



4η Σειρά Ασκήσεων - Δυναμική Παραχώρηση Μνήμης

Άσκηση 4.1

Να γραφεί πρόγραμμα που να βρίσκει και να εμφανίζει το μεγαλύτερο από n σε πλήθος στοιχείων ενός πίνακα, χρησιμοποιώντας δυναμική παραχώρηση μνήμης. Η τιμή του n να καθορίζεται κατά τη διάρκεια της εκτέλεσης του προγράμματος μέσω της `scanf`, με τον περιορισμό $n > 0$. Να υλοποιήσετε τη λύση χρησιμοποιώντας: (α) τη συνάρτηση `malloc()` και (β) τη συνάρτηση `calloc()`.

Άσκηση 4.2

Να γραφεί συνάρτηση που να δημιουργεί έναν πίνακα με τυχαίους ακέραιους αριθμούς στο διάστημα `[50-100]`. Το μέγεθος του πίνακα να καθορίζεται κατά την εκτέλεση του προγράμματος από τη συνάρτηση `main()`, ελέγχοντας παράλληλα ότι είναι θετικός αριθμός. Η συνάρτηση `main()`, θέλουμε να εμφανίζει τους τυχαίους αριθμούς.

Άσκηση 4.3

Να γραφεί συνάρτηση `void remove_min` που να διαγράφει το μικρότερο ή τα μικρότερα σε τιμή στοιχεία ενός μονοδιάστατου αριθμητικού πίνακα N θέσεων ($N = \text{γνωστό}$). Στη συνέχεια να γραφεί κατάλληλο πρόγραμμα που να κάνει χρήση αυτής της συνάρτησης.

Άσκηση 4.4

Χρησιμοποιώντας δυναμική παραχώρηση μνήμης, να γραφεί πρόγραμμα που:

1. δημιουργεί έναν μονοδιάστατο, μη ταξινομημένο πίνακα a , μεγέθους n . Η τιμή του n θα καθορίζεται κατά τη διάρκεια της εκτέλεσης του προγράμματος μέσω της `scanf`, με τον περιορισμό $n > 0$.
2. διαβάζει μία τιμή, έστω s .
3. βρίσκει και εμφανίζει έναν υποπίνακα b του a , (που αποτελείται από συνεχόμενα στοιχεία του a), το άθροισμα των οποίων θα ισούται με s .

Παράδειγμα:

`n=8, s=20`

`a={4, 2, 10, 3, -3, 10, 7, 5}`

`b={10, 3, -3, 10}`

Άσκηση 4.5

Θεωρείστε έναν μονοδιάστατο πίνακα ακεραίων θετικών αριθμών. Ο πίνακας περιέχει N , ($N > 200$) ζεύγη ακεραίων θετικών τιμών. Η τιμή του N καθορίζεται κατά τη διάρκεια της εκτέλεσης

της `main()`. Κάθε ζεύγος τιμών, αντιστοιχεί στον αύξοντα αριθμό εβδομάδας (επιτρεπτές τιμές 1-52, όλες οι εβδομάδες αφορούν το ίδιο ημερολογιακό έτος) και στις ώρες εκτός λειτουργίας, εντός της συγκεκριμένης εβδομάδας, μιας οποιασδήποτε εργαλειομηχανής σε μια βιομηχανική μονάδα. Κάθε αύξων αριθμός εβδομάδας, μπορεί να υπάρχει περισσότερες από μια φορές και ο πίνακας δεν είναι ταξινομημένος ως προς τον αύξοντα αριθμό εβδομάδας, δηλ. Τα ζεύγη υπάρχουν με τυχαία σειρά. Ενδέχεται επίσης, να μην υπάρχουν όλοι οι δυνατικοί αύξοντες αριθμοί εβδομάδας στον πίνακα.

Να γραφεί μια συνάρτηση με όνομα `weeks_off` που να δέχεται ως όρισμα εισόδου αυτό τον πίνακα. Η συνάρτηση θέλουμε να επιστρέφει στη `main()` τα παρακάτω:

- Ένα νέο μονοδιάστατο πίνακα που να περιλαμβάνει σε αύξουσα σειρά (χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης), όλους τους υπάρχοντες αύξοντες αριθμούς εβδομάδας, καθέναν από μια φορά.
- Ένα νέο μονοδιάστατο πίνακα που να περιλαμβάνει σε αύξουσα σειρά (χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης), ζεύγη θετικών τιμών: (α/α εβδομάδας, συνολικός χρόνος εκτός λειτουργίας για την εβδομάδα, για όλες τις εργαλειομηχανές), για όλους τους υπάρχοντες α/α εβδομάδας του αρχικού πίνακα εισόδου.

Σημείωση: η συνάρτηση ΔΕΝ θα πρέπει να περιέχει εντολές `printf`.

Παράδειγμα με `N=8`:

Αρχικός Πίνακας:

13	2	19	4	33	8	8	11	13	9	8	5	19	4	50	50
----	---	----	---	----	---	---	----	----	---	---	---	----	---	----	----

Νέος πίνακας μόνο με τους κωδικούς:

8	13	19	33	50
---	----	----	----	----

Νέος πίνακας ζευγών:

8	16	13	11	19	8	33	8	50	40
---	----	----	----	----	---	----	---	----	----

Στη συνέχεια να γραφεί πρόγραμμα που:

- να δίνει τιμές στα ορίσματα εισόδου της συνάρτησης `weeks_off`, εντός των αποδεκτών ορίων τιμών με κατάλληλη χρήση της συνάρτησης δημιουργίας τυχαίων αριθμών `rand()`.
- να καλεί τη συνάρτηση `weeks_off`, και να εμφανίζει τα σχετικά αποτελέσματα.

Άσκηση 4.6

Δύο μονοδιάστατοι πίνακες, έστω `a` και `b`, είναι όμοιοι εάν περιέχουν ακριβώς τα ίδια στοιχεία, όχι απαραίτητα στις ίδιες θέσεις.

Να γραφεί μια συνάρτηση με όνομα `check_identical` με ορίσματα εισόδου δύο μονοδιάστατους πίνακες ακεραίων θετικών αριθμών με ίσιο μέγεθος `N` (`N=γνωστό`).

Η συνάρτηση `check_identical` θα επιστρέφει στη `main()` τα παρακάτω:

- την τιμή 0 εάν οι δύο πίνακες περιέχουν ακριβώς τα ίδια στοιχεία.
- την τιμή 1 εάν οι δύο πίνακες ΔΕΝ περιέχουν ακριβώς τα ίδια στοιχεία, καθώς και έναν νέο πίνακα που θα περιέχει σε αύξουσα διάταξη, (χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης, όλα τα μη κοινά στοιχεία των δύο πινάκων).

Σημείωση: η συνάρτηση ΔΕΝ θα πρέπει να περιέχει εντολές `printf`.

Στη συνέχεια να γραφεί ένα πρόγραμμα που να:

1. γεμίζει τους πίνακες `a` και `b`, με κατάλληλη χρήση της συνάρτησης δημιουργίας τυχαίων αριθμών `rand()`, ώστε οι τιμές να ανήκουν στο διάστημα `[1, 30]`
2. καλεί τη συνάρτηση `check_identical`, και εμφανίζει τα σχετικά αποτελέσματα.

Παράδειγμα με `N=8`:

Πίνακας `a`:

4	3	8	14	11	9	3	17
---	---	---	----	----	---	---	----

Πίνακας `b`:

14	9	7	19	17	3	9	8
----	---	---	----	----	---	---	---

Αποτελέσματα στη `main()`:

1 (δηλαδή, οι πίνακες δεν είναι όμοιοι)

Μη κοινά στοιχεία σε αύξουσα διάταξη: 4, 7, 11, 19

Άσκηση 4.7

Η επιφάνεια ενός πολυγώνου στο επίπεδο (2 διαστάσεις) υπολογίζεται με τη μέθοδο του **πολυγώνου Gauss (Shoelace theorem)**, ως εξής:

$$A = \frac{1}{2} \left| \sum_{i=1}^n (x_i \cdot y_{i+1} - x_{i+1} \cdot y_i) \right|$$

όπου $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ και $(x_{n+1}, y_{n+1}) = (x_1, y_1)$, οι συντεταγμένες των κορυφών.

Να υλοποιήσετε συνάρτηση `calculateArea` που να δέχεται ως είσοδο τις συντεταγμένες ενός πολυγώνου και να υπολογίζει την επιφάνειά του.

Στη συνέχεια να γραφεί ένα πρόγραμμα που:

1. Ο χρήστης θα εισάγει τον αριθμό των κορυφών του πολυγώνου.
2. Το πρόγραμμα θα ζητήσει τις συντεταγμένες (x, y) για κάθε κορυφή.
3. Θα καλείται τη συνάρτηση `calculateArea`
4. Θα εμφανίζει την επιφάνεια του πολυγώνου με ακρίβεια δύο δεκαδικών ψηφίων.

Άσκηση 4.8

Να γραφεί μια συνάρτηση με όνομα `find_primes` που να δέχεται ως είσοδο ένα μονοδιάστατο πίνακα ακεραίων αριθμών μεγέθους N (N =γνωστό), που θα γεμίζει στη συνάρτηση `main()` με τυχαίες τιμές στο διάστημα $[0, 999]$.

Η συνάρτηση θα επιστρέφει στη `main()` ένα νέο πίνακα, που θα περιλαμβάνει μόνο τα στοιχεία του αρχικού πίνακα που είναι πρώτοι (primes) αριθμοί. Ο νέος πίνακας θα πρέπει να είναι ταξινομημένος σε αύξουσα διάταξη, χωρίς να χρησιμοποιηθεί διαδικασία ταξινόμησης. Αν υπάρξουν πολλαπλές ίδιες τιμές, αυτές θα πρέπει να εμφανίζονται όλες.

Σημείωση: η συνάρτηση ΔΕΝ θα πρέπει να περιέχει εντολές `printf`.

Στη συνέχεια, να γραφεί πρόγραμμα που:

1. να γεμίζει ένα μονοδιάστατο πίνακα N θέσεων με τυχαίους ακέραιους και θετικούς αριθμούς στο διάστημα $[0, 999]$. Η τιμή του N , να καθορίζεται κατά τη διάρκεια εκτέλεσης του προγράμματος.
2. να καλεί τη συνάρτηση `find_primes`.
3. να εμφανίζει τα αποτελέσματα που επιστρέφει η συνάρτηση `find_primes`.

Άσκηση 4.9

Δύο μονοδιάστατοι αριθμητικοί πίνακες θετικών ακεραίων αριθμών, έστω a , b με πλήθος στοιχείων m , n αντίστοιχα, περιέχουν θετικούς ακέραιους αριθμούς στην περιοχή $[0-99]$. Να εξασφαλίσετε ότι: κάθε ακέραιος αριθμός εμφανίζεται μόνον μία φορά σε κάθε πίνακα και ότι οι δύο πίνακες a , b δεν έχουν κανένα κοινό στοιχείο.

Να γραφεί μια συνάρτηση με όνομα `check_successive` με ορίσματα εισόδου τους πίνακες a , b . Η συνάρτηση αυτή να επιστρέφει στη συνάρτηση `main()`:

- Έναν νέο πίνακα που θα περιέχει σε αύξουσα διάταξη, ΧΩΡΙΣ να χρησιμοποιηθεί διαδικασία ταξινόμησης, όλα τα στοιχεία των δύο πινάκων a , b .
- Αν τα στοιχεία του νέου πίνακα δεν είναι απολύτως διαδοχικά θα πρέπει να επιστρέφονται επίσης σε μορφή μονοδιάστατου αριθμητικού πίνακα οι τιμές που λείπουν, στο διάστημα τιμών που οριοθετούν το πρώτο και το τελευταίο στοιχείο του νέου πίνακα.

(ΠΡΟΣΟΧΗ: η συνάρτηση ΔΕΝ θα πρέπει να περιέχει εντολές `printf`)

Στη συνέχεια να γραφεί ένα πρόγραμμα σε γλώσσα C που :

- θα γεμίζει τους πίνακες a , b με κατάλληλη χρήση της συνάρτησης δημιουργίας τυχαίων αριθμών `rand()` ώστε οι τιμές να ανήκουν στο διάστημα $[0-99]$. Το πλήθος των στοιχείων κάθε πίνακα θα προσδιορίζεται κατά τη διάρκεια της εκτέλεσης της `main()`.
- θα καλεί τη συνάρτηση `check_successive` και θα εμφανίζει τα σχετικά αποτελέσματα.

Παράδειγμα με $m = 8$ και $