θα διαβάζει τους τίτλους των βιβλίων του πρώτου και του δεύτερου καταστήματος σε δύο πίνακες και στη συνέχεια θα δημιουργεί νέους πίνακες που θα περιέχουν:
(Α) Τους τίτλους των βιβλίων που υπάρχουν και στα δυο καταστήματα, ταξινομημένους, αλφαβητικά ως προς τον τίτλο βιβλίου.
(Β) Τους τίτλους των βιβλίων που υπάρχουν στο πρώτο και όχι στο δεύτερο κατάστημα.

**ΛΥΣΗ**

```
Αλγόριθμος Αναζήτηση
Δεδομένα //Πλήθος_Α, Πλήθος_Β//
Για i από 1 μέχρι Πλήθος_Α
    Διάβασε Α[i]
Τέλος_επανάληψης
Για j από 1 μέχρι Πλήθος_Β
    Διάβασε Β[j]
Τέλος_επανάληψης
κ ← 0
λ ← 0
Για i από 1 μέχρι Πλήθος_Α
    j ← 1
    ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
    Όσο j <= Πλήθος_Β και ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ επανάλαβε
        Αν Β[j] = Α[i] τότε
                ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ
        αλλιώς
```

```
                    j ← j + 1
            Τέλος_αν
        Τέλος_επανάληψης
        Αν ΒΡΕΘΗΚΕ = ΑΛΗΘΗΣ τότε
            κ ← κ + 1
            ΚΟΙΝΑ[κ] ← Α[i]
        αλλιώς
            λ ← λ + 1
            ΔΙΑΦ[λ] ← Α[i]
        Τέλος_αν
Τέλος_επανάληψης
Για i από 2 μέχρι κ
        Για j από κ μέχρι i με_βήμα -1
            Αν ΚΟΙΝΑ[j-1] > ΚΟΙΝΑ[j] τότε
                ΤΕΜΡ ← ΚΟΙΝΑ[j-1]
                ΚΟΙΝΑ[j-1] ← ΚΟΙΝΑ[j]
                ΚΟΙΝΑ[j] ← ΤΕΜΡ
            Τέλος_αν
        Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος Αναζήτηση
```

**4ο ΘΕΜΑ**

Ένας εκδοτικός οίκος χρησιμοποιεί 35 διανομείς για τη διακίνηση των βιβλίων του. Στο τέλος κάθε μήνα καταγράφονται οι πωλήσεις που πραγματοποιήθηκαν από κάθε διανομέα ώστε να υπολογιστεί και το μπόνους που θα του αποδοθεί.
Είναι ευνόητο ότι οι πωλήσεις ενός διανομέα σε χρονικό διάστημα ενός μηνός δεν είναι κατ' ανάγκην 30. Το ποσό του μπόνους υπολογίζεται **κλιμακωτά** ανάλογα με το ποσό των μηνιαίων πωλήσεων κάθε διανομέα σύμφωνα με τον παρακάτω πίνακα:

| Μηνιαίες πωλήσεις διανομέα (€) | Μπόνους % |
|---|---|
| Μέχρι και 200 | 0 |
| Άνω των 200 μέχρι και 1000 | 1.5 |
| Άνω των 1000 | 4 |

Να αναπτύξετε αλγόριθμο που θα επιτελεί τις παρακάτω ενέργειες:
1. Για κάθε διανομέα:
α) θα διαβάζει το όνομά του και θα το καταχωρεί σε πίνακα ΟΝ καθώς και το μηνιαίο βασικό μισθό του και θα το καταχωρεί σε πίνακα Β.
β) θα διαβάζει επαναληπτικά τα ποσά των πωλήσεων που πέτυχε τον προηγούμενο μήνα. Η επαναληπτική διαδικασία θα ολοκληρώνεται όταν εισαχθεί αρνητικός αριθμός ή το μηδέν.
γ) θα υπολογίζει το μπόνους που θα λάβει και θα το εκτυπώνει.
2. Θα δημιουργεί πίνακα Τ, που θα περιέχει τις τελικές μηνιαίες απολαβές κάθε διανομέα.
3. Θα ελέγχει ποιος διανομέας έχει τον υψηλότερο τελικό μισθό που να είναι ταυτόχρονα μικρότερος από 600 €. Αν δεν υπάρχει τέτοιος, να εκτυπώνεται κατάλληλο μήνυμα.
4. Να εκτυπώνονται τα ονόματα όσων πωλητών είχαν το δεύτερο μεγαλύτερο μισθό μεταξύ των υπαλλήλων του εκδοτικού οίκου.

**ΛΥΣΗ**

```
Αλγόριθμος Εκδόσεις
Για i από 1 μέχρι 35
    Διάβασε ΟΝ[i], Β[i]
    S ← 0
    Διάβασε ποσό
    Όσο ποσό > 0 επανάλαβε
        S ← S + ποσό
        Διάβασε ποσό
    Τέλος_επανάληψης
    Αν S <= 200 τότε
        bonus ←  0
    Αλλιώς_αν S <= 1000 τότε
        bonus ← (S – 200) * 1.5/100
    Αλλιώς
        bonus ← 800 * 1.5/100 + (S – 1000) * 4/100
    Τέλος_αν
    Εκτύπωσε bonus
    Τ[i] ← Β[i] + bonus
Τέλος_επανάληψης
max ← -1
Για i από 1 μέχρι 35
    Αν Τ[i] < 600 και Τ[i] > max τότε
        max ← Τ[i]
        θ ← i
    Τέλος_αν
Τέλος_επανάληψης
Αν max = -1 τότε
    Εκτύπωσε "Κανείς με μισθό <600"
Αλλιώς
    Εκτύπωσε ΟΝ[θ]
Τέλος_αν
Για i από 2 μέχρι 35
    Για j από 35 μέχρι i με_βήμα –1
        Αν Τ[j–1] < Τ[j] τότε
            Αντιμετάθεσε Τ[j–1], Τ[j]
            Αντιμετάθεσε ΟΝ[j–1], Ο[j]
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
διακοπή ← ψευδής
Ι ← 1
πρώτος ← 0
Όσο (διακοπή = ψευδής) και (i <= 35) επανάλαβε
    Αν Τ[i] = Τ[1] τότε
        πρώτος ← πρώτος + 1
        i ← i + 1
    Αλλιώς
        διακοπή ← αληθής
    Τέλος_αν
Τέλος_επανάληψης
διακοπή ← ψευδής
i ← πρώτος + 1
Όσο (διακοπή =ψευδής) και (i <= 35) επανάλαβε
    Αν Τ[i] = Τ[πρώτος + 1] τότε
        Εκτύπωσε Ο[i]
        i ← i + 1
    Αλλιώς
        διακοπή ← αληθής
    Τέλος_αν
Τέλος_επανάληψης
Τέλος Εκδόσεις
```

**4ο ΘΕΜΑ**

Ο Ελληνικός Οργανισμός Τουρισμού αποφάσισε να διεξαγάγει μια έρευνα για τα ελληνικά ξενοδοχεία. Στην έρευνα συμμετέχουν τα 7000 ξενοδοχεία της επικράτειας, και γι' αυτά καταγράφονται οι μηνιαίες εισπράξεις για το περασμένο έτος. Να αναπτυχθεί αλγόριθμος που:
(α) Θα διαβάζει τα απαραίτητα στοιχεία.
(β) Θα υπολογίζει τις ετήσιες εισπράξεις κάθε ξενοδοχείου και θα τις εκχωρεί σε έναν νέο πίνακα.
(γ) Θα εκτυπώνει το όνομα του ξενοδοχείου με τις περισσότερες εισπράξεις.
(δ) Θα υπολογίζει και θα εκτυπώνει το όνομα κάθε ξενοδοχείου συνοδευόμενο από το πλήθος των μηνών που λειτούργησε (δηλαδή είχε έσοδα) την περασμένη χρονιά.
(ε) Θα δημιουργεί και θα εκτυπώνει τον πίνακα ΤΡΙΜΗΝΑ[4], που περιέχει τις εισπράξεις που παρουσιάστηκαν ανά τρίμηνο.

