



3^ο ΦΥΛΛΑΔΙΟ ΑΣΚΗΣΕΩΝ - ΠΙΝΑΚΕΣ

ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C

ΑΣΚΗΣΗ - 1

Μονοδιάστατος πίνακας 50 θέσεων περιέχει τις ποσότητες παραγωγής (τύπου int) ενός προϊόντος. Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα παρακάτω :

1. Εισαγωγή των στοιχείων του πίνακα με χρήση τυχαίων αριθμών (περιοχή τιμών 0-100).
2. Εμφάνιση των στοιχείων του πίνακα.
3. Εύρεση της μέσης τιμής των στοιχείων του πίνακα.
4. Δημιουργία νέου πίνακα που θα περιέχει όλες τις ποσότητες των οποίων η διαφορά από τη μέση τιμή των στοιχείων του αρχικού πίνακα είναι μεγαλύτερη κατά 20% της μέσης τιμής.
5. Εμφάνιση των στοιχείων του νέου πίνακα καθώς και του πλήθους των στοιχείων του.

ΑΣΚΗΣΗ - 2

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει N ακεραίους αριθμούς σε μονοδιάστατο πίνακα N θέσεων (N = γνωστό). Στη συνέχεια θα εισάγεται ένας ακέραιος αριθμός z και το πρόγραμμα θα βρίσκει αν ο αριθμός υπάρχει στον πίνακα και σε ποια θέση. Αν ο αριθμός υπάρχει περισσότερες από μία φορές να εμφανίζεται μόνον η θέση όπου εμφανίζεται για πρώτη φορά.

ΑΣΚΗΣΗ - 3

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει N ακεραίους αριθμούς σε μονοδιάστατο πίνακα N θέσεων (N = γνωστό). Στη συνέχεια θα εισάγεται στο πρόγραμμα ένας ακέραιος αριθμός z και το πρόγραμμα θα βρίσκει αν ο αριθμός z υπάρχει στον πίνακα και σε ποια θέση. Αν ο αριθμός υπάρχει περισσότερες από μία φορές θα εμφανίζονται όλες οι θέσεις του πίνακα όπου υπάρχει ο αριθμός z.

ΑΣΚΗΣΗ - 4

Να γραφεί πρόγραμμα σε γλώσσα C που θα εναλλάσσει τα περιεχόμενα ενός μονοδιάστατου πίνακα ακεραίων N θέσεων (N = γνωστό) ως εξής :

- Το 1ο στοιχείο με το N-οστό στοιχείο
- Το 2ο στοιχείο με το N-1 στοιχείο κλπ

Το πρόγραμμα θα εμφανίζει τον πίνακα πριν και μετά την εναλλαγή των στοιχείων του.

ΑΣΚΗΣΗ - 5

Να γραφεί πρόγραμμα σε γλώσσα C που θα διαχωρίζει ένα δεδομένο πίνακα ακεραίων τιμών N θέσεων (N =γνωστό) σε δύο νέους πίνακες που θα περιέχουν αντίστοιχα τις άρτιες και τις περιττές τιμές του αρχικού πίνακα. Το πρόγραμμα θα εμφανίζει τον αρχικό πίνακα, τους δύο νέους πίνακες καθώς και το πλήθος των στοιχείων τους.

ΑΣΚΗΣΗ - 6

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει N ακεραίους αριθμούς σε μονοδιάστατο πίνακα N θέσεων ($N = \text{γνωστό}$) και θα τον ταξινομεί κατά αύξουσα σειρά των στοιχείων του. Στη συνέχεια θα εισάγεται ένας ακέραιος αριθμός A και :

- αν ο αριθμός A υπάρχει στον πίνακα θα βρῖσκεται η θέση του, με τις λιγότερες δυνατές προσπάθειες (δυναμική αναζήτηση). Αν ο αριθμός υπάρχει περισσότερες από μία φορές θα εμφανίζεται μόνον η θέση όπου θα εντοπιστεί για πρώτη φορά.
- αν ο αριθμός A δεν υπάρχει στον πίνακα να εισάγεται σε αυτόν στην κατάλληλη θέση ώστε ο πίνακας να διατηρείται ταξινομημένος.

ΑΣΚΗΣΗ - 7

Για ορισμένους ακέραιους αριθμούς ισχύει ότι το άθροισμα των ψηφίων του αριθμού διαιρεί τον ίδιο τον αριθμό π.χ. για τον αριθμό 1729 ισχύει : $1+7+2+9=19$ και $1729/19=91$.

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει σε έναν μονοδιάστατο πίνακα όλους αυτούς τους αριθμούς που πληρούν την παραπάνω ιδιότητα και είναι μικρότεροι του 10000. Ο αλγόριθμος θα εμφανίζει στο τέλος τον πίνακα καθώς και το πλήθος των στοιχείων του.

ΑΣΚΗΣΗ - 8

Θεωρήστε έναν ακέραιο και θετικό αριθμό n που έχει d ψηφία. Διαχωρίστε το τετράγωνο του αριθμού σε δύο τμήματα, το πρώτο από τα δεξιά που έχει d ψηφία και το δεύτερο από τα αριστερά που έχει d ή $d-1$ ψηφία. Προσθέστε αυτούς τους δύο αριθμούς και ελέγξτε αν το άθροισμά τους είναι ίσο με τον αρχικό αριθμό n . Παραδείγματα 9 ($9^2 = 81$, $8 + 1 = 9$), 45 ($45^2 = 2025$, $20 + 25 = 45$).

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει σε έναν μονοδιάστατο πίνακα όλους αυτούς τους αριθμούς που πληρούν την παραπάνω ιδιότητα και είναι μικρότεροι του 100. Ο αλγόριθμος θα εμφανίζει στο τέλος τον πίνακα καθώς και το πλήθος των στοιχείων του.

ΑΣΚΗΣΗ - 9

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει όλες τις τετράδες ακεραίων αριθμών, έστω $(\alpha, \beta, \gamma, \delta)$, που :

- ανήκουν στο διάστημα $[1,20]$,
- είναι διαφορετικές μεταξύ τους (δηλαδή να αποφύγετε την επανάληψη τετράδων όπως $(\alpha, \beta, \gamma, \delta)$, $(\alpha, \gamma, \delta, \beta)$, $(\gamma, \alpha, \beta, \delta)$ κλπ), και
- ικανοποιούν την ιδιότητα: $\alpha^2 + \beta^2 + \gamma^2 = \delta^2$

Παράδειγμα : $1^2+4^2+8^2=9^2$, αλλά όχι $1^2+2^2+2^2=3^2$, διότι επαναλαμβάνεται το 2.

Επίσης:

1. Να βρεθεί και να εμφανιστεί το πλήθος αυτών των διαφορετικών μεταξύ τους τετράδων.
2. Να δημιουργηθεί ένας μονοδιάστατος αριθμητικός πίνακας ακεραίων που θα περιέχει τις πρώτες 10 τέτοιες τετράδες (αν υπάρχουν λιγότερες τότε θα περιέχει ακριβώς όσες υπάρχουν)

ΑΣΚΗΣΗ - 10

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει όλες τις θετικές Πυθαγόρειες τριάδες $(m^2 - n^2, 2mn, m^2 + n^2)$ για $1 \leq m, n \leq 30$.

Οι τριάδες αυτές θα καταχωρούνται διαδοχικά σε έναν μονοδιάστατο αριθμητικό πίνακα. Στη συνέχεια θα εμφανίζονται όλες οι τριάδες που έχουν καταχωρηθεί στον πίνακα καθώς και το πλήθος αυτών των τριάδων.

ΑΣΚΗΣΗ - 11

Να γραφεί πρόγραμμα σε γλώσσα C που θα απομακρύνει από ένα μονοδιάστατο αριθμητικό πίνακα ακεραίων N θέσεων (N = γνωστό) όλες τις πολλαπλές ίδιες τιμές δημιουργώντας ένα νέο πίνακα όπου κάθε αριθμός θα εμφανίζεται μόνο μία φορά.

ΑΣΚΗΣΗ - 12

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει N ακεραίους αριθμούς σε μονοδιάστατο πίνακα N θέσεων (N = γνωστό) και θα βρίσκει το μεγαλύτερο στοιχείο καθώς και τη θέση που κατέχει στον πίνακα. Σε περίπτωση που υπάρχουν περισσότερα από ένα μέγιστα στοιχεία θα βρίσκονται και θα εμφανίζονται όλες οι αντίστοιχες θέσεις, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης του πίνακα.

ΑΣΚΗΣΗ - 13

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει N ακεραίους αριθμούς σε μονοδιάστατο πίνακα N θέσεων (N = γνωστό). Το πρόγραμμα θα βρίσκει το μεγαλύτερο στοιχείο, τη θέση που κατέχει στον πίνακα και στη συνέχεια το αμέσως μικρότερο διαφορετικό στοιχείο καθώς και τη θέση που κατέχει. Σε περίπτωση που υπάρχουν περισσότερα από ένα στοιχεία σε κάθε μία από τις δύο αναζητήσεις θα βρίσκονται και θα εμφανίζονται όλα, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης του πίνακα.

ΑΣΚΗΣΗ - 14

Πρώτοι αριθμοί (prime numbers) είναι εκείνοι που έχουν ως γνήσιους διαιρέτες τον εαυτό τους και την μονάδα. Ο αλγόριθμος του Ερατοσθένη βρίσκει όλους τους πρώτους αριθμούς που είναι μικρότεροι ή ίσοι ενός δοθέντος αριθμού N. Να γραφεί πρόγραμμα σε γλώσσα C που θα εμφανίζει όλους τους πρώτους αριθμούς μεταξύ 1 και 1000. Για την εύρεση να χρησιμοποιηθεί ο αλγόριθμος του Ερατοσθένη που περιγράφεται στη συνέχεια υπό μορφή βημάτων.

- Δημιουργείστε μια λίστα των αριθμών (δηλ. 1-N)
- Ο αριθμός 2 είναι πρώτος αλλά όχι τα πολλαπλάσιά του (4,6,8...). Διαγράψτε όλα τα πολλαπλάσια του 2 (δηλ. 4,6,8,...)
- Βρείτε τον πρώτο κατά σειρά αριθμό που απέμεινε στη λίστα (δηλ. 3,...) μετά από αυτούς που έχουν διαγραφεί, ελέγξτε αν είναι πρώτος και διαγράψτε όλα τα πολλαπλάσιά του.
- Επαναλάβετε το βήμα 3 μέχρι να βρεθεί ο πρώτος ακέραιος αριθμός που δεν έχει απορριφθεί και το τετράγωνό του είναι μεγαλύτερο του N (συνθήκη τερματισμού).
- Όλοι οι αριθμοί που απομένουν στη λίστα είναι οι πρώτοι αριθμοί μεταξύ 2 και N.

ΑΣΚΗΣΗ - 15

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει σε μονοδιάστατο πίνακα 300 θέσεων τις ποσότητες παραγωγής ενός προϊόντος για ένα έτος. Κατά σειρά σε κάθε 25 θέσεις αντιστοιχεί ένας μήνας του έτους δηλ. οι θέσεις 1-25 αντιστοιχούν στον Ιανουάριο, οι θέσεις 26-50 στον Φεβρουάριο κλπ. Να γίνει επίσης εισαγωγή της επιθυμητής μηνιαίας ποσότητας παραγωγής για κάθε έναν από τους 12 μήνες. Το πρόγραμμα θα βρίσκει και θα εμφανίζει:

- σε ποια ημέρα κάθε μήνα η συνολική μηνιαία ποσότητα παραγωγής υπερβαίνει την επιθυμητή
- ποια είναι η μηνιαία πλεονάζουσα παραγωγή.

ΑΣΚΗΣΗ - 16

Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει σε δύο μονοδιάστατους πίνακες N ζεύγη ακεραίων αριθμών (N = γνωστό), έστω M, P που αντιστοιχούν στον αριθμό μήνα M (αποδεικτές τιμές 1-12) και σε μία ποσότητα παραγωγής P που παρήχθη τον μήνα M (P>0 και P>>N). Για κάθε μήνα υπάρχουν περισσότερα από ένα και άγνωστα σε πλήθος ζεύγη τιμών. Τα δεδομένα εισάγονται με τυχαία σειρά, δηλαδή δεν εισάγονται κατά σειρά δεδομένα του μήνα 1 μετά του μήνα 2 κλπ. Να βρεθεί σε ποιόν μήνα του έτους ξεκινώντας από την αρχή του έτους η συνολική παραγωγή υπερβαίνει μια δεδομένη τιμή X.

ΑΣΚΗΣΗ - 17

Η παραγωγή ενός προϊόντος για N περιόδους παραγωγής (N = γνωστό) καταχωρείται σε ένα μονοδιάστατο πίνακα ακεραίων θετικών αριθμών A. Να γραφεί πρόγραμμα σε γλώσσα C που :

- θα εισάγει στον πίνακα A τα στοιχεία πραγματοποιώντας όλους τους απαραίτητους ελέγχους ώστε οι αριθμοί να είναι θετικοί.
- θα δημιουργεί στη συνέχεια ένα νέο πίνακα B που θα περιέχει στις αντίστοιχες θέσεις την τρέχουσα μέση τιμή των στοιχείων που έχουν εισαχθεί.
- θα βρίσκει και θα εμφανίζει πόσες και ποιες περιόδους η παραγωγή διαφέρει κατά μέγιστο 10% της τρέχουσας μέσης τιμής της ίδιας περιόδου.

ΠΑΡΑΔΕΙΓΜΑ για N=6

| Περίοδος | 1 | 2 | 3 | 4 | 5 | 6 |
|--------------------|-----|-----|-----|------|-----|-----|
| Παραγωγή | 100 | 80 | 90 | 80 | 200 | 170 |
| Τρέχουσα μέση τιμή | 100 | 90 | 90 | 87.5 | 110 | 120 |
| Διαφορά <10% | ναι | όχι | ναι | ναι | όχι | όχι |

ΑΣΚΗΣΗ - 18

Η χρονική διάρκεια λειτουργίας και η χρονική διάρκεια συντήρησης μιας εργαλειομηχανής (σε ώρες) είναι γνωστές για κάθε μια από N χρονικές περιόδους (N = γνωστό). Να γραφεί πρόγραμμα σε γλώσσα C που θα εισάγει τα στοιχεία αυτά σε δύο μονοδιάστατους πίνακες. Ο υπεύθυνος παραγωγής επιθυμεί η συνολική χρονική διάρκεια συντήρησης να μην υπερβαίνει το 15% της συνολικής χρονικής διάρκειας λειτουργίας της εργαλειομηχανής. Για τους λόγους αυτούς θέλει να γνωρίζει ποια είναι για κάθε χρονική περίοδο η τρέχουσα συνολική χρονική διάρκεια λειτουργίας και η τρέχουσα συνολική χρονική διάρκεια συντήρησης. Το πρόγραμμα θα πρέπει να εμφανίζει τα δεδομένα καθώς και τις τιμές που υπάρχουν στο παράδειγμα που ακολουθεί, μέχρι εκείνη τη χρονική περίοδο στην οποία η συνολική χρονική διάρκεια συντήρησης γίνεται για πρώτη φορά μικρότερη του 15% της συνολικής χρονικής διάρκειας λειτουργίας.

Παράδειγμα:

| ΧΡΟΝΙΚΗ ΠΕΡΙΟΔΟΣ | 1 | 2 | 3 | 4 | 5 | |
|--------------------------------------|------|------|------|------|-----|--|
| Λειτουργία | 150 | 110 | 140 | 130 | ... | |
| Τρέχουσα συνολική λειτουργία | 150 | 260 | 400 | 530 | | |
| Συντήρηση | 28 | 22 | 20 | 5 | ... | |
| Τρέχουσα συνολική συντήρηση | 28 | 50 | 70 | 75 | | |
| % συνολ. συντήρηση / συν. λειτουργία | 18.6 | 19.2 | 17.5 | 14.1 | | |

ΑΣΚΗΣΗ - 19

Η ποσότητα παραγωγής ενός προϊόντος και η ποσότητα του προϊόντος που απορρίπτεται από τον ποιοτικό έλεγχο είναι γνωστές για N σε πλήθος ώρες ($N =$ γνωστό). Η μονάδα παραγωγής του προϊόντος επιθυμεί η συνολική ποσότητα που απορρίπτεται να μην υπερβαίνει το 20% της συνολικής ποσότητας παραγωγής για όλη την χρονική περίοδο των N ωρών. Ωστόσο διάφορα γεγονότα δεν επιτρέπουν στην παραγωγή να επιτύχει από την πρώτη ώρα λειτουργίας τον επιθυμητό στόχο. Για τους λόγους αυτούς ο προϊστάμενος της παραγωγής θέλει να γνωρίζει ποιο είναι για κάθε ώρα το ποσοστό % απόρριψης της τρέχουσας συνολικής παραγωγής.

Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα παρακάτω:

- θα εισάγει σε δύο μονοδιάστατους πίνακες ακεραίων θετικών αριθμών N τιμές που αφορούν την ποσότητα παραγωγής και την αντίστοιχη απορριπτόμενη ποσότητα για N ώρες παραγωγής. Κατά την εισαγωγή πρέπει να γίνονται οι απαραίτητοι έλεγχοι εγκυρότητας τιμών.
- θα εμφανίζει στην οθόνη τις τιμές που έχουν εισαχθεί καθώς και τις υπολογιζόμενες τιμές που υπάρχουν στο παρακάτω παράδειγμα μέχρι την ώρα (αν αυτή υπάρχει) κατά την οποία η συνολική απορριπτόμενη ποσότητα παραγωγής γίνεται για πρώτη φορά μικρότερη του 20% της συνολικής ποσότητας παραγωγής.

Παράδειγμα:

| ΩΡΑ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... |
|--|-----|-----|-----|-----|------|---|---|-----|
| Ποσότητα παραγωγής | 100 | 150 | 140 | 80 | 110 | | | ... |
| Τρέχουσα συνολ. ποσότητα παραγωγής | 100 | 250 | 390 | 470 | 580 | | | |
| Απορριπτόμενη ποσότητα | 25 | 45 | 35 | 5 | 3 | | | ... |
| Τρέχουσα συνολ. απορ. ποσότητα παραγωγής | 25 | 70 | 105 | 110 | 113 | | | |
| ποσοστό απόρριψης % | 25 | 28 | 27 | 23 | 19.5 | | | |

ΑΣΚΗΣΗ - 20

Για τη μέτρηση της κυκλοφοριακής πυκνότητας σε διασταυρώσεις οδών μετρώνται τα αυτοκίνητα που διέρχονται σε ορισμένα χρονικά διαστήματα και υπολογίζεται ο μέσος όρος. Τα δεδομένα των μετρήσεων περιλαμβάνουν τον κωδικό της διασταύρωσης, τον χρόνο έναρξης και λήξης της μέτρησης και το πλήθος των αυτοκινήτων, όπως φαίνεται στον παρακάτω πίνακα :

| Κωδικός διασταύρωσης | Χρόνος έναρξης | Χρόνος λήξης | Πλήθος αυτοκινήτων |
|----------------------|----------------|--------------|--------------------|
| 27 | 1330 | 1337 | 27 |
| 14 | 1422 | 1435 | 52 |
| 27 | 1407 | 1502 | 95 |
| 27 | 1507 | 1514 | 67 |
| 14 | 1503 | 1509 | 57 |
| 44 | 2117 | 2233 | 65 |

Οι χρόνοι έναρξης και λήξης αποθηκεύονται ως τετραψήφιοι ακέραιοι όπου τα 2 πρώτα ψηφία δηλώνουν την ώρα και τα 2 επόμενα τα πρώτα λεπτά. Να θεωρήσετε ως δεδομένο ότι ο χρόνος λήξης είναι πάντοτε μεγαλύτερος του χρόνου έναρξης (δηλαδή δεν χρειάζεται έλεγχος κατά την εισαγωγή των τιμών).

Τα δεδομένα καταχωρούνται σε 4 μονοδιάστατους πίνακες N στοιχείων ($N =$ γνωστό) και αφορούν K διασταυρώσεις (η τιμή του K ($K \leq N$) δεν είναι γνωστή και κατά την εισαγωγή των δεδομένων πρέπει να προκύπτει τιμή $K > 2$).

Να γράψετε πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει τη μέση τιμή αυτοκινήτων ανά λεπτό για μια καθορισμένη διασταύρωση, έστω D , ο κωδικός της οποίας θα εισάγεται από το πληκτρολόγιο (χρήση της scanf). Αν ο κωδικός της διασταύρωσης δεν υπάρχει στα δεδομένα, τότε θα εμφανίζεται το μήνυμα : «ΔΕΝ ΥΠΑΡΧΟΥΝ ΜΕΤΡΗΣΕΙΣ ΓΙΑ ΤΗ ΔΙΑΣΤΑΥΡΩΣΗ» D .

ΑΣΚΗΣΗ - 21

Να βρεθούν τα αποτελέσματα που θα προκύψουν από την εκτέλεση του παρακάτω προγράμματος. Για να δείξετε την εξέλιξη των υπολογισμών να δημιουργήσετε έναν πίνακα των μεταβλητών όπου θα καταγράψετε όλες τις διαδοχικές τους τιμές.

```
#include <stdio.h>
#define N 10
int i=0,j, k, a[N],b[N], element=0, start=0, finish=0;

void main()
{
    a[start]=1; a[finish]=1; printf("%4d \n",a[start]);
    for (k=0;k<N;k++) b[k]=a[k];

    while ( (element <50) && (i<N) )
    {
        i++;
        a[start]=b[finish];
        printf("%4d ",a[start]);
        for (j=1;j<=i;j++)
            {
                a[j]=a[j-1]+b[j-1];
                printf(" %3d ",a[j]);
            }
        printf("\n");
        for (k=0;k<N;k++) b[k]=a[k];
        element=a[i];finish=i;
    }
}
```

ΑΣΚΗΣΗ - 22

Να γίνει εισαγωγή τιμών σε 2 μονοδιάστατους πίνακες θετικών πραγματικών αριθμών N θέσεων ($N =$ γνωστό). Ο 1ος πίνακας περιέχει τιμές για το κόστος λειτουργίας μιας εργαλειομηχανής για κάθε μία από N εβδομάδες λειτουργίας της (με αριθμούς εβδομάδων από 1 έως N) ενώ ο 2ος το κόστος συντήρησής της αντίστοιχα. Κατά την εισαγωγή να γίνουν όλοι οι απαραίτητοι έλεγχοι ώστε οι τιμές που εισάγονται να είναι θετικές ή μηδενικές.

Το πρόγραμμα, σε γλώσσα C :

1. θα βρίσκει και θα εμφανίζει το πλήθος των εβδομάδων με κόστος λειτουργίας μεγαλύτερο μιας δεδομένης γνωστής τιμής $LIMIT$.
2. θα δημιουργεί ένα νέο πίνακα που θα περιλαμβάνει τους αριθμούς εβδομάδων με κόστος συντήρησης μικρότερο του μέσου κόστους συντήρησης των N εβδομάδων.
3. θα εμφανίζει τα περιεχόμενα του νέου πίνακα.

ΑΣΚΗΣΗ - 23

Να γίνει εισαγωγή τιμών σε μονοδιάστατο πίνακα ακεραίων θετικών αριθμών στην περιοχή [1-99], N στοιχείων (N = γνωστό και άρτιος αριθμός). Η εισαγωγή των αριθμών μπορεί να γίνει με οποιονδήποτε τρόπο (εντολή scanf, χρήση συνάρτησης δημιουργίας τυχαίων αριθμών), αρκεί να υπάρχει έλεγχος εγκυρότητας τιμών.

Να θεωρήσετε ότι δίνονται επίσης :

- μία δεδομένη γνωστή τιμή t , $0 < t < N$ και
- ένα γνωστό άνω όριο M.

Το πρόγραμμα, σε γλώσσα C, θα δημιουργεί νέο πίνακα που θα περιλαμβάνει τα αθροίσματα των στοιχείων του αρχικού πίνακα που ξεκινούν από τις θέσεις 1, 2, 3,...N και απέχουν απόσταση t. Αν η διαδικασία φθάσει ή υπερβεί το τέλος του πίνακα θα συνεχίζεται πάλι από την αρχή μέχρι το άθροισμα να υπερβεί την τιμή M.

Παράδειγμα. Αρχικός πίνακας , N=10, t=4, M=200

| | | | | | | | | | |
|----|---|----|----|----|----|----|----|----|----|
| 14 | 7 | 91 | 27 | 53 | 72 | 18 | 52 | 33 | 44 |
|----|---|----|----|----|----|----|----|----|----|

Αθροίσματα :

1° άθροισμα : $14+53+33+91+18=209$

2° άθροισμα : $7+72+44+27+52=202$

Νέος πίνακας :

| | | | | |
|-----|-----|-----|-----|------|
| 209 | 202 | ... | ... | |
|-----|-----|-----|-----|------|

ΑΣΚΗΣΗ - 24

Να γίνει εισαγωγή τιμών σε μονοδιάστατο πίνακα ακεραίων θετικών αριθμών στην περιοχή [1-999], N στοιχείων (N = γνωστό). Η εισαγωγή των αριθμών μπορεί να γίνει με οποιονδήποτε τρόπο (εντολή scanf, χρήση συνάρτησης δημιουργίας τυχαίων αριθμών), αρκεί να υπάρχει έλεγχος εγκυρότητας τιμών.

Να δημιουργηθεί νέος πίνακας που θα περιλαμβάνει τις διαφορές των στοιχείων του αρχικού πίνακα, ανά ζεύγη, κατ' απόλυτη τιμή, ως εξής (διαφορά του 1ου με το 2ο στοιχείο, του 3ου στοιχείου με το 4ο στοιχείο κ.ο.κ.), ταξινομημένες σε αύξουσα διάταξη, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης. Αν ο αριθμός N είναι περιττός τότε θα υπολογιστούν οι διαφορές για τα πρώτα N-1 στοιχεία.

Παράδειγμα:

Αρχικός πίνακας , N=10 :

| | | | | | | | | | |
|----|----|----|----|----|----|---|---|----|----|
| 37 | 71 | 19 | 25 | 53 | 19 | 8 | 5 | 77 | 24 |
|----|----|----|----|----|----|---|---|----|----|

Διαφορές :

$$|37 - 71|=34, \quad |19 - 25|=6, \quad |53 - 19|=34, \quad |8 - 5|=3, \quad |77 - 24|=53,$$

Νέος πίνακας σε αύξουσα διάταξη:

| | | | | |
|---|---|----|----|----|
| 3 | 6 | 34 | 34 | 53 |
|---|---|----|----|----|

ΑΣΚΗΣΗ - 25

Θεωρείστε τους 10 πρώτους ακέραιους αριθμούς (1-10) σε μία σειρά με αριθμό σειράς $i=1$. Η επόμενη σειρά περιλαμβάνει 10 ακέραιους αριθμούς που είναι οι συνεχόμενοι ακέραιοι αριθμοί που ξεκινούν από τον $i+1$ αριθμό της αμέσως προηγούμενης σειράς. Οι σειρές δημιουργούνται επαναληπτικά μέχρις ότου το άθροισμα των στοιχείων που βρίσκονται στην i θέση κάθε σειράς υπερβεί μια δεδομένη τιμή n (n =γνωστό, δίνεται ως σταθερά στην αρχή του αλγορίθμου ή του προγράμματος). Π.χ. για $n=30$ δημιουργούνται οι παρακάτω σειρές:

| | | | | | | | | | | |
|-------|----|----|----|----|----|----|----|----|----|----|
| $i=1$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $i=2$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| $i=3$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| $i=4$ | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| $i=5$ | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

Το άθροισμα $1+3+6+10+15=35$ είναι μεγαλύτερο του $n=30$ και η επανάληψη σταματά.

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρρίσκει και θα εμφανίζει τις σειρές αυτές (τα στοιχεία σε πλαίσιο προφανώς δεν θα εμφανίζονται σε πλαίσιο!). Στο τέλος θα πρέπει να εμφανίζεται και το τελικό άθροισμα των στοιχείων που φαίνονται σε πλαίσιο στο παραπάνω παράδειγμα.

ΑΣΚΗΣΗ - 26

Ένας αριθμητικός πίνακας ακεραίων N θέσεων ($N=γνωστό$) περιέχει τιμές στο διάστημα $[1-99]$. Να γίνει εισαγωγή τιμών στον πίνακα χρησιμοποιώντας τυχαίους αριθμούς (συνάρτηση $rand()$).

Θεωρείστε ότι οι τιμές του πίνακα:

- Είναι μοναδικές δηλ. δεν υπάρχει τιμή που επαναλαμβάνεται
- Οι τιμές ανά περιοχές, είναι διατεταγμένες σε αύξουσα σειρά και κάθε νέα αύξουσα ακολουθία τιμών ξεκινά με τιμή που είναι μεγαλύτερη από την πρώτη τιμή της προηγούμενης αύξουσας ακολουθίας τιμών

π.χ. για $N=15$ μια μορφή του πίνακα μπορεί να είναι:

| | | | | | | | | | | | | | | |
|---|---|---|---|----|----|---|---|----|----|---|----|----|----|----|
| 1 | 3 | 5 | 9 | 12 | 16 | 2 | 6 | 10 | 15 | 4 | 13 | 14 | 11 | 17 |
|---|---|---|---|----|----|---|---|----|----|---|----|----|----|----|

Οι αύξουσες ακολουθίες είναι οι εξής τέσσερις :

1, 3, 5, 9, 12, 16

2, 6, 10, 15

4, 13, 14

11, 17

Να γίνει εισαγωγή μιας τιμής που δεν υπάρχει στον πίνακα (θα γίνεται έλεγχος μη ύπαρξης της τιμής αυτής) με τέτοιο τρόπο ώστε:

- Το πλήθος των τιμών σε κάθε υπάρχουσα αύξουσα ακολουθία να παραμείνει το ίδιο
- Οι τιμές σε όλες τις ακολουθίες να παραμείνουν σε αύξουσα σειρά
- Η νέα τιμή να εισαχθεί στην κατάλληλη θέση στην 1^η ακολουθία, εκτοπίζοντας μία τιμή η οποία θα μεταφερθεί στην κατάλληλη θέση στην 2^η ακολουθία κ.ο.κ.

π.χ. αν εισαχθεί η τιμή 8 τότε ο νέος πίνακας θα έχει τη μορφή:

| | | | | | | | | | | | | | | | |
|---|---|---|---|----|----|---|---|---|----|---|----|----|----|----|----|
| 1 | 3 | 5 | 8 | 12 | 16 | 2 | 6 | 9 | 15 | 4 | 10 | 14 | 11 | 13 | 17 |
|---|---|---|---|----|----|---|---|---|----|---|----|----|----|----|----|

και οι αύξουσες ακολουθίες θα είναι οι εξής πέντε :

1, 3, 5, 8, 12, 16

2, 6, 9, 15

4, 10, 14

11, 13

17

Το πρόγραμμα, σε γλώσσα C, θα πρέπει να εμφανίζει:

- την αρχική μορφή του πίνακα
- τη νέα μορφή του πίνακα
- το πλήθος των αυξουσών ακολουθιών πριν και μετά από την εισαγωγή της νέας τιμής.

ΑΣΚΗΣΗ - 27

Η παραγωγή ενός προϊόντος για N περιόδους παραγωγής ($N = \text{γνωστό}$) καταχωρείται σε ένα μονοδιάστατο πίνακα πραγματικών θετικών αριθμών A . Σε κάθε μία από τις N περιόδους αντιστοιχούν 2 βάρδια παραγωγής, η ημερήσια βάρδια και η νυχτερινή βάρδια. Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα εξής:

- θα εισάγει στον πίνακα A τα δεδομένα παραγωγής, σε μορφή ζευγών ανά περίοδο παραγωγής. Το 1^ο στοιχείο κάθε ζεύγους αντιστοιχεί στην ημερήσια παραγωγή και το 2^ο στοιχείο στη νυχτερινή παραγωγή. Πρέπει να εξασφαλίσετε ότι όλα τα δεδομένα θα είναι θετικοί πραγματικοί αριθμοί στην περιοχή $[30.0, 200.0]$.
- θα δημιουργεί στη συνέχεια δύο νέους πίνακες με ονόματα H και B που θα περιέχουν :
 - ο πίνακας H τις ποσότητες ημερήσιας παραγωγής που είναι μεγαλύτερες κατά 15% από τον μέσο όρο της συνολικής ημερήσιας παραγωγής των N περιόδων.
 - Αν $LIMIT (= \text{ΓΝΩΣΤΟ})$ είναι η επιθυμητή συνολική παραγωγή των N περιόδων κατά τη νυχτερινή βάρδια, ο πίνακας B θα περιέχει μόνον εκείνες τις ποσότητες νυχτερινής παραγωγής που αποτελούν το πλεόνασμα της επιθυμητής συνολικής παραγωγής κατά τη νυχτερινή βάρδια.
- Θα εμφανίζει τα στοιχεία των πινάκων H και B .

ΠΑΡΑΔΕΙΓΜΑ για $N=6$, $LIMIT = 250.0$

| Περίοδος | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | |
|----------|------|-----|------|-----|------|------|-----|-----|------|-----|------|------|
| Βάρδια | H | B | H | B | H | B | H | B | H | B | H | B |
| Παραγωγή | 150. | 80. | 190. | 70. | 195. | 170. | 30. | 55. | 100. | 95. | 145. | 125. |

Για το ερώτημα 2b : $80.0 + 70.0 + 170.0 = 320.0 > 250.0$ οπότε ο πίνακας B θα είναι :

| | | | |
|--------------------------|------|------|-------|
| 70.0 (= 320.0 - 250.0) | 55.0 | 95.0 | 125.0 |
|--------------------------|------|------|-------|

ΑΣΚΗΣΗ - 28

Να δημιουργηθεί ένας πίνακας ακέραιων θετικών αριθμών, έστω A , πλήθους N θέσεων ($N = \text{γνωστό}$) στον οποίο θα καταχωρούνται τιμές στο διάστημα $[0-99]$ με τους απαραίτητους ελέγχους εγκυρότητας.

Θεωρώντας γνωστό έναν αριθμό M ($M > 100$):

1. να δημιουργηθούν δύο νέοι πίνακες B και C :
 - a. Ο πίνακας B θα περιλαμβάνει εκείνες τις θέσεις του αρχικού πίνακα στις οποίες το άθροισμα των στοιχείων γίνεται μεγαλύτερο του M . Όταν βρισκείται μια τέτοια θέση οι υπολογισμοί, για το υπόλοιπο του πίνακα, θα γίνονται από την αρχή, συμπεριλαμβάνοντας και την τρέχουσα θέση.
 - b. Ο πίνακας C θα περιλαμβάνει τις αντίστοιχες τιμές των αθροισμάτων, όπως αυτά προκύπτουν από τη σχετική διαδικασία.
2. Να εμφανιστούν τα περιεχόμενα των πινάκων B και C καθώς και η μέση τιμή των στοιχείων του πίνακα C .

Παράδειγμα: για $N=14$, $M=110$

Αρχικός πίνακας A :

| | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| ΘΕΣΗ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| ΤΙΜΗ | 11 | 45 | 20 | 32 | 99 | 34 | 27 | 77 | 11 | 5 | 89 | 72 | 65 | 59 |

Νέος πίνακας B

| | | | | | |
|---|---|---|----|----|----|
| 5 | 6 | 8 | 11 | 12 | 14 |
|---|---|---|----|----|----|

$11+45+20+32+99=207 > 110$, $99+34=133 > 110$, $34+27+77=138 > 110$ κλπ

Νέος πίνακας C

| | | | | | | |
|-----|-----|-----|-----|-----|-----|------|
| 207 | 133 | 138 | 182 | 161 | ... | |
|-----|-----|-----|-----|-----|-----|------|

ΑΣΚΗΣΗ - 29

Η παραγωγή ενός προϊόντος απαιτεί N φάσεις κατεργασίας από M σε πλήθος διαφορετικές εργαλειομηχανές ($N =$ γνωστό, $M =$ γνωστό). Κάθε εργαλειομηχανή μπορεί να χρησιμοποιηθεί περισσότερες από μία φορές για το ίδιο προϊόν. Για κάθε φάση κατεργασίας είναι γνωστά:

1. ο κωδικός της φάσης κατεργασίας (ακέραιος αριθμός στην περιοχή $[1-999]$),
2. ο κωδικός της εργαλειομηχανής που χρησιμοποιείται (ακέραιος αριθμός στην περιοχή $[1-9]$),
3. ο χρόνος κατεργασίας της φάσης, σε sec (ακέραιος αριθμός >0)

Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα εξής:

1. Θα καταχωρεί δεδομένα σε έναν πίνακα ακεραίων με $3*N$ σε πλήθος θέσεις, σύμφωνα με τα παραπάνω. (ΠΡΟΣΟΧΗ!! Εάν χρησιμοποιηθεί η εντολή `scanf_s` είναι απαραίτητος ο έλεγχος εγκυρότητας τιμών. Συνιστάται η κατάλληλη χρήση της συνάρτησης `rand()` για να αποφευχθεί η διαδικασία ελέγχου εγκυρότητας τιμών).
2. Θεωρώντας ότι η σειρά εκτέλεσης των φάσεων κατεργασίας σε κάθε εργαλειομηχανή καθορίζεται από τον κωδικό της φάσης κατεργασίας, με προτεραιότητα στις φάσεις με μικρότερο κωδικό φάσης κατεργασίας:
 - a. Να βρείτε και να εμφανίσετε τη σειρά εκτέλεσης των φάσεων κατεργασίας σε κάθε εργαλειομηχανή, ΧΩΡΙΣ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΕΤΕ ΔΙΑΔΙΚΑΣΙΕΣ ΤΑΞΙΝΟΜΗΣΗΣ! Συγχρόνως να υπολογίσετε και να εμφανίσετε και το συνολικό χρόνο απασχόλησης της κάθε εργαλειομηχανής.

Αριθμητικό ΠΑΡΑΔΕΙΓΜΑ , $N=8$.

Οι χρησιμοποιούμενες εργαλειομηχανές ($M = 4$) είναι οι : 2, 5, 7, 9.

| | | | | | | | | | | | | | | | | | | | | | | | |
|-----|---|----|----|---|----|----|---|----|----|---|-----|-----|---|----|----|---|----|-----|---|---|-----|---|----|
| 157 | 2 | 45 | 19 | 7 | 10 | 14 | 9 | 77 | 91 | 2 | 120 | 305 | 2 | 17 | 72 | 5 | 71 | 345 | 7 | 5 | 773 | 5 | 88 |
|-----|---|----|----|---|----|----|---|----|----|---|-----|-----|---|----|----|---|----|-----|---|---|-----|---|----|

Αποτελέσματα:

| | | |
|--------------------|--------------|-----------------------|
| ΕΡΓΑΛΕΙΟΜΗΧΑΝΗ : 2 | 91, 157, 305 | συνολικός χρόνος: 182 |
| ΕΡΓΑΛΕΙΟΜΗΧΑΝΗ : 5 | 72, 773 | συνολικός χρόνος: 159 |
| ΕΡΓΑΛΕΙΟΜΗΧΑΝΗ : 7 | 19, 345 | συνολικός χρόνος: 15 |
| ΕΡΓΑΛΕΙΟΜΗΧΑΝΗ : 9 | 14 | συνολικός χρόνος: 77 |

ΑΣΚΗΣΗ - 30

Σε ένα Πανεπιστημιακό Τμήμα πρόκειται να καταρτιστεί μία λίστα για τα μαθήματα της εξεταστικής περιόδου. Για κάθε μάθημα είναι γνωστά:

1. Ο κωδικός του μαθήματος (ακέραιος στην περιοχή [1-99])
2. Το εξάμηνο στο οποίο ανήκει (ακέραιος στην περιοχή [1-9])
3. Το πλήθος των φοιτητών & φοιτητριών που πρόκειται να δώσουν εξετάσεις (ακέραιος 1-999)

Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα εξής :

1. Θα εισάγει δεδομένα σε έναν πίνακα ακεραιών με $3*N$ σε πλήθος θέσεις, σύμφωνα με τα παραπάνω. Ο γνωστός αριθμός $N > 0$ δηλώνει το πλήθος των μαθημάτων για τα οποία θα πραγματοποιηθούν εξετάσεις (δηλ. ενδεχομένως για κάποια μαθήματα δεν θα διενεργηθούν εξετάσεις). Η εισαγωγή θα γίνει αποκλειστικά και μόνο με χρήση τυχαίων αριθμών - συνάρτηση `rand()` - στα όρια που έχουν δοθεί για κάθε τριάδα στοιχείων. **ΠΡΟΣΟΧΗ!! Ο κάθε κωδικός μαθήματος θα πρέπει να υπάρχει μόνον μία φορά στον πίνακα!!**
2. Θα εμφανίζει μία λίστα των μαθημάτων που πρόκειται να εξεταστούν, ανά εξάμηνο. Η λίστα θα είναι ταξινομημένη κατά αύξουσα σειρά εξαμήνου και κατά αύξουσα σειρά του κωδικού μαθήματος μέσα στο κάθε εξάμηνο, ΧΩΡΙΣ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΕΤΕ ΔΙΑΔΙΚΑΣΙΕΣ ΤΑΞΙΝΟΜΗΣΗΣ! Στο τέλος κάθε εξαμήνου θα πρέπει να εμφανίζεται ο μέσος όρος των ατόμων που πρόκειται να δώσουν εξετάσεις.

Αριθμητικό ΠΑΡΑΔΕΙΓΜΑ , $N=8$.

| | | | | | | | | | | | | | | | | | | | | | | | |
|----|---|----|----|---|-----|----|---|----|----|---|-----|----|---|-----|----|---|----|----|---|-----|----|---|----|
| 15 | 2 | 45 | 49 | 7 | 190 | 74 | 8 | 77 | 91 | 1 | 120 | 30 | 1 | 170 | 72 | 8 | 71 | 55 | 7 | 115 | 77 | 8 | 70 |
|----|---|----|----|---|-----|----|---|----|----|---|-----|----|---|-----|----|---|----|----|---|-----|----|---|----|

Αποτελέσματα :

| | | |
|-------------|-----------------------|----------------------------------|
| ΕΞΑΜΗΝΟ : 1 | μαθήματα : 30, 91 | μέσος όρος: 145 $(=(170+120)/2)$ |
| ΕΞΑΜΗΝΟ : 2 | μαθήματα : 15 | μέσος όρος: 45 |
| ΕΞΑΜΗΝΟ : 7 | μαθήματα : 49, 55 | μέσος όρος: 153 $(=(190+115)/2)$ |
| ΕΞΑΜΗΝΟ : 8 | μαθήματα : 72, 74, 77 | μέσος όρος: 73 $(=(71+77+70)/3)$ |

ΑΣΚΗΣΗ - 31

Ορισμένοι πρώτοι αριθμοί (prime numbers), έστω p_n , ονομάζονται good primes εφόσον ικανοποιούν τη συνθήκη:

$$p_n^2 > p_{n-1} \cdot p_{n+1}$$

Μερικοί τέτοιοι πρώτοι αριθμοί είναι οι : 5, 11, 17, 29, 37, 41, 53, ... (δηλ. πχ $5^2 = 25 > 4 \cdot 6 = 24$)

Να γραφεί αλγόριθμος ή πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εισάγει όλους αυτούς τους πρώτους αριθμούς που είναι μικρότεροι του 1000 σε ένα μονοδιάστατο πίνακα. Στο τέλος να εμφανίσετε τα στοιχεία αυτού του πίνακα. (ΥΠΟΔΕΙΞΗ : Το παρακάτω τμήμα κώδικα σε γλώσσα C βρίσκει αν ένας ακέραιος και θετικός αριθμός $k \geq 2$ είναι πρώτος (prime) αριθμός).

```
i=2; flag=0;
while ((i<=k/2) && (flag==0))
{
    if (k%i==0) flag=1;
    i++;
}
if (flag==0) printf("number %4d is prime \n",k);
```

ΑΣΚΗΣΗ - 32

Δύο μονοδιάστατοι αριθμητικοί πίνακες ακεραίων έχουν μέγεθος m και n αντίστοιχα. Οι τιμές των m και n δίνονται ως σταθερές στην αρχή του προγράμματος. Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα παρακάτω :

1. Θα γεμίζει τους δύο πίνακες με τυχαίους θετικούς ακέραιους αριθμούς στην περιοχή τιμών 1-99, με κατάλληλη χρήση της συνάρτησης rand().
2. Θα δημιουργεί και θα εμφανίζει έναν νέο πίνακα που θα περιέχει όλα τα στοιχεία των δύο αρχικών πινάκων, το καθένα από μια φορά (σε περίπτωση ύπαρξης διπλών, τριπλών κλπ τιμών) σε αύξουσα διάταξη, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης.

Παράδειγμα

1^{ος} Πίνακας $m = 11$

| | | | | | | | | | | |
|----|----|----|---|----|----|---|----|----|----|----|
| 45 | 19 | 22 | 7 | 88 | 19 | 4 | 22 | 59 | 32 | 27 |
|----|----|----|---|----|----|---|----|----|----|----|

2^{ος} Πίνακας $m = 15$

| | | | | | | | | | | | | | | |
|----|---|----|----|----|----|---|----|----|----|----|----|----|----|----|
| 33 | 2 | 22 | 88 | 65 | 99 | 4 | 27 | 59 | 11 | 33 | 14 | 16 | 14 | 25 |
|----|---|----|----|----|----|---|----|----|----|----|----|----|----|----|

Νέος Πίνακας - 3^{ος} Πίνακας

| | | | | | | | | | | | | | | | | |
|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 2 | 4 | 7 | 11 | 14 | 16 | 19 | 22 | 25 | 27 | 32 | 33 | 45 | 59 | 65 | 88 | 99 |
|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

ΑΣΚΗΣΗ - 33

Να δημιουργηθεί αριθμητικός πίνακας ακεραίων, N θέσεων (N = γνωστό και N>100), που θα περιέχει ακέραιους τυχαίους (χρήση της συνάρτησης rand()) θετικούς αριθμούς [1-1999]. Στη συνέχεια, να χρησιμοποιήσετε κατάλληλα τον παρακάτω κώδικα προγράμματος σε γλώσσα C που βρίσκει εάν ένας θετικός ακέραιος αριθμός είναι πρώτος αριθμός (prime number) ώστε :

1. να βρείτε τη συχνότητα εμφάνισης κάθε πρώτου αριθμού που περιέχεται στον πίνακα. Να εμφανίσετε σε κάθε σειρά, σε αύξουσα διάταξη, χωρίς να χρησιμοποιήσετε διαδικασία ταξινόμησης, τον πρώτο αριθμό και τη συχνότητα εμφάνισης
2. να καταχωρήσετε σε έναν νέο πίνακα όλους τους πρώτους αριθμούς που υπάρχουν στον αρχικό πίνακα, σε αύξουσα σειρά, τον καθένα όσες φορές υπάρχει

ΠΑΡΑΔΕΙΓΜΑ: Αρχικός πίνακας για N=16

| | | | | | | | | | | | | | | | |
|-----|----|-----|------|-----|------|-----|---|-----|------|-----|------|-----|-----|----|------|
| 257 | 18 | 120 | 1627 | 400 | 1217 | 990 | 8 | 541 | 1627 | 701 | 1804 | 257 | 541 | 90 | 1024 |
|-----|----|-----|------|-----|------|-----|---|-----|------|-----|------|-----|-----|----|------|

Συχνότητα εμφάνισης

257 2

541 2

701 1

1217 1

1627 2

Νέος πίνακας

| | | | | | | | |
|-----|-----|-----|-----|-----|------|------|------|
| 257 | 257 | 541 | 541 | 701 | 1217 | 1627 | 1627 |
|-----|-----|-----|-----|-----|------|------|------|

```
#include <stdio.h>
void main() {
    int num, i, isPrime=0;
    printf("Enter a positive number\n");
    scanf_s("%d",&num);
    /* Check whether num is divisible by any number between 2 to (num/2) */
    for(i = 2; i <=(num/2); ++i) {
        if (num%i==0) {
            isPrime=1;
            break;
        }
    }
    if (isPrime==0) printf("%d is a Prime Number",num);
    else printf("%d is NOT a Prime Number",num);
}
```

ΑΣΚΗΣΗ – 34

Πίνακας ακεραίων N θέσεων ($N =$ γνωστό και $N > 300$), έστω a , περιέχει ακέραιες θετικές τιμές στην περιοχή $[1,9]$. Να θεωρήσετε ότι οι τιμές αυτές αντιστοιχούν σε ορισμένα χρώματα. Ένας δεύτερος πίνακας, έστω b , με πλήθος θέσεων $M=4$ περιέχει ένα συγκεκριμένο μοτίβο τεσσάρων οποιωνδήποτε χρωμάτων από αυτά που υπάρχουν στον πίνακα a .

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει:

Σε ποιες θέσεις του πίνακα a υπάρχει το μοτίβο που περιέχεται στον πίνακα b . Το πρόγραμμα θα πρέπει να εμφανίζει, για κάθε εμφάνιση του συγκεκριμένου μοτίβου, τη θέση έναρξης του μοτίβου και τη θέση πέρας του μοτίβου

ΠΑΡΑΔΕΙΓΜΑ

Πίνακας a , με $N = 21$

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 6 | 7 | 8 | 1 | 4 | 9 | 4 | 3 | 7 | 8 | 1 | 4 | 5 | 5 | 7 | 1 | 8 | 3 | 2 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Πίνακας b :

| | | | |
|---|---|---|---|
| 7 | 8 | 1 | 4 |
|---|---|---|---|

ΑΠΟΤΕΛΕΣΜΑΤΑ

1^η εμφάνιση : έναρξη θέση 3, πέρας θέση 6

2^η εμφάνιση : έναρξη θέση 10, πέρας θέση 13

ΑΣΚΗΣΗ – 35

Θεωρείστε έναν πίνακα που περιέχει δυαδικά ψηφία, δηλ. 0 και 1, υπό μορφή αριθμών τύπου short int, μεγέθους N ($N =$ γνωστό, άρτιος αριθμός και $N > 200$).

Να υλοποιήσετε υπό μορφή προγράμματος σε γλώσσα C την παρακάτω διαδικασία:

Εξέταση στο αρχικό ήμισυ του πίνακα, του πλήθους των 1. Εάν το πλήθος αυτό είναι μονό τότε η εξέταση θα συνεχιστεί στο πρώτο ήμισυ του εξεταζόμενου μέρους του πίνακα (δηλ. στο αρχικό $\frac{1}{4}$ των στοιχείων του αρχικού πίνακα), ενώ, εάν το πλήθος αυτό είναι άρτιο, η εξέταση θα συνεχιστεί στο δεύτερο ήμισυ δηλ. στο 2^ο τέταρτο των στοιχείων του αρχικού πίνακα). Σε κάθε εξέταση (διάσχιση) μέρους του εξεταζόμενου τμήματος του πίνακα θα εμφανίζετε: τη θέση του 1^{ου} και του τελευταίου στοιχείου της εξεταζόμενης περιοχής και το πλήθος των 1. Η διαδικασία ολοκληρώνεται όταν το μέγεθος του εξεταζόμενου πίνακα γίνει 1 (δηλ. ελάχιστο).

ΠΑΡΑΔΕΙΓΜΑ για $N=16$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

ΒΗΜΑ 1

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|

Πλήθος των 1 = 5, Θέση του 1^{ου} στοιχείου 1^η, θέση τελευταίου στοιχείου 8η

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
|---|---|---|---|

Πλήθος των 1 = 3, Θέση του 1^{ου} στοιχείου 1^η, θέση τελευταίου στοιχείου 4η

| | |
|---|---|
| 1 | 1 |
|---|---|

Πλήθος των 1 = 2, Θέση του 1^{ου} στοιχείου 1^η, θέση τελευταίου στοιχείου 2η

| |
|---|
| 1 |
|---|

Πλήθος των 1 = 1, Θέση του 1^{ου} στοιχείου 1^η, θέση τελευταίου στοιχείου 1η,

μέγεθος του πίνακα = 1, ΤΕΛΟΣ ΔΙΑΔΙΚΑΣΙΑΣ

ΑΣΚΗΣΗ – 36

Ένας τριγωνικός αριθμός (triangular number), έστω x_n , προσδιορίζεται από τη σχέση $x_n = n(n+1)/2$, όπου $n = 1, 2, 3, \dots$. Μερικοί τριγωνικοί αριθμοί είναι : 1, 3, 6, 10, 15, 21, 28, 36, 45,...

Ένας αριθμός λέγεται τέλειος (perfect number) αν είναι ίσος με το άθροισμα των γνησίων διαιρετών του. Π.χ. οι αριθμοί 6 και 28 είναι τέλειοι διότι : $6=1+2+3$ και $28=1+2+4+7+14$.

Να γράψετε έναν αλγόριθμο ή ένα πρόγραμμα σε γλώσσα C που:

1. Θα εισάγει κατά σειρά N (N = γνωστό, ορίζεται ως σταθερά στην αρχή) τυχαίες θετικές ακέραιες τιμές μέσω της συνάρτησης rand() στην περιοχή [1,200].
2. Θα δημιουργεί τον αντίστοιχο τριγωνικό αριθμό.
3. Θα ελέγχει αν ο τριγωνικός αριθμός είναι τέλειος και εφόσον είναι θα τον καταχωρεί σε έναν μονοδιάστατο αριθμητικό πίνακα.
4. Θα εμφανίζει τα στοιχεία του πίνακα, καθώς και το πλήθος των μη μηδενικών στοιχείων του.

ΑΣΚΗΣΗ – 37

Η απόσταση δύο σημείων (x_1, y_1, z_1) και (x_2, y_2, z_2) στον τρισδιάστατο καρτεσιανό χώρο είναι :

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Δύο μονοδιάστατοι αριθμητικοί πίνακες, έστω a και b, περιέχουν τριάδες τιμών double που αντιστοιχούν στις συντεταγμένες N και M σημείων αντίστοιχα (N, M = γνωστά, ορίζονται ως σταθερές του προγράμματος).

1. Να εισάγετε δεδομένα στους πίνακες a και b μόνον με χρήση τυχαίων αριθμών (συνάρτηση rand()) στην περιοχή τιμών [0.0, 99.9].
2. Να θεωρήσετε ένα γνωστό σημείο C στον χώρο με γνωστές συντεταγμένες x_C, y_C, z_C (ορίζονται ως σταθερές του προγράμματος).
 - a. Χρησιμοποιώντας όλα τα στοιχεία του πίνακα a να βρείτε τη συνολική απόσταση όλων των σημείων του πίνακα a από το σημείο C.
 - b. Χρησιμοποιώντας όλα τα στοιχεία του πίνακα b να βρείτε ποιο σημείο του πίνακα b απέχει τη μικρότερη απόσταση από το σημείο C, με ακρίβεια υπολογισμού 10^{-3} (ένα σημείο προσδιορίζεται εντός του πίνακα από τη θέση που έχει στον πίνακα). Αν τα σημεία με τη μικρότερη απόσταση είναι περισσότερα από ένα να τα εμφανίσετε όλα. Στο τέλος να εμφανίσετε και την τιμή αυτής της ελάχιστης απόστασης.

ΑΣΚΗΣΗ – 38

Θεωρείστε έναν μονοδιάστατο πίνακα ακεραίων θετικών αριθμών N θέσεων ($N = \text{γνωστό}$).

1. Να εισάγετε τιμές στον πίνακα με χρήση τυχαίων αριθμών στην περιοχή $[1,99]$.
2. Ορίζουμε ως ρινότο το στοιχείο που υπάρχει στην αρχική ($1^{\text{η}}$ θέση) του πίνακα. Να διαμερίσετε τον πίνακα, δημιουργώντας έναν νέο πίνακα, σε δύο τμήματα ως εξής:

ο νέος πίνακας περιλαμβάνει στο πρώτο του τμήμα, σε αύξουσα διάταξη, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης, όλες τις τιμές του αρχικού πίνακα που είναι μικρότερες ή και ίσες από την τιμή που υπάρχει στη θέση ρινότο. Στη συνέχεια, υπάρχει η τιμή της θέσης ρινότο και ακολουθούν, στο δεύτερο τμήμα του, οι τιμές του αρχικού πίνακα που είναι μεγαλύτερες της τιμής που υπάρχει στη θέση ρινότο, όπως εμφανίζονται στον αρχικό πίνακα, από αριστερά προς τα δεξιά.

3. Να εμφανίσετε τους δύο πίνακες

ΠΑΡΑΔΕΙΓΜΑ για $N=10$

αρχικός πίνακας

| | | | | | | | | | |
|---|---|---|----|----|---|---|---|----|---|
| 7 | 2 | 7 | 14 | 16 | 3 | 8 | 1 | 27 | 5 |
|---|---|---|----|----|---|---|---|----|---|

νέος πίνακας

| | | | | | | | | | |
|---|---|---|---|---|---|----|----|---|----|
| 1 | 2 | 3 | 5 | 7 | 7 | 14 | 16 | 8 | 27 |
|---|---|---|---|---|---|----|----|---|----|

ΑΣΚΗΣΗ – 39

Ένας κυκλικός πρώτος (prime) αριθμός είναι εκείνος που παραμένει πρώτος (prime) μετά από μια επαναληπτική τοποθέτηση όλων των ψηφίων του σε κάθε πιθανή θέση του αριθμού, π.χ. ο αριθμός 197 είναι κυκλικός prime διότι οι αριθμοί 197, 971 και 719 είναι πρώτοι (prime) αριθμοί.

Να γραφεί αλγόριθμος ή πρόγραμμα σε γλώσσα C/C++ που θα βρίσκει και θα καταχωρεί σε μονοδιάστατο αριθμητικό πίνακα τις πρώτες 30 ακολουθίες κυκλικών πρώτων αριθμών, υπό την προϋπόθεση σε κάθε ακολουθία όλοι να είναι μικρότεροι του 5000. Στη συνέχεια θα πρέπει να εμφανίζονται όλες αυτές οι ακολουθίες, μία σε κάθε γραμμή της οθόνης του ΗΥ. (ΥΠΟΔΕΙΞΗ : Το παρακάτω τμήμα κώδικα σε γλώσσα C βρίσκει αν ένας ακέραιος και θετικός αριθμός $k \geq 2$ είναι πρώτος (prime) αριθμός).

```
i=2; flag=0;
while ((i<=k/2) && (flag==0))
{
    if (k%i==0) flag=1;
    i++;
}
if (flag==0) printf("number %4d is prime \n",k);
```

ΑΣΚΗΣΗ – 40

Να γραφεί κώδικας σε γλώσσα C για τον παρακάτω αλγόριθμο:

1. Δημιουργία ενός θετικού τυχαίου αριθμού m : $2 \leq m \leq 1500$
2. Εύρεση όλων των διαιρετών του m που είναι πρώτοι αριθμοί (prime numbers), βάσει του κώδικα που παρατίθεται στη συνέχεια
3. Προσδιορισμός του αριθμού n που θα είναι το γινόμενο των γνήσιων διαιρετών του αριθμού m με περιττό πλήθος εμφανίσεων. Κάθε τέτοιος γνήσιος διαιρέτης θα συμμετέχει μόνον μία φορά στον προσδιορισμό του αριθμού n .
4. Προσδιορισμός του αριθμού $z = m \cdot n$
5. Εύρεση του αριθμού k για τον οποίο ισχύει ότι $k^2 = z = m \cdot n$
6. Εμφάνιση των αριθμών m, n, k σύμφωνα με την προδιαγραφή του παραδείγματος.

Παράδειγμα, έστω $m = 90$

Διαιρέτες (γνήσιοι) του 90 : 2, 3, 3, 5

Διαιρέτες με περιττό πλήθος εμφανίσεων: 2, 5

$n = 2 \cdot 5 = 10$, $z = 90 \cdot 10 = 900$, $k = 30$ ($30^2 = 900$)

$m = 90$, $n = 10$, $k = 30$

(ΥΠΟΔΕΙΞΗ : Το παρακάτω τμήμα κώδικα σε γλώσσα C βρίσκει αν ένας ακέραιος και θετικός αριθμός $k \geq 2$ είναι πρώτος (prime) αριθμός).

```
i=2; flag=0;
while ((i<=k/2) && (flag==0))
{
    if (k%i==0) flag=1;
    i++;
}
if (flag==0) printf("number %4d is prime \n",k);
```

ΑΣΚΗΣΗ – 41

1. Να δημιουργηθεί αριθμητικός πίνακας ακεραιών, N θέσεων ($N =$ γνωστό και $N > 20$), που θα περιέχει τριψήφιους τυχαίους ακέραιους θετικούς αριθμούς (χρήση της συνάρτησης `rand()`) στην περιοχή $[1, 999]$.
2. Για κάθε στοιχείο του πίνακα να βρείτε τους διαιρέτες του που είναι πρώτοι (prime) χρησιμοποιώντας κατάλληλα τον παρακάτω κώδικα προγράμματος σε γλώσσα C, που βρίσκει εάν ένας θετικός ακέραιος αριθμός είναι πρώτος αριθμός (prime number). Να εισάγετε όλους αυτούς τους διαιρέτες που είναι πρώτοι (prime), όλων των στοιχείων του πίνακα, μαζί με τη συχνότητα εμφάνισής τους, σε έναν νέο πίνακα, σε ζεύγη. Κάθε τέτοιο ζεύγος θα αποτελείται από τον διαιρέτη και τη συχνότητα εμφάνισής του. Τα στοιχεία του νέου πίνακα θα είναι ταξινομημένα σε αύξουσα διάταξη του πρώτου στοιχείου (δηλ. του διαιρέτη) κάθε ζεύγους, χωρίς να χρησιμοποιήσετε κάποια μέθοδο ταξινόμησης. (ΥΠΟΔΕΙΞΗ : Ο πρώτος prime αριθμός είναι το 2).

ΠΑΡΑΔΕΙΓΜΑ: Αρχικός πίνακας για $N=5$

| | | | | |
|---|----|----|----|----|
| 2 | 84 | 63 | 62 | 15 |
|---|----|----|----|----|

| αριθμός | διαιρέτες |
|---------|---------------------------------------|
| 2 | 2 |
| 84 | 2, 3, 4, 6, 7, 12, 14, 21, 28, 42, 84 |
| 63 | 3, 7, 9, 21, 63 |
| 62 | 2, 31, 62 |
| 15 | 3, 5, 15 |

Οι διαιρέτες που είναι prime, με τις συχνότητές του είναι (ζεύγη) : (2,3), (3,3), (5,1), (7,2), (31,1)

Νέος πίνακας ζευγών, ταξινομημένος

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|----|---|
| 2 | 3 | 3 | 3 | 5 | 1 | 7 | 2 | 31 | 1 |
|---|---|---|---|---|---|---|---|----|---|

```
#include <stdio.h>
```

```
int n, i, c = 0;
```

```
void main()
```

```
{
```

```
printf("Enter any number n:");
```

```
scanf_s("%d", &n);
```

```
for (i = 1; i <= n; i++)
```

```
{ if (n % i == 0) c++; }
```

```
if (c == 2) printf("%5d is a Prime number\n",n);
```

```
else printf("%5d is not a Prime number\n",n);
```

```
}
```

ΑΣΚΗΣΗ – 42

1. Να δημιουργηθεί πίνακας θετικών ακέραιων αριθμών, κατά μέγιστο M θέσεων ($M =$ γνωστό και $M > 20$), που θα περιέχει ακέραιους θετικούς αριθμούς που θα δημιουργούνται επαναληπτικά από τον τύπο $2n^2 + 1$ ($n = 0, 1, 2, \dots$) μέχρις ότου να προκύψει τιμή μεγαλύτερη του 5000 (η τιμή αυτή δεν θα περιλαμβάνεται στον αριθμητικό πίνακα).
2. Χρησιμοποιώντας τον ανωτέρω πίνακα να δημιουργήσετε και να εμφανίσετε έναν νέο πίνακα που θα περιλαμβάνει εκείνες τις τιμές του αρχικού πίνακα που είναι πρώτοι (prime) αριθμοί χρησιμοποιώντας κατάλληλα τον παρακάτω κώδικα προγράμματος σε γλώσσα C, που βρίσκει εάν ένας θετικός ακέραιος αριθμός είναι πρώτος αριθμός (prime number). (ΥΠΟΔΕΙΞΗ : Ο πρώτος prime αριθμός είναι το 2).

ΠΑΡΑΔΕΙΓΜΑ: Αρχικός πίνακας :

| | | | | | | | | | | |
|---|---|---|----|----|----|----|-------|------|------|------|
| 1 | 3 | 9 | 19 | 33 | 51 | 73 | | 4609 | 4803 | 5001 |
|---|---|---|----|----|----|----|-------|------|------|------|

Νέος πίνακας με τους primes :

| | | | |
|---|----|----|-------|
| 3 | 19 | 73 | |
|---|----|----|-------|

```
#include <stdio.h>

int n, i, c = 0;
void main()
{
printf("Enter any number n:");
scanf_s("%d", &n);

for (i = 1; i <= n; i++)
{ if (n % i == 0) c++; }

if (c == 2) printf("%5d is a Prime number\n",n);
else printf("%5d is not a Prime number\n",n);
}
```

ΑΣΚΗΣΗ – 43

Αριθμητικός πίνακας ακεραιών, Ν θέσεων περιέχει όλους τους πρώτους (primes) ακέραιους θετικούς αριθμούς που είναι μικρότεροι του αριθμού 1000. Δίνεται ότι το πλήθος των αριθμών αυτών δεν είναι μεγαλύτερο από 200. Χρησιμοποιώντας τον κώδικα που δίνεται στη συνέχεια:

1. Να δημιουργήσετε αυτόν τον πίνακα και να καταχωρήσετε, σε αύξουσα διάταξη, τους πρώτους αριθμούς που ζητούνται (ΥΠΟΔΕΙΞΗ : Ο πρώτος prime αριθμός είναι το 2).
2. Να βρείτε και να εμφανίσετε το πλήθος αυτών των αριθμών.
3. Για κάθε ζεύγος διαδοχικών πρώτων αριθμών του πίνακα αυτού να βρείτε τη διαφορά τους και να την εμφανίσετε μαζί με τους δύο πρώτους αριθμούς, ανά γραμμή.
4. Να βρείτε και να εμφανίσετε, σε αύξουσα διάταξη, τη συχνότητα εμφάνισης όλων αυτών των διαφορών.

ΠΑΡΑΔΕΙΓΜΑ: Για ένα τμήμα του πίνακα

| | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-------|-----|
| | 353 | 359 | 367 | 373 | 379 | 383 | 389 | | 997 |
|-------|-----|-----|-----|-----|-----|-----|-----|-------|-----|

| Αριθμοί | ΔΙΑΦΟΡΑ |
|----------|---------|
| 353, 359 | 6 |
| 359, 367 | 8 |
| 367, 373 | 6 |
| 373, 379 | 6 |
| 379, 383 | 4 |
| 383, 389 | 6 |

| | | | |
|---------------------|---|---|---|
| ΔΙΑΦΟΡΑ | 4 | 6 | 8 |
| Συχνότητα εμφάνισης | 1 | 4 | 1 |

```
#include <stdio.h>

int n, i, c = 0;
void main()
{
    printf("Enter any number n:");
    scanf_s("%d", &n);

    for (i = 1; i <= n; i++)
    { if (n % i == 0) c++; }

    if (c == 2) printf("%5d is a Prime number\n",n);
    else     printf("%5d is not a Prime number\n",n);
}
```

ΑΣΚΗΣΗ – 44

Αριθμητικός πίνακας ακεραίων περιέχει N τριάδες τιμών ($N =$ γνωστό και $N > 50$) που αντιστοιχούν :

- στον κωδικό ανεμογεννήτριας (επιτρεπτοί κωδικοί 1 - 10)
- στην ημερομηνία παραγωγής, στη μορφή ΜΜΗΗ (ΜΜ = μήνας, ΗΗ = ημέρα) για το ίδιο έτος
- στην ημερήσια παραγόμενη ενέργεια (θετικές ακέραιες τιμές στην περιοχή $[0,999]$)

Κάθε κωδικός ανεμογεννήτριας και κάθε ημερομηνία μπορούν να εμφανίζεται περισσότερες από μία φορές στον πίνακα. Μικρό παράδειγμα για $N=7$ τριάδες:

| | | | | | | | | | | | | | | | | | | | | |
|---|-----|-----|---|-----|-----|---|------|-----|---|-----|-----|---|-----|-----|---|-----|-----|---|-----|-----|
| 5 | 211 | 220 | 3 | 525 | 500 | 8 | 1111 | 250 | 8 | 814 | 400 | 3 | 709 | 500 | 8 | 729 | 300 | 5 | 529 | 600 |
|---|-----|-----|---|-----|-----|---|------|-----|---|-----|-----|---|-----|-----|---|-----|-----|---|-----|-----|

Να γραφεί πρόγραμμα σε γλώσσα C που:

1. Θα εισάγει τυχαία δεδομένα στον πίνακα, με χρήση της συνάρτησης `rand()`, στα όρια των τιμών που έχουν δοθεί.
2. Θα βρίσκει και θα εμφανίζει, σε αύξουσα διάταξη, πόσοι και ποιοι κωδικοί ανεμογεννητριών υπάρχουν στα δεδομένα (στο παράδειγμα υπάρχουν οι κωδικοί 3, 5, 8).
3. Θα βρίσκει, θα εισάγει σε ένα νέο πίνακα, και στη συνέχεια θα εμφανίζει τη συνολική ανά μήνα παραγόμενη ενέργεια για όλες τις ανεμογεννήτριες.
4. Θα βρίσκει, θα εισάγει σε ένα νέο πίνακα, και στη συνέχεια θα εμφανίζει τη συνολική ανά ανεμογεννήτρια παραγόμενη ενέργεια για όλο το έτος

Οι δύο νέοι πίνακες των ερωτημάτων 3 & 4 για το ανωτέρω παράδειγμα θα είναι:

Συνολική ανά μήνα παραγόμενη ενέργεια:

| Ιαν | Φεβρ | Μαρτ | Απρ | Μαι | Ιουν | Ιουλ | Αυγ | Σεπτ | Οκτ | Νοε | Δεκ |
|-----|------|------|-----|------|------|------|-----|------|-----|-----|-----|
| 0 | 220 | 0 | 0 | 1100 | 0 | 800 | 400 | 0 | 0 | 250 | 0 |

Συνολική ανά ανεμογεννήτρια ετήσια παραγόμενη ενέργεια :

| Κωδ. Ανεμ. 3 | Κωδ. Ανεμ.5 | Κωδ. Ανεμ. 8 |
|--------------|-------------|--------------|
| 1000 | 820 | 950 |

ΑΣΚΗΣΗ – 45

Θεωρείστε έναν πίνακα που περιέχει δυαδικά ψηφία, δηλ. 0 και 1, υπό μορφή αριθμών τύπου short int, μεγέθους N ($N =$ γνωστό, άρτιος αριθμός και $N > 100$).

Να υλοποιήσετε υπό μορφή προγράμματος σε γλώσσα C την παρακάτω διαδικασία, θεωρώντας τα 2 τμήματα του πίνακα που περιλαμβάνουν, το 1^ο τα πρώτα $N/2$ στοιχεία, και το 2^ο τα υπόλοιπα $N/2$ στοιχεία.

Να συγκρίνετε, ανά ζεύγη, ελέγχοντας όλα τα δημιουργούμενα ζεύγη

- το 1^ο στοιχείο του 1^{ου} τμήματος με το τελευταίο στοιχείο του 2^{ου} τμήματος
- το 2^ο στοιχείο του 1^{ου} τμήματος με το προτελευταίο στοιχείο του 2^{ου} τμήματος
-

Εάν οι τιμές στα ζεύγη δεν είναι ίδιες να καταχωρείτε σε ένα νέο πίνακα όλα τα ζεύγη τιμών στα οποία οι δύο τιμές δεν είναι ίδιες.

Στη συνέχεια, να υπολογιστεί η αριθμητική τιμή του δυαδικού αριθμού που προκύπτει, στο δεκαδικό σύστημα αρίθμησης, έστω x , χρησιμοποιώντας τον τύπο:

$$x = \sum_{i=0}^{n-1} a_i \cdot 2^{(n-i-1)}$$

Όπου :

a_i είναι τα στοιχεία του νέου πίνακα

n είναι το μέγεθος του νέου πίνακα

ΠΑΡΑΔΕΙΓΜΑ για $N=16$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

1^ο τμήμα

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|

2^ο τμήμα

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|

Ζεύγη τιμών από τα δύο τμήματα : (1, 0), (1, 1), (0, 1), (1, 0), (0, 1), (1, 0), (1, 1), (0, 0)

ΝΕΟΣ ΠΙΝΑΚΑΣ

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|

ΑΡΙΘΜΗΤΙΚΗ ΤΙΜΗ $x = 614$

ΑΣΚΗΣΗ – 46

Έστω n, k είναι δύο μη αρνητικοί ακέραιοι αριθμοί με $0 \leq k \leq n$. Ο διωνυμικός συντελεστής $\binom{n}{k}$ ορίζεται ως

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{1 \cdot 2 \cdot 3 \cdots k}$$

Επίσης ορίζονται :

$$\binom{n}{0} = \binom{n}{n} = 1, \quad \binom{n}{1} = \binom{n}{n-1} = n$$

Παράδειγμα :

$$\binom{14}{9} = \frac{14 \cdot 13 \cdot 12 \cdot 11 \cdot 10 \cdot 9 \cdot 8 \cdot 7 \cdot 6}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9} = 2002$$

Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα παρακάτω:

1. Θα εισάγει σε έναν μονοδιάστατο πίνακα ακεραιών θετικών αριθμών N ($N = \text{γνωστό}$) ζεύγη τιμών n, k . Οι τιμές των n, k θα πρέπει να ικανοποιούν και τους δύο ακόλουθους περιορισμούς:

$$0 \leq k \leq n \quad \text{και} \quad 0 \leq n \leq 15$$

Προτείνεται η εισαγωγή των τιμών να γίνεται με κατάλληλη χρήση της συνάρτησης `rand()`.

2. Για κάθε ζεύγος τιμών του πίνακα αυτού:

- a. να καταχωρούνται σε ένα νέο μονοδιάστατο πίνακα, κατά σειρά, οι αριθμοί που συνθέτουν τον παρονομαστή και τον αριθμητή του τύπου του διωνυμικού συντελεστή, δηλ. για το ανωτέρω παράδειγμα ο νέος πίνακας θα είναι

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|---|---|---|---|

- b. Να εμφανίζονται τα περιεχόμενα του νέου πίνακα, μία σειρά για κάθε ζεύγος του αρχικού πίνακα
- c. Να υπολογίζεται και να εμφανίζεται το αποτέλεσμα των πράξεων (δηλ. για το παράδειγμα η τιμή 2002). Οι πράξεις θα πρέπει να γίνονται χρησιμοποιώντας `MONON` τα στοιχεία του νέου πίνακα.

(Στις απαντήσεις σας πρέπει να ληφθούν υπόψη όλες οι δυνατές περιπτώσεις για τις τιμές των ζευγών n, k)

ΑΣΚΗΣΗ – 47

Να δημιουργηθεί αριθμητικός πίνακας ακεραίων, μεγέθους N ζευγών ($N =$ γνωστό). Κάθε ζεύγος τιμών (έστω a, b τα ονόματα των τιμών κάθε ζεύγους) περιέχει:

- Έναν τριψήφιο ακέραιο αριθμό $[100,999]$, είναι ο a που αναφέρεται ανωτέρω
- Έναν μονοψήφιο ακέραιο αριθμό $[1,9]$, είναι ο b που αναφέρεται ανωτέρω

Να γράψετε πρόγραμμα σε γλώσσα C που θα υλοποιεί τα παρακάτω:

1. Για κάθε ζεύγος τιμών (a, b) του πίνακα να βρείτε και να εμφανίσετε όπως φαίνεται στο παράδειγμα που ακολουθεί, τους b σε πλήθος πρώτους (primes) αριθμούς που ικανοποιούν τη συνθήκη:

$$100 \leq \text{prime number} \leq a.$$

Να εξετάσετε και την περίπτωση να προκύπτουν λιγότεροι σε πλήθος πρώτοι αριθμοί λόγω αυτού του περιορισμού.

2. Επίσης, το πρόγραμμα θα βρίσκει και θα εμφανίζει, συμπεριλαμβάνοντας τα αποτελέσματα που προκύπτουν για όλα τα ζεύγη τιμών του πίνακα:
 - Το πλήθος των primes που βρέθηκαν (για το παράδειγμα το πλήθος είναι 21)
 - Το πλήθος των διαφορετικών primes που βρέθηκαν (για το παράδειγμα το πλήθος είναι 18)
 - Τη συχνότητα εμφάνισης κάθε prime, σε αύξουσα διάταξη των primes, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης

Για να διαπιστώσετε εάν ένας αριθμός είναι prime θα χρησιμοποιήσετε κατάλληλα και **MONON** το παρακάτω πρόγραμμα που βρίσκει εάν ένας ακέραιος θετικός αριθμός, μεγαλύτερος ή και ίσος του 2, είναι πρώτος (prime).

ΠΑΡΑΔΕΙΓΜΑ: Αρχικός πίνακας για $N=4$ ζεύγη (η 1^η γραμμή στον πίνακα του παραδείγματος, με τα ονόματα a και b , , είναι για διευκόλυνσή σας)

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| a | b | a | b | a | b | a | b |
| 110 | 7 | 411 | 4 | 127 | 5 | 770 | 8 |

| ζεύγος | primes αριθμοί | Πλήθος primes |
|--------|--|---------------|
| 110, 7 | 109, 107, 103, 101 | 4 |
| 411, 4 | 409, 401, 397, 389 | 4 |
| 127, 5 | 127, 113, 109, 107, 103 | 5 |
| 770, 8 | 769, 761, 757, 751, 743, 739, 733, 727 | 8 |

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    int n, i, flag = 1;
    printf("Enter a number: \n"); scanf_s("%d", &n);

    for (i = 2; i <= sqrt(n) / 2; i++)
        if (n % i == 0) { flag = 0; break; }

    if (flag == 1) printf("%d is a prime number", n);
    else printf("%d is not a prime number", n);
}
```

ΑΣΚΗΣΗ – 48

1. Να δημιουργηθεί αριθμητικός πίνακας ακεραίων, N θέσεων ($N =$ γνωστό και $N > 50$), που θα περιέχει τετραψήφιους τυχαίους ακέραιους θετικούς αριθμούς, πεδίο τιμών $[1000, 9999]$ (χρήση της `rand()`).
2. Να βρείτε και να εμφανίσετε πόσοι και ποιοι από αυτούς τους αριθμούς είναι πρώτοι (primes), χρησιμοποιώντας κατάλληλα τον παρακάτω κώδικα σε γλώσσα C. (Ο πρώτος prime αριθμός είναι το 2)
3. Για κάθε στοιχείο του πίνακα να βρείτε και να εμφανίσετε, όπως δείχνει το παράδειγμα:
 - a. τους γνήσιους διαιρέτες του που είναι prime (πρώτοι) αριθμοί, χρησιμοποιώντας κατάλληλα τον παρακάτω κώδικα. Κάθε γνήσιος prime διαιρέτης θα ληφθεί μόνον μία φορά.
 - b. το πλήθος αυτών των γνήσιων prime διαιρετών.
4. Παράλληλα με το βήμα 3, να δημιουργήσετε έναν νέο πίνακα που θα περιέχει τις συχνότητες αυτών των γνήσιων prime διαιρετών. Τα στοιχεία του νέου πίνακα θα είναι ταξινομημένα σε αύξουσα διάταξη, χωρίς να χρησιμοποιήσετε κάποια μέθοδο ταξινόμησης. (ΥΠΟΔΕΙΞΗ : Ο πρώτος prime αριθμός είναι το 2).
5. Στο τέλος να εμφανίσετε τον ταξινομημένο πίνακα συχνοτήτων, υπό τη μορφή ζευγών (αριθμός, συχνότητα), όπως φαίνεται στο παράδειγμα.

ΠΑΡΑΔΕΙΓΜΑ: Αρχικός πίνακας για $N=6$

| | | | | | |
|------|------|------|------|------|------|
| 7686 | 5533 | 4777 | 3329 | 7580 | 9246 |
|------|------|------|------|------|------|

Primes : 3329 ΠΛΗΘΟΣ primes 1

| αριθμός | prime διαιρέτες | πλήθος |
|---------|-----------------|--------|
| 7686 | 2, 3, 7, 61 | 4 |
| 5533 | 11, 503 | 2 |
| 4777 | 17, 281 | 2 |
| 3329 | | 0 |
| 7580 | 2, 5, 379 | 3 |
| 9246 | 2, 3, 23, 67 | 4 |

Νέος πίνακας συχνοτήτων, ταξινομημένος 2,3, 3,2, 5,1, 7,1, 11,1, 17,1, 23,1, 61, 1, 67,1, 281,1, 379,1, 503, 1

```
#include <stdio.h>
int n, i, c = 0;
void main()
{
    printf("Enter any number n:");
    scanf_s("%d", &n);

    for (i = 1; i <= n; i++)
        { if (n % i == 0) c++; }

    if (c == 2) printf("%d is a Prime number\n",n);
    else printf("%d is not a Prime number\n",n);
}
```

ΑΣΚΗΣΗ – 49

1. Να δημιουργηθεί αριθμητικός πίνακας ακεραιών, N θέσεων ($N =$ γνωστό και $N > 30$), που θα περιέχει τριψήφιους τυχαίους ακέραιους θετικούς αριθμούς, πεδίο τιμών $[100, 999]$ (χρήση της `rand()`).
2. Να βρείτε και να εμφανίσετε πόσοι και ποιοι από αυτούς τους αριθμούς είναι πρώτοι (primes), χρησιμοποιώντας κατάλληλα τον παρακάτω κώδικα σε γλώσσα C. (Ο πρώτος prime αριθμός είναι το 2).
3. Για κάθε στοιχείο του πίνακα να βρείτε και να εμφανίσετε, όπως δείχνει το παράδειγμα (ΜΟΝΟΝ ΤΑ ΣΤΟΙΧΕΙΑ ΤΩΝ ΣΤΗΛΩΝ 1, 3 & 4, χωρίς τις επικεφαλίδες των στηλών):
 - a. Τον μεγαλύτερο prime γνήσιο διαιρέτη του, χρησιμοποιώντας κατάλληλα τον παρακάτω κώδικα.
 - b. Το πλήθος των γνήσιων, μη πρώτων (primes), διαιρετών του.
4. Παράλληλα με το βήμα 3, να δημιουργήσετε έναν νέο πίνακα που θα περιέχει τις συχνότητες αυτών των μεγαλύτερων primes γνήσιων διαιρετών του βήματος 3a. Τα στοιχεία του νέου πίνακα θα είναι ταξινομημένα σε αύξουσα διάταξη, χωρίς να χρησιμοποιήσετε κάποια μέθοδο ταξινόμησης.
5. Στο τέλος να εμφανίσετε τον ταξινομημένο πίνακα συχνοτήτων, υπό τη μορφή ζευγών (αριθμός, συχνότητα) που φαίνεται στο παράδειγμα.

ΠΑΡΑΔΕΙΓΜΑ: Αρχικός πίνακας για $N=5$

| | | | | |
|-----|-----|-----|-----|-----|
| 369 | 730 | 449 | 405 | 614 |
|-----|-----|-----|-----|-----|

Primes : 449 ΠΛΗΘΟΣ primes 1

| αριθμός | ΓΝΗΣΙΟΙ ΔΙΑΙΡΕΤΕΣ (με έντονο χρώμα είναι οι primes) | Μεγαλύτερος γνήσιος prime διαιρέτης | Πλήθος μη πρώτων διαιρετών |
|---------|--|---|-------------------------------|
| 369 | 3 , 9, 41 , 123 | 41 | 2 |
| 730 | 2 , 5 , 10, 73 , 146, 365 | 73 | 3 |
| 449 | EINAI prime | | 0 |
| 405 | 3 , 5 , 9, 15, 27, 45, 81, 135 | 5 | 6 |
| 614 | 2 , 307 | 307 | 0 |

Νέος πίνακας συχνοτήτων, ταξινομημένος: 5, 1, 41, 1, 73, 1, 307, 1

```
#include <stdio.h>
int n, i, c = 0;
void main()
{
    printf("Enter any number n:");
    scanf_s("%d", &n);

    for (i = 1; i <= n; i++) { if (n % i == 0) c++; }

    if (c == 2) printf("%5d is a Prime number\n",n);
    else printf("%5d is not a Prime number\n",n);
}
```

ΑΣΚΗΣΗ – 50

Αριθμητικός πίνακας ακεραίων, μεγέθους N τριάδων ($N = \text{γνωστό}$) περιέχει σε κάθε τριάδα τιμών (έστω a, b, c τα ονόματα των τιμών κάθε τριάδας) στοιχεία για τη βαθμολογία φοιτητών & φοιτητριών:

- Έναν τριψήφιο ακέραιο αριθμό $[1,999]$, είναι ο a που αναφέρεται ανωτέρω, που συμβολίζει τον αριθμό μητρώου ενός φοιτητή / μίας φοιτήτριας.
- Έναν τριψήφιο ακέραιο αριθμό $[101,999]$, είναι ο b που αναφέρεται ανωτέρω, που συμβολίζει τον αριθμό ενός μαθήματος του προγράμματος σπουδών. Το πρώτο ψηφίο αφορά τον αριθμό του εξαμήνου $[1,9]$. Εάν το τελευταίο ψηφίο είναι άρτιος αριθμός το μάθημα είναι υποχρεωτικό, ενώ αν είναι περιττός αριθμός το μάθημα είναι κατ' επιλογήν.
- Το βαθμό του μαθήματος, ακέραιες τιμές $[0,10]$, είναι ο c που αναφέρεται ανωτέρω. Εάν ο βαθμός είναι ίσος με μηδέν τότε θεωρείται ότι ο φοιτητής / φοιτήτρια δεν έχει εξεταστεί στο μάθημα αυτό.

ΠΑΡΑΔΕΙΓΜΑΤΑ : (759, 104, 8), (778, 307,4), (415, 802, 0), (759, 107,3),

Να γράψετε πρόγραμμα σε γλώσσα C που θα υλοποιεί τα παρακάτω, χρησιμοποιώντας όλα τα δεδομένα του αρχικού πίνακα:

1. Θα εισάγει τις τριάδες τιμών στον αρχικό πίνακα, σύμφωνα με τους περιορισμούς, χρησιμοποιώντας κατάλληλα τη συνάρτηση `rand()`.
2. Θα βρίσκει και θα εμφανίζει για ένα εξάμηνο, ο αριθμός του οποίου εισάγεται μέσω της `scanf`, με τον κατάλληλο έλεγχο εγκυρότητας τιμής, έναν μονοδιάστατο αριθμητικό πίνακα, έστω q , που θα περιέχει υπό μορφή ζευγών - (κωδικός μαθήματος, μέσος όρος βαθμολογίας (στρογγυλοποιημένος στον μικρότερο ακέραιο)) όλα τα μαθήματα αυτού του εξαμήνου.
3. Θα δημιουργεί και θα εμφανίζει, έναν ταξινομημένο κατά αύξοντα αριθμό μητρώου νέο μονοδιάστατο πίνακα, έστω Z , που θα περιέχει για κάθε φοιτητή/ φοιτήτρια που υπάρχει στον αρχικό πίνακα, υπό μορφή τριάδων:
 - a. Τον αριθμό μητρώου
 - b. Το πλήθος των υποχρεωτικών μαθημάτων στα οποία έχει βαθμό ≥ 5 .
 - c. Το πλήθος των κατ' επιλογήν μαθημάτων στα οποία έχει βαθμό ≥ 5

ΔΕΝ ΠΡΕΠΕΙ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΕΤΕ ΚΑΠΟΙΟΝ ΑΛΓΟΡΙΘΜΟ ΤΑΞΙΝΟΜΗΣΗΣ !!!!

Η εμφάνιση του πίνακα Z θα γίνεται με έναν αριθμό μητρώου σε κάθε γραμμή της οθόνης

ΑΣΚΗΣΗ – 51

Σχετικοί πρώτοι αριθμοί (relative primes) είναι ζεύγη ακεραίων θετικών αριθμών οι οποίοι έχουν ως μοναδικό κοινό διαιρέτη τον αριθμό 1. Παράδειγμα:

Οι αριθμοί 14 και 15 είναι σχετικοί πρώτοι αριθμοί επειδή:

Διαιρέτες του 14 : 1, 2, 7, 14

Διαιρέτες του 15: 1, 3, 5, 15

(επομένως ο μόνον κοινός διαιρέτης τους είναι το 1).

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει όλα τα ζεύγη σχετικών πρώτων αριθμών, υπό τον περιορισμό: και οι δύο αριθμοί να είναι μικρότεροι του 100.

Τα ζεύγη αυτά θα πρέπει να καταχωρούνται σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραίων. Το πρόγραμμά σας θα πρέπει στο τέλος να εμφανίζει τον πίνακα αυτό, ένα ζεύγος ανά γραμμή της οθόνης, καθώς και το πλήθος των ζευγών.

ΑΣΚΗΣΗ – 52

Να βρεθούν όλοι οι πρώτοι (primes) θετικοί ακέραιοι αριθμοί που δημιουργούνται από τη σχέση :

$$n^m - (n-1)^m$$

και είναι μικρότεροι του 5000, για τιμές $2 \leq n \leq 10$ και $2 \leq m \leq 9$.

Οι τιμές πρέπει να εμφανίζονται σε αύξουσα διάταξη στην οθόνη, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης. Η κάθε τιμή που προκύπτει από τον εν λόγω τύπο πρέπει να εμφανίζεται μόνον μία φορά. Πόσοι μοναδικοί πρώτοι (primes) αριθμοί προκύπτουν;

Για να ελέγξετε εάν ένας ακέραιος θετικός αριθμός $z \geq 2$ είναι prime να χρησιμοποιήσετε την ακόλουθη συνάρτηση:

```
bool findprime(int z)
{
    int i, flag=1;
    if (z>=2)
    {
        for(i=2; i<=z/2; i++)
        {
            if (z%i==0) { flag=0; return false; }
        }
        if (flag==1) return true;
        else return false;
    }
    else return false;
}
```

Για να ελέγξετε τα αποτελέσματά σας η αλληλουχία των αριθμών είναι:

3, 5, 7, 11, 13, ..., 271, 4651

ΑΣΚΗΣΗ – 53

Να βρεθούν όλα τα ζεύγη **primes** θετικών ακεραιών αριθμών, έστω (p, q) , που ικανοποιούν τη σχέση

$$|p^3 - q^7| = |p - q| \text{ για τιμές } 2 \leq p \leq 20, \quad 2 \leq q \leq 10$$

Τίθεται ο περιορισμός καμία τιμή του τύπου να μην υπερβαίνει το 5000.

Για τον έλεγχο εάν ένας θετικός ακέραιος αριθμός $z \geq 2$ είναι prime να χρησιμοποιήσετε τη συνάρτηση:

```
bool findprime(int z)
{
    int i, flag=1;
    if (z>=2)
    {
        for(i=2; i<=z/2; i++)
        {
            if (z%i==0) { flag=0; return false; }
        }
        if (flag==1) return true;
        else return false;
    }
    else return false;
}
```

Τα ζεύγη που προκύπτουν πρέπει να καταχωρούνται σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραιών, ο οποίος και θα χρησιμοποιηθεί για την εμφάνιση των ζευγών, με ένα ζεύγος ανά γραμμή της οθόνης. Πόσα ζεύγη προκύπτουν; Για να ελέγξετε τα αποτελέσματά σας το μοναδικό ζεύγος που θα προκύψει είναι (13,3).

ΑΣΚΗΣΗ – 54

Να γράψετε ένα πρόγραμμα σε γλώσσα C που θα βρίσκει όλους τους πρώτους (prime) αριθμούς που προκύπτουν από τη σχέση

$$\frac{n(n+1)}{2} + 1$$

όπου n θετικός ακέραιος αριθμός, $n > 1$.

Π.χ. ($n = 3, prime = 7$), ($n = 4, prime = 11$), ..., ($n = 47, prime = 1129$), ...

Το πρόγραμμα:

1. Θα εισάγει όλα αυτά τα ζεύγη $(n, prime)$ σε αύξουσα σειρά του n , σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραιών, ΜΕΧΡΙ να προκύψει $prime$ μεγαλύτερος του 5000 (ο πρώτος $prime > 5000$ είναι ο 5051 ο οποίος δεν θα συμπεριληφθεί στον πίνακα).
2. Θα εμφανίζει, ένα ζεύγος ανά γραμμή της οθόνης, μαζί με τον αύξοντα αριθμό του ζεύγους στον πίνακα.
3. Να βρείτε και να εμφανίσετε σε ποια ζεύγη και οι δύο αριθμοί είναι primes. Στο τέλος να εμφανίσετε και το πλήθος αυτών των ζευγών.

ΥΠΟΔΕΙΞΗ: για να βρείτε εάν ένας θετικός ακέραιος αριθμός είναι πρώτος (prime) να χρησιμοποιήσετε κάποιο έτοιμο τμήμα κώδικα σε γλώσσα C, όπως έχει δοθεί κατά τη διάρκεια των εργαστηριακών ασκήσεων του μαθήματος.

ΑΣΚΗΣΗ – 55

Ένας πρώτος (prime) αριθμός ονομάζεται balanced prime εάν ισαπέχει από τον αμέσως προηγούμενο και τον αμέσως επόμενο πρώτο αριθμό. Π.χ. ο αριθμός 53 ισαπέχει από τον αμέσως προηγούμενο prime, είναι ο αριθμός 47, και από τον αμέσως επόμενο prime, είναι ο 59. Προφανώς ισχύει η σχέση

$$p_n = \frac{p_{n-1} + p_{n+1}}{2}$$

Το πρόγραμμα σε γλώσσα C που θα πρέπει να γράψετε:

- 1) Θα εισάγει όλους αυτούς τους balanced primes, σε αύξουσα σειρά, σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραιών, ΜΕΧΡΙ να προκύψει balanced prime μεγαλύτερος του 5000 (ο πρώτος balanced prime >5000 είναι ο 5107 ο οποίος δεν θα συμπεριληφθεί στον πίνακα). Η ακολουθία που πρέπει να προκύψει είναι : 5, 53, 157, 173,
- 2) Θα εμφανίζει, έναν τέτοιο αριθμό ανά γραμμή της οθόνης, μαζί με τον άξοντα αριθμό του στον πίνακα.

ΥΠΟΔΕΙΞΗ: για να βρείτε εάν ένας θετικός ακέραιος αριθμός είναι πρώτος (prime) να χρησιμοποιήσετε κάποιο έτοιμο τμήμα κώδικα σε γλώσσα C, όπως έχει δοθεί κατά τη διάρκεια των εργαστηριακών ασκήσεων του μαθήματος.

ΑΣΚΗΣΗ – 56

Μια τριάδα πρώτων αριθμών (prime triplet) είναι ένα σύνολο τριών primes αριθμών στην οποία ο μικρότερος και ο μεγαλύτερος prime διαφέρουν κατά 6. Η πρώτη τέτοια τριάδα είναι (5, 7, 11). Η ακολουθία των prime triplets έχει στην αρχή τις εξής τριάδες:

$$(5, 7, 11), (7, 11, 13), (11, 13, 17), (13, 17, 19), (17, 19, 23), (37, 41, 43), \dots$$

Το πρόγραμμα σε γλώσσα C που θα πρέπει να γράψετε:

- 1) Θα εισάγει όλες αυτές τις τριάδες, σε αύξουσα σειρά, σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραιών, ΜΕΧΡΙ να βρεθεί η τελευταία τριάδα της οποίας και οι τρεις πρώτοι αριθμοί θα είναι μικρότεροι του 1000. Για τον έλεγχο του κώδικά σας η τελευταία τριάδα είναι η {881, 883, 887}.
- 2) Θα εμφανίζει, μία τριάδα ανά γραμμή της οθόνης, τις τριάδες του πίνακα, μαζί με τον άξοντα αριθμό κάθε τριάδας
- 3) Θα βρίσκει και θα εμφανίζει τη συχνότητα κάθε prime αριθμού που υπάρχει στον πίνακα (π.χ. ο αριθμός 5 έχει συχνότητα 1, ο αριθμός 7 έχει συχνότητα 2 κ.ο.κ.)

ΥΠΟΔΕΙΞΗ: για να βρείτε εάν ένας θετικός ακέραιος αριθμός είναι πρώτος (prime) να χρησιμοποιήσετε κάποιο έτοιμο τμήμα κώδικα σε γλώσσα C, όπως έχει δοθεί κατά τη διάρκεια των εργαστηριακών ασκήσεων του μαθήματος.

ΑΣΚΗΣΗ – 57

Για ορισμένους θετικούς ακέραιους αριθμούς, έστω n , ο αριθμός $2^n + 1$ είναι ένας περιττός πρώτος (prime) αριθμός. Π.χ. για $n = 2$, $2^2 + 1 = 5$, ο 5 είναι ένας περιττός prime αριθμός.

Να γραφεί πρόγραμμα σε γλώσσα C το οποίο, θα βρίσκει όλους αυτούς τους περιττούς primes αριθμούς για $n = 1, 2, 3, \dots$ υπό την προϋπόθεση:

όλοι αυτοί οι περιττοί primes να είναι μικρότεροι του 2000.

Να εισάγετε όλα αυτά τα ζεύγη των τιμών $(n, prime)$ σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραίων αριθμών, γνωστού μεγέθους N .

Στο τέλος να εμφανίσετε τον πίνακα των ζευγών $(n, prime)$, ένα ζεύγος ανά γραμμή της οθόνης.

Στη συνέχεια να βρείτε και να εμφανίσετε πόσοι και ποιοι από τους αριθμούς n που υπάρχουν στα ζεύγη του πίνακα είναι ακέραιες δυνάμεις του 2. (π.χ. για $n = 2$, ισχύει ότι ο n είναι ακέραια δύναμη του 2 ($= 2^1$)).

ΑΣΚΗΣΗ – 58

Να βρεθούν όλες οι Πυθαγόρειες τριάδες, έστω x, y, z που δημιουργούνται από ζεύγη πρώτων (primes) αριθμών, έστω (a, b) , υπό τους περιορισμούς:

- $a > b$
- $z < 1000$, (z είναι η μεγαλύτερη τιμή κάθε Πυθαγόρειας τριάδας).

Οι τιμές των x, y, z δημιουργούνται από τις σχέσεις:

$$x = a^2 - b^2$$

$$y = 2ab$$

$$z = a^2 + b^2$$

ΠΑΡΑΔΕΙΓΜΑ

Για $a = 3, b = 2$:

$$x = a^2 - b^2 = 3^2 - 2^2 = 9 - 4 = 5$$

$$y = 2ab = 2 \cdot 3 \cdot 2 = 12$$

$$z = a^2 + b^2 = 3^2 + 2^2 = 9 + 4 = 13$$

Να εισάγετε όλες αυτές τις τριάδες τιμών x, y, z σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραίων αριθμών, γνωστού μεγέθους N τριάδων. Στο τέλος να εμφανίσετε όλες τις τριάδες που έχουν εισαχθεί στον πίνακα, μία τριάδα ανά γραμμή της οθόνης, μαζί με τον αύξοντα αριθμό κάθε τριάδας.

Εμφάνιση αποτελεσμάτων, από την εκτέλεση του πηγαίου κώδικα:

$$1) \quad a = 3 \quad b = 2 \quad x = 5 \quad y = 12 \quad z = 13$$

$$2) \quad a = 5 \quad b = 2 \quad x = 21 \quad y = 20 \quad z = 29$$

.....

ΑΣΚΗΣΗ – 59

Να γραφεί πρόγραμμα σε γλώσσα C που :

1. θα βρίσκει και θα εμφανίζει όλους τους τετραψήφιους prime (πρώτους) ακεραίους αριθμούς, οι οποίοι παραμένουν primes εάν αντιστρέψουμε τη σειρά των ψηφίων τους.

Π.χ. 3917, 7193 είναι και οι δύο primes

Τα ζεύγη αυτά θα πρέπει να εισάγονται σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραίων αριθμών, γνωστού μεγέθους N . Να εμφανίσετε τον πίνακα των ζευγών, ένα ζεύγος ανά γραμμή της οθόνης.

2. Στη συνέχεια, χρησιμοποιώντας όλα τα στοιχεία του μονοδιάστατου αριθμητικού πίνακα:

να βρείτε και να εμφανίσετε τη συχνότητα όλων των αριθμών που σχηματίζονται από τα δύο τελευταία ψηφία κάθε τέτοιου prime. Στο τέλος να εμφανιστούν όλοι αυτοί οι διψήφιοι αριθμοί μαζί με τη συχνότητά τους, σε αύξουσα διάταξη, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης. Η εμφάνιση θα περιλαμβάνει, σε κάθε γραμμή της οθόνης, τον διψήφιο αριθμό και τη συχνότητά του.

ΑΣΚΗΣΗ – 60

Ορισμένοι ακέραιοι θετικοί αριθμοί προκύπτουν ως γινόμενο δύο διαδοχικών πρώτων (primes) αριθμών. Ένας τέτοιος αριθμός είναι ο **2021** ($2021 = 43 * 47$), οι 43 και 47 είναι δύο διαδοχικοί πρώτοι αριθμοί.

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει όλους τους αριθμούς που πληρούν αυτήν την ιδιότητα, υπό την προϋπόθεση να είναι όλοι μικρότεροι του 5000.

Η λίστα που πρέπει να προκύπτει ως αποτέλεσμα του πηγαίου κώδικα είναι :

6, 15, 35, 2021, 2491,.....

Ο αριθμός μαζί με τους δύο διαδοχικούς primes συνθέτουν μία τριάδα: (αριθμός, πρώτος prime, διαδοχικός prime), π.χ. **2021, 43, 47**.

Να εισάγετε κατά σειρά όλες αυτές τις τριάδες σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραίων αριθμών, γνωστού μεγέθους N .

Στο τέλος να εμφανίσετε τον πίνακα των τριάδων, μία τριάδα ανά γραμμή της οθόνης.

ΑΣΚΗΣΗ – 61

Ένας αριθμός Proth είναι ένας φυσικός αριθμός z της μορφής : $z = k \cdot 2^n + 1$, όπου :

k, n είναι θετικοί ακέραιοι αριθμοί,

ο k είναι περιττός και πρέπει να ισχύει ότι $2^n > k$.

Ένας Proth prime είναι κάθε αριθμός Proth που είναι ταυτόχρονα και prime (πρώτος).

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει, σε αύξουσα σειρά, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης, όλους τους Proth primes που είναι μικρότεροι του 3000.

Η ακολουθία των Proth primes που θα προκύψει από την εκτέλεση του πηγαίου κώδικα είναι :

3, 5, 13,, 2753

ΑΣΚΗΣΗ – 62

Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα εξής:

1. Δημιουργία δύο αριθμητικών πινάκων, έστω **a** και **b**, με πλήθος στοιχείων **N** και **M** (**N** και **M** είναι γνωστά), με τυχαίους αριθμούς στην περιοχή [10,999].
2. Εύρεση της τιμής **k** που είναι ο πρώτος (prime) αριθμός που είναι ο αμέσως μεγαλύτερος από τη μέση τιμή των τιμών που περιέχονται και στους δύο πίνακες **a** και **b**.
3. Δημιουργώντας όλους τους συνδυασμούς αθροισμάτων $a_i + b_j$ (δηλ. ένα στοιχείο του πίνακα **a** και ένα στοιχείο του πίνακα **b**), να βρείτε και να καταχωρήσετε σε έναν νέο αριθμητικό πίνακα όλα αυτά τα ζεύγη τιμών των οποίων το άθροισμα είναι ίσο με **k-1**, **k** ή **k+1**.
4. Να εμφανίσετε τον νέο αριθμητικό πίνακα ως εξής:
 - a. Πρώτα όλα τα ζεύγη με άθροισμα ίσο με **k**, καθώς και το πλήθος τους
 - b. Στη συνέχεια όλα τα ζεύγη με άθροισμα ίσο με **k-1**, καθώς και το πλήθος τους
 - c. Τέλος όλα τα ζεύγη με άθροισμα ίσο με **k+1**, καθώς και το πλήθος τους

ΠΑΡΑΔΕΙΓΜΑ

a = {8, 12, 9, 20, 30}

b = {9, 10, 3, 20, 29, 17, 158}

μέση τιμή των στοιχείων από τους **a** και **b**: 27.08333 (πλησιέστερος ακέραιος αριθμός = 27) και ο αμέσως επόμενος prime **k** = 29.

Ζεύγη (που προκύπτουν από το βήμα 3, πρώτο στοιχείο του ζεύγους από τον πίνακα **a**, δεύτερο στοιχείο από τον πίνακα **b**):

(8,20), (12,17), (9,20), (20,9), (20,10) (θα ακολουθούν τα υπόλοιπα αποτελέσματα)

ΑΣΚΗΣΗ – 63

Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα εξής:

1. Δημιουργία τριών θετικών ακέραιων τιμών: **s**, **p**, **n** με τους ακόλουθους περιορισμούς, με κατάλληλη χρήση της συνάρτησης rand().
$$2 \leq n \leq 5$$

p είναι prime (πρώτος) αριθμός και $p < 100$

s αντιπροσωπεύει ένα άθροισμα, $p < s$
2. Να βρεθούν οι **n** σε πλήθος πρώτοι (primes) αριθμοί που είναι μεγαλύτεροι του **p**, των οποίων το άθροισμα ισούται με **s**. Όλα αυτά τα σύνολα πρώτων αριθμών θα πρέπει να αποθηκεύονται διαδοχικά σε έναν αριθμητικό πίνακα
3. Να εμφανιστούν τα στοιχεία του αριθμητικού πίνακα, με ένα σύνολο πρώτων αριθμών σε κάθε γραμμή της οθόνης.

ΠΑΡΑΔΕΙΓΜΑΤΑ

Input: $n = 3, p = 2, s = 23$
Output: $[3, 7, 13]$
 $[5, 7, 11]$

Input: $n = 4, p = 3, s = 54$
Output:
 $[5, 7, 11, 31]$
 $[5, 7, 13, 29]$
 $[5, 7, 19, 23]$
 $[5, 13, 17, 19]$
 $[7, 11, 13, 23]$
 $[7, 11, 17, 19]$

ΑΣΚΗΣΗ – 64

Δύο πραγματικοί αριθμοί, έστω x και y , δημιουργούν, ο καθένας, μία ακολουθία τιμών. Οι δύο ακολουθίες ορίζονται ως εξής:

Ο αριθμός x ορίζει την ακολουθία a_0, a_1, a_2, \dots ενώ ο αριθμός y την ακολουθία b_0, b_1, b_2, \dots σύμφωνα με τους παρακάτω τύπους:

$$a_{n+1} = \frac{a_n + b_n}{2}$$
$$b_{n+1} = \sqrt{a_n \cdot b_n}$$

Να γραφεί πρόγραμμα σε γλώσσα C που :

1. θα δημιουργεί N (N =γνωστό) ζεύγη τυχαίων θετικών πραγματικών αριθμών (τύπου double) x, y υπό την προϋπόθεση ότι σε κάθε ζεύγος οι δύο αριθμοί διαφέρουν τουλάχιστον κατά 10.0.
2. Για κάθε ζεύγος πραγματικών αριθμών x και y : θα καταχωρεί κάθε στοιχείο της κάθε ακολουθίας σε δύο αντίστοιχους μονοδιάστατους αριθμητικούς πίνακες **μέχρις ότου** η διαφορά δύο ομόλογων όρων στις δύο ακολουθίες γίνει μικρότερη από 10^{-6} .
3. θα εμφανίζει όλα τα στοιχεία των δύο πινάκων (ένα ζεύγος τιμών a_i, b_i σε κάθε γραμμή της οθόνης) καθώς και το πλήθος αυτών των ζευγών, για κάθε ζεύγος πραγματικών αριθμών x και y .

ΠΑΡΑΔΕΙΓΜΑ για $x = 24.0$ και $y = 6.0$

| n | a_n | b_n |
|-----|-----------|-----------|
| 0 | 24.0 | 6.0 |
| 1 | 15.0 | 12.0 |
| 2 | 13.500000 | 13.416408 |
| ... | | |

ΑΣΚΗΣΗ – 65

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει (μία τριάδα σε κάθε γραμμή της οθόνης) όλες τις διαφορετικές τριάδες ακεραίων θετικών αριθμών a, b, n για τις οποίες ισχύει ότι

$$a^n + b^n \text{ είναι ένας πρώτος (prime) αριθμός}$$

Το πρόγραμμα θα σταματά όταν βρεθούν οι 100 μικρότεροι και διαφορετικοί prime αριθμοί ή δημιουργηθεί prime αριθμός μεγαλύτερος από το 30000.

Όλοι αυτοί οι πρώτοι (primes) αριθμοί θα πρέπει να εισάγονται σε έναν μονοδιάστατο αριθμητικό πίνακα. Τα στοιχεία αυτού του πίνακα, στο τέλος, θα εμφανίζονται σε αύξουσα σειρά (χωρίς να χρησιμοποιηθεί διαδικασία – αλγόριθμος ταξινόμησης).

ΠΑΡΑΔΕΙΓΜΑ

$$2^2 + 3^2 = 4 + 9 = 13 \text{ είναι πρώτος (prime)}$$

$$3^2 + 2^2 = 9 + 4 = 13 \text{ είναι πρώτος (prime) αλλά έχει ήδη βρεθεί!!!!}$$

...

$$6^4 + 7^4 = 3697$$

...

ΑΣΚΗΣΗ – 66

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει όλους τους θετικούς ακέραιους αριθμούς $n > 1$ για τους οποίους ισχύει ότι υπάρχει ένας τουλάχιστον πρώτος (prime) αριθμός p που ικανοποιεί τη σχέση

$$n^2 \leq p \leq (n+1)^2$$

Το πρόγραμμα θα σταματά σε εκείνη την τιμή του n στην οποία θα βρεθεί για 1^n φορά prime αριθμός p μεγαλύτερος από 32000.

Για κάθε τιμή n το πρόγραμμα:

1. θα εμφανίζει σε αύξουσα διάταξη τους σχετικούς πρώτους (primes) αριθμούς
2. θα καταχωρεί όλους τους primes, που θα δημιουργηθούν κατά την εκτέλεση του προγράμματος, σε έναν μονοδιάστατο αριθμητικό πίνακα
3. θα εμφανίζει κατά φθίνουσα διάταξη τη συχνότητα εμφάνισης κάθε prime χωρίς να χρησιμοποιηθεί διαδικασία – αλγόριθμος ταξινόμησης. Σε κάθε γραμμή της οθόνης θα πρέπει να εμφανίζεται ένας prime και η συχνότητά του.

ΑΣΚΗΣΗ – 67

Να γραφεί πρόγραμμα σε γλώσσα C που για κάθε πρώτο (prime) αριθμό $p > 5$ θα βρίσκει τον μεγαλύτερο ακέραιο αριθμό n ο οποίος διαιρεί το πολυώνυμο $p^4 - 1$.

Το πρόγραμμα θα ελέγχει τους 200 πρώτους prime αριθμούς και θα καταχωρεί σε έναν μονοδιάστατο αριθμητικό πίνακα όλους τους σχετικούς αριθμούς n .

Στο τέλος θα πρέπει να εμφανίζονται οι συχνότητες αυτών των αριθμών n σε φθίνουσα διάταξη χωρίς να χρησιμοποιηθεί διαδικασία – αλγόριθμος ταξινόμησης. Σε κάθε γραμμή της οθόνης θα εμφανίζεται η τιμή του n και η συχνότητά της.

ΑΣΚΗΣΗ – 68

Ο αριθμός 2022 είναι σύνθετος τετραψήφιος αριθμός. Δημιουργείται από δύο διαδοχικούς άρτιους αριθμούς σε σειρά, τους 20 και 22. Επόμενος τέτοιος αριθμός είναι ο 2224 και προηγούμενος ο 1820. Επίσης είναι πλούσιος (ή υπερτέλειος) αριθμός (abundant number) δηλαδή είναι ένας αριθμός που είναι μικρότερος από το άθροισμα των γνησίων διαιρητών του. Οι γνήσιοι διαιρέτες του είναι 1, 2, 3, 6, 337, 674, 1011 που έχουν άθροισμα το 2034.

Να γραφεί πρόγραμμα σε γλώσσα C το οποίο θα υλοποιεί τα ακόλουθα:

1. Θα καταχωρεί σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραίων θετικών αριθμών όλους τους τετραψήφιους θετικούς αριθμούς (περιοχή τιμών [1000,9999]) που πληρούν και τις ανωτέρω δύο ιδιότητες, όπως ο αριθμός 2022.
2. Θα εμφανίζει όλους αυτούς τους αριθμούς καθώς και το πλήθος τους.

ΑΣΚΗΣΗ – 69

Ο αριθμός 2022 είναι σύνθετος τετραψήφιος αριθμός. Δημιουργείται από δύο διαδοχικούς άρτιους αριθμούς σε σειρά, τους 20 και 22, και βρίσκεται μεταξύ δύο διαδοχικών πρώτων αριθμών, των 2017 και του 2027, και μάλιστα είναι ο μέσος όρος τους, $2022 = (2017 + 2027) / 2$.

Να γραφεί πρόγραμμα σε γλώσσα C το οποίο θα υλοποιεί τα ακόλουθα:

1. Θα καταχωρεί σε έναν μονοδιάστατο αριθμητικό πίνακα ακεραίων θετικών αριθμών όλους τους τετραψήφιους θετικούς αριθμούς (περιοχή τιμών [1000,9999]) που πληρούν και τις ανωτέρω δύο ιδιότητες, όπως ο αριθμός 2022.
2. Θα εμφανίζει όλους αυτούς τους αριθμούς καθώς και το πλήθος τους.

(ΥΠΟΔΕΙΞΗ : Το παρακάτω τμήμα κώδικα σε γλώσσα C βρίσκει αν ένας ακέραιος και θετικός αριθμός $k \geq 2$ είναι πρώτος (prime) αριθμός).

```
i=2; flag=0;
while ((i<=k/2) && (flag==0))
{
    if (k%i==0) flag=1;
    i++;
}
if (flag==0) printf("number %4d is prime \n",k);
```

ΑΣΚΗΣΗ – 70

1. Χρησιμοποιώντας κατάλληλα τη συνάρτηση **rand()** να δημιουργήσετε δύο τυχαίους αριθμούς, έστω a, m για τους οποίους θα πρέπει να ισχύει ότι:

$$100 < a < 1000 \quad \text{και} \quad 7 \leq m \leq 40$$

2. Χρησιμοποιώντας τον κώδικα που υπάρχει στο τέλος για την εύρεση ενός prime (πρώτου) αριθμού:

Να δημιουργήσετε δύο μονοδιάστατους πίνακες, μεγέθους $N = 200$ ο καθένας, οι οποίοι θα περιέχουν :

- Ο πρώτος όλους τους συνεχόμενους primes, σε πλήθος m , που είναι μικρότεροι του a
- Ο δεύτερος όλους τους συνεχόμενους primes, σε πλήθος m , που είναι μεγαλύτεροι του a

Παράλληλα να βρείτε και το ακριβές πλήθος των μη μηδενικών τιμών που περιέχονται σε καθέναν από τους ανωτέρω δύο πίνακες.

Στη συνέχεια να εμφανίσετε κάθε πίνακα ξεχωριστά, μόνον τα μη μηδενικά στοιχεία του.

3. Χρησιμοποιώντας όλα τα στοιχεία των δύο πινάκων:

- a. να βρείτε για κάθε στοιχείο τον αριθμό που αποτελείται από τα 2 τελευταία ψηφία του (π.χ. από τον αριθμό 277 θα προκύπτει ο αριθμός 77)
- b. β. Τη συχνότητα εμφάνισης όλων αυτών των διψήφιων αριθμών που προκύπτουν από τους 2 πίνακες. Η εμφάνιση των συχνοτήτων θα πρέπει να γίνει κατά αύξουσα σειρά των διψήφιων αριθμών. Σε κάθε γραμμή της οθόνης θα εμφανίζεται ένας διψήφιος και η συχνότητά του (δηλ. 2 τιμές).

ΠΑΡΑΔΕΙΓΜΑ Αν : $a = 271, m = 9$

Τα αποτελέσματα για τους 2 πίνακες θα είναι

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-------|
| 269 | 263 | 257 | 251 | 241 | 239 | 233 | 229 | 227 | 0 | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-------|

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-------|
| 277 | 281 | 283 | 293 | 307 | 311 | 313 | 317 | 331 | 0 | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-------|

ΑΣΚΗΣΗ – 71

Η βαθμολογία σε ιστιοπλοϊκούς αγώνες εξαρτάται από:

- την απόσταση x κάθε ιστιοδρομίας, σε ναυτικά μίλια,
- το πλήθος των συμμετοχών s , και
- τη σειρά τερματισμού t

Ο τύπος που παρέχει τη βαθμολογία b ανάλογα με τη σειρά τερματισμού είναι:

$$b = 1000 \cdot e^{(-0.15 \cdot (t-1))} \cdot \sqrt{(x / 100)} \cdot \sqrt{s / 10}$$

(*Σημείωση* : Δεν τερματίζουν πάντοτε όλα τα σκάφη που συμμετέχουν)

Να γράψετε ένα πρόγραμμα σε γλώσσα C που θα δημιουργεί τυχαίες τιμές για τις μεταβλητές ως εξής:

$$50 \leq x \leq 90 \quad \text{και} \quad 8 \leq s \leq 15$$

και θα εμφανίζει τον πίνακα βαθμολογίας, ανάλογα με τη σειρά τερματισμού ως εξής :

Παράδειγμα για $x = 70$, $s = 10$

```
distance = 70 symmetoxes= 10
```

| | | | | | | | | | | | | | | | | | | | | |
|------------|----|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|--|--|--|--|--|--|--|
| TERMATISAN | 1 | SKAFH: | 264.6 | | | | | | | | | | | | | | | | | |
| TERMATISAN | 2 | SKAFH: | 374.2 | 322.0 | | | | | | | | | | | | | | | | |
| TERMATISAN | 3 | SKAFH: | 458.3 | 394.4 | 339.5 | | | | | | | | | | | | | | | |
| TERMATISAN | 4 | SKAFH: | 529.2 | 455.4 | 392.0 | 337.4 | | | | | | | | | | | | | | |
| TERMATISAN | 5 | SKAFH: | 591.6 | 509.2 | 438.3 | 377.2 | 324.7 | | | | | | | | | | | | | |
| TERMATISAN | 6 | SKAFH: | 648.1 | 557.8 | 480.1 | 413.2 | 355.7 | 306.1 | | | | | | | | | | | | |
| TERMATISAN | 7 | SKAFH: | 700.0 | 602.5 | 518.6 | 446.3 | 384.2 | 330.7 | 284.6 | | | | | | | | | | | |
| TERMATISAN | 8 | SKAFH: | 748.3 | 644.1 | 554.4 | 477.2 | 410.7 | 353.5 | 304.2 | 261.9 | | | | | | | | | | |
| TERMATISAN | 9 | SKAFH: | 793.7 | 683.2 | 588.0 | 506.1 | 435.6 | 374.9 | 322.7 | 277.8 | 239.1 | | | | | | | | | |
| TERMATISAN | 10 | SKAFH: | 836.7 | 720.1 | 619.8 | 533.5 | 459.2 | 395.2 | 340.2 | 292.8 | 252.0 | 216.9 | | | | | | | | |

ΑΣΚΗΣΗ – 72

Η δημιουργία των προσωπικών κωδικών στις τραπεζικές κάρτες γίνεται με την ακόλουθη διαδικασία:

1. Η τράπεζα αποστέλλει σε κάθε πελάτη μία σειρά από 16 ακέραια ψηφία (0-9).
2. Ο πελάτης λαμβάνει στο κινητό του τηλέφωνο ένα μήνυμα που αποτελείται από 4 αριθμούς στο διάστημα 1-16.
3. Με βάση τα δεδομένα του 1^{ου} βήματος ο πελάτης, επιλέγοντας τα ψηφία που αντιστοιχούν στις θέσεις που προκύπτουν από το βήμα 2, δημιουργεί τον τετραψήφιο αριθμό PIN. **Εάν το 1^ο ψηφίο του PIN προκύψει ίσο με το μηδέν να αντικαθίσταται από το 1^ο μη μηδενικό ψηφίο από τα 16 ακέραια ψηφία του βήματος 1.**

Να γράψετε ένα πρόγραμμα σε γλώσσα C που θα υλοποιεί τα ανωτέρω τρία βήματα για έναν πελάτη, χρησιμοποιώντας δεδομένα που θα προκύψουν από κατάλληλη χρήση της συνάρτησης `rand()`. Τα δεδομένα και τα αποτελέσματα πρέπει να εμφανίζονται στην οθόνη.

Για τον υπολογισμό του PIN πρέπει να χρησιμοποιήσετε τον τύπο του αθροιστή.

Στη συνέχεια να βρείτε τους 10 επόμενους του PIN πρώτους αριθμούς, υπό την προϋπόθεση να είναι όλοι <10000, και να τους καταχωρήσετε σε έναν μονοδιάστατο πίνακα, χρησιμοποιώντας κατάλληλα τον κώδικα που υπάρχει στο τέλος. Να εμφανίσετε τα στοιχεία του πίνακα αυτού.

ΠΑΡΑΔΕΙΓΜΑ

```
Digits from bank
1 9 6 8 0 8 6 9 8 2 4 5 1 9 8 1

locations for PIN
12 2 13 8

PIN = 5919
Primes : 5923 5927 5939 5953 5981 5987 6007 6011 6029 6037
```

(ΥΠΟΔΕΙΞΗ : Το παρακάτω τμήμα κώδικα σε γλώσσα C βρίσκει αν ένας ακέραιος και θετικός αριθμός $k \geq 2$ είναι πρώτος (prime) αριθμός).

```
i=2; flag=0;
while ((i<=k/2) && (flag==0))
{
    if (k%i==0) flag=1;
    i++;
}
if (flag==0) printf("number %4d is prime \n",k);
```

ΑΣΚΗΣΗ – 73

Η ανάληψη της κατασκευής ενός τεχνικού έργου προϋποθέτει:

- A) την κατάθεση τεχνικής μελέτης και οικονομικής προσφοράς και
- B) την αξιολόγησή τους και τελικά τη συνολική τους βαθμολόγηση.

Η βαθμολόγηση αξιολογεί 10 διαφορετικά κριτήρια. Κάθε κριτήριο βαθμολογείται από 0 έως 10 (10 είναι η μεγαλύτερη βαθμολογία)

Να θεωρήσετε ότι έχουν κατατεθεί Ν προσφορές (η τιμή του Ν ορίζεται ως σταθερά)

Για κάθε κατατεθειμένη προσφορά να δημιουργήσετε με χρήση της συνάρτησης rand() τα εξής δεδομένα:

1. Κωδικός προσφοράς (τριψήφιος ακέραιος 100-999). Δεν απαιτείται έλεγχος ώστε ο κωδικός κάθε προσφοράς να είναι μοναδικός.
2. Ποσό οικονομικής προσφοράς, θετικός αριθμός τύπου double
3. Βαθμολόγηση της προσφοράς, 10 ακέραιες τιμές 0 -10 (σύμφωνα με τα παραπάνω)

Π.χ.

```
397 45000.0      8  8  7  9  10  3  8  5  7  6
714 47800.0      10 5  2  8  7  9  7  3  4  9
.....
```

Η τελική βαθμολογία προκύπτει από το άθροισμα :

$$\text{Πλήθος } 10 * 10 + \text{πλήθος } 9*9 + \text{πλήθος } 8*8 + \dots \text{πλήθος } 5*5$$

Στην τελική βαθμολογία δεν συμμετέχουν οι βαθμολογίες που είναι μικρότερες από 5.

Παράδειγμα βαθμολογίας για τις 2 σειρές δεδομένων:

| Βαθμολογίες | Κωδ. Προσφοράς 397 | Κωδ. Προσφοράς 714 |
|--------------------------|--------------------|--------------------|
| 10 | 1 | 10 |
| 9 | 1 | 9 |
| 8 | 3 | 24 |
| 7 | 2 | 14 |
| 6 | 1 | 6 |
| 5 | 1 | 5 |
| Τελική βαθμολογία | | 68 |

Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα εξής :

1. θα εισάγει διαδοχικά τα δεδομένα για Ν προσφορές με βάση όσα αναφέρονται ανωτέρω
2. θα αποθηκεύει σε τρεις μονοδιάστατους πίνακες, κατά σειρά, για κάθε προσφορά τον κωδικό προσφοράς, το ποσό οικονομικής προσφοράς, την τελική βαθμολογία
3. Θα βρίσκει και θα εμφανίζει ποια προσφορά έχει την υψηλότερη τελική βαθμολογία. Να θεωρήσετε ότι θα υπάρχει μόνον μία τέτοια προσφορά. Η εμφάνιση θα περιλαμβάνει, σε μία γραμμή της οθόνης :
τον κωδικό προσφοράς, το ποσό οικονομικής προσφοράς, την τελική βαθμολογία

ΑΣΚΗΣΗ – 74

Ένας ακέραιος θετικός αριθμός, έστω n , ονομάζεται 2-σχεδόν πρώτος (2-almost prime) εάν έχει δύο (όχι απαραίτητα διαφορετικούς) πρώτους (prime) γνήσιους διαιρέτες (proper divisors). Γνήσιος διαιρέτης ενός θετικού ακέραιου αριθμού θεωρείται κάθε διαιρέτης του αριθμού για τον οποίο ισχύει :

$$1 < \text{γνήσιος διαιρέτης} < n$$

Η σχετική ακολουθία είναι των 2-almost primes είναι:

4, 6, 9, 10, 14, 15, 21, 22, 25, ..., 119, 121, 122, ...

(4 = 2*2, 6=2*3, 9=3*3, 10=2*5, ..., 119=7*17)

Το τμήμα κώδικα σε γλώσσα C που ακολουθεί βρίσκει και εμφανίζει όλους τους γνήσιους διαιρέτες ενός ακέραιου θετικού αριθμού $n \geq 2$:

```
do
{
    printf("n="); scanf("%d",&n);
}
while (n <= 1);

c = n;
while (c > 1)
{
    for (i = 2; i <= c; i++)
        if (c % i == 0)
        {
            printf("proper divisor of %4d is %3d =\n",n,i);
            c = c / i;
        }
}
```

Ένας ακέραιος θετικός αριθμός, έστω k , λέγεται πρώτος (prime) εάν διαιρείται μόνον από τον εαυτό του και τη μονάδα.

Το παρακάτω τμήμα κώδικα σε γλώσσα C βρίσκει αν ένας ακέραιος και θετικός αριθμός $k \geq 2$ είναι πρώτος (prime) αριθμός).

```
i=2; flag=0;
while ((i<=k/2) && (flag==0))
{
    if (k%i==0) flag=1;
    i++;
}
if (flag==0) printf("number %4d is prime \n",k);
```

Να γράψετε ένα πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει τους **50 (κατά αύξουσα σειρά)** σε πλήθος 2-almost primes, ξεκινώντας από τον πρώτο 2-almost prime που είναι ο αριθμός 4.

(ΥΠΟΔΕΙΞΗ 1: πρέπει να χρησιμοποιήσετε κατάλληλα τα ανωτέρω δύο τμήματα κώδικα).

(ΥΠΟΔΕΙΞΗ 2: Υπάρχουν περισσότεροι από 2-almost primes θετικοί ακέραιοι αριθμοί).

Προτεινόμενη αλγοριθμική διαδικασία:

Για κάθε ακέραιο θετικό αριθμό $n \geq 2$

- Βρίσκω και αποθηκεύω σε κατάλληλη δομή όλους τους γνήσιους και πρώτους (prime) διαιρέτες του (ΥΠΟΔΕΙΞΗ : να χρησιμοποιήσετε συνδυαστικά τα ανωτέρω δύο τμήματα κώδικα)
- Βρίσκω αν το πλήθος αυτών των γνήσιων και πρώτων διαιρέτων είναι ακριβώς 2.
- Εάν το πλήθος αυτό είναι 2 τότε αποθηκεύω τον θετικό ακέραιο αριθμό σε μία κατάλληλη δομή

Επαναλαμβάνω τη διαδικασία μέχρι να μετρήσω 50 τέτοιους 2-almost prime αριθμούς.

Εμφανίζω τη δομή των αριθμών που προέκυψαν.