

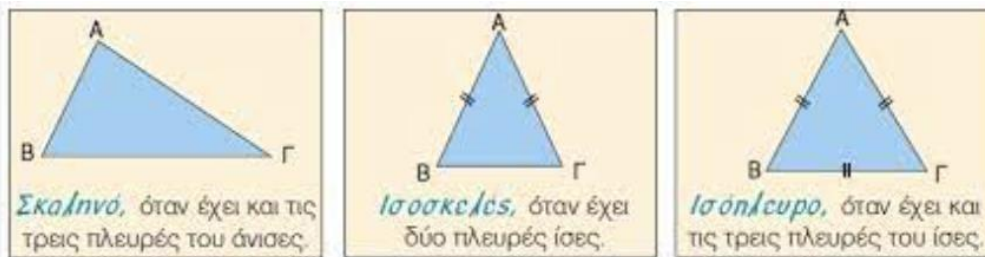
1η Σειρά Ασκήσεων (ασκήσεις επιλογής)

1. Αν κάποιος **φορολογούμενος** υποβάλλει τη δήλωσή του ηλεκτρονικά μέσω του Διαδικτύου, έχει έκπτωση 2.5% στον φόρο που του αναλογεί, με τον περιορισμό το ποσό της έκπτωσης να μην είναι μεγαλύτερο από 118€ (αν το ποσό της έκπτωσης είναι μεγαλύτερο από 118€ να προσαρμόζεται αυτομάτως στα 118€). Να γραφεί πρόγραμμα που διαβάζει το ποσό του φόρου (€) ενός φορολογούμενου και αν έκανε τη δήλωσή του ηλεκτρονικά (ΝΑΙ/ΟΧΙ) και υπολογίζει και εμφανίζει το φόρο που τελικά θα πληρώσει.
2. Να γραφεί πρόγραμμα το οποίο διαβάζει τρεις ακέραιους αριθμούς και ελέγχει κατά πόσο αυτοί είναι **τιμές μιας συνεχόμενης τριάδας ακεραίων**. π.χ. με είσοδο: 7, 5, 6 να εμφανίζει 'συνεχόμενοι'.
3. Ένα **τυπογραφείο** έχει το εξής κοστολόγιο:

Είδος εκτύπωσης	Επιφάνεια	Κόστος
Ασπρόμαυρη	μέχρι 250 τ. εκ.	0.35 € ανά τ. εκ.
	πάνω από 250 τ. εκ.	0.45 € ανά τ. εκ.
Έγχρωμη	μέχρι 150 τ. εκ.	0.55 € ανά τ. εκ.
	πάνω από 150 τ. εκ. μέχρι 850 τ. εκ.	0.70 € ανά τ. εκ.
	πάνω από 850 τ. εκ.	0.88 € ανά τ. εκ.

Να αναπτύξετε πρόγραμμα που θα διαβάζει το είδος και την επιφάνεια της εκτύπωσης και θα εμφανίζει το κόστος της με επιπλέον επιβάρυνση φόρου 4 %.

4. Τρεις θετικοί αριθμοί α, β, γ είναι πλευρές τριγώνου όταν καθένας από αυτούς είναι μικρότερος από το άθροισμα των άλλων δύο. Επιπλέον:



- αν έχει και τις τρεις πλευρές ίσες, τότε το τρίγωνο είναι ισόπλευρο
- αν έχει δύο μόνο πλευρές ίσες, τότε το τρίγωνο είναι ισοσκελές
- αν έχει και τις τρεις πλευρές διαφορετικές, τότε το τρίγωνο είναι σκαληνό

Να αναπτύξετε πρόγραμμα που θα διαβάζει τρεις αριθμούς και αν αποτελούν πλευρές τριγώνου, να εμφανίζει **το είδος του τριγώνου** αυτού, αλλιώς να εμφανίζει το μήνυμα «Οι αριθμοί αυτοί δεν αποτελούν τρίγωνο»

5. Ένα αυτοκίνητο κάνει **service** ανά 7500km. Κάνει μικρό και μεγάλο service εναλλάξ (ξεκινώντας με το μικρό). Να γραφεί πρόγραμμα το οποίο διαβάζει τα km που έχουν διανυθεί. Εμφανίζει σε πόσα km είναι το επόμενο service και τι είδους (μικρό/μεγάλο).

2η Σειρά Ασκήσεων (ασκήσεις επιλογής)

1. Το κόστος των **αεροπορικών εισιτηρίων** για το Κάιρο είναι 125 € το άτομο, ωστόσο το παιδικό εισιτήριο είναι 80 €. Κάθε άτομο μπορεί να μεταφέρει 15 κιλά αποσκευών, ωστόσο αν αθροιστικά υπάρχουν περισσότερα κιλά από το επιτρεπόμενο όριο τότε κάθε επιπλέον κιλό χρεώνεται με 1.5 €. Παρ' όλα αυτά αν τα έξοδα για τα παραπάνω ξεπεράσουν τα 450 € υπάρχει έκπτωση 9%. Τέλος, ο φόρος αεροδρομίου είναι 20 € το άτομο. Μια οικογένεια αποφάσισε να πάει διακοπές στο Κάιρο. Να γραφεί πρόγραμμα το οποίο διαβάζει το πλήθος των ενηλίκων, το πλήθος των παιδιών και το συνολικό βάρος των αποσκευών και να εμφανίζει τα χρήματα που πρέπει να καταβάλλει η οικογένεια.



2. Να γραφεί πρόγραμμα το οποίο διαβάζει τον αριθμό των αγοριών και των κοριτσιών που θα πάνε σε σχολική εκδρομή, θα ταξιδέψουν με πλοίο και θα διανυκτερεύσουν σε **4κλινες καμπίνες** (ξεχωριστά τα αγόρια και τα κορίτσια). Να υπολογίζει και να εμφανίζει τον αριθμό των 4κλινων καμπινών που απαιτούνται ξεχωριστά για τα αγόρια και για τα κορίτσια και συνολικά. Να εμφανίζει επιπλέον και τον αριθμό των ελεύθερων κλινών των τελευταίων 4κλινων καμπινών (αν υπάρχουν) ξεχωριστά για τα αγόρια και για τα κορίτσια.

```
Duse agoria
121
Duse koritsia
78
31 kampines agoriwn 20 kampines koritsiwn 51 synolika
Eleytheres klines agoriwn: 3
Eleytheres klines koritsiwn: 2
```

π.χ.

3. Στον προκριματικό γύρο για το **κύπελο ποδοσφαίρου** αγωνίζονται 2 ομάδες σε 2 αγώνες, έναν στην έδρα της κάθε ομάδας. Προκρίνεται εκείνη η ομάδα που έχει πετύχει τα περισσότερα συνολικά γκολ στους 2 αγώνες. Εάν τα συνολικά γκολ των 2 ομάδων είναι ίσα, τότε προκρίνεται η ομάδα εκείνη που έχει πετύχει τα περισσότερα γκολ εκτός έδρας. Εάν και αυτά είναι ίσα, οδηγούνται στην παράταση. Να γραφεί πρόγραμμα το οποίο δέχεται τα ονόματα των 2 ομάδων (ο1, ο2), τα γκολ που πέτυχαν εντός (εντ1, εντ2) και εκτός έδρας (εκτ1, εκτ2) και εμφανίζει την ομάδα που προκρίνεται, ή το μήνυμα "Παράταση".
4. Σε ένα δήμο, η **δημοτική εταιρεία ύδρευσης** χρεώνει κλιμακωτά, με βάση τον παρακάτω πίνακα:

Περιγραφή	Χρέωση	
	Κατανάλωση σε m ³	Χρέωση ανά m ³
Κόστος νερού	0 – 100	0.08 €
	101 – 500	0.12 €
	501 – 1000	0.16 €
	1001 – άνω	0.20 €
Τέλη ύδρευσης	60% (επί της χρεώσεως του νερού)	
Τέλη αποχέτευσης	80% (επί της χρεώσεως του νερού)	
Φ.Π.Α.	13% (επί της συνολικής χρεώσεως)	

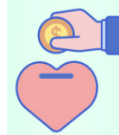
Να γραφεί πρόγραμμα που διαβάζει την κατανάλωση του συνδρομητή και εμφανίζει το ποσό που οφείλει ο συνδρομητής.

5. Το **Κοινοβούλιο** μιας χώρας έχει συνολικά 300 βουλευτές. Μία συνέλευση είναι σε απαρτία όταν είναι παρόντα τουλάχιστον τα 3/4 του συνόλου των βουλευτών. Για να υπερψηφιστεί μία πρόταση θα πρέπει περισσότερα από τα 2/3 των παρόντων βουλευτών να ψηφίσουν θετικά. Να γραφεί πρόγραμμα που διαβάζει τον αριθμό των παρόντων βουλευτών και αν η συνέλευση είναι σε απαρτία να διαβάζει τον αριθμό αυτών που ψήφισαν υπέρ της πρότασης. Να εμφανίζει κατάλληλο μήνυμα αν η πρόταση υπερψηφίστηκε ή καταψηφίστηκε ή αν υπήρξε έλλειψη απαρτίας.



3η Σειρά Ασκήσεων (ασκήσεις επανάληψης)

1. Να γραφεί πρόγραμμα που διαβάζει επαναληπτικά τα ποσά που προσφέρονται από δωρητές σε έναν **έρανο** μέχρι να δοθεί ως ποσό το -1. Να εμφανίζει α. το μέσο όρο των ποσών που προσφέρθηκαν β. το μεγαλύτερο ποσό που προσφέρθηκε γ. πόσοι δωρητές έδωσαν μεγαλύτερο ποσό από τον προηγούμενο από αυτούς δωρητή.



2. Μία **κάβα** πουλάει λευκό και κόκκινο κρασί. Να γραφεί πρόγραμμα το οποίο α) διαβάζει το αρχικό απόθεμα του λευκού και του κόκκινου κρασιού σε λίτρα και με έλεγχο εγκυρότητας ώστε να είναι θετικοί αριθμοί β) διαβάζει την τιμή του λίτρου κάθε είδους γ) επαναληπτικά και για κάθε παραγγελία πελάτη διαβάζει την ποσότητα σε λίτρα που επιθυμεί και το είδος του κρασιού ('Λ' για λευκό ή 'Κ' για κόκκινο) και αν υπάρχει αρκετό απόθεμα να εκτελεί την παραγγελία και να εμφανίζει το κόστος της. Να σταματάει όταν δοθεί παραγγελία που δεν μπορεί να εξυπηρετηθεί. δ) τελικά να εμφανίζει: τα συνολικά έσοδα και το σύνολο των λίτρων που πουλήθηκαν για κάθε είδος.

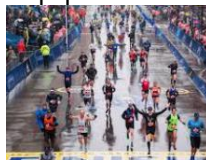


3. Να γραφεί πρόγραμμα που διαβάζει τις 1000 τυχαίες ρίψεις ενός **νομίσματος** ('Κ'/'Γ'). Να εμφανίζει την πλευρά ('Κ'/'Γ') που είχε το μεγαλύτερο σερί εμφανίσεων ή το μήνυμα 'Ισοπαλία'.



4. Να γραφεί πρόγραμμα το οποίο διαβάζει επαναληπτικά τους χρόνους τερματισμού των δρομέων ενός **μαραθωνίου** σε ώρες-λεπτά-δευτερόλεπτα με τυχαία σειρά τερματισμού και μέχρι να δοθεί ως χρόνος τερματισμού το: 0,0,0. Εάν η

πρώτη καταχώρηση χρόνου αφορά τον νικητή του περσινού μαραθώνιου δρόμου, να εμφανίζει τη σειρά κατάταξής του στον φετινό. Να θεωρήσετε ότι όλοι οι χρόνοι είναι διαφορετικοί.



5. Κάποια εταιρεία έχει καθορίσει τα εξής κριτήρια για την απονομή «δώρου» (**bonus**) στους πωλητές της: για πωλήσεις πάνω από 3000 € δίνεται δώρο 15% επί των πωλήσεων. Για πωλήσεις πάνω από 4500 € δίνεται δώρο 20% και για πωλήσεις πάνω από 6000 € δίνεται δώρο 25%. Να γραφεί πρόγραμμα που διαβάσει τις πωλήσεις που έχει κάνει ο πωλητής και εμφανίζει το δώρο που παίρνει. Το πρόγραμμα να σταματά όταν ο χρήστης επιλέξει ΟΧΙ σε μία ερώτηση του τύπου “Θέλεις να συνεχίσεις;” η οποία και να γίνεται μετά από το τέλος κάθε επανάληψης.



4η Σειρά Ασκήσεων (ασκήσεις επανάληψης)

1. Να γραφεί πρόγραμμα προσομοίωσης ενός μηχανήματος επαναληπτικής λήψης της τιμής της εξωτερικής θερμοκρασίας ενός χώρου και διακοπής της λειτουργίας του όταν διαβασθούν **10 συνεχόμενες ίδιες** θερμοκρασίες. Να εμφανίζει το συνολικό πλήθος των θερμοκρασιών που διαβάστηκαν.



2. Σε μία **δημοπρασία** συμμετέχουν 2 ενδιαφερόμενοι για έναν πίνακα ζωγραφικής που έχει τιμή εκκίνησης τα 10000€. Οι 2 ενδιαφερόμενοι κάνουν εναλλάξ προσφορές, κάθε μία από τις οποίες πρέπει να είναι κατά τουλάχιστον 10% προσαυξημένη σε σχέση με την προηγούμενη. Να γραφεί πρόγραμμα που δέχεται τα ονόματα των 2 ενδιαφερομένων και επαναληπτικά τις προσφορές τους, μέχρι να δοθεί σαν προσφορά η τιμή -1, οπότε και λήγει η δημοπρασία και ο πίνακας πωλείται σε εκείνον που έκανε την τελευταία έγκυρη προσφορά. Να εμφανίζει τον ενδιαφερόμενο που αγόρασε τελικά τον πίνακα και σε ποιά τιμή. Θεωρείστε ότι γίνεται τουλάχιστον μία προσφορά και γίνεται από τον 1^ο ενδιαφερόμενο.