**ΛΥΣΗ**

```
Αλγόριθμος ΕΟΤ
Για i από 1 μέχρι 7000     ! ερώτημα α
    Διάβασε ΟΝΟΜΑ[i]
    Για j από 1 μέχρι 12
        Διάβασε ΕΙΣΠΡΑΞΕΙΣ[i, j]
    Τέλος_επανάληψης
Τέλος_επανάληψης
Για i από 1 μέχρι 7000     ! ερώτημα β
    άθροισμα[i] ← 0
    Για j από 1 μέχρι 12
        άθροισμα[i] ← άθροισμα[i] + ΕΙΣΠΡΑΞΕΙΣ[i, j]
    Τέλος_επανάληψης
Τέλος_επανάληψης
max ← άθροισμα[1]       ! ερώτημα γ
θέση ← 1
Για i από 2 μέχρι 7000
    Αν άθροισμα[i] > max τότε
        max ← άθροισμα[i]
        θέση ← i
    Τέλος_αν
Τέλος_επανάληψης
Εκτύπωσε ΟΝΟΜΑ[θέση]
Για i από 1 μέχρι 7000     ! ερώτημα δ
    μη_μηδεν ← 0
    Για j από 1 μέχρι 12
        Αν ΕΙΣΠΡΑΞΕΙΣ[i, j] <> 0 τότε
            μη_μηδεν ← μη_μηδεν + 1
        Τέλος_αν
    Τέλος_επανάληψης
```

```
    Εκτύπωσε ΟΝΟΜΑ[i], "λειτούργησε", μη_μηδεν, "μήνες"
Τέλος_επανάληψης
Για j από 1 μέχρι 12      ! ερώτημα ε
    sum[j] ← 0
    Για i από 1 μέχρι 7000
        sum[j] ← sum[j] + ΕΙΣΠΡΑΞΕΙΣ[i, j]
    Τέλος_επανάληψης
Τέλος_επανάληψης
κ ← 0
Για j από 1 μέχρι 10 με_βήμα 3
    κ ← κ + 1
    ΤΡΙΜΗΝΑ[κ] ← sum[j] + sum[j+1] + sum[j+2]
    Εκτύπωσε ΤΡΙΜΗΝΑ[κ]
Τέλος_επανάληψης
Τέλος ΕΟΤ
```

**4ο ΘΕΜΑ**

Στο πρωτάθλημα μπάσκετ της Α1 κατηγορίας ανδρών μια ομάδα συμμετέχει με 20 παίκτες δίνοντας συνολικά 40 αγώνες.
Να γραφεί αλγόριθμος ο οποίος :
α) να καταχωρεί σ' έναν πίνακα ακεραίων Π[20,40] τους πόντους που πέτυχε ο κάθε παίκτης σε κάθε αγώνα του πρωταθλήματος. Όταν ένας παίκτης δε συμμετέχει σ' έναν αγώνα τότε καταχωρούμε την τιμή -1 στον πίνακα. Θεωρήστε ότι οι τιμές που εισάγονται στον πίνακα είναι μεγαλύτερες ή ίσες του -1 και μικρότερες ή ίσες του 25. Μην ασχολείστε με τον έλεγχο εγκυρότητας δεδομένων.
β) να καταχωρεί σ' έναν πίνακα Ο[20] τα ονόματα των παικτών της ομάδας.
γ) να βρίσκει σε πόσους από τους 40 αγώνες , ένας μόνο παίκτης έφερε την καλύτερη επίδοση στον αγώνα.
δ) να εμφανίζει τα ονόματα των παικτών που δεν έχασαν κανέναν αγώνα στο πρωτάθλημα.
ε) να διαβάζει το όνομα ενός παίκτη και να εμφανίζει για το συγκεκριμένο παίκτη τους έξι αγώνες στους οποίους είχε τις καλύτερες επιδόσεις. Θεωρήστε ότι δεν υπάρχει περίπτωση ισοβαθμίας. Αν ο παίκτης αυτός έχει αγωνιστεί σε συνολικά λιγότερους από έξι αγώνες να εμφανίζεται κατάλληλα διαμορφωμένο μήνυμα.

**ΛΥΣΗ**

```
Αλγόριθμος Μπάσκετ
! ερωτήματα α, β
Για i από 1 μέχρι 20
    Διάβασε Ο[i]
    Για j από 1 μέχρι 40
        Διάβασε Π[i, j]
    Τέλος_επανάληψης
Τέλος_επανάληψης
! ερώτημα γ
πόσοι ← 0
Για j από 1 μέχρι 40
    ! βρες max στήλης
    max ← Π[1, j]
    Για i από 2 μέχρι 20
        Αν Π[i, j] > max τότε
            max ← Π[i, j]
        Τέλος_αν
    Τέλος_επανάληψης
    ! πόσα max;
    α ← 0
    Για i από 1 μέχρι 20
        Αν Π[i, j] = max τότε
            α ← α + 1
        Τέλος_αν
    Τέλος_επανάληψης
    Αν α = 1 τότε    ! μόνο 1 max
        πόσοι ← πόσοι + 1
    Τέλος_αν
Τέλος_επανάληψης
Εμφάνισε πόσοι
! ερώτημα δ
Για i από 1 μέχρι 20
    έχασε ← ΨΕΥΔΗΣ
    Για j από 1 μέχρι 40
        Αν Π[i, j] = -1 τότε
            έχασε ← ΑΛΗΘΗΣ
        Τέλος_αν
    Τέλος_επανάληψης
    Αν έχασε = ΨΕΥΔΗΣ    ! δεν έχασε κανέναν αγώνα
        Εμφάνισε Ο[i]
    Τέλος_αν
Τέλος_επανάληψης
! ερώτημα ε
Διάβασε όνομα_αναζ        ! ερώτημα ε
done ← ψευδής
pos ← 0
i ← 1
Όσο done = ψευδής και i <= 20 επανάλαβε
    Αν Ο[i] = όνομα_αναζ τότε
        done ← αληθής
        pos ← i
    αλλιώς
        i ← i + 1
    Τέλος_αν
Τέλος_επανάληψης
Αν done = αληθής τότε
   ! Στον πίνακα ΑΓ[40] κρατάω τους αριθμούς των αγώνων
   Για i από 1 μέχρι 40
       ΑΓ[Ι] ← i
   Τέλος_επανάληψης
  ! φθίνουσα ταξινόμηση γραμμής pos πίνακα Π
   Για i από 2 μέχρι 40
       Για j από 40 μέχρι i με_βήμα -1
           Αν Π[pos, j-1] < Π[pos, j] τότε
               Αντιμετάθεσε Π[pos, j-1], Π[pos, j]
               Αντιμετάθεσε ΑΓ[j-1], ΑΓ[j]
           Τέλος_αν
       Τέλος_επανάληψης
    Τέλος_επανάληψης
    α ← 0
    Για j από 1 μέχρι 6    ! κοίτα μόνο τις 6 πρώτες θέσεις
        Αν Π[pos, j] <> -1 τότε
            α ← α + 1
        Τέλος_αν
    Τέλος_επανάληψης
    Αν α = 6 τότε          ! έχει 6 καλύτερους
        Εμφάνισε 'Οι 6 καλύτεροι αγώνες'
        Για j από 1 μέχρι 6
            Εμφάνισε ΑΓ[j]
        Τέλος_επανάληψης
    αλλιώς
        Εμφάνισε " δεν έχει καν συμμετοχή σε 6 αγώνες"
    Τέλος_αν
αλλιώς
    Εμφάνισε "Δεν υπάρχει παίκτης με αυτό το όνομα"
Τέλος_αν
Τέλος Μπάσκετ
```

**4ο ΘΕΜΑ**

Να γραφεί πρόγραμμα το οποίο:
(Α) Να διαβάζει για τους 100 κατοίκους ενός χωριού τα ονόματα και το έτος γέννησης και να τα αποθηκεύει στους δύο πίνακες ΟΝΟΜΑ και ΕΤΟΣ.
(Β) Να εκτυπώνει τα ονόματα των κατοίκων που έχουν δικαίωμα ψήφου (πρέπει να είναι μεγαλύτεροι ή ίσοι με 18) καθώς και το πλήθος τους.
(Γ) Με δεδομένα την αποχή από τις εκλογές που κυμάνθηκε στο 20%, το ποσοστό των άκυρων ψηφοδελτίων 12% και το γεγονός ότι ψηφίστηκαν και τα 6 κόμματα που κατέβηκαν στις εκλογές, να υπολογίζει τη συχνότητα εμφάνισης κάθε ψήφου.
*Σημείωση: Οι δυνατοί ψήφοι είναι 1, 2, 3, 4, 5, 6 ανάλογα με το κόμμα προτίμησης του κάθε ψηφοφόρου και αποθηκεύονται στο μονοδιάστατο πίνακα ΨΗΦΟΙ[6].*

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ ΧΩΡΙΟ
ΣΤΑΘΕΡΕΣ
        ΤΡΕΧΟΝ = 2007
        ΑΠΟΧΗ = 0.2
        ΑΚΥΡΑ = 0.12
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: Ι, ΕΤΟΣ[100], ΗΛΙΚΙΑ[100], ΠΛΗΘΟΣ,
                  ΣΥΧΝ[6], ΨΗΦΟΙ[100], ΕΓΚΥΡΑ
        ΧΑΡΑΚΤΗΡΕΣ: ΟΝΟΜΑ[100]
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    ΓΡΑΨΕ ΄Δώσε το όνομα του κατοίκου΄
    ΔΙΑΒΑΣΕ ΟΝΟΜΑ[Ι]
    ΓΡΑΨΕ ΄Δώσε το έτος γέννησης του κατοίκου ΄
    ΔΙΑΒΑΣΕ ΕΤΟΣ[Ι]
    ΗΛΙΚΙΑ[Ι] ← ΤΡΕΧΟΝ – ΕΤΟΣ[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛΗΘΟΣ ← 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
```

```
   ΑΝ ΗΛΙΚΙΑ[Ι] >= 18 ΤΟΤΕ
        ΠΛΗΘΟΣ ← ΠΛΗΘΟΣ + 1
        ΓΡΑΨΕ ΄Ο ΄, ΟΝΟΜΑ[Ι], ΄ ψηφίζει΄
   ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΠΛΗΘΟΣ, ΄ κάτοικοι έχουν δικαίωμα ψήφου΄
ΕΓΚΥΡΑ ← ΠΛΗΘΟΣ * (1 – ΑΠΟΧΗ – ΑΚΥΡΑ)
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Α_Μ(ΕΓΚΥΡΑ)
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΓΡΑΨΕ ΄Δώσε το ψήφο προτίμησης΄
        ΔΙΑΒΑΣΕ ΨΗΦΟΙ[Ι]
    ΜΕΧΡΙΣ_ΟΤΟΥ ΨΗΦΟΙ[Ι] >= 1 ΚΑΙ ΨΗΦΟΙ[Ι] <= 6
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6      !Δημιουργία πίνακα συχνοτήτων
      ΣΥΧΝ[Ι] ← 0
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ ΕΓΚΥΡΑ
      ΣΥΧΝ[ΨΗΦΟΙ[Ι]] ← ΣΥΧΝ[ΨΗΦΟΙ[Ι]] + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
      ΓΡΑΨΕ ΄Το κόμμα ΄, Ι, ΄ έλαβε ΄, ΣΥΧΝ[Ι], ΄ ψήφους΄
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΧΩΡΙΟ
```

**Δεύτερος τρόπος για τον πίνακα συχνοτήτων:**

```
Για i από 1 μέχρι 6
      ΣΥΧΝ[i] ← 0
Τέλος_επανάληψης
Για i από 1 μέχρι ΕΓΚΥΡΑ
      Για j από 1 μέχρι 6
            Αν  ΨΗΦΟΙ[i] = j  τότε
                  ΣΥΧΝ[j] ← ΣΥΧΝ[j] + 1
            Τέλος_αν
      Τέλος_επανάληψης
Τέλος_επανάληψης
```

## 4ο ΘΕΜΑ

Τα τελευταία χρόνια ολοένα και συζητείται στους επιστημονικούς κύκλους το ενδεχόμενο της σταδιακής αύξησης της θερμοκρασίας του πλανήτη μας λόγω του φαινομένου του θερμοκηπίου. Στα πλαίσια της μελέτης των καταγραφομένων από τη μετεωρολογική υπηρεσία θερμοκρασιών να κατασκευάσετε πρόγραμμα που:

(Α) θα δέχεται για 20 πόλεις της Ελλάδας τα ονόματά τους και τις θερμοκρασίες τους όπως αυτές έχουν καταγραφεί για τα έτη από 1981 μέχρι και 2005 (25 έτη) και αποθηκεύει τα δεδομένα αυτά σε κατάλληλους πίνακες

- ✓ Τα ονόματα των πόλεων σε μονοδιάστατο πίνακα ΟΝ[20]
- ✓ Τα έτη σε μονοδιάστατο πίνακα ΕΤΗ[25]
- ✓ Τις θερμοκρασίες σε δισδιάστατο πίνακα Θ[20,25]

(Β) θα δέχεται το όνομα μιας πόλης και αν αυτή είναι ανάμεσα στις 20 της μελέτης, θα ταξινομεί τις θερμοκρασίες που σημειώθηκαν σ' αυτή κατά φθίνουσα διάταξη

(Γ) και τέλος θα εμφανίζει τις ταξινομένες αυτές θερμοκρασίες ακολουθούμενη η κάθε μια από τη χρονιά που σημειώθηκε.

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ_4
ΜΕΤΑΒΛΗΤΕΣ
      ΠΡΑΓΜΑΤΙΚΕΣ: Θ[20,25], temp1
      ΑΚΕΡΑΙΕΣ: k, i, j, POS, ΕΤΗ[25], temp2
      ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[20], πόλη
      ΛΟΓΙΚΕΣ: DONE
ΑΡΧΗ
! Εισαγωγή δεδομένων
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 25
      ΔΙΑΒΑΣΕ ΕΤΗ[j]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
      ΔΙΑΒΑΣΕ ΟΝ[i]
      ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 25
            ΔΙΑΒΑΣΕ Θ[i, j]
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Δώσε όνομα πόλης'
ΔΙΑΒΑΣΕ πόλη
i ← 1
DONE ← ΨΕΥΔΗΣ
POS ← 0
ΟΣΟ (i <= 20) ΚΑΙ (DONE = ΨΕΥΔΗΣ)  ΕΠΑΝΑΛΑΒΕ
      ΑΝ  ΟΝ[i] = πόλη  ΤΟΤΕ
            DONE ← ΑΛΗΘΗΣ
            POS ← i
      ΑΛΛΙΩΣ
            i ← i + 1
      ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ  DONE = ΨΕΥΔΗΣ  ΤΟΤΕ
      ΓΡΑΨΕ  'Δεν υπάρχει πόλη με όνομα: ', πόλη
ΑΛΛΙΩΣ
ΓΙΑ k ΑΠΟ 2 ΜΕΧΡΙ 25
      ΓΙΑ j ΑΠΟ 25 ΜΕΧΡΙ k ΜΕ ΒΗΜΑ –1
                  ΑΝ Θ[POS, j-1] < Θ[POS, j] ΤΟΤΕ
                        Temp1 ← Θ[POS, j-1]
                        Θ[POS, j-1] ← Θ[POS, j]
                        Θ[POS, j] ← Temp1
                        Temp2 ← ΕΤΗ[j-1]
                        ΕΤΗ[j-1] ← ΕΤΗ[j]
                        ΕΤΗ[j] ← Temp2
                  ΤΕΛΟΣ_ΑΝ
            ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 25
       ΓΡΑΨΕ  Θ[POS, j], ΕΤΗ[j]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

## 4ο ΘΕΜΑ

Τα αποτελέσματα των αγώνων του Eurobasket (νίκη ή ήττα) καθώς και η διαφορά πόντων για τον όμιλο που ήταν και η Εθνική μας περιέχονται στους παρακάτω πίνακες:

- ✓ Τα ονόματα των ομάδων σε μονοδιάστατο πίνακα ΟΝ[6]
- ✓ Τον χαρακτηρισμό κάθε αγώνα (”Ν” ή ”Η”) σε δισδιάστατο πίνακα Α[6,6]
- ✓ Τη διαφορά πόντων κάθε αγώνα σε δισδιάστατο πίνακα Π[6,6]

Να γραφεί πρόγραμμα που διαβάζει τα δεδομένα στους παραπάνω πίνακες και εκτυπώνει τη τελική βαθμολογία του ομίλου (ονόματα και βαθμούς) ξεκινώντας από τον καλύτερο

*Σημείωση: Σε περίπτωση ισοβαθμίας προηγείται η ομάδα που έχει την καλύτερη διαφορά πόντων από τις ισόβαθμες. Αν έχουν την ίδια διαφορά πόντων, τότε η κατάταξη γίνεται ονομαστικά.*

*Για παράδειγμα οι πίνακες θα έχουν τη μορφή*

| ***Πιν. Α*** | *Γερμανία* | *Ελλάδα* | *Σερβία* | *Ιταλία* | *Ρωσία* | *Γαλλία* |
|---|---|---|---|---|---|---|
| *Γερμανία* | | | | | | |
| *Ελλάδα* | *Ν* | | | | | |
| *Σερβία* | *Η* | *Η* | | | | |
| *Ιταλία* | *Ν* | *Ν* | *Η* | | | |
| *Ρωσία* | *Η* | *Η* | *Η* | *Ν* | | |
| *Γαλλία* | *Ν* | *Ν* | *Ν* | *Ν* | *Η* | |

| ***Πιν. Π*** | *Γερμανία* | *Ελλάδα* | *Σερβία* | *Ιταλία* | *Ρωσία* | *Γαλλία* |
|---|---|---|---|---|---|---|
| *Γερμανία* | | | | | | |
| *Ελλάδα* | *5* | | | | | |
| *Σερβία* | *6* | *7* | | | | |
| *Ιταλία* | *3* | *4* | *8* | | | |
| *Ρωσία* | *8* | *6* | *9* | *9* | | |
| *Γαλλία* | *1* | *8* | *2* | *2* | *7* | |

| ***Πιν. ΟΝ*** | *Γερμανία* | *Ελλάδα* | *Σερβία* | *Ιταλία* | *Ρωσία* | *Γαλλία* |
|---|---|---|---|---|---|---|

*Κάθε κελί αντιστοιχεί σε έναν αγώνα μεταξύ της ομάδας που αντιστοιχεί στη γραμμή και της ομάδας που αντιστοιχεί στη στήλη.*

*Το αποτέλεσμα ”Ν” ή ”Η” σημαίνει ότι η ομάδα που αντιστοιχεί στη γραμμή νίκησε ή αντίστοιχα έχασε από την ομάδα που αντιστοιχεί στη στήλη. Ο αριθμός δείχνει τη διαφορά πόντων. Π.χ. η Ελλάδα νίκησε τη Γερμανία με 5 πόντους διαφορά ενώ η Σερβία ηττήθηκε από την Ελλάδα με 7 πόντους διαφορά.*

*Τα στοιχεία της κύριας διαγωνίου δεν περιέχουν καμία πληροφορία αφού καμία ομάδα δεν παίζει με τον εαυτό της.*

*Ο πίνακας περιέχει στοιχεία μόνο κάτω από τη διαγώνιό του, είναι δηλαδή τετραγωνικός.*

*Για κάθε αγώνα η νικήτρια ομάδα παίρνει 2 βαθμούς και η ηττημένη 1 βαθμό.*

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ4
```

```
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: ι,j,Π[6,6],ΒΑΘ[6],ΔΙΑΦ[6],Help
        ΧΑΡΑΚΤΗΡΕΣ:ΟΝ[6],Α[6,6],Help2
ΑΡΧΗ
 ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 6
    ΓΡΑΨΕ 'Δώσε το όνομα της ομάδας',ι
    ΔΙΑΒΑΣΕ ΟΝ[ι]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 6
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 6
        ΑΝ (ι>j) ΤΟΤΕ
            ΓΡΑΨΕ 'Δώσε αποτέλεσμα αγώνα',ΟΝ[ι],'με',ΟΝ[j]
            ΔΙΑΒΑΣΕ Α[ι,j]
            ΓΡΑΨΕ 'Δώσε διαφορά πόντων',ΟΝ[ι],'με',ΟΝ[j]
            ΔΙΑΒΑΣΕ Π[ι,j]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 6
    ΒΑΘ[ι] ← 0      ! Δημιουργία πινάκων με βαθμούς
    ΔΙΑΦ[ι] ← 0     ! και διαφορές πόντων
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 6
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 6
        ΑΝ (ι>j) ΤΟΤΕ
            ΑΝ Α[ι,j] = 'Ν' ΤΟΤΕ
                ΒΑΘ[ι] ← ΒΑΘ[ι] + 2
                ΒΑΘ[j] ← ΒΑΘ[j] + 1
                ΔΙΑΦ[ι] ← ΔΙΑΦ[ι] + Π[ι,j]
                ΔΙΑΦ[j] ← ΔΙΑΦ[j] - Π[ι,j]
            ΤΕΛΟΣ_ΑΝ
            ΑΝ Α[ι,j] = 'Η' ΤΟΤΕ
                ΒΑΘ[ι] ← ΒΑΘ[ι] + 1
                ΒΑΘ[j] ← ΒΑΘ[j] + 2
                ΔΙΑΦ[ι] ← ΔΙΑΦ[ι] - Π[ι,j]
                ΔΙΑΦ[j] ← ΔΙΑΦ[j] + Π[ι,j]
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΙΑ ι ΑΠΟ 2 ΜΕΧΡΙ 30
    ΓΙΑ j ΑΠΟ 30 ΜΕΧΡΙ ι ΜΕ_ΒΗΜΑ −1
        ΑΝ ΒΑΘ[j-1] < ΒΑΘ[j] ΤΟΤΕ
            Help ← ΒΑΘ[j]
            ΒΑΘ[j] ← ΒΑΘ[j-1]
            ΒΑΘ[j-1] ← Help
            Help ← ΔΙΑΦ[j]
            ΔΙΑΦ[j] ← ΔΙΑΦ[j-1]
            ΔΙΑΦ[j-1] ← Help
            Help2 ← ΟΝ[j]
            ΟΝ[j] ← ΟΝ[j-1]
            ΟΝ[j-1] ← Help2
        ΤΕΛΟΣ_ΑΝ
        ΑΝ (ΒΑΘ[j-1]=ΒΑΘ[j]) ΚΑΙ (ΔΙΑΦ[j-1]<ΔΙΑΦ[j]) ΤΟΤΕ
            Help ← ΔΙΑΦ[j]
            ΔΙΑΦ[j] ← ΔΙΑΦ[j-1]
            ΔΙΑΦ[j-1] ← Help
            Help2 ← ΟΝ[j]
            ΟΝ[j] ← ΟΝ[j-1]
            ΟΝ[j-1] ← Help2
        ΤΕΛΟΣ_ΑΝ
        ΑΝ (ΒΑΘ[j-1]=ΒΑΘ[j]) ΚΑΙ (ΔΙΑΦ[j-1]=ΔΙΑΦ[j]) ΚΑΙ
            & (ΟΝ[j-1]>ΟΝ[j]) ΤΟΤΕ
            Help2 ← ΟΝ[j]
            ΟΝ[j] ← ΟΝ[j-1]
            ΟΝ[j-1] ← Help2
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΡΑΨΕ 'Τελική κατάταξη ομίλου'
 ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 6
    ΓΡΑΨΕ ΟΝ[ι],ΒΑΘ[ι]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΘΕΜΑ4
```

**4ο ΘΕΜΑ**

Σε μια εταιρεία, καταγράφονται τα τηλεφωνήματα που έκαναν οι 10 υπάλληλοι κατά τη διάρκεια ενός μήνα. Συνολικά έκαναν 250 κλήσεις.

Για τη καταγραφή των δεδομένων αυτών, χρησιμοποιούνται τρεις πίνακες. Ένας μονοδιάστατος **ΟΝ**[10], με τα ονοματεπώνυμα των υπαλλήλων. Ένας δισδιάστατος, **ΚΛΗΣΕΙΣ**[250,2], σε κάθε γραμμή του οποίου αποθηκεύουμε πληροφορίες για κάθε κλήση που έγινε κατά τη διάρκεια του μήνα ως εξής: στην πρώτη στήλη αποθηκεύεται το ονοματεπώνυμο του υπαλλήλου που έκανε την κλήση, στην δεύτερη στήλη το γράμμα 'Α' αν η κλήση ήταν αστική και το γράμμα 'Υ' αν κλήση ήταν υπεραστική. Τέλος, ένας μονοδιάστατος πίνακας **ΔΙΑΡΚΕΙΑ**[250] με τη διάρκεια της κάθε κλήσης σε δευτερόλεπτα.

| | ΟΝ |
|---|---|
| 1 | ΗΛΙΑΣ |
| 2 | ΚΩΣΤΑΣ |
| 3 | ΓΙΑΝΝΗΣ |
| | ..... |
| | ..... |
| 10 | ΓΕΩΡΓΙΟΣ |

| | ΚΛΗΣΕΙΣ | | ΔΙΑΡΚΕΙΑ |
|---|---|---|---|
| 1 | ΗΛΙΑΣ | Α | 60 |
| 2 | ΗΛΙΑΣ | Α | 45 |
| 3 | ΑΡΗΣ | Α | 128 |
| 4 | ΓΙΑΝΝΗΣ | Υ | 36 |
| 5 | ΗΛΙΑΣ | Υ | 30 |
| 6 | ΓΙΑΝΝΗΣ | Α | 60 |
| | ..... | | .... |
| | ..... | | .... |
| | ..... | | .... |
| 250 | ΗΛΙΑΣ | Υ | 32 |

Παρακάτω δίνεται ο πίνακας της χρέωσης των κλήσεων για την εταιρεία (η χρέωση της κλήσης είναι **κλιμακωτή**).

| ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΚΛΗΣΗΣ | ΔΙΑΡΚΕΙΑ (σε sec) | ΧΡΕΩΣΗ / sec |
|---|---|---|
| ΑΣΤΙΚΗ | 1-200 | 0,025 |
| | 201-500 | 0,020 |
| | περισσότερα από 500 | 0,015 |
| ΥΠΕΡΑΣΤΙΚΗ | 1-250 | 0,030 |
| | 251-600 | 0,025 |
| | περισσότερα από 600 | 0,020 |

Να γράψετε πρόγραμμα το οποίο:

(α) Θα δημιουργεί τους πίνακες ΟΝ[10] και ΚΛΗΣΕΙΣ[250,2], ελέγχοντας ότι στην δεύτερη στήλη του πίνακα ΚΛΗΣΕΙΣ αποθηκεύονται μόνο τα γράμματα 'Υ' ή 'Α', και ότι στον πίνακα ΔΙΑΡΚΕΙΑ[250] αποθηκεύονται αριθμοί μεγαλύτεροι του μηδενός.

(β) Θα δημιουργεί ένα νέο πίνακα ΧΡ[10] με τα χρήματα που πληρώνει η εταιρεία για τον κάθε υπάλληλο.

(γ) Θα διαβάζει το όνομα ενός υπαλλήλου και θα εμφανίζει τα λεφτά που πληρώνει για αυτόν η εταιρεία. Αν ο υπάλληλος δεν υπάρχει θα πρέπει να εμφανίζεται ανάλογο μήνυμα. Η εύρεση του υπαλλήλου θα πρέπει να γίνεται με τη βοήθεια υποπρογράμματος που θα φτιάξετε για το σκοπό αυτό.

(δ) Θα εμφανίζει τους υπαλλήλους και τη χρέωσή τους, ξεκινώντας από αυτόν που έκανε τη μεγαλύτερη χρέωση στην εταιρεία, προς αυτόν που έκανε τη μικρότερη. Αν δυο υπάλληλοι είχαν την ίδια χρέωση, να τους εμφανίζει αλφαβητικά.

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ ΚΛΗΣΕΙΣ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, ΔΙΑΡΚΕΙΑ[250]
    ΠΡΑΓΜΑΤΙΚΕΣ: ΠΟΣΟ[250], ΧΡ[10], ΤΕΜΡ1
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[10], ΚΛΗΣΕΙΣ[250,2], Χ, ΤΕΜΡ2
ΑΡΧΗ
 ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
     ΔΙΑΒΑΣΕ ΟΝ[Ι]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 250
     ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
         ΔΙΑΒΑΣΕ ΔΙΑΡΚΕΙΑ[Ι]
     ΜΕΧΡΙΣ_ΟΤΟΥ ΔΙΑΡΚΕΙΑ[Ι] > 0
     ΔΙΑΒΑΣΕ ΚΛΗΣΕΙΣ[Ι,1]
     ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
         ΔΙΑΒΑΣΕ ΚΛΗΣΕΙΣ[Ι,2]
     ΜΕΧΡΙΣ_ΟΤΟΥ ΚΛΗΣΕΙΣ[Ι,2]='Α' Ή ΚΛΗΣΕΙΣ[Ι,2]= 'Υ'
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 250
     ΑΝ ΚΛΗΣΕΙΣ[Ι,2]='Α' ΤΟΤΕ
         ΑΝ ΔΙΑΡΚΕΙΑ[Ι] <= 200 ΤΟΤΕ
             ΠΟΣΟ[Ι] ← 0.025 * ΔΙΑΡΚΕΙΑ[Ι]
         ΑΛΛΙΩΣ_ΑΝ ΔΙΑΡΚΕΙΑ[Ι] <= 500 ΤΟΤΕ
             ΠΟΣΟ[Ι] ← 0.025*200 + 0.020*(ΔΙΑΡΚΕΙΑ[Ι] – 200)
         ΑΛΛΙΩΣ
             ΠΟΣΟ[Ι] ← 0.025*200 + 0.020*300 +
             &       0.015*(ΔΙΑΡΚΕΙΑ[Ι] – 500)
         ΤΕΛΟΣ_ΑΝ
     ΑΛΛΙΩΣ
         ΑΝ ΔΙΑΡΚΕΙΑ[Ι] <= 250 ΤΟΤΕ
             ΠΟΣΟ[Ι] ← 0.030*ΔΙΑΡΚΕΙΑ[Ι]
         ΑΛΛΙΩΣ_ΑΝ ΔΙΑΡΚΕΙΑ[Ι] <= 600 ΤΟΤΕ
             ΠΟΣΟ[Ι] ← 0.030*250 + 0.025*(ΔΙΑΡΚΕΙΑ[Ι] – 250)
         ΑΛΛΙΩΣ
```

```
            ΠΟΣΟ[Ι] ← 0.030*250 + 0.025*350 +
            &     0.020*(ΔΙΑΡΚΕΙΑ[Ι] – 600)
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΧΡ[Ι] ← 0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 250
        ΑΝ ΚΛΗΣΕΙΣ[J,1] = ΟΝ[Ι] ΤΟΤΕ
            ΧΡ[Ι] ← ΧΡ[Ι] + ΠΟΣΟ[J]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ Χ
  ΚΑΛΕΣΕ ΕΥΡΕΣΗ(ΟΝ, Χ)
  ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 10
    ΓΙΑ J ΑΠΟ 10 ΜΕΧΡΙ Ι ΜΕ ΒΗΜΑ -1
        ΑΝ ΧΡ[J-1]<ΧΡ[J] ΤΟΤΕ
            ΤΕΜΡ1 ← ΧΡ[J-1]
            ΧΡ[J-1] ← ΧΡ[J]
            ΧΡ[J] ← ΤΕΜΡ1
            ΤΕΜΡ2 ← ΟΝ[J-1]
            ΟΝ[J-1] ← ΟΝ[J]
            ΟΝ[J] ← ΤΕΜΡ2
        ΑΛΛΙΩΣ_ΑΝ ΧΡ[J-1]=ΧΡ[J] ΤΟΤΕ
            ΑΝ ΟΝ[J-1]>ΟΝ[J] ΤΟΤΕ
                ΤΕΜΡ2 ← ΟΝ[J-1]
                ΟΝ[J-1] ← ΟΝ[J]
                ΟΝ[J] ← ΤΕΜΡ2
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ 1 ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΡΑΨΕ ΟΝ[Ι], ΧΡ[Ι]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ


ΔΙΑΔΙΚΑΣΙΑ ΕΥΡΕΣΗ(ΟΝ, Χ)
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, ΘΕΣΗ
    ΛΟΓΙΚΕΣ: ΒΡΕΘΗΚΕ
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[10], Χ
ΑΡΧΗ
  Ι ← 1
  ΘΕΣΗ ← 0
  ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
  ΌΣΟ Ι <= 10 ΚΑΙ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
    ΑΝ ΟΝ[Ι]=Χ ΤΟΤΕ
        ΘΕΣΗ ← Ι
        ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
    ΑΛΛΙΩΣ
        Ι ← Ι + 1
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΝ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'δεν υπάρχει ο υπάλληλος'
  ΑΛΛΙΩΣ
    ΓΡΑΨΕ ΧΡ[ΘΕΣΗ]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

**4ο ΘΕΜΑ**

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 αναπτύξετε πρόγραμμα το οποίο:

(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 εμφανίζει για κάθε πόλη το όνομά της και τη μέγιστη θερμοκρασία που καταγράφηκε σε αυτήν, στη διάρκεια του μήνα. Ο υπολογισμός της μέγιστης θερμοκρασίας να γίνεται με τη χρήση υποπρογράμματος που θα κατασκευάσετε για το σκοπό αυτό.

(γ) Θα διαβάζει το όνομα μιας πόλης και θα εμφανίζει τη μέγιστη θερμοκρασία που καταγράφηκε στην πόλη αυτή στη διάρκεια του μήνα. Ο υπολογισμός της μέγιστης θερμοκρασίας να γίνεται με τη βοήθεια του υποπρογράμματος που αναπτύχθηκε στο προηγούμενο ερώτημα.

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ ΕΜΥ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: pos, i, j
    ΠΡΑΓΜΑΤΙΚΕΣ: ΘΕΡΜ[20, 31], μέγιστη
    ΧΑΡΑΚΤΗΡΕΣ: ΠΟΛΗ[20], όνομα_πόλης
    ΛΟΓΙΚΕΣ: done
ΑΡΧΗ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20  ! ερώτημα α
    ΔΙΑΒΑΣΕ ΠΟΛΗ[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 31
        ΔΙΑΒΑΣΕ ΘΕΡΜ[i, j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20   ! ερώτημα β
     μέγιστη  ← Υπολόγισε_max (i, ΘΕΡΜ)
     ΓΡΑΨΕ ΠΟΛΗ[i], μέγιστη
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ όνομα_πόλης ! σειριακή αναζήτηση, ερώτημα γ
  done ← ΨΕΥΔΗΣ
  pos ← 0
  i ← 1
  ΌΣΟ (i <= 20) ΚΑΙ (done = ΨΕΥΔΗΣ) ΕΠΑΝΑΛΑΒΕ
     ΑΝ (ΠΟΛΗ[i] = όνομα_πόλης) ΤΟΤΕ
        done ← ΑΛΗΘΗΣ
        pos ← i
     ΑΛΛΙΩΣ
        i ← i + 1
     ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΝ (done = ΑΛΗΘΗΣ) ΤΟΤΕ
     μέγιστη  ← Υπολόγισε_max (pos, ΘΕΡΜ)
     ΓΡΑΨΕ ΠΟΛΗ[pos], μέγιστη
  ΑΛΛΙΩΣ
     ΓΡΑΨΕ 'Δεν υπάρχει πόλη με αυτό το όνομα'
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΕΜΥ


ΣΥΝΑΡΤΗΣΗ Υπολόγισε_max (γραμμή, ΠΙΝΑΚΑΣ): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: j
    ΠΡΑΓΜΑΤΙΚΕΣ: ΠΙΝΑΚΑΣ[20, 31], max
ΑΡΧΗ
  max ← ΠΙΝΑΚΑΣ[γραμμή, 1] ! εύρεση μεγίστου στη γραμμή
  ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 31
     ΑΝ (ΠΙΝΑΚΑΣ[γραμμή, j] > max) ΤΟΤΕ
        max ← ΠΙΝΑΚΑΣ[γραμμή, j]
     ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  Υπολόγισε_max ← max
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

**4ο ΘΕΜΑ**

Το Υπουργείο Ναυτιλίας σε συνεργασία με το Υπουργείο Τουρισμού σχεδιάζουν να προωθήσουν πληροφορίες ιστιοπλοΐας για τις αποστάσεις σε ναυτικά μίλια 15 απομακρυσμένων νησιών του Αιγαίου έτσι ώστε να ενισχυθεί ο θαλασσινός τουρισμός. Για το έργο αυτό χρειάζεται μονοδιάστατος πίνακας Ονόματα 15 θέσεων που περιέχει τα ονόματα 15 νησιών καθώς και δισδιάστατος τετραγωνικός πίνακας Απόσταση με 15 γραμμές και 15 στήλες που περιέχει τις μεταξύ τους αποστάσεις σε ναυτικά μίλια ως εξής:

Στο στοιχείο Απόσταση [5,3] καταχωρείται η απόσταση μεταξύ των νησιών που τα ονόματά τους βρίσκονται στην 5η και την 3η θέση του πίνακα Ονόματα. Για παράδειγμα στον πίνακα του σχήματος που καταχωρούνται οι αποστάσεις για 6 μόνο νησιά, η απόσταση μεταξύ της Κιμώλου και της Αστυπάλαιας είναι 88 ναυτικά μίλια.

Πίνακες με τα ονόματα και τις αποστάσεις για 6 μόνο νησιά του Αιγαίου

| Ονόματα | Απόσταση | | | | | |
|---|---|---|---|---|---|---|
| | **Φολέγανδρος** | **Ανάφη** | **Αστυπάλαια** | **Σίκινος** | **Κίμωλος** | **Τήλος** |
| **Φολέγανδρος** | - | - | - | - | - | - |
| **Ανάφη** | **43** | **-** | **-** | **-** | **-** | - |
| **Αστυπάλαια** | **68** | **31** | **-** | **-** | **-** | - |
| **Σίκινος** | **9** | **36** | **59** | **-** | **-** | - |
| **Κίμωλος** | **18** | **64** | **88** | **28** | **-** | - |
| **Τήλος** | **118** | **79** | **50** | **110** | **137** | - |

Να υλοποιήσετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

(α) Διαβάζει τα ονόματα των νησιών και τα αποθηκεύει σε μονοδιάστατο πίνακα

(β) Εισάγει σε δισδιάστατο πίνακα τις αποστάσεις μεταξύ των νησιών ως εξής :

Διαβάζει από το πληκτρολόγιο και τοποθετεί τις αποστάσεις μόνο για τις θέσεις του πίνακα που βρίσκονται κάτω της κυρίας διαγωνίου ελέγχοντας να είναι θετικοί αριθμοί.

(γ) Εντοπίζει και εμφανίζει τα ονόματα των 2 νησιών με τη μικρότερη μεταξύ τους απόσταση

(δ) Να κατασκευαστεί συνάρτηση η οποία θα δέχεται σαν παράμετρο έναν πίνακα[15,15] με στοιχεία συμπληρωμένα μόνο κάτω από την κύρια διαγώνιο και έναν αριθμό από το 1 ως το 15 που αντιστοιχεί στη θέση ενός νησιού στον πίνακα Ονόματα. Η συνάρτηση θα επιστρέφει το μέσο όρο όλων των στοιχείων του πίνακα που βρίσκονται στην ίδια γραμμή καθώς και αυτών που βρίσκονται στην ίδια στήλη με τον αριθμό που δόθηκε σαν παράμετρο και βρίσκονται κάτω από την κύρια διαγώνιο.

(ε) Θα υπολογίζει και θα εμφανίζει το νησί που έχει τον ελάχιστο μέσο όρο αποστάσεων από τα άλλα νησιά έτσι ώστε να κατασκευαστεί εκεί ένας σταθμός ανεφοδιασμού. Για τον υπολογισμό του μέσου όρου της απόστασης κάποιου νησιού από όλα τα άλλα να χρησιμοποιηθεί η συνάρτηση του προηγούμενου ερωτήματος.

*<u>Σημείωση:</u> Θεωρήστε ότι όλες οι αποστάσεις είναι διαφορετικές μεταξύ τους.*

*Στην απάντηση η συνάρτηση θα γράφεται μετά το τέλος του κυρίως προγράμματος σύμφωνα με τους συντακτικούς κανόνες της ΓΛΩΣΣΑΣ.*

**<u>ΛΥΣΗ</u>**

```
ΠΡΟΓΡΑΜΜΑ  ΘΕΜΑ_4
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: νησί, γραμμή, στήλη, Απ[15, 15], min, i, j
    ΧΑΡΑΚΤΗΡΕΣ: Ον[15]
ΑΡΧΗ
  ΓΙΑ νησί ΑΠΟ 1 ΜΕΧΡΙ 15
      ΔΙΑΒΑΣΕ Ον[νησί]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ γραμμή ΑΠΟ 1 ΜΕΧΡΙ 15
      ΓΙΑ στήλη ΑΠΟ 1 ΜΕΧΡΙ γραμμή – 1
          Αρχή_Επανάληψης
              ΔΙΑΒΑΣΕ Απ[γραμμή, στήλη]
          Μέχρις_ότου Απ[γραμμή, στήλη] > 0
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  i ← 2
  j ← 1
  ΓΙΑ γραμμή ΑΠΟ 3 ΜΕΧΡΙ 15
      ΓΙΑ στήλη ΑΠΟ 1 ΜΕΧΡΙ γραμμή – 1
          ΑΝ Απ[γραμμή, στήλη] < Απ[i ,j] Τότε
              i ← γραμμή
              j ← στήλη
          ΤΕΛΟΣ_ΑΝ
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ Ον[i], Ον[j]
  min ← Μέσος_Όρος(Απ, 1)
  θέση ← 1
  ΓΙΑ νησί ΑΠΟ 2 ΜΕΧΡΙ 15
      ΑΝ Μέσος_Όρος(Απ, νησί) < min ΤΟΤΕ
          θέση ← νησί
          min ← Μέσος_Όρος(Απ, νησί)
      ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ  Ον[θέση]
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ  Μέσος_Όρος(Π, θέση) : ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: γραμμή, στήλη, Π[15, 15], θέση, Σ
ΑΡΧΗ
  Σ ← 0
  ΓΙΑ στήλη ΑΠΟ 1 ΜΕΧΡΙ θέση – 1
      Σ ← Σ + Π[θέση, στήλη]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ γραμμή ΑΠΟ θέση+1 ΜΕΧΡΙ 15
      Σ ← Σ + Π[γραμμή, θέση]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  Μέσος_Όρος ← Σ / 14
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

| | | | | | | |
|---|---|---|---|---|---|---|
| **Φολέγανδρος** | - | - | - | - | - | - |
| **Ανάφη** | **43** | **-** | **-** | **-** | **-** | - |
| **Αστυπάλαια** | **68** | **31** | **-** | **-** | **-** | - |
| **Σίκινος** | **9** | **36** | **59** | **-** | **-** | - |
| **Κίμωλος** | **18** | **64** | **88** | **28** | **-** | - |
| **Τήλος** | **118** | **79** | **50** | **110** | **137** | - |

*Στο παραπάνω σχήμα φαίνονται οι αποστάσεις που πρέπει να προστεθούν στο Σ για να υπολογιστεί ο μέσος όρος των αποστάσεων της Σικίνου από τα υπόλοιπα νησιά, όπου **θέση = 4***

**4ο ΘΕΜΑ**

Δύο εταιρίες πρόκειται να συγχωνευτούν οπότε πρέπει να συγχωνεύσουν το μητρώο υπαλλήλων τους. Η πρώτη εταιρία διατηρεί σε έναν πίνακα 50x5 το μητρώο υπαλλήλων της ως εξής: στην 1η στήλη επώνυμο, στη 2η όνομα, στην τρίτη διεύθυνση κατοικίας, στην τέταρτη αριθμό δελτίου ταυτότητας και στην πέμπτη τηλέφωνο ενώ η δεύτερη εταιρία διατηρεί σε έναν πίνακα 30x5 το μητρώο υπαλλήλων της ως εξής: στην 1η στήλη επώνυμο, στη 2η όνομα, στην τρίτη διεύθυνση κατοικίας, στην τέταρτη τηλέφωνο και στην πέμπτη αριθμό δελτίου ταυτότητας.

Να αναπτύξετε πρόγραμμα το οποίο:

Α) να εισάγει σε δυο πίνακες το μητρώο κάθε εταιρίας

Β) να δημιουργήσει πίνακα 80x5 με το ενιαίο μητρώο της νέας εταιρίας.

Γ) να εμφανίζει τα στοιχεία του νέου ενιαίου μητρώου αλφαβητικά ως προς τα επώνυμα.

Να χρησιμοποιήσετε δυο διαδικασίες, μια για την αντιμετάθεση δυο στοιχείων (να κληθεί 2 φορές από το πρόγραμμα) και μια για συγχώνευση πινάκων.

**<u>ΛΥΣΗ</u>**

```
ΠΡΟΓΡΑΜΜΑ ΕΤΑΙΡΙΕΣ
ΜΕΤΑΒΛΗΤΕΣ
    ΧΑΡΑΚΤΗΡΕΣ: Α[50,5], Β[30,5], Γ[80,5]
    ΑΚΕΡΑΙΕΣ: Ι,J,K
ΑΡΧΗ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
      ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
          ΔΙΑΒΑΣΕ Α[Ι,J]
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
     ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
          ΔΙΑΒΑΣΕ Β[Ι,J]
     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ! αντιμετάθεση 4ης με 5η στήλη του Β
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
          ΚΑΛΕΣΕ ΑΝΤΙΜΕΤΑΘΕΣΗ(Β[Ι,4], Β[Ι,5])
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ! Συγχώνευση πινάκων
  ΚΑΛΕΣΕ ΣΥΓΧΩΝΕΥΣΗ (Α,Β,Γ)
  ! ταξινόμηση ενιαίου μητρώου
  ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 80
     ΓΙΑ J ΑΠΟ 80 ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
          ΑΝ Γ[J-1, 1] > Γ[J, 1] ΤΟΤΕ
               ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 5
                    ΚΑΛΕΣΕ ΑΝΤΙΜΕΤΑΘΕΣΗ(Γ[J-1, K], Γ[J, K])
               ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
          ΤΕΛΟΣ_ΑΝ
     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 80
     ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
          ΓΡΑΨΕ Γ[Ι,J]
     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΝΤΙΜΕΤΑΘΕΣΗ (Χ, Υ)
ΜΕΤΑΒΛΗΤΕΣ
    ΧΑΡΑΚΤΗΡΕΣ: Χ , Υ, ΤΕΜΡ
ΑΡΧΗ
  ΤΕΜΡ ← Χ
  Χ ← Υ
  Υ ← ΤΕΜΡ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΣΥΓΧΩΝΕΥΣΗ (Α, Β, Γ)
ΜΕΤΑΒΛΗΤΕΣ
    ΧΑΡΑΚΤΗΡΕΣ: Α[50,5], Β[30,5], Γ[80,5]
    ΑΚΕΡΑΙΕΣ: Ι,J
ΑΡΧΗ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 50
     ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
          Γ[Ι,J] ← Α[Ι,J]
     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
     ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
          Γ[Ι+50, J] ← Β[Ι,J]
     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

**4ο ΘΕΜΑ**

Να αναπτύξετε προγραμμα το οποίο :

Α) να τοποθετεί σε δύο μονοδιάστατους πίνακες τα επώνυμα και τα εταιρικά τηλέφωνα 10 συνεργατών μιας εταιρίας (θεωρήστε ότι δεν υπάρχουν συνωνυμίες).

Β) να τοποθετεί σε 4 πίνακες τα στοιχεία των κλήσεων των 10 συνεργατών κατά τη διάρκεια ενός μήνα ως εξής: i) τηλέφωνα αποστολέα ii) τον 10ψηφιο αριθμό προορισμού iii) την πληροφορία αν πρόκειται για κλήση (1) ή για μήνυμα (0) και iv) το χρόνο κλήσης σε δευτερόλεπτα (0 για μήνυμα), σταματώντας την εισαγωγή στοιχείων όταν δοθεί ως κινητό αποστολέα (συνεργάτη) το -1.

Γ) να διαβάζει το επώνυμο ενός υπαλλήλου και να εμφανίζει το συνολικό χρόνο κλήσεων του και τον αριθμό των μηνυμάτων κατά τη διάρκεια του μήνα.

Δ) να δημιουργεί νέο πίνακα με τις συνολικές μηνιαίες χρεώσεις των 10 συνεργατών λαμβάνοντας υπόψη τα εξής: i) χρέωση μηνύματος 0,1 € ii) χρέωση προς κινητό (αριθμός που αρχίζει από 6) 0,0015 €/sec iii) χρέωση προς σταθερό 0,0007 €/sec και iv) 8 € πάγιο

Ε) να εμφανίζει κατά αλφαβητική σειρά τα ονοματεπώνυμα των υπαλλήλων και τις συνολικές χρεώσεις του μήνα κάθε ενός.

**ΛΥΣΗ**

```
ΠΡΟΓΡΑΜΜΑ ΕΤΑΙΡΙΕΣ
ΜΕΤΑΒΛΗΤΕΣ
    ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[10], ΤΗΛ[10], ΤΗΛ_ΣΥΝ[10000],
                ΤΗΛ_ΠΡ[10000], ΕΠΩΝΥΜΟ, ΤΕΜΡ
    ΑΚΕΡΑΙΕΣ: Ι, J, ΠΛ[10000], ΧΡ[10000], ΠΛΗΘΟΣ, ΘΕΣΗ,
              ΠΛΗΘΟΣ_ΜΗΝ, ΣΥΝΟΛΟ_ΧΡΟΝΟΥ
    ΠΡΑΓΜΑΤΙΚΕΣ: ΧΡΕΩΣΗ[10], ΤΕΜΡ1
    ΛΟΓΙΚΕΣ: ΒΡΕΘΗΚΕ
ΑΡΧΗ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
     ΔΙΑΒΑΣΕ ΟΝ[Ι], ΤΗΛ[Ι]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ! Β ΕΡΩΤΗΜΑ
  Ι ← 1
  ΔΙΑΒΑΣΕ ΤΗΛ_ΣΥΝ[Ι], ΤΗΛ_ΠΡ[Ι], ΠΛ[Ι], ΧΡ[Ι]
  ΌΣΟ ΤΗΛ_ΣΥΝ[Ι]<> '-1' ΕΠΑΝΑΛΑΒΕ
      Ι ← Ι+1
     ΔΙΑΒΑΣΕ ΤΗΛ_ΣΥΝ[Ι], ΤΗΛ_ΠΡ[Ι], ΠΛ[Ι], ΧΡ[Ι]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΠΛΗΘΟΣ ← Ι-1
  ! Γ ΕΡΩΤΗΜΑ
  ΔΙΑΒΑΣΕ ΕΠΩΝΥΜΟ
  ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ
  ΘΕΣΗ ← 0
  Ι ← 1
  ΌΣΟ Ι<=10 ΚΑΙ ΒΡΕΘΗΚΕ= ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
     ΑΝ ΟΝ[Ι]=ΕΠΩΝΥΜΟ ΤΟΤΕ
          ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ
          ΘΕΣΗ ← Ι
     ΑΛΛΙΩΣ
          Ι ← Ι+1
     ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΝ ΒΡΕΘΗΚΕ=ΑΛΗΘΗΣ ΤΟΤΕ
     ΣΥΝΟΛΟ_ΧΡΟΝΟΥ ← 0
     ΠΛΗΘΟΣ_ΜΗΝ ← 0
     ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ ΠΛΗΘΟΣ
          ΑΝ ΤΗΛ[ΘΕΣΗ]=ΤΗΛ_ΣΥΝ[Ι] ΤΟΤΕ
             ΣΥΝΟΛΟ_ΧΡΟΝΟΥ ← ΣΥΝΟΛΟ_ΧΡΟΝΟΥ +ΧΡ[Ι]
             ΑΝ ΠΛ[Ι]=0 ΤΟΤΕ
                  ΠΛΗΘΟΣ_ΜΗΝ ← ΠΛΗΘΟΣ_ΜΗΝ + 1
             ΤΕΛΟΣ_ΑΝ
          ΤΕΛΟΣ_ΑΝ
     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
     ΓΡΑΨΕ ΣΥΝΟΛΟ_ΧΡΟΝΟΥ, ΠΛΗΘΟΣ_ΜΗΝΥΜΑΤΩΝ
  ΤΕΛΟΣ_ΑΝ
  ! Δ ΕΡΩΤΗΜΑ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
     ΧΡΕΩΣΗ[Ι] ← 8
     ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ ΠΛΗΘΟΣ
          ΑΝ ΠΛ[J] =1 ΤΟΤΕ
             ΑΝ ΤΗΛ_ΠΡ[J]DIV10^9=6 ΤΟΤΕ
                  ΧΡΕΩΣΗ[Ι] ← ΧΡΕΩΣΗ[Ι]+ΧΡ[J] * 0,0015
             ΑΛΛΙΩΣ
                  ΧΡΕΩΣΗ[Ι] ← ΧΡΕΩΣΗ[Ι]+ΧΡ[J] * 0,0007
             ΤΕΛΟΣ_ΑΝ
          ΑΛΛΙΩΣ
             ΧΡΕΩΣΗ[Ι] ← ΧΡΕΩΣΗ[Ι]+ 0,1
          ΤΕΛΟΣ_ΑΝ
     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ! Ε ΕΡΩΤΗΜΑ
  ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ 10
     ΓΙΑ J ΑΠΟ 10 ΜΕΧΡΙ Ι ΜΕ_ΒΗΜΑ -1
          ΑΝ ΟΝ[J-1]> ΟΝ[J] ΤΟΤΕ
             ΤΕΜΡ ← ΟΝ[J-1]
             ΟΝ[J-1] ← ΟΝ[J]
             ΟΝ[J] ← ΤΕΜΡ
             ΤΕΜΡ ← ΤΗΛ[J-1]
             ΤΗΛ[J-1] ← ΤΗΛ[J]
             ΤΗΛ[J] ← ΤΕΜΡ
             ΤΕΜΡ1 ← ΧΡΕΩΣΗ[J-1]
             ΧΡΕΩΣΗ[J-1] ← ΧΡΕΩΣΗ[J]
             ΧΡΕΩΣΗ[J] ← ΤΕΜΡ1
          ΤΕΛΟΣ_ΑΝ
     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
     ΓΡΑΨΕ ΟΝ[Ι], ΧΡΕΩΣΗ[Ι]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

**☺ !!! ΕΥΧΟΜΑΣΤΕ ΕΠΙΤΥΧΙΑ !!! ☺**