3. Να γραφεί πρόγραμμα το οποίο διαβάσει έναν πραγματικό αριθμό μεγαλύτερο του 1 και εμφανίζει το **πλήθος των ψηφίων** του ακέραιου και του δεκαδικού του μέρους. Π.χ. για είσοδο της τιμής 354.19238 εμφανίζει: 3 ψηφία το ακέραιο μέρος και 5 το δεκαδικό.
4. Ένας **δήμος** προκηρύσσει **μειοδοτικό διαγωνισμό** για την κατασκευή ενός δρόμου. Να γραφεί πρόγραμμα το οποίο α) διαβάσει το μέγιστο κόστος ΜΕΚ (€) που μπορεί να διατεθεί και το μέγιστο αποδεκτό χρόνο παράδοσης του έργου ΜΕΧ (ημέρες) β) διαβάσει επαναληπτικά και για κάθε προσφορά: το όνομα της εταιρείας που κατεβάζει προσφορά, το κόστος της (€) και το χρόνο παράδοσης του έργου (ημέρες) μέχρι να δοθεί ως όνομα η λέξη 'ΤΕΛΟΣ' οπότε και να μην διαβάσει προσφορά και χρόνο παράδοσης. γ) εμφανίζει την εταιρεία που έκανε την καλύτερη προσφορά. Καλύτερη θεωρείται η προσφορά με το μικρότερο κόστος. Σε περίπτωση που δύο ή παραπάνω εταιρείες κατέβασαν προσφορά με το ίδιο μικρότερο κόστος να προτιμηθεί η εταιρεία με το μικρότερο χρόνο παράδοσης (έστω διαφορετικοί για όλες τις προσφορές). Τα παραπάνω να εμφανίζονται εάν η καλύτερη προσφορά είναι κόστους το πολύ όσο το ΜΕΚ και χρόνου παράδοσης το πολύ όσο το ΜΕΧ. Διαφορετικά να εμφανίζεται το μήνυμα 'Το έργο δεν ανατίθεται σε καμία εταιρεία'.



5. Ένα **πλοίο** έχει τη δυνατότητα μεταφοράς οχημάτων με μέγιστο όριο συνολικού βάρους τα 400.000 κιλά. Να γραφεί πρόγραμμα το οποίο διαβάσει επαναληπτικά το είδος ('Φ' για φορτηγό, 'Ε' για επιβατικό και 'Δ' για δίκυκλο) και το βάρος (σε κιλά) ενός οχήματος και επιτρέπει την είσοδό του, εφόσον το βάρος του δεν υπερβαίνει το διαθέσιμο υπόλοιπο του συνολικού βάρους. Διαφορετικά να τερματίζει τυτώνοντας το μήνυμα «ΤΕΛΟΣ ΕΙΣΟΔΟΥ». ΣΗΜΕΙΩΣΗ: Δεν απαιτείται έλεγχος εγκυρότητας για τα δεδομένα εισόδου. Να εμφανίζει: α) το συνολικό βάρος των οχημάτων που θα ταξιδέψουν. β) το πλήθος των φορτηγών, των επιβατικών και των δίκυκλων που θα ταξιδέψουν. γ) το διαθέσιμο βάρος που περίσσεψε, εφόσον υπάρχει, διαφορετικά το μήνυμα «ΠΛΗΡΟΤΗΤΑ 100%».



5η Σειρά Ασκήσεων (πίνακες τιμών - διαγράμματα ροής)

Δίνονται τα παρακάτω τμήματα προγράμματος:

<p>X ← 1 Όσο X<5 επανάλαβε A ← X+2 B ← 3*A-4 C ← B-A+4 Αν A > B τότε Αν A > C τότε MAX ← A Αλλιώς MAX ← C Τέλος αν αλλιώς Αν B > C τότε MAX ← B Αλλιώς MAX ← C Τέλος αν Τέλος αν Γράψε X, A, B, C, MAX X ← X+2 Τέλος επανάληψης</p>	<p>X ← K Y ← L Αν X < Y τότε TEMP ← X X ← Y Y ← TEMP Τέλος_αν Όσο Y<>0 επανάλαβε TEMP ← Y Y ← X MOD Y X ← TEMP Γράψε X, Y Τέλος_επανάληψης Y ← (K * L) DIV X Γράψε X, Y</p>	<p>A ← 1 B ← 1 N ← 0 M ← 2 Όσο B < 6 επανάλαβε X ← A + B Αν X MOD 2=0 τότε N ← N + 1 Αλλιώς M ← M + 1 Τέλος_αν A ← B B ← X Γράψε N, M, B Τέλος επανάληψης Γράψε X</p>	<p>D ← 2 για X από 2 μέχρι 5 μεβήμα 2 A ← 10 * X B ← 5 * X + 10 C ← A + B - (5 * X) D ← 3 * D - 5 Y ← A + B - C + D Τέλος_επανάληψης</p>	<p>x ← 1 i ← 1 ΑρχήΕπανάληψης y ← x * i Αν y mod 2 = 0 τότε x ← x + 2 z ← x + y Αλλιώς x ← x + 1 z ← x * y ΤέλοςΑν Γράψε x, y, z i ← i + 1 ΜέχριςΌτου x>7</p>
<p>Ποιες είναι οι τιμές των μεταβλητών X, A, B, C, MAX που θα εμφανιστούν κατά την εκτέλεση του παραπάνω τμήματος προγράμματος;</p>	<p>Για K = 24 και L = 40, ποιές θα είναι οι τιμές των μεταβλητών X, Y καθώς αυτές τυπώνονται με την εντολή Γράψε X, Y (τόσο μέσα στη δομή επανάληψης όσο και στο τέλος του προγράμματος);</p>	<p>Να γράψετε τις τιμές των μεταβλητών N, M και B, όπως αυτές τυπώνονται σε κάθε επανάληψη, και την τιμή της μεταβλητής X που τυπώνεται μετά το τέλος της επανάληψης.</p>	<p>Να βρείτε τις τιμές των μεταβλητών A, B, C, D, X και Y σε όλες τις επαναλήψεις.</p>	<p>Ποιές είναι οι τιμές των μεταβλητών x, y και z που θα εμφανισθούν κατά την εκτέλεση του παραπάνω τμήματος προγράμματος;</p>

Να σχεδιάσετε τα αντίστοιχα διαγράμματα ροής



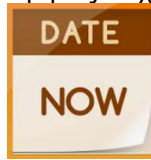
6η Σειρά Ασκήσεων (μονοδιάστατοι πίνακες)

1. Ένας **συγγραφέας** έχει γράψει 30 βιβλία. Να γραφεί πρόγραμμα που α) διαβάζει τους τίτλους των βιβλίων σε πίνακα T[30], καθώς και τα έτη έκδοσής τους σε πίνακα X[30]. β) διαβάζει δύο έτη X1 και X2. γ) εμφανίζει τους τίτλους των βιβλίων που εκδόθηκαν μεταξύ των ετών X1 και X2 δ) υπολογίζει και εμφανίζει το πλήθος των βιβλίων που βρέθηκαν στο ερώτημα γ.

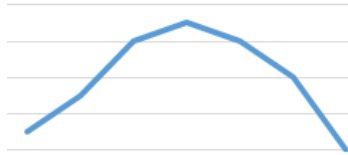


2. Να γραφεί πρόγραμμα που καταχωρεί στον πίνακα ΗΜΕΡ[12] τον αριθμό των ημερών (28/30/31) του κάθε μήνα ενός μη δίσεκτου έτους (365 ημερών) ως εξής: ο μήνας 2 έχει 28 ημέρες, οι μήνες 4,6,9,11 έχουν 30 ημέρες και οι υπόλοιποι 31.

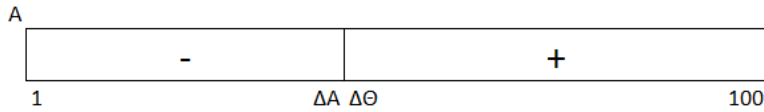
Διαβάζει τον αριθμό των ημερών που έχουν περάσει από την αρχή του έτους (1-365 με έλεγχο εγκυρότητας) και εμφανίζει την αντίστοιχη **ημερομηνία** σε μορφή: ημέρα/μήνας. Π.χ. για αριθμό των ημερών = 54 να εμφανίζει: 23/2.



3. Να γραφεί πρόγραμμα που διαβάζει τον ακέραιο A[100] (έστω με διαφορετικές τιμές). Να ελέγχει εάν οι τιμές του έχουν **“τα κοίλα άνω”**. Δηλ. α) οι τιμές του από το 1^ο κελί ως τη θέση μεγίστου είναι κατά αύξουσα σειρά και β) οι τιμές του από τη θέση μεγίστου ως το 100^ο κελί είναι κατά φθίνουσα σειρά.



4. Να γραφεί πρόγραμμα το οποίο διαβάζει στον ακέραιο A[100] τιμές έτσι ώστε **οι αρνητικές τιμές να καταχωρούνται από την αρχή** και από αριστερά προς τα δεξιά, ενώ **οι θετικές τιμές να καταχωρούνται από το τέλος** και από δεξιά προς τα αριστερά, μέχρι να γεμίσει πλήρως ο πίνακας. Οι μηδενικές τιμές να αγνοούνται. Τελικά να εμφανίζει με κατάλληλο μήνυμα ποιες τιμές ήταν οι περισσότερες (θετικές/αρνητικές) ή αν ήταν ίσες.



5. Να γραφεί πρόγραμμα που διαβάζει τον A[100] χαρακτήρων με τις σωστές απαντήσεις ενός **test 100 ερωτήσεων σωστού/λάθους (Σ/Λ)** και τον M[100] χαρακτήρων με τις απαντήσεις ενός διαγωνιζόμενου. Κάθε σωστή απάντηση μετράει: για τις ερωτήσεις 1-20 από 1 μονάδα, για τις ερωτήσεις 21-40 από 2 μονάδες, για τις ερωτήσεις 41-60 από 3 μονάδες, για τις ερωτήσεις 61-80 από 4 μονάδες, για τις ερωτήσεις 81-100 από 5 μονάδες, ενώ κάθε λάθος απάντηση μετράει για τις ερωτήσεις 1-20 από -1 μονάδα, για τις ερωτήσεις 21-40 από -2 μονάδες, για τις ερωτήσεις 41-60 από -3 μονάδες, για τις ερωτήσεις 61-80 από -4 μονάδες, για τις ερωτήσεις 81-100 από -5 μονάδες. Να υπολογίζει και να εμφανίζει τη βαθμολογία του (από -300 ως 300) και να τον χαρακτηρίζει: 'άριστος' με βαθμολογία: [280,300], 'πολύ καλός': [250,279], 'καλός': [200,249], 'μέτριος': [180,199], 'κακός': [-300,179].

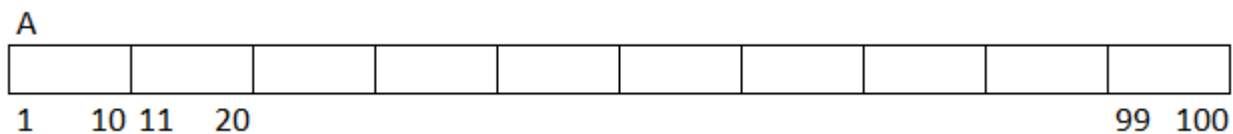


7η Σειρά Ασκήσεων (μονοδιάστατοι πίνακες)

1. Να γραφεί πρόγραμμα που διαβάζει τους πίνακες ακεραίων A[100] και B[100]. Να ελέγχει εάν **όλες** οι τιμές του ενός πίνακα είναι **μεγαλύτερες** από **όλες** τις τιμές του άλλου (και όχι 1-1).
2. Να γραφεί πρόγραμμα το οποίο: α) διαβάζει στον ακέραιο K[10] τις ενδείξεις του χιλιομετρική (**κοντέρ**) ενός αυτοκινήτου για τα 10 πρώτα έτη της λειτουργίας του (π.χ. στο K[3] η ένδειξη του χιλιομετρική στο τέλος του 3ου έτους), με έλεγχο εγκυρότητας έτσι ώστε κάθε τιμή να είναι μεγαλύτερη ή ίση της προηγούμενης. Για κάθε λανθασμένη καταχώρηση να εμφανίζει κατάλληλο μήνυμα β) εμφανίζει το μικρότερο και το μεγαλύτερο σύνολο χιλιομέτρων που διανύθηκαν σε ένα έτος. γ) ελέγχει εάν υπήρξε έτος κατά το οποίο το αυτοκίνητο παρέμεινε ακινητοποιημένο δ) εμφανίζει το μέσο ετήσιο αριθμό χιλιομέτρων που διένυσε το αυτοκίνητο. Παρατήρηση: το αυτοκίνητο αγοράστηκε με το κοντέρ μηδενισμένο.



3. Να γραφεί πρόγραμμα που διαβάζει τον ακέραιο A[100]. Να ελέγχει εάν τα **10 αθροίσματα** των κελιών του στις περιοχές 1-10, 11-20, ... , 91-100 είναι μεταξύ τους ίσα.



4. Να γραφεί πρόγραμμα το οποίο διαβάζει τον ακέραιο A[100] και έναν ακέραιο Sum και εμφανίζει τα **ζεύγη των στοιχείων** (όχι υποχρεωτικά διαδοχικών) του με άθροισμα ίσο με τον Sum. Αν δεν υπάρχει κανένα τέτοιο ζεύγος να εμφανίζει κατάλληλο μήνυμα.
5. Να γραφεί πρόγραμμα που διαβάζει τους πίνακες ακεραίων A[100] και B[200]. Να βρίσκει και να εμφανίζει τη **συνολική μέγιστη και την ελάχιστη** τιμή των δύο πινάκων (σαν να ήταν ένας πίνακας).

πόντων τους σε όλα τα παιχνίδια και για όσους εντοπισθούν στον πίνακα O[10]. Θεωρείστε ότι τα ονόματα είναι διαφορετικά.



15	BLOCK	21
11	TURNOVER	7
23	STEAL	25
45	ASSIST	51
32	FOULS DRAWN	31
43	FB SUCCESS	27
13/21	FB FAILED	11/17
4	FREE THROW	7

12^η Σειρά Ασκήσεων (πίνακες)

1. Να γραφεί πρόγραμμα το οποίο διαβάζει στον M[1000] (ακέрайος) τους μηνιαίους μισθούς 1000 υπαλλήλων. Να εμφανίζει τον «**πιο συχνό**» μισθό (αυτόν που εμφανίζεται τις περισσότερες φορές) και πόσοι υπάλληλοι τον έχουν. Θεωρείστε ότι είναι μοναδικός.



2. Να γραφεί πρόγραμμα το οποίο διαβάζει τον ακέрайο πίνακα A[100] και δημιουργεί νέο πίνακα όπου θα έχουν **διαχωριστεί** τα μη μηδενικά με τα μηδενικά στοιχεία. Παρατήρηση: πρώτα να τοποθετηθούν τα μηδενικά και να μην αλλοιωθεί η διάταξη των αριθμών.
3. Στις μαθητικές **εκλογές** του σχολείου σας, αναλάβετε να φτιάξετε ένα πρόγραμμα για την ευκολότερη εξαγωγή των αποτελεσμάτων. Το σχολείο σας έχει συνολικά 500 παιδιά, εκ των οποίων έθεσαν υποψηφιότητα τα 32. Δεδομένου ότι ο κάθε υποψήφιος έχει έναν κωδικό αριθμό από το 1 έως το 32, το πρόγραμμά σας θα πρέπει να λειτουργεί ως εξής: α) να διαβάζει για καθέναν από τους ψηφοφόρους τον κωδικό αριθμό του υποψηφίου που ψήφισε. β) να υπολογίζει και να εμφανίζει τον κωδικό αριθμό του υποψηφίου που πήρε τις περισσότερες ψήφους. (δεν υπάρχει ισοβαθμία και κάθε ψηφοφόρος ψηφίζει έναν μόνο υποψήφιο).



4. Να γραφεί πρόγραμμα το οποίο διαβάζει στον πίνακα χαρακτήρων Σ[100] 100 σύμβολα (γράμματα, ψηφία κλπ.) και στον παράλληλο πίνακα χαρακτήρων Κ[100] την αντίστοιχη κρυπτογραφημένη τιμή τους. Διαβάζει στον M[1000] ένα μήνυμα 1000 χαρακτήρων. Να γεμίζει τον KM[1000] με το αρχικό μήνυμα, **κρυπτογραφημένο**. Θεωρείστε ότι τα σύμβολα του αρχικού μηνύματος περιέχονται όλα στον Σ[100].



5. Να γραφεί πρόγραμμα το οποίο **διαχειρίζεται έναν ακέрайο A[10]**. Αρχικά ο πίνακας είναι άδειος οπότε και να εκχωρεί στην ακέрайα μεταβλητή-στάθμη N την τιμή μηδέν(0). Με κατάλληλο επαναληπτικό μενού να εκτελεί τις παρακάτω λειτουργίες: 1. **Προσθήκη** 2. **Παρεμβολή** 3. **Διαγραφή** 4. **Έξοδος**. Αν επιλεγθεί το 1 (Προσθήκη) και εφόσον έχει ελεύθερο χώρο, να διαβάζει μια νέα τιμή στο τέλος του πίνακα. Αν επιλεγθεί το 2 (Παρεμβολή) και εφόσον έχει στοιχεία και ελεύθερο χώρο, να διαβάζει τη θέση παρεμβολής (με έλεγχο εγκυρότητας ώστε να είναι από 1 έως N) και μια νέα τιμή την οποία και να παρεμβάλει στη δοθείσα θέση ολισθαίνοντας τα επόμενα στοιχεία μία θέση δεξιά. Αν επιλεγθεί το 3 (Διαγραφή) και εφόσον έχει στοιχεία, να διαβάζει τη θέση του υπό διαγραφή στοιχείου (με έλεγχο εγκυρότητας ώστε να είναι από 1 έως N) και να το διαγράφει ολισθαίνοντας τα επόμενα στοιχεία μία θέση αριστερά. Σε κάθε επανάληψη να εμφανίζει τα στοιχεία του πίνακα.



13^η Σειρά Ασκήσεων (πίνακες)

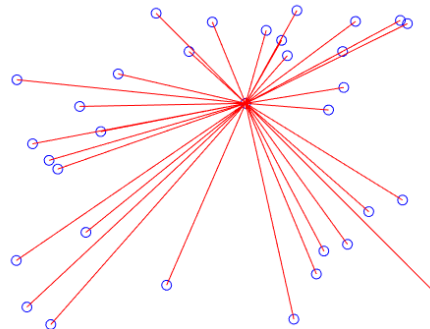
- Μια εταιρεία 200 εργαζομένων θέλει να υπολογίσει **στατιστικά** στοιχεία σχετικά με το μισθό, το φύλο και την ηλικία των εργαζομένων της. Να γραφεί πρόγραμμα το οποίο α) διαβάζει τα παραπάνω στοιχεία (μισθός, φύλο, ηλικία) και θα ελέγχει την αξιοπιστη καταχώρησή τους (φύλο = 'Α' ή 'Γ' και μισθός και ηλικία θετικοί αριθμοί) β) εμφανίζει τη μέση ηλικία των εργαζομένων που παίρνουν πάνω από 1000 € το μήνα γ) εμφανίζει το ποσοστό των εργαζομένων που είναι μικρότεροι των 35 ετών και έχουν μισθό μεγαλύτερο από το μέσο μισθό που πληρώνει η εταιρεία δ) εμφανίζει το ποσοστό των αντρών και των γυναικών στους 30 πιο καλοπληρωμένους εργαζόμενους.



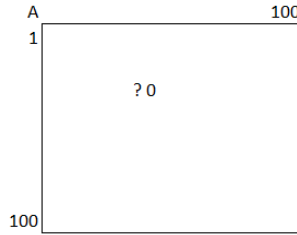
- Σε ένα **νέο αγώνισμα ακοντισμού** συμμετέχουν 20 αθλητές με 10 ρίψεις ο καθένας και τον εξής κανονισμό: νικητής αναδεικνύεται εκείνος που έκανε **την καλύτερη βολή σε κάθε ρίψη (1-10) τις περισσότερες φορές**. Να γραφεί πρόγραμμα το οποίο διαβάζει σε κατάλληλους πίνακες τα ονόματα και τις ρίψεις των αθλητών. Να εμφανίζει το όνομα ή τα ονόματα των νικητών.



- Στο νησιωτικό σύμπλεγμα των **Κυκλάδων** ανήκουν 33 νησιά. Να γραφεί πρόγραμμα το οποίο: α. διαβάζει σε πίνακα $O[33]$ τα ονόματα των νησιών. β. διαβάζει σε πίνακα $A[33, 33]$ τις αποστάσεις μεταξύ των νησιών (μόνο στα στοιχεία κάτω της 1ης κύριας διαγωνίου) γ. καταχωρεί τις ίδιες αποστάσεις στα συμμετρικά στοιχεία άνω της 1ης κύριας διαγωνίου δ. υπολογίζει και εμφανίζει το νησί (έστω ένα) που έχει τον ελάχιστο μέσο όρο αποστάσεων από τα άλλα νησιά έτσι ώστε να τοποθετηθεί εκεί ένας τηλεοπτικός **αναμεταδότης**. Στη συνέχεια να διαβάζει την εμβέλεια του αναμεταδότη και να ελέγχει εάν μπορεί να καλυφθεί και το πιο απομακρυσμένο νησί από το νησί με τον αναμεταδότη.



- Να γραφεί πρόγραμμα το οποίο διαβάζει τον ακέραιο $A[100]$. Να ελέγχει εάν όλες οι τιμές που περιέχει είναι **ίσες μεταξύ τους**.
- Να γραφεί πρόγραμμα το οποίο διαβάζει στον ακέραιο πίνακα $A[100, 100]$ τις τιμές 1-10.000 σε τυχαίες θέσεις, εκτός από μία τιμή σε ένα κελί που θα διαβάσει την τιμή 0. Να εντοπίζει το κελί αυτό και να καταχωρεί σε αυτό την **τιμή που λείπει**.

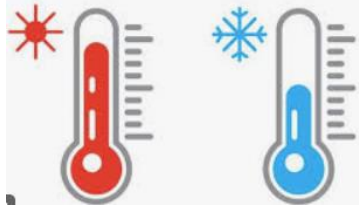


14^η Σειρά Ασκήσεων (διαδικασίες - συναρτήσεις)

Να σχηματίσετε τους πίνακες τιμών των παρακάτω προγραμμάτων. Τί θα εμφανίσουν;				
ΠΡΟΓΡΑΜΜΑ ΠΤ1 ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: Α, Β, Γ ΑΡΧΗ Α ← 3 Β ← 13 Γ ← 2 ΓΡΑΨΕ Α, Β, Γ ΚΑΛΕΣΕ Διαδ (Β, Γ) ΓΡΑΨΕ Α, Β, Γ ΚΑΛΕΣΕ Διαδ (Γ, Α) ΓΡΑΨΕ Α, Β, Γ ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ! ===== ΔΙΑΔΙΚΑΣΙΑ Διαδ (α1, α2) ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: α1, α2 ΑΡΧΗ α1 ← α1 DIV 2 α2 ← α2 ^ 3 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	ΠΡΟΓΡΑΜΜΑ ΠΤ2 ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: x, y ΑΡΧΗ x ← 5 y ← 13 ΚΑΛΕΣΕ Διαδ (x, y) ΓΡΑΨΕ x, y ΚΑΛΕΣΕ Διαδ (y, x) ΓΡΑΨΕ x, y ΚΑΛΕΣΕ Διαδ (x, y) ΓΡΑΨΕ x, y ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ! ===== ΔΙΑΔΙΚΑΣΙΑ Διαδ (α1, α2) ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: α1, α2 ΑΡΧΗ ΑΝ (α1 MOD 2 = 0) ΤΟΤΕ α1 ← α1 - 3 α2 ← α2 ^ 2 ΑΛΛΙΩΣ α1 ← α1 ^ 2 α2 ← α2 + 5 ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	ΠΡΟΓΡΑΜΜΑ ΠΤ3 ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: Α, Β, Γ ΑΡΧΗ Α ← 27 Β ← 2 Γ ← 13 ΚΑΛΕΣΕ Διαδ (Α, Β, Γ) ΓΡΑΨΕ Α, Β, Γ ΚΑΛΕΣΕ Διαδ (Γ, Α, Β) ΓΡΑΨΕ Α, Β, Γ ΚΑΛΕΣΕ Διαδ (Β, Γ, Α) ΓΡΑΨΕ Α, Β, Γ ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ! ===== ΔΙΑΔΙΚΑΣΙΑ Διαδ (α1, α2, α3) ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: α1, α2, α3 ΑΡΧΗ α3 ← α1 + α2 - α3 α2 ← α2 - α1 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	ΠΡΟΓΡΑΜΜΑ ΠΤ4 ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: Α, Β ΑΡΧΗ Α ← 2 Β ← 19 ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ΚΑΛΕΣΕ Διαδ (Β, Α) ΓΡΑΨΕ Α, Β ΜΕΧΡΙΣ_ΟΤΟΥ (Α > Β) ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ! ===== ΔΙΑΔΙΚΑΣΙΑ Διαδ (α1, α2) ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: α1, α2 ΑΡΧΗ α1 ← α1 - 2 α2 ← α2 + 5 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	ΠΡΟΓΡΑΜΜΑ ΠΤ5 ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: α, β, αποτ ΑΡΧΗ α ← 2 β ← 13 αποτ ← Υπ(α, β) Γράψε αποτ α ← 5 * α + αποτ β ← Α_Τ(αποτ - 20) αποτ ← Υπ(β, α) - 3 Γράψε α, β, αποτ αποτ ← Υπ(α, β) Γράψε αποτ ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ! ===== Συνάρτηση Υπ(x, y): Ακέραιο ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: x, y, π ΑΡΧΗ π ← (x + y) div x Υπ ← 2 - π ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

15^η Σειρά Ασκήσεων (διαδικασίες - συναρτήσεις)

- α) Να γραφεί η διαδικασία Ταξινόμηση(A, αρχή, τέλος) η οποία δέχεται σαν παραμέτρους έναν ακέραιο πίνακα A[360] και 2 δείκτες αρχή και τέλος. Να ταξινομεί κατά αύξουσα σειρά την υποπεριοχή του A που οριοθετείται από τους δείκτες αρχή και τέλος. β) Να γραφεί η διαδικασία Εμφάνιση(A, αρχή, τέλος) η οποία δέχεται σαν παραμέτρους έναν ακέραιο πίνακα A[360] και 2 δείκτες αρχή και τέλος. Να εμφανίζει στην οθόνη την υποπεριοχή του A που οριοθετείται από τους δείκτες αρχή και τέλος. γ) Να γραφεί πρόγραμμα το οποίο δεδομένου του πίνακα Θ[360] με τις **θερμοκρασίες** μιας πόλης στις 12:00 το μεσημέρι για κάθε ημέρα ενός έτους (1 μήνας = 30 ημέρες) και με τη βοήθεια των παραπάνω υποπρογραμμάτων, εκτελεί τα παρακάτω: i. εμφανίζει τις 3 πιο χαμηλές θερμοκρασίες του Ιανουαρίου ii. εμφανίζει τις 10 πιο υψηλές θερμοκρασίες του καλοκαιριού iii. εμφανίζει τις 5 πιο χαμηλές και τις 5 πιο υψηλές θερμοκρασίες της Άνοιξης.



- α) Να γραφεί συνάρτηση η οποία δέχεται σαν παραμέτρους έναν πραγματικό πίνακα A[20, 360], 2 έγκυρους αριθμούς γραμμών γ1, γ2 και 2 έγκυρους αριθμούς στηλών σ1, σ2. Να υπολογίζει το **μέσο όρο της περιοχής** των κελιών που ορίζουν οι παραπάνω γραμμές και στήλες. β) Να γραφεί πρόγραμμα που διαβάζει σε κατάλληλο πίνακα τις εισπράξεις 20 πωλητών για τις 360 ημέρες ενός έτους (να θεωρηθεί ότι κάθε μήνας έχει 30 ημέρες) και με χρήση της παραπάνω συνάρτησης υπολογίζει και εμφανίζει: i. το μέσο όρο των πωλήσεων των 10 πρώτων πωλητών για το μήνα Απρίλιο, ii. το μέσο όρο των πωλήσεων του 12^{ου} πωλητή για την καλοκαιρινή περίοδο, iii. το μέσο όρο των πωλήσεων των 5 τελευταίων πωλητών για το πρώτο 10ήμερο του Αυγούστου.

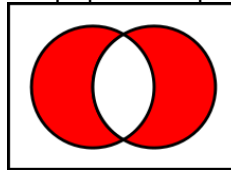


- Έστω ο ακέραιος A[500, 1000] που συμβολίζει μία έκταση 500 x 1000 μέτρων προς πώληση. Να γραφεί κύριο πρόγραμμα το οποίο: α) αρχικοποιεί τον A με μηδέν(0) υποδηλώνοντας ότι αρχικά όλη η έκταση είναι διαθέσιμη προς

πώληση β) επαναληπτικά: (i) ζητάει τις διαστάσεις ενός οικοπέδου που επιθυμεί να αγοράσει κάποιος (γρ x στ) με έλεγχο εγκυρότητας ώστε $1 \leq \gamma\rho \leq 500$ και $1 \leq \sigma\tau \leq 1000$ (ii) καλεί τη διαδικασία Πώληση(A, γρ, στ, ok) η οποία ψάχνει το **1ο διαθέσιμο οικόπεδο** διαστάσεων γρ x στ και αν το βρει, το διαθέτει εκχωρώντας σε όλα τα στοιχεία του την τιμή ένα(1) και επιστρέφει μέσω της λογικής παραμέτρου ok την τιμή Αληθής. Διαφορετικά, επιστρέφει μέσω της λογικής παραμέτρου ok την τιμή Ψευδής. Σε κάθε περίπτωση η διαδικασία εμφανίζει το μήνυμα "Πωλήθηκε" ή "Μη διαθέσιμο". Να σταματάει όταν έχει πωληθεί τουλάχιστον το 80% της συνολικής έκτασης. γ) τελικά να εμφανίζει (i) το % ποσοστό της έκτασης που πωλήθηκε (ii) τον μέσο όρο της έκτασης των οικοπέδων που πωλήθηκαν. Να γράψετε τη διαδικασία Πώληση.



4. Η **συμμετρική διαφορά** δύο συνόλων A, B ορίζεται ως τα στοιχεία του A που δεν ανήκουν στο B και τα στοιχεία του B που δεν ανήκουν στο A. π.χ. $A = \{1,2,3,4,5\}$ $B = \{4,5,6,7,8\}$ => συμμετρική διαφορά = $\{1,2,3,6,7,8\}$. Να γραφεί πρόγραμμα που διαβάζει τους πίνακες A[5] και B[5] και εμφανίζει τη συμμετρική διαφορά τους καθώς και το πλήθος των στοιχείων της. Υπόδειξη: φτιάξτε κατάλληλη διαδικασία που δέχεται ως παραμέτρους 2 ακέραιους πίνακες 5 στοιχείων και εμφανίζει τα στοιχεία του 1ου που δεν ανήκουν στον 2ο και επιστρέφει το πλήθος τους. Καλέστε την 2 φορές.

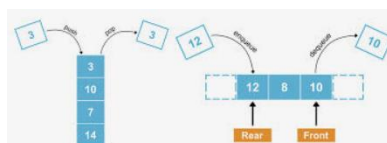


5. Να γράψετε πρόγραμμα το οποίο προετοιμάζει μία νέα πίστα για το παιχνίδι «**Ναρκαλιευτής**» ως εξής: α) να αρχικοποιεί έναν ακέραιο A[10, 10] με μηδενικά. β) να τοποθετεί στον A 20 «νάρκες» (τιμή -1) σε τυχαίες θέσεις. Θεωρείστε τη συνάρτηση ΤΥΧ(α, β) η οποία επιστρέφει μία τυχαία τιμή στο διάστημα [α, β]. Για κάθε μία από τις 20 νάρκες να καλεί επαναληπτικά τη συνάρτηση ΤΥΧ μία φορά για να λάβει μία τυχαία συντεταγμένη γραμμής (1-10) και μία φορά για να λάβει μία τυχαία συντεταγμένη στήλης (1-10) μέχρι να καθορισθεί ένα κελί του πίνακα A στο οποίο δεν έχει ξανατοποθετηθεί νάρκη. γ) στα υπόλοιπα κελιά του πίνακα A που δεν περιέχουν νάρκη, να τοποθετεί μία τιμή (0-8) που ισοδυναμεί με τον αριθμό των γειτονικών (οριζόντια ή κάθετα ή διαγώνια) κελιών που περιέχουν νάρκη. Για το σκοπό αυτό φτιάξτε την ακέραια συνάρτηση «Νάρκες» με παραμέτρους τις ακέραιες i, j και έναν ακέραιο A[10, 10]. Για το κελί A[i, j] με τιμή ≠ -1 να επιστρέφει τον αριθμό των γειτονικών κελιών με νάρκες (τιμή -1).



16^η Σειρά Ασκήσεων (στοίβα + ουρά + υποπρογράμματα)

1. Να γραφεί πρόγραμμα το οποίο αρχικοποιεί **στοίβα** Σ[100] και **ουρά** Ο[200] ως άδειες και μέσω κατάλληλου **μενού** εκτελεί τα παρακάτω: 1. Ωθηση, 2. Απώθηση, 3. Εισαγωγή, 4. Εξαγωγή, 5. Έξοδος. Τελικά να εμφανίζει: πλήθος υπερχειλίσεων, υποχειλίσεων, αποτυχημένων εισαγωγών, αποτυχημένων εξαγωγών και πλήθος στοιχείων της στοίβας και της ουράς.



2. Να γραφεί η συνάρτηση DaysOfMonth(m, y): Ακέραια η οποία δέχεται τον αριθμό έναν μήνα και ένα έτος (ακέραιοι) και επιστρέφει τις ημέρες του μήνα. Οι μήνες 4,6,9,11 έχουν 30 ημέρες. Ο Φεβρουάριος έχει 28 ημέρες για τα μη δίσεκτα έτη και 29 για τα δίσεκτα (έτη που είναι πολλαπλάσια του 4 και όχι του 100 ή είναι πολλαπλάσια του 400). Όλοι οι υπόλοιποι μήνες έχουν 31 ημέρες. Να γραφεί η διαδικασία ReadDate(d, m, y) που διαβάζει μία **ημερομηνία** (ημέρα d, μήνας m, έτος y: ακέραιοι) **με έλεγχο εγκυρότητας** ώστε ο μήνας m να ανήκει στο [1,12] και η ημέρα d στο [1, ημΜην], όπου ημΜην: οι ημέρες του μήνα (να υπολογίζονται με τη συνάρτηση DaysOfMonth). Να γραφεί πρόγραμμα που **διαβάζει 2 έγκυρες ημερομηνίες** καλώντας τη διαδικασία ReadDate **και τις συγκρίνει** (μεγαλύτερη θεωρείται η μεταγενέστερη).



3. Σε έναν σταθμό **διοδίων** λειτουργεί μία πύλη και εξυπηρετεί σε **ουρά** 1 όχημα ανά $\frac{1}{2}$ λεπτό. Να γραφεί πρόγραμμα το οποίο για κάθε λεπτό ενός 24ώρου: α) εξάγει 2 οχήματα (αν υπάρχουν) β) διαβάσει τον αριθμό των νέων οχημάτων που προσέρχονται (≥ 0) και για κάθε όχημα εισάγει τον αριθμό κυκλοφορίας του γ) εμφανίζει πόσα και ποια οχήματα είναι στην ουρά. Τελικά να εμφανίζει πόσα οχήματα εξυπηρετήθηκαν συνολικά. Το πρόγραμμα να χρησιμοποιεί την αλφαριθμητική ουρά $O[1000]$ και τις διαδικασίες Εισαγωγή(O , front, rear, x) και Εξαγωγή(O , front, rear, x, s) τις οποίες και να υλοποιήσετε. (O η ουρά, front και rear οι ακέραιοι δείκτες της, x αριθμός κυκλοφορίας που εισάγεται/εξάγεται και s η λογική παράμετρος που επιστρέφεται με την τιμή Αληθής/Ψευδής ανάλογα με το αν η Εξαγωγή έγινε με επιτυχία).



4. Να γραφεί πρόγραμμα που προσομοιώνει τη λειτουργία μίας **ουράς σε τράπεζα** με ένα ταμείο ως εξής: η ουρά να ξεκινάει περιέχοντας τον αριθμό προτεραιότητας 1 ο οποίος και εξυπηρετείται. Στην ουρά εισάγεται ένας νέος πελάτης ανά 2' με αριθμό προτεραιότητας που διαρκώς αυξάνεται κατά 1 και η εξυπηρέτησή του διαρκεί 3'. Για κάθε λεπτό της δωρης λειτουργίας της τράπεζας, το πρόγραμμα να κάνει τα εξής: α) εμφανίζει τα λεπτά που πέρασαν από την έναρξη της λειτουργίας β) εμφανίζει ποιος αριθμός προτεραιότητας εισάγεται και ποιος εξάγεται (αν υπάρχει) γ) εμφανίζει ποιος αριθμός εξυπηρετείται δ) εμφανίζει πόσοι πελάτες είναι στην ουρά. Το πρόγραμμα να χρησιμοποιεί την ακέραια ουρά $O[1000]$ και τις διαδικασίες Εισαγωγή(O , front, rear, x) και Εξαγωγή(O , front, rear, x) τις οποίες και να υλοποιήσετε. (O η ουρά, front και rear οι ακέραιοι δείκτες της και x η ακέραια τιμή που εισάγεται/εξάγεται).



5. Να γραφεί πρόγραμμα το οποίο διαβάσει στον $O[30]$ τα ονόματα 30 ομάδων **ποδοσφαίρου** και στον $B[30,100]$ τις **κατατάξεις** τους σε ένα **πρωτάθλημα** τα τελευταία 100 χρόνια (0 αν δεν συμμετείχαν). Να εμφανίζει ποια ομάδα έχει α) τα περισσότερα πρωταθλήματα και πόσα β) ποια τα περισσότερα σερί πρωταθλήματα πόσα γ) ποια τις περισσότερες κατατάξεις στις θέσεις 1-5 και πόσες και δ) ποια τις περισσότερες κατατάξεις στη θέση 16 και κάτω και πόσες. Να προβλεφθεί η ισοβαθμία. Να χρησιμοποιηθεί το υποπρόγραμμα $\text{maxTimes}(O, T)$, όπου $O[30]$ χαρακτήρων και παράλληλος $T[30]$ ακέραιος, το οποίο εμφανίζει τη μέγιστη τιμή του T και τα αντίστοιχα στοιχεία του O με τη μέγιστη τιμή του T το οποίο και να υλοποιήσετε.



17^η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

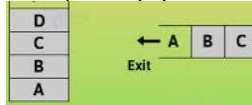
1. Να γραφεί πρόγραμμα το οποίο διαβάζει αρχικά τους **μισθούς** 100 υπαλλήλων και στη συνέχεια διαβάζει μισθούς μέχρι να δοθεί ως μισθός η τιμή -1. Εμφανίζει: το μέγιστο μισθό και το μέσο όρο των μισθών.



2. Έστω έτοιμη διαδικασία $\Omega_{\text{ρα}}(\omega, \lambda, \delta)$ που επιστρέφει την τρέχουσα ώρα. Να γραφεί η διαδικασία Δευτερόλεπτα_Ωρα($\chi, \omega, \lambda, \delta$) η οποία δέχεται έναν χρόνο χ σε δευτερόλεπτα και τον μετατρέπει στις περισσότερες δυνατές ώρες, λεπτά και δευτερόλεπτα. Να γραφεί η ακέραια συνάρτηση $\Omega_{\text{ρα_Δευτερόλεπτα}}(\omega, \lambda, \delta)$ η οποία δέχεται μία ώρα σε ώρες(ω), λεπτά(λ) και δευτερόλεπτα(δ) και τη μετατρέπει σε δευτερόλεπτα. Να γραφεί πρόγραμμα που διαχειρίζεται ακέραια **ουρά** $Q[100]$ για την εξυπηρέτηση πολιτών **σε δημόσια υπηρεσία** που λειτουργεί 08:00 - 16:00 και μέσω κατάλληλου μενού και με χρήση των παραπάνω, εκτελεί τα εξής: 1. Εισαγωγή: εισαγωγή (αν γίνεται) της ώρας προσέλευσης (σε δευτερόλεπτα). 2. Εξαγωγή: εξαγωγή (αν γίνεται) και εμφάνιση του **χρόνου αναμονής** από την ώρα προσέλευσης. 3. Έξοδος: τερματισμός και εμφάνιση του μέσου χρόνου αναμονής σε ώρες, λεπτά και δευτερόλεπτα. Να θεωρηθεί ότι θα εξυπηρετηθεί ένας τουλάχιστον πολίτης.



3. Δίνεται η ακέραια ουρά $Q[100]$ με δείκτες front και rear και η ακέραια στοίβα $S[50]$ με δείκτη top. Να γραφεί τμήμα κώδικα που **εξάγει επαναληπτικά τα στοιχεία της ουράς και τα ωθεί στη στοίβα** μέχρι να αδειάσει η ουρά ή να γεμίσει η στοίβα. Τελικά να εμφανίζει τι από τα δύο (ή και τα δύο) συνέβη.



4. Να γραφεί πρόγραμμα το οποίο διαβάζει τα λίτρα βενζίνης της **δεξαμενής** ενός πρατηρίου (θετικά και το πολύ έως 10000 με έλεγχο εγκυρότητας). Επαναληπτικά διαβάζει παραγγελίες σε λίτρα μέχρι να εξαντληθεί το απόθεμα ή δεν εξυπηρετηθούν 3 συνεχόμενα οχήματα. Τελικά να εμφανίζει το μέσο όρο λίτρων ανά εξυπηρετούμενη παραγγελία.

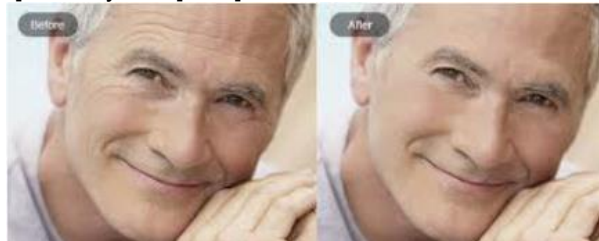


5. Στο **σκάκι** η βασίλισσα κινείται οριζόντια/κάθετα/διαγωνίως. Να γραφεί το υποπρόγραμμα Αναζήτηση(Σ , β , γ , σ) που δέχεται τον $\Sigma[8,8]$ χαρακτήρων και τη β χαρακτήρων. Επιστρέφει στις ακέραίες γ , σ τη γραμμή και τη στήλη εντοπισμού της β στον Σ (ή -1 αν δεν βρεθεί). Να γραφεί το υποπρόγραμμα ΔύοΒασίλισσες(Σ) που δέχεται τον $\Sigma[8,8]$ χαρακτήρων. Αναζητά τις δύο βασίλισσες την άσπρη ('AB') και τη μαύρη ('MB') και αν τις βρει να ελέγχει αν η μία "τρώει" την άλλη (αν δηλ. βρίσκονται στην ίδια γραμμή/στήλη/διαγώνιο). Σημείωση: ο $\Sigma[8,8]$ χαρακτήρων περιέχει μόνο τις **δύο βασίλισσες**. Δύο στοιχεία $\Sigma[\gamma_1, \sigma_1]$ και $\Sigma[\gamma_2, \sigma_2]$ βρίσκονται στην ίδια διαγώνιο εάν $|\gamma_1 - \gamma_2| = |\sigma_1 - \sigma_2|$.



18^η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Η εξομάλυνση (**smoothing**) χρησιμοποιείται για τη μείωση του θορύβου μιας εικόνας. Μπορείτε να εξομαλύνετε μια εικόνα με την αντικατάσταση κάθε τιμής της (pixel) με τη μέση τιμή των γειτόνων της, συμπεριλαμβανομένης και της ίδιας. Να γραφεί πρόγραμμα που διαβάζει τις τιμές (pixel) μιας εικόνας $N \times M$ έστω 600×800 σε πραγματικό πίνακα $A[600, 800]$. Να εξομαλύνει την εικόνα σε νέο πίνακα $B[600, 800]$ τον οποίο και να εμφανίζει. Για το σκοπό αυτό να υλοποιήσετε και να χρησιμοποιήσετε τα παρακάτω υποπρογράμματα: α) συνάρτηση $MO(A, i, j)$: Πραγματική με παραμέτρους έναν πραγματικό πίνακα $A[600, 800]$ και τον αριθμό γραμμής και στήλης ενός στοιχείου του (i, j) . Να επιστρέφει τη μέση τιμή των γειτόνων του, συμπεριλαμβανομένης και του ίδιου. β) συνάρτηση $Εντός(N, M, x, y)$: Λογική που ελέγχει αν το x ανήκει στο $[1, N]$ και το y στο $[1, M]$.



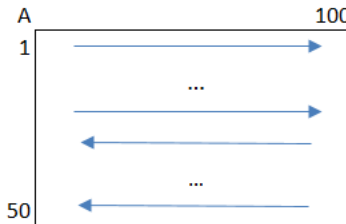
2. Να γραφεί πρόγραμμα το οποίο διαβάζει τους γραπτούς **βαθμούς** 100 μαθητών σε 9 πανελλαδικώς εξεταζόμενα μαθήματα, σε πίνακα $A[100, 9]$. Να δέχεται σαν είσοδο ένα βαθμό, **να ελέγχει εάν ο βαθμός αυτός έχει γραφτεί από όλους τους μαθητές τουλάχιστον μία φορά** (εάν δηλ. ο βαθμός υπάρχει σε κάθε γραμμή του πίνακα A) και να εμφανίζει κατάλληλο μήνυμα.



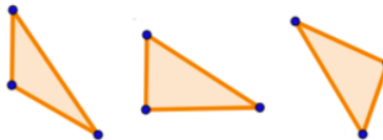
3. Ένα **ζάρι** θεωρείται "**πειραγμένο**" αν δεν εμφανίζει τις τιμές 1-6 σχεδόν ισοπίθανα. Διαφορετικά θεωρείται κανονικό. Να γραφεί πρόγραμμα που διαβάζει τις 100.000 ενδείξεις ενός ζαριού και αν οι πιθανότητες όλων των τιμών 1-6 κυμαίνονται από 15% έως και 18% να το χαρακτηρίζει "κανονικό". Διαφορετικά να το χαρακτηρίζει "πειραγμένο".



4. Να γραφεί πρόγραμμα που διαβάζει ακέραιο $A[50, 100]$ και εμφανίζει τα στοιχεία του κατά γραμμές **εναλλάξ μία από την αρχή του και μία από το τέλος του**: $A[1,1]$, $A[50,100]$, $A[1,2]$, $A[50,99]$, ... $A[25,100]$, $A[26,1]$



5. Τρεις αριθμοί α , β , γ είναι **πλευρές τριγώνου** όταν καθένας από αυτούς είναι μικρότερος από το άθροισμα των άλλων δύο. Επιπλέον: α) αν το τετράγωνο ενός από αυτούς είναι ίσο με το άθροισμα των τετραγώνων των άλλων δύο, τότε το τρίγωνο είναι ορθογώνιο β) αν το τετράγωνο του καθενός είναι μικρότερο από το άθροισμα των τετραγώνων των άλλων δύο, τότε το τρίγωνο είναι οξυγώνιο γ) αν το τετράγωνο ενός από αυτούς είναι μεγαλύτερο από το άθροισμα των τετραγώνων των άλλων δύο, τότε το τρίγωνο είναι αμβλυγώνιο. Να γραφεί πρόγραμμα που **για κάθε τριάδα α , β , γ** στο ακέραιο διάστημα $[1, 10]$ υπολογίζει το πλήθος εκείνων που αποτελούν α) πλευρές τριγώνου β) πλευρές ορθογώνιου τριγώνου γ) πλευρές οξυγώνιου τριγώνου δ) πλευρές αμβλυγώνιου τριγώνου.



19^η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Ένας σύλλογος αποφάσισε να διοργανώσει **διαγωνισμό χορού** με ζευγάρια μελών ως εξής: θα δέχεται επαναληπτικά τις αιτήσεις συμμετοχών με το όνομα και το φύλο του κάθε ενδιαφερόμενου. Θα συντηρεί 2 ουρές: $OA[100]$ για τους άντρες και $OG[100]$ για τις γυναίκες, στις οποίες θα εισάγονται τα ονόματα των μελών. Τελικά στο διαγωνισμό θα συμμετέχουν όσα ζευγάρια μπορούν να σχηματισθούν και με σειρά προτεραιότητας των αιτήσεών τους.

Να γραφεί πρόγραμμα το οποίο: α) αρχικοποιεί τις 2 ουρές ως άδειες β) διαβάζει επαναληπτικά τις αιτήσεις συμμετοχών με το όνομα και το φύλο ('Α'/Γ' με έλεγχο εγκυρότητας) του κάθε ενδιαφερόμενου και ενημερώνει κατάλληλα την αντίστοιχη ουρά. Να τερματίζει την εισαγωγή όταν σαν όνομα δοθεί η λέξη 'ΤΕΛΟΣ' ή όταν συμπληρωθούν 100 αιτήσεις. γ) τελικά να εμφανίζει: i. τα ζευγάρια του διαγωνισμού ii. τα ονόματα και το φύλο των μελών που δεν βρήκαν ταίρι (αν υπάρχουν). π.χ. OA : Δημήτρης, Βασίλης, Γιάννης, Πέτρος, Κώστας OG : Άννα, Μαρία, Ελένη => ζευγάρια του διαγωνισμού: (Δημήτρης-Άννα), (Βασίλης-Μαρία), (Γιάννης-Ελένη) και τα μέλη που δεν βρήκαν ταίρι: Πέτρος, Κώστας (άντρες).

Για τα παραπάνω να χρησιμοποιήσετε (χωρίς να τις υλοποιήσετε) τις διαδικασίες:

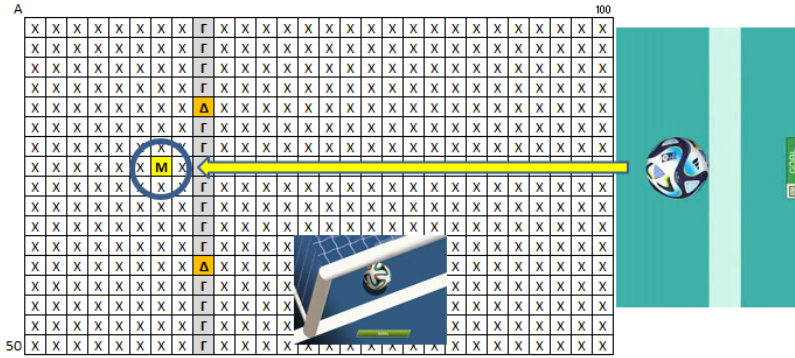
Εισαγωγή(O , f , r , on) που εισάγει αν μπορεί το on σε ουρά $O[100]$ με δείκτες f και r

Εξαγωγή(O , f , r , on , ok) που εξάγει αν μπορεί το on από ουρά $O[100]$ με δείκτες f και r και επιστρέφει στη λογική παράμετρο ok τιμή ανάλογη με το αν έγινε η εξαγωγή

Να θεωρήσετε ότι θα γίνει μία τουλάχιστον αίτηση.



2. Η τεχνολογία γραμμής (**Goal-line technology**) είναι σύστημα ηλεκτρονικού ελέγχου, το οποίο χρησιμοποιείται στο ποδόσφαιρο για την άμεση διακρίβωση της επίτευξης ή όχι τέρματος (goal) όταν υπάρχουν αμφιβολίες εάν και κατά πόσο η μπάλα έχει διέλθει με όλη της τη μάζα, την τελική γραμμή της εστίας. Έστω ο $A[50, 100]$ χαρακτήρων που απεικονίζει μία εικόνα (κάτοψη) μιας φάσης. Κάθε στοιχείο του A αντιστοιχεί σε απόσταση 1cm. Είναι γεμάτος με το γράμμα 'X' εκτός: i) μιας στήλης που είναι γεμάτη με το γράμμα 'Γ' που αντιστοιχεί στην γραμμή της αριστερής εστίας ii) δύο στοιχείων με το γράμμα 'Δ' που αντιστοιχούν στα δοκάρια της εστίας (στην ίδια στήλη με τη γραμμή) και iii) ενός στοιχείου με το γράμμα 'Μ' που αντιστοιχεί στο κέντρο της μπάλας (έστω ακτίνας 11cm). Να γραφεί τμήμα προγράμματος το οποίο: α) εντοπίζει στον A τη γραμμή της εστίας β) εντοπίζει τις θέσεις των δοκαριών (πάνω και κάτω) γ) εντοπίζει τη θέση της μπάλας δ) εμφανίζει κατάλληλο μήνυμα κατά πόσο "μετράει το γκολ". Αν δηλ. η μπάλα έχει περάσει ολόκληρη τη γραμμή (προς τα αριστερά) και το κέντρο της βρίσκεται μεταξύ των δοκαριών.



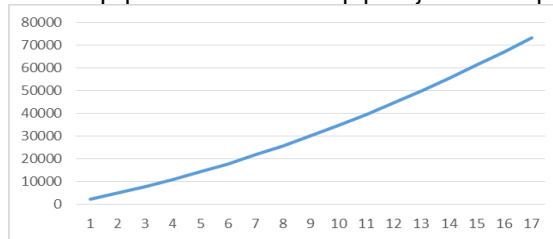
3. Να γραφεί πρόγραμμα το οποίο διαβάζει στον $O[100]$ τα ονόματα 100 μαθητών, στον $M[10]$ τα ονόματα 10 μαθημάτων και στον $B[100, 10]$ τους **βαθμούς** των 100 μαθητών στα 10 μαθήματα. Να εμφανίζει: α) τον κάθε μαθητή με τη μέση βαθμολογία του β) όλους τους βαθμούς (1000) κατά φθίνουσα σειρά μαζί με τα αντίστοιχα ονόματα των μαθητών και των μαθημάτων. Όσοι μαθητές έχουν τον ίδιο βαθμό να εμφανίζονται αλφαβητικά. **Υπόδειξη:** αντιγράψτε τους 1000 βαθμούς στον $B2[1000]$ και στους παράλληλους $O2[1000]$ και $M2[1000]$ τα αντίστοιχα ονόματα των μαθητών και των μαθημάτων.



4. Να γραφεί πρόγραμμα το οποίο διαβάζει επαναληπτικά: τον μήνα (1-12) και τα κέρδη (€) που είχε μία **αλυσίδα Super Market**, μέχρι να δοθεί ως μήνας η τιμή -1. Να υπολογίζει τα συνολικά κέρδη του κάθε μήνα. Τελικά να εμφανίζει τους μήνες με τα περισσότερα συνολικά κέρδη. Σημείωση: οι τιμές των μηνών μπορεί να επαναλαμβάνονται.



5. Ένας **εργαζόμενος** προσλαμβάνεται σε μια εταιρεία με αρχικό μηνιαίο μισθό 1000 €. Κάθε χρόνο ο εργαζόμενος παίρνει μηνιαία αύξηση 100 €. Από τα χρήματα που συγκεντρώνει το χρόνο αποταμιεύει το 20% με σκοπό να αγοράσει ένα διαμέρισμα το οποίο κοστίζει 70000€. Να γραφεί πρόγραμμα που α) υπολογίζει και εμφανίζει σε πόσα χρόνια ο εργαζόμενος θα αποκτήσει το δικό του σπίτι και β) αν η επίπλωση του σπιτιού κοστίζει 4500 €, εξετάζει αν υπάρχει περίσσειμα χρημάτων για να μπορέσει να αγοράσει έπιπλα και εμφανίζει κατάλληλο μήνυμα.



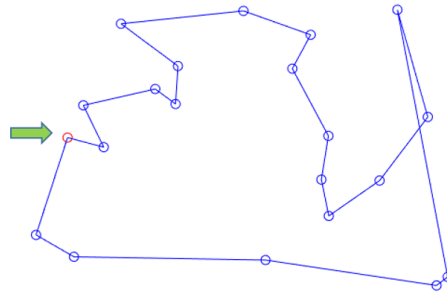
20^η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Σε έναν διαγωνισμό οι **επιδόσεις ήταν τριών κατηγοριών: Α,Β,Γ**. Να γραφεί πρόγραμμα το οποίο διαβάζει επαναληπτικά τα ονόματα και τις επιδόσεις των διαγωνιζόμενων με έλεγχο εγκυρότητας (Α/Β/Γ) **μέχρι να συμπληρωθούν 100** σε οποιαδήποτε από τις τρεις βαθμολογικές κατηγορίες, στους πίνακες $O_A[100]$, $O_B[100]$ και $O_G[100]$. Με κλήση προς κατάλληλο υποπρόγραμμα το οποίο και να υλοποιήσετε, να εμφανίζει αλφαβητικά τα ονόματα των διαγωνιζόμενων της κατηγορίας που συμπληρώθηκε πρώτη.
2. Ένα σχολείο πάει εκδρομή και οι μαθητές θα μετακινηθούν με **πλοίο και σε 4κλινες καμπίνες** ξεχωριστά τα αγόρια από τα κορίτσια. Να γραφεί πρόγραμμα το οποίο διαβάζει επαναληπτικά τα στοιχεία των μαθητών: όνομα και φύλο ('Α'/'Κ') μέχρι να δοθεί ως όνομα η λέξη 'ΤΕΛΟΣ'. Κάθε φορά που συμπληρώνεται μία 4κλινη καμπίνα, να εμφανίζει τα ονόματα των αγοριών/κοριτσιών που θα ταξιδέψουν σε αυτήν. Να εμφανίζει επιπλέον τον αριθμό των καμπινών που θα χρειασθούν ξεχωριστά για τα αγόρια και για τα κορίτσια. Σημείωση: θα κλείνεται ξεχωριστή καμπίνα ακόμη και αν στο τέλος περισσέψουν 1-3 αγόρια/κορίτσια.



3. Το **Πρόβλημα του Πλανόδιου Πωλητή**. Να γραφεί πρόγραμμα το οποίο διαβάζει τις συντεταγμένες ενός σημείου εκκίνησης $A(x, y)$ και 10 άλλων σημείων στον $P[10, 2]$. Να εμφανίζει τη διαδρομή που ξεκινάει από το A, περνάει από όλα τα σημεία του P από μία φορά μεταβαίνοντας κάθε φορά στο πλησιέστερο σημείο και επιστρέφει πίσω στο A. Να εμφανίζει τη συνολική απόσταση που διένυσε. **Σημείωση:** η διαδρομή να εμφανίζεται με τους αριθμούς των σημείων 1-

10. Η απόσταση δύο σημείων $(x_1, y_1), (x_2, y_2)$ είναι $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$



4. α. Να γραφεί ένα υποπρόγραμμα με όνομα «**Αναζήτηση Τιμής**» και παραμέτρους ένα μονοδιάστατο πραγματικό πίνακα Π και έναν πραγματικό x. Το υποπρόγραμμα να βρίσκει και να επιστρέφει την πρώτη θέση εμφάνισης της τιμής x στον πίνακα Π ή την τιμή 0 εάν το x δε βρεθεί.
β. Να γραφεί το πρόγραμμα «**Στατιστικά**» το οποίο να δέχεται σαν είσοδο τους γραπτούς βαθμούς 150 μαθητών ενός σχολείου στην 100θμια κλίμακα. Το πρόγραμμα να εμφανίζει με χρήση του παραπάνω υποπρογράμματος, τους βαθμούς (1-100) που έτυχε να μην έχει γράψει κανείς από τους 150 μαθητές.









5. Να γραφεί πρόγραμμα το οποίο διαβάζει τον Π[100] με τις πωλήσεις (€) 100 πωλητών με έλεγχο εγκυρότητας (>0) και τον Κ[100] με τις **οικογενειακές τους καταστάσεις** (με έλεγχο εγκυρότητας 'Ε': έγγαμος, 'Α': άγαμος) και υπολογίζει και εμφανίζει: α) τις μέσες πωλήσεις των έγγαμων και των άγαμων και β) την οικογενειακή κατάσταση του καλύτερου πωλητή (χωρίς ισοτιμία).



21^η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Να γραφεί πρόγραμμα που διαβάζει τις **συντεταγμένες 2 δίσκων** (κέντρα (x_1, y_1) και (x_2, y_2) και ακτίνες R1 και R2).

Στη συνέχεια να υπολογίζει την απόσταση d των δύο κέντρων ($\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$) καθώς και το ελάχιστο minR και το μέγιστο maxR των δύο ακτίνων. Τελικά, να ελέγχει τη σχετική θέση των δύο δίσκων εμφανίζοντας κατάλληλο μήνυμα για κάθε μία από τις παρακάτω 6 περιπτώσεις:

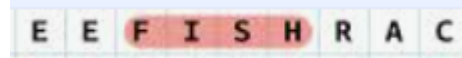
- a.  ομόκεντροι δίσκοι ($d = 0$)
- b.  ο ένας δίσκος εσωκλείεται στον άλλο ($d + \min R < \max R$)
- c.  εφάπτονται εσωτερικά ($d + \min R = \max R$)
- d.  τέμνονται ($d + \min R > \max R$ και $d - \min R < \max R$)
- e.  εφάπτονται εξωτερικά ($d = \min R + \max R$)
- f.  είναι «ξένοι» μεταξύ τους ($d > \min R + \max R$)

Παρατήρηση: θεωρείστε ότι οι δύο ακτίνες R1 και R2 είναι διαφορετικές μεταξύ τους.

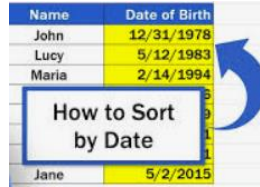
3. Να γραφεί πρόγραμμα που διαβάζει πίνακα ακεραίων $A[50, 100]$. Να διαβάζει μία ακέραια τιμή και την αναζητά **σειριακά** κατά γραμμές, μέχρι να την εντοπίσει για 1η φορά. Εάν τη βρει να εμφανίζει τις συντεταγμένες του κελιού όπου εντοπίστηκε.



4. Να γραφεί πρόγραμμα που διαβάζει στον $K[1000]$ τους 1000 χαρακτήρες ενός κειμένου και στον $\Phi[20]$ τους 20 χαρακτήρες μιας φράσης. Να βρίσκει πόσες φορές υπάρχει η **φράση μέσα στο κείμενο** και σε ποια σημεία.

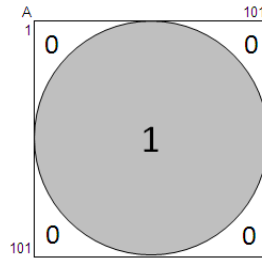


5. Να γραφεί πρόγραμμα που διαβάζει στον $HM[100, 3]$ 100 ημερομηνίες (ημέρα, μήνας, έτος). Να τις ταξινομεί κατά αύξουσα **χρονολογική σειρά**.



23η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

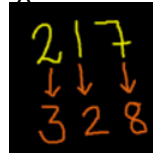
1. Να γραφεί πρόγραμμα το οποίο γεμίζει τον ακέραιο $A[101, 101]$ με 1 στον **κυκλικό δίσκο** που εφάπτεται εσωτερικά του 4γώνου και με 0 τα υπόλοιπα κελιά. Σημείωση: να γραφεί η συνάρτηση $Dist(x1, y1, x2, y2)$: Πραγματική η οποία υπολογίζει και επιστρέφει την απόσταση των σημείων $(x1, y1)$ και $(x2, y2)$. Ένα σημείο (x, y) βρίσκεται εντός του ανωτέρω κυκλικού δίσκου εάν η απόστασή του από το κέντρο $(51, 51)$ είναι μικρότερη ή ίση της ακτίνας του (50)



2. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν θετικό ακέραιο και ελέγχει εάν είναι **"ισορροπημένος"**. Ένας αριθμός λέγεται ισορροπημένος εάν το άθροισμα των ψηφίων στο πρώτο μισό του είναι ίσο με το άθροισμα των ψηφίων στο δεύτερο μισό του. π.χ. οι αριθμοί 2956, 73628, 234126, 342598213 είναι ισορροπημένοι.



3. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν θετικό ακέραιο και εμφανίζει τον ακέραιο που προκύπτει εάν **αυξηθούν** τα ψηφία του αρχικού **κατά 1** (το ψηφίο 9 να γίνεται 0). Π.χ. 12389 => 23490

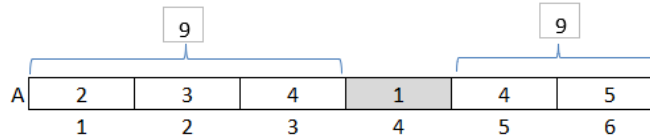


4. Να γραφεί συνάρτηση που δέχεται έναν θετικό ακέραιο και επιστρέφει το άθροισμα των "γνήσιων" διαιρετών του (εκτός του 1 και του εαυτού του). Να γραφεί πρόγραμμα το οποίο διαβάζει δύο θετικούς ακεραίους και ελέγχει αν είναι **φίλιοι** αριθμοί: έχουν το ίδιο άθροισμα γνήσιων διαιρετών.



5. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν ακέραιο $A[100]$ και στη συνέχεια δημιουργεί τον ακέραιο $B[100]$ έτσι ώστε να περιέχει τις τιμές του A **διαχωρισμένες** ως εξής: πρώτα οι αρνητικές τιμές, μετά οι μηδενικές και στο τέλος οι θετικές (χωρίς τη χρήση άλλων πινάκων και χωρίς να αλλάξει η διάταξη των θετικών και των αρνητικών τιμών. π.χ. για $A[10]$ με τιμές: 4,0,-5,1,0,-2,3,2,10,8 → τιμές του B : -5,-2,0,0,4,1,3,2,10,8).

2,3,4,1,4,5 σημείο ισορροπίας είναι το 4ο (1) διότι τα στοιχεία στα αριστερά (2,3,4) και τα στοιχεία στα δεξιά (4,5) έχουν το ίδιο άθροισμα (9).



25η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Να γραφεί πρόγραμμα που γεμίζει έναν ακέραιο A[50,100] με τις τιμές 1,2,...,5000 κατά γραμμές και κατασκευάζει τον **επίκεντρο χάρτη** X[N,M] έτσι ώστε το κάθε στοιχείο X[i, j] να ισούται με το άθροισμα του στοιχείου A[i, j] με όλα τα γειτονικά στοιχεία του π.χ. για A[4, 5]:

A					
	1	2	3	4	5
	6	7	8	9	10
	11	12	13	14	15
	16	17	18	19	20

X					
	16	27	33	39	28
	39	63	72	81	57
	69	108	117	126	87
	56	87	93	99	68

Υπόδειξη: για τους σκοπούς της άσκησης να υλοποιήσετε συνάρτηση f(A, γρ, στ): Ακέραια με A[50,100], γρ,στ ακέραιες. Να επιστρέφει το A[γρ,στ] εάν οι γρ,στ είναι εντός των ορίων του A[50,100], διαφορετικά το 0.

2. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο A[100] και **τον ανακατεύει στο μέγιστο βαθμό**. Τοποθετεί δηλ. στο A[1] το 1ο μεγαλύτερο, στο A[2] το 1ο μικρότερο, στο A[3] το 2ο μεγαλύτερο, στο A[4] το 2ο μικρότερο κ.ο.κ. π.χ. εάν ο A[10] έχει τις τιμές: 1,2,3,4,5,6,7,8,9,10 καταλήγει με τις τιμές: 10,1,9,2,8,3,7,4,6,5



3. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν αριθμό ατόμων και έναν αριθμό ομάδων (<= άτομα) και χωρίζει τα άτομα σε **"ισοδύναμες" ομάδες**. π.χ. 27 άτομα, 4 ομάδες => 1 ομάδα των 6 και 3 ομάδες των 7 π.χ. 20 άτομα, 5 ομάδες => 5 ομάδες των 4 π.χ. 21 άτομα, 6 ομάδες => 3 ομάδες των 3 και 3 ομάδες των 4.



4. Σε **super market** υπάρχουν 12 ταμεία που εξυπηρετούν μία ουρά χωρητικότητας 80 πελατών. Να γραφεί πρόγραμμα το οποίο α) εμφανίζει επαναληπτικό μενού των επιλογών: 1. Είσοδος πελάτη, 2. Εξυπηρέτηση πελάτη, 3. Τερματισμός. Διαβάζει μία επιλογή με έλεγχο εγκυρότητας. β) αν επιλεγθεί το 1 να εισάγεται στην ουρά ο αύξοντας αριθμός του πελάτη γ) αν επιλεγθεί το 2 να εξάγεται ο πρώτος στην ουρά και να εισάγεται ο αριθμός του ταμείου εξυπηρέτησης (1-12). Να καλεί το υποπρόγραμμα ΥΠΟ για τον υπολογισμό του ποσού που οφείλει ο πελάτης δ) αν επιλεγθεί το 3 να εμφανίζει το ταμείο με τα περισσότερα έσοδα (μόνο ένα) ε) να υλοποιηθεί το υποπρόγραμμα ΥΠΟ το οποίο διαβάζει επαναληπτικά τον αριθμό τεμαχίων και το κόστος ανά τεμάχιο, μέχρι να δοθεί ως αριθμός τεμαχίων το μηδέν (0). Να επιστρέφει το συνολικό ποσό.



5. Το **μαγικό τετράγωνο** αποτελεί διάταξη αριθμών σε συστοιχία ίσου συνόλου γραμμών, στηλών και διαγωνίων. π.χ.

2	7	6	→15
9	5	1	→15
4	3	8	→15
↙15	↓15	↓15	↘15

Να γραφεί πρόγραμμα το οποίο διαβάζει έναν ακέραιο A[100, 100] και ελέγχει εάν είναι μαγικό τετράγωνο.

26η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Εταιρεία διοργανώνει παιχνίδι **στοιχηματισμού** για τον διαγωνισμό τραγουδιού της **Eurovision**. Να γραφεί πρόγραμμα το οποίο: α) διαβάσει σε πίνακες τα ονόματα των 32 υποψήφιων χωρών και τους συντελεστές απόδοσης της καθεμιάς (πραγματικές με εγκυρότητα ώστε να είναι > 1). Ο συντελεστής αυτός πολλαπλασιάζεται με το ποσό που ποντάρεται στη νικήτρια χώρα και αποδίδεται στους δικαιούχους παίκτες. Τα ποσά που ποντάρονται στις υπόλοιπες χώρες εισπράττονται από την εταιρεία. β) διαβάσει επαναληπτικά τα πονταρίσματα παικτών: όνομα χώρας (με εγκυρότητα) και ποσό μέχρι να δοθεί ως όνομα η λέξη 'τέλος' γ) τηρεί σε παράλληλο πίνακα τα συνολικά πονταρίσματα της κάθε χώρας. δ) μετά τη λήξη της διαδικασίας πονταρισμάτων διαβάσει το όνομα της νικήτριας χώρας μέχρι να δοθεί κάτι έγκυρο ε) υπολογίζει τα κέρδη/ζημιές της εταιρείας: συνολικά πονταρίσματα σε όλες τις χώρες πλην της νικήτριας - σύνολο πονταρισμάτων στην νικήτρια χώρα * συντελεστή απόδοσης. Για τις ανάγκες της άσκησης να φτιάξετε τη συνάρτηση Βρες(Ο, χώρα):Ακέραια που ψάχνει μία χώρα στον πίνακα O[32] και επιστρέφει τη θέση εντοπισμού της (1-32) ή την τιμή 0.



2. Σε **κουζίνα** εστιατορίου εισάγονται οι παραγγελίες σε **2 ουρές**. Να γραφεί πρόγραμμα το οποίο: α) αρχικοποιεί 2 ουρές O1 και O2 χαρακτήρων και χωρητικότητας 100 ως άδειες β) εμφανίζει επαναληπτικά το παρακάτω μενού επιλογών: 1. Εισαγωγή 2. Εξαγωγή 3. Τερματισμός μέχρι να επιλεγθεί το 3 γ) αν επιλεγθεί το 1 να διαβάσει μία παραγγελία και να την εισάγει στην ουρά με τα λιγότερα στοιχεία. Αν είναι γεμάτες και οι 2 να εμφανίζει κατάλληλο μήνυμα δ) αν επιλεγθεί το 2 να διαβάσει από ποια ουρά (1/2) γίνεται η εξαγωγή και να εμφανίζει την εξαγόμενη παραγγελία. Αν η ουρά αυτή είναι άδεια να εμφανίζει κατάλληλο μήνυμα ε) κατά τον τερματισμό να εμφανίζει πόσες παραγγελίες ολοκληρώθηκαν από κάθε ουρά. Πρόταση υλοποίησης με υποπρογράμματα: α) Συνάρτηση ΣτοιχείαΟυράς(O, f, r): Ακέραια (δέχεται μία ουρά χαρακτήρων O[100] με δείκτες f και r και επιστρέφει τον αριθμό των στοιχείων της) β) Διαδικασία Εισαγωγή(O, f, r, x) (δέχεται μία ουρά χαρακτήρων O[100] με δείκτες f και r και μία τιμή x χαρακτήρων. Εισάγει το x στην ουρά) γ) Διαδικασία Εξαγωγή(O, f, r, x) (δέχεται μία ουρά χαρακτήρων O[100] με δείκτες f και r και μία ακέραια παράμετρο x. Κάνει μία εξαγωγή στην ουρά και αν γίνει με επιτυχία, αυξάνει το x κατά 1)



3. Σε κέντρο τεχνικού ελέγχου οχημάτων (**ΚΤΕΟ**) προσέρχονται οχήματα 3 κατηγοριών: επιβατηγά, φορτηγά και μοτοσυκλέτες. Για κάθε κατηγορία οχήματος υπάρχει μία **ουρά** χωρητικότητας 100. Να γραφεί πρόγραμμα το οποίο: α) αρχικοποιεί 3 ουρές OE, OF, OM χαρακτήρων και χωρητικότητας 100 ως άδειες β) εμφανίζει επαναληπτικά το παρακάτω μενού επιλογών: 1. Εισαγωγή 2. Εξαγωγή 3. Τερματισμός μέχρι να επιλεγθεί το 3 γ) αν επιλεγθεί το 1 να διαβάσει την κατηγορία του οχήματος (επιβατηγό/φορτηγό/μοτοσυκλέτα) την οποία και να εισάγει στην αντίστοιχη ουρά. Αν είναι γεμάτη να εμφανίζει κατάλληλο μήνυμα δ) αν επιλεγθεί το 2 να διαβάσει από ποια ουρά (1/2/3) γίνεται η εξαγωγή και να εμφανίζει τον εξαγόμενο αριθμό κυκλοφορίας. Αν η ουρά αυτή είναι άδεια να εμφανίζει κατάλληλο μήνυμα ε) κατά τον τερματισμό να εμφανίζει τα συνολικά έσοδα του ΚΤΕΟ εάν το κόστος ελέγχου είναι 50€ για τα επιβατηγά, 150€ για τα φορτηγά και 30€ για τις μοτοσυκλέτες. Πρόταση υλοποίησης με υποπρογράμματα: α) Διαδικασία Εισαγωγή(O, f, r, x) (δέχεται μία ουρά χαρακτήρων O[100] με δείκτες f και r και μία τιμή x χαρακτήρων. Εισάγει το x στην ουρά) β) Διαδικασία Εξαγωγή(O, f, r, x) (δέχεται μία ουρά χαρακτήρων O[100] με δείκτες f και r και μία ακέραια παράμετρο x. Κάνει μία εξαγωγή στην ουρά και αν γίνει με επιτυχία, αυξάνει το x κατά το κόστος ελέγχου 50€/150€/30€ της κατηγορίας του εξαγόμενου οχήματος).

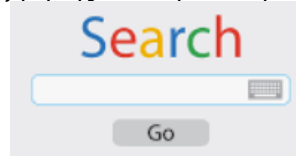


4. Να γραφεί πρόγραμμα το οποίο διαβάσει στους πίνακες O[80] και Π[80, 12] τα ονόματα 80 πωλητών και τις μηνιαίες πωλήσεις τους (€) για ένα έτος. Να ταξινομήσει τις 12 πωλήσεις καθενός πωλητή κατά αύξουσα σειρά και να εμφανίσει τους πωλητές στους οποίους έγινε ο μεγαλύτερος **αριθμός αντιμεταθέσεων** για την ταξινόμηση των πωλήσεών τους.



5. Να γραφεί πρόγραμμα προσομοίωσης μίας **μηχανής αναζήτησης** το οποίο: α) διαβάσει στον πίνακα U[1000] τις διευθύνσεις 1000 διαφορετικών sites και στον πίνακα KW[1000, 100] 100 λέξεις κλειδιά για κάθε site β) διαβάσει το πολυ

10 λέξεις αναζήτησης ή να μέχρι να δοθεί το κενό γ) εμφανίζει τα 10 πιο σχετικά sites. Αυτά δηλαδή που περιέχουν τις περισσότερες φορές τις εισαγόμενες λέξεις αναζήτησης. Σε περίπτωση ισοβαθμίας να εμφανίζονται αλφαβητικά.



27^η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

Μετατρέψετε τα παρακάτω μπλοκ μη δομημένου κώδικα σε δομημένο. Υπόδειξη: μετατρέψετε τα πρώτα σε διάγραμμα ροής.

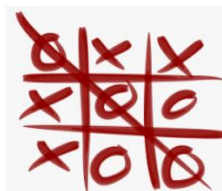
1	2	3	4	5
5: αν (συνθ 1) τότε εντ 1 αλλιώς GOTO 10 τέλος_αν GOTO 5 10: εντ 2	5: αν (συνθ 1) τότε εντ 1 αλλιώς GOTO 20 τέλος_αν αν (συνθ 2) τότε εντ 2 GOTO 10 τέλος_αν εντ 3 10: εντ 4 GOTO 5 20: εντ 5	5: αν (συνθ 1) τότε εντ 1 10: εντ 2 αν (συνθ 2) τότε εντ 3 GOTO 10 τέλος_αν GOTO 5 αλλιώς εντ 4 τέλος_αν	Διάβασε F C ← 5/9*(F-32) Αν C < 10 τότε GOTO 10 Αν C > 25 τότε GOTO 20 GOTO 30 10: Εκτύπωσε 'Χαμηλή' GOTO 40 20: Εκτύπωσε 'Υψηλή' GOTO 40 30: Εκτύπωσε 'Μέτρια' 40: !συνέχεια	5: αν (συνθ 1) τότε εντ 1 αν (συνθ 2) τότε εντ 2 εντ 3 GOTO 10 τέλος_αν εντ 4 10: εντ 5 GOTO 5 τέλος_αν εντ 6

```

BEGIN
  GOTO label_1;
  DBMS_OUTPUT.PUT_LINE("This is skipped");
  <<label_1>>
  DBMS_OUTPUT.PUT_LINE("It is here!");
END;
    
```

28η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Να γραφεί πρόγραμμα το οποίο διαβάζει τον T[100, 100] χαρακτήρων με τιμές 'X'/'O'/. Στη συνέχεια να ελέγχει αν υπάρχει νικητής σε μορφή **τρίλιζας** (αν υπάρχει γραμμή ή στήλη ή διαγώνιος γεμάτη με 'X'/'O') και αν ναι, ποιος είναι ('X'/'O') και σε ποια γραμμή ή στήλη ή διαγώνιο. Να εμφανισθούν όλοι οι πιθανοί "νικητές". Αν δεν υπάρχει νικητής να εμφανίζει κατάλληλο μήνυμα.



2. Όταν ρίχνουμε δύο **ζάρια** η πιθανότητα να πετύχουμε ένα συγκεκριμένο αποτέλεσμα εξαρτάται από το πλήθος των συνδυασμών που μπορεί να μας δώσουν το συγκεκριμένο **άθροισμα**. Για παράδειγμα, το αποτέλεσμα 11 δίνεται από δύο συνδυασμούς. Επομένως, η πιθανότητά του είναι 2/36, γιατί 36 είναι όλοι οι πιθανοί συνδυασμοί. Να γραφεί πρόγραμμα το οποίο εμφανίζει για όλα τα πιθανά αθροίσματα (2-12) την % πιθανότητα εμφάνισής τους σε μία ρίψη δύο ζαριών.



3. Ένας πίνακας χαρακτήρων Γ[70, 100] προσομοιώνει ένα **γήπεδο ποδοσφαίρου** σε οριζόντια θέση. Σε ένα στοιχείο περιέχει το γράμμα 'Π' που υποδηλώνει τη θέση ενός υπό εξέταση παίκτη. Σε δέκα άλλα στοιχεία περιέχει το γράμμα 'Α' που υποδηλώνει τη θέση των 10 αντιπάλων του (πλην του τερματοφύλακα). Να γραφεί πρόγραμμα το οποίο διαβάζει τον παραπάνω πίνακα και να εξετάζει εάν ο παίκτης 'Π' είναι σε θέση **οφσάιντ** (offside): εάν βρίσκεται στο αντίπαλο μισό του γηπέδου και ταυτόχρονα είναι πλησιέστερα στην αντίπαλη γραμμή τέρματος και από τους 10 αντιπάλους του. Να θεωρηθεί ότι η αντίπαλη γραμμή τέρματος του παίκτη 'Π' βρίσκεται στη δεξιά μεριά του γηπέδου. (Παρατήρηση: ο κανονισμός του οφσάιντ παρουσιάζεται στην άσκηση σε πιο απλή μορφή από τον κανονικό)

Γ	1			50				100
1								
			A		A			
			A		A	Π		
		A		A			A	
			A		A			
70					A			

4. Να γραφεί πρόγραμμα το οποίο διαβάσει στους πίνακες χαρακτήρων $U[100]$, $P[100]$ τα ονόματα χρήστη και τους κωδικούς (usernames & passwords) 100 εγγεγραμμένων μελών ενός **κοινωνικού δικτύου** καθώς και στον ακέραιο $F[100, 100]$ τις μεταξύ τους **σχέσεις φιλίας** (1/0) στην άνω τριγωνική περιοχή του. Το 1 στο στοιχείο $F[i,j]$ υποδηλώνει φιλία των μελών i και j . Να καταχωρεί στο συμμετρικό στοιχείο $F[j,i]$ την τιμή του $F[i,j]$. Να διαβάζει username & password ενός μέλους που θέλει να εισέλθει μέχρι να δοθούν έγκυρα στοιχεία. Εμφανίζει μενού με τις επιλογές: 1. Φίλοι: εμφανίζει τα usernames των φίλων του και πόσοι είναι. 2. Αίτημα φιλίας: διαβάζει ένα username μέχρι να δοθεί κάποιο έγκυρο με το οποίο δεν υπάρχει ήδη σχέση φιλίας καθώς και μία απάντηση (ΝΑΙ/ΟΧΙ) για έγκριση του αιτήματος. Να ενημερώνει τον πίνακα F σε περίπτωση έγκρισης. 3. Διαγραφή φιλίας: διαβάζει ένα username μέχρι να δοθεί κάποιο έγκυρο με το οποίο υπάρχει σχέση φιλίας και ενημερώνει τον πίνακα F . 4. Έξοδος. Θεωρείστε ότι τα usernames είναι διαφορετικά



5. Να γραφεί πρόγραμμα το οποίο διαβάσει στον ακέραιο πίνακα $P[48]$ τα **ραντεβού** (1 για κλεισμένο ή 0 για ελεύθερο) ενός **εμβολιαστικού κέντρου** για κάθε τέταρτο της 12ωρης λειτουργίας του (08:00-20:00) και στον παράλληλο πίνακα χαρακτήρων $AMKA[48]$ το $AMKA$ του ατόμου που έκλεισε το ραντεβού ή το κενό αν είναι ελεύθερο. ($P[1]$ ραντεβού των 08:00, $P[2]$ ραντεβού των 08:15, ..., $P[48]$ ραντεβού των 19:45). Εμφανίζει μενού με τις επιλογές: 1. Κλείσιμο ραντεβού: εμφανίζει τις ώρες των διαθέσιμων ραντεβού και διαβάζει μία ώρα (ώρα-λεπτό) μέχρι να δοθεί κάτι έγκυρο. Διαβάζει ένα $AMKA$ και ενημερώνει τους πίνακες P και $AMKA$. Εάν δεν υπάρχει διαθέσιμο ραντεβού να εμφανίζει κατάλληλο μήνυμα. 2. Ακύρωση ραντεβού: διαβάζει ένα $AMKA$ και αν εντοπισθεί στον πίνακα $AMKA$ ενημερώνει τους πίνακες P και $AMKA$. 3. Έξοδος



29η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Να γραφεί πρόγραμμα που προσομοιώνει τη λειτουργία ενός **ταμείου Super Market** ως εξής: διαβάζει σε κατάλληλους πίνακες τους κωδικούς και τις τιμές 1000 προϊόντων. Διαβάζει για έναν πελάτη επαναληπτικά τον κωδικό και τον αριθμό των τεμαχίων για τα προϊόντα που αγόρασε μέχρι να δοθεί ως κωδικός η τιμή -1. Υπολογίζει και εμφανίζει το συνολικό κόστος των αγορών μειωμένο κατά 1% σε περίπτωση που υπερβαίνει τα 200€. Διαβάζει το ποσό πληρωμής του πελάτη με εγκυρότητα έτσι ώστε να είναι τουλάχιστον όσο και το χρέος του. Εμφανίζει τα ρέστα που πρέπει να του επιστραφούν. Θεωρείστε ότι οι κωδικοί που εισάγονται είναι σωστοί, τα τεμάχια θετικά και ότι εισάγεται ένας τουλάχιστον κωδικός.



2. **Τράπεζα** έχει **5 ταμεία** και διατηρεί την ουρά αναμονής του καθενός (μεγέθους 10) στον ακέραιο $Q[5, 10]$ και τους αντίστοιχους ακέραιους δείκτες τους στον ακέραιο $D[5,2]$ (1η σειρά οι δείκτες εμπρός και πίσω της 1ης ουράς κοκ.). Να γραφεί πρόγραμμα το οποίο αρχικοποιεί τις 5 ουρές ώστε να είναι άδειες και εκτελεί επαναληπτικά μέσω μενού τις εξής λειτουργίες: 1. Εισαγωγή, εισάγει το νούμερο προτεραιότητας του επόμενου πελάτη (ξεκινώντας με το 1) στην ουρά με τους λιγότερους πελάτες εκείνη τη στιγμή. 2. Εξαγωγή, διαβάζει τον αριθμό του ταμείου (1-5) και εξαγάγει έναν πελάτη από την

αντίστοιχη ουρά εμφανίζοντας το νούμερό του. 3. Τερματισμός, εξάγει έναν-έναν όλους τους πελάτες και των 5 ουρών εμφανίζοντας τα νούμερά τους. Τερματίζει τη λειτουργία και εμφανίζει πόσοι πελάτες εξυπηρετήθηκαν συνολικά.



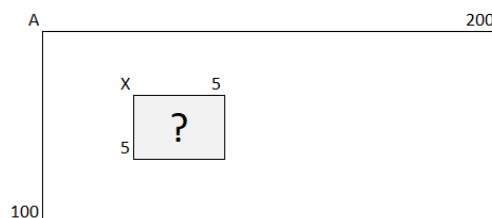
3. Στο αγώνισμα της **κολύμβησης μικτής ατομικής 400 μέτρων**, ο αθλητής αγωνίζεται σε αποστάσεις των 100 μέτρων, στα 4 στυλ της κολύμβησης. Η σειρά που ακολουθείται είναι: πεταλούδα, ύπτιο, πρόσθιο και ελεύθερο. Να γραφεί πρόγραμμα που διαβάσει σε κατάλληλους πίνακες τα ονόματα 8 κολυμβητών και τους χρόνους που πέτυχαν σε καθένα από τα 4 στυλ του μικτού ατομικού (πραγματικοί σε sec). Καταχωρεί σε κατάλληλο πίνακα τα 4 στυλ. Εμφανίζει: α) καθένα από τα 4 στυλ και τον κολυμβητή με την καλύτερη επίδοση σε αυτό β) την τελική κατάταξη γ) σε πόσα από τα 4 στυλ ο τελικός νικητής είχε την καλύτερη επίδοση. Θεωρείστε ότι όλοι οι χρόνοι είναι διαφορετικοί.



4. Το **δέκαθλο ανδρών** είναι ένα σύνθετο αγώνισμα του στίβου στο οποίο οι αθλητές αγωνίζονται σε 10 αγώνισματα. Δρομικά(4): 100 μέτρα, 400 μέτρα, 110 μέτρα μετ' εμποδίων, 1500 μέτρα. Άλματα(3): άλμα εις μήκος, άλμα εις ύψος, άλμα επί κοντώ. Ρίψεις(3): σφαιροβολία, δισκοβολία, ακοντισμός. Για τον υπολογισμό της βαθμολογίας έχουν καθορισθεί 3 σταθερές (A,B,C) για κάθε αγώνισμα. Η σταθερά B ορίζει για τα δρομικά αγώνισματα το μέγιστο όριο χρόνου πάνω από το οποίο η βαθμολογία για αυτό το αγώνισμα μηδενίζεται, ενώ για τα υπόλοιπα αγώνισματα ορίζει το ελάχιστο όριο επίδοσης κάτω από το οποίο η βαθμολογία για αυτό το αγώνισμα επίσης μηδενίζεται. Σε κάθε άλλη περίπτωση, η βαθμολογία για τα δρομικά αγώνισματα δίνεται από τον τύπο $A*(B-P)^C$ και για τα υπόλοιπα από τον τύπο $A*(P-B)^C$, όπου P η επίδοση του αθλητή για τα δρομικά και η καλύτερη επίδοση για τα υπόλοιπα. Η τελική βαθμολογία κάθε αθλητή αποτελεί το άθροισμα των 10 βαθμολογιών του. Να γραφεί πρόγραμμα το οποίο α) διαβάσει σε κατάλληλους πίνακες τα ονόματα 20 δεκαθλητών, τις επιδόσεις τους σε καθένα από τα 10 αγώνισματα και τις σταθερές (A,B,C) των 10 αγωνισμάτων (πρώτα τα 4 δρομικά και μετά τα 6 υπόλοιπα). β) υπολογίζει την τελική βαθμολογία τους γ) εμφανίζει το όνομα του νικητή δ) διαβάσει το παγκόσμιο ρεκόρ και ελέγχει εάν το έσπασε ή το ισοφάρισε. Θεωρείστε ότι δεν υπάρχει ισοβαθμία στην 1η θέση.



5. Να γραφεί πρόγραμμα που διαβάσει έναν πίνακα A[100, 200] και έναν X[5,5]. **Αναζητά τον X στον A** εμφανίζοντας τα σημεία στα οποία εντοπίζεται (τις συντεταγμένες του επάνω αριστερά στοιχείου) ή κατάλληλο μήνυμα σε περίπτωση αποτυχημένης αναζήτησης.



30^η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

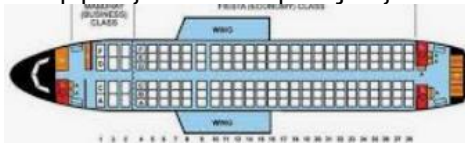
1. Να γραφεί πρόγραμμα που προσομοιώνει μία αποστολή αντιμετώπισης δασικής πυρκαγιάς από ένα **πυροσβεστικό αεροπλάνο**. Να διαβάσει α) τις συντεταγμένες (x, y) 10 εστιών σε πίνακα Σ[10,2] και β) ένα όριο ασφαλείας των καυσίμων που επιβάλλει την αναγκαστική επιστροφή του αεροπλάνου στη βάση του. Αποστολή του αεροπλάνου είναι να κάνει 3 τουλάχιστον επιτυχημένες ρίψεις σε κάθε εστία. Μία ρίψη θεωρείται επιτυχημένη για μία εστία όταν γίνεται σε απόσταση μικρότερη των 100 μέτρων από αυτήν. Το πρόγραμμα να διαβάσει επαναληπτικά τις συντεταγμένες (x, y) της κάθε ρίψης και να ελέγχει εάν ήταν επιτυχημένη για κάποια από τις 10 εστίες. Μετά από κάθε ρίψη να διαβάσει το απόθεμα των καυσίμων. Να τερματίζει τις ρίψεις όταν ολοκληρωθεί η αποστολή ή όταν τα καύσιμα πέσουν κάτω από το όριο ασφαλείας. Τελικά να εμφανίζει πόσες ρίψεις έγιναν και αναλόγως, το μήνυμα "Αποστολή εξετελέσθη" ή το % ποσοστό των εστιών που δέχθηκαν 3 τουλάχιστον ρίψεις. Θεωρείστε ότι οι αποστάσεις των 10 εστιών είναι μεγαλύτερη των 100 μέτρων. Για τις ανάγκες της

άσκησης να υλοποιηθεί η συνάρτηση Απόσταση(x1,y1,x2,y2) που επιστρέφει την απόσταση 2 σημείων με τον τύπο:

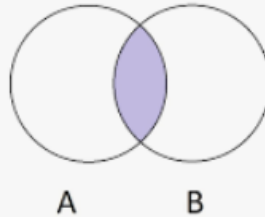
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



2. Μία αεροπορική πτήση έχει 100 θέσεις επιβατών. Να γραφεί πρόγραμμα το οποίο: α) αρχικοποιεί τον Θ[100] με το μηδέν β) εμφανίζει επαναληπτικά το εξής μενού επιλογών: 1. Αγορά εισιτηρίου 2. Ακύρωση εισιτηρίου 3. Τερματισμός μέχρι να επιλεγεί το 3. γ) αν επιλεγεί το 1 να εμφανίζει όλες τις διαθέσιμες θέσεις της πτήσης (τους δείκτες των μηδενικών στοιχείων του πίνακα Θ). Αν υπάρχει διαθεσιμότητα να διαβάσει τον επιθυμητό αριθμό θέσης με εγκυρότητα ώστε να είναι διαθέσιμος (0). Να δεσμεύει τη θέση καταχωρώντας την τιμή 1. Διαφορετικά να εμφανίζει το μήνυμα 'πτήση γεμάτη'. δ) αν επιλεγεί το 2 να ελέγχει αν υπάρχει τουλάχιστον μία δεσμευμένη θέση (1). Αν υπάρχει να διαβάσει έναν αριθμό θέσης με εγκυρότητα ώστε να είναι δεσμευμένος (1) και στη συνέχεια να τον αποδεσμεύει καταχωρώντας την τιμή 0. Διαφορετικά να εμφανίζει το μήνυμα 'πτήση άδεια'. ε) Τελικά να εμφανίζει πόσοι επιβάτες ταξιδεύουν και το μέγιστο σερί κενών θέσεων.



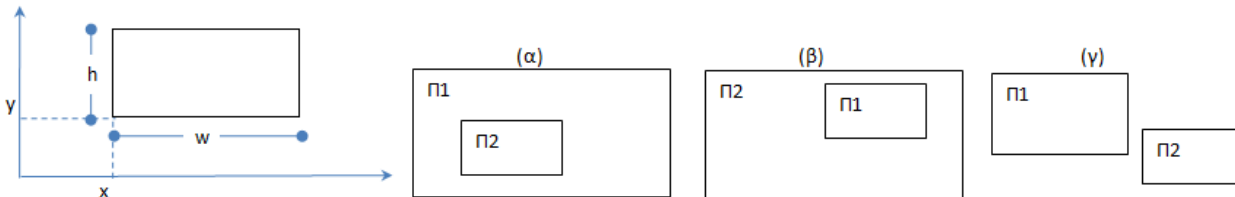
3. Να γραφεί πρόγραμμα το οποίο διαβάσει τους ακέραιους A[20,30] και B[60,90] και εμφανίζει τα στοιχεία της "τομής" τους. Τις τιμές δηλαδή που βρίσκονται και στους δύο πίνακες. Αν δεν υπάρχει καμία να εμφανίζει κατάλληλο μήνυμα.



4. Σε μία δημόσια υπηρεσία για να εκδοθεί ένα πιστοποιητικό πρέπει οι πολίτες να περάσουν πρώτα από το γραφείο 1 και στη συνέχεια από το γραφείο 2. Σε κάθε γραφείο υπάρχει ουρά μεγέθους 100. Να γραφεί πρόγραμμα το οποίο α) αδειάζει τις ουρές O1 και O2 τύπου χαρακτήρων. β) εμφανίζει επαναληπτικά το εξής μενού επιλογών: 1. Εισαγωγή 2. Εξαγωγή 3. Τερματισμός μέχρι να επιλεγεί το 3. γ) αν επιλεγεί το 1 να διαβάσει τον αριθμό ταυτότητας ενός πολίτη και να τον εισάγει στην ουρά O1 (γραφείο 1). δ) αν επιλεγεί το 2 να διαβάσει τον αριθμό του γραφείου (1/2) από το οποίο γίνεται η εξαγωγή. Αν δοθεί το γραφείο 1 και εφόσον μπορεί να γίνει εισαγωγή στο γραφείο 2, να κάνει εξαγωγή από την ουρά O1 και εισαγωγή του εξαχθέντος στοιχείου στην ουρά O2. Αν δοθεί το γραφείο 2 να κάνει εξαγωγή από την ουρά O2. ε) Τελικά να εμφανίζει πόσοι πολίτες εξέδωσαν το πιστοποιητικό.



5. Ένα ορθογώνιο παραλληλόγραμμο μπορεί να αναπαρασταθεί από 4 αριθμούς: x,y,w,h όπου (x,y) οι συντεταγμένες του κάτω αριστερά σημείου, w το πλάτος του και h το ύψος του. Να γραφεί η λογική συνάρτηση PointInRect(px, py, x, y, w, h) που δέχεται τις συντεταγμένες (px, py) ενός σημείου και ένα ορθογώνιο παραλληλόγραμμο (x, y, w, h). Ελέγχει εάν το σημείο βρίσκεται εντός του παραλληλογράμμου. Να γραφεί πρόγραμμα το οποίο διαβάσει τις τιμές δύο ορθογώνιων παραλληλογράμμων Π1 (x1, y1, w1, h1) και Π2 (x2, y2, w2, h2). Ελέγχει εάν α) το Π2 εσωκλείεται εντός του Π1 ή β) το Π1 εσωκλείεται εντός του Π2 ή γ) τα παραλληλόγραμμα είναι "ξένα" μεταξύ τους (δεν έχουν κανένα κοινό σημείο)



31η Σειρά Ασκήσεων (ανακεφαλαιωτικές)

1. Να γραφεί πρόγραμμα το οποίο διαβάσει τον ακέραιο A[10] με θετικές τιμές και τον ταξινομεί κατά φθίνουσα σειρά με βάση τον αριθμό των διαιρετών τους. Οι τιμές με ίσο αριθμό διαιρετών να ταξινομούνται κατά φθίνουσα σειρά (sort_by_diviaretos.psc). Π.χ. για τιμές εισόδου: 1,2,...,10:
 Arithmos 10 diaretos 4, Arithmos 8 diaretos 4, Arithmos 6 diaretos 4, Arithmos 9 diaretos 3, Arithmos 4 diaretos 3, Arithmos 7 diaretos 2, Arithmos 5 diaretos 2, Arithmos 3 diaretos 2, Arithmos 2 diaretos 2, Arithmos 1 diaretos 1

2. Από μία **χαλαζόπτωση** επλήγησαν 100 ελαιοπαραγωγοί και το κράτος αποφάσισε να τους αποζημιώσει **αναλογικά** και με βάση των αριθμών των δέντρων του καθενός. Να γραφεί πρόγραμμα που διαβάσει σε πίνακες: τα ονόματα και των αριθμών των δέντρων των 100 ελαιοπαραγωγών, καθώς και το συνολικό ποσό της αποζημίωσης. Να εμφανίζει τους ελαιοπαραγωγούς και την αποζημίωση που τους αναλογεί κατά φθίνουσα σειρά αποζημιώσεων. Όσοι λαμβάνουν την ίδια αποζημίωση να εμφανίζονται αλφαβητικά.



3. Το **Παιχνίδι της Ζωής**: ένας οργανισμός αποτελείται από μία επιφάνεια με $N \times M$ κύτταρα. Σε κάθε βήμα του παιχνιδιού υπολογίζουμε τα κύτταρα τα οποία θα υπάρχουν στην επόμενη γενιά. Το αν ένα κύτταρο θα υπάρχει στην επόμενη γενιά εξαρτάται από τον αριθμό των γειτονικών του κυττάρων που έχει ο οργανισμός ζωντανά σε αυτήν τη γενιά. Πιο συγκεκριμένα, οι κανόνες που καθορίζουν αν ένα κύτταρο θα είναι ζωντανό ή όχι στην επόμενη γενιά είναι οι ακόλουθοι:

- κάθε ζωντανό κύτταρο με λιγότερο από δύο ζωντανούς γείτονες πεθαίνει από έλλειψη γειτόνων,
- κάθε ζωντανό κύτταρο με δύο ή τρεις ζωντανούς γείτονες επιβιώνει και στην επόμενη γενιά,
- κάθε ζωντανό κύτταρο με περισσότερους από τρεις ζωντανούς γείτονες πεθαίνει από υπερπληθυσμό,
- κάθε νεκρό κύτταρο με ακριβώς τρεις ζωντανούς γείτονες μετατρέπεται στην επόμενη γενιά σε ζωντανό κύτταρο.

Να γραφεί πρόγραμμα που διαβάσει έναν ακέραιο $A[100, 200]$ με τιμές 0/1 (0: νεκρό, 1: ζωντανό κύτταρο). Διαβάσει τη θέση ενός κυττάρου (γραμμή, στήλη) και αλλάζει και εμφανίζει την κατάσταση του κυττάρου (ζωντανό/νεκρό) στην επόμενη γενιά. Η εισαγωγή των δεδομένων να γίνεται με έλεγχο εγκυρότητας.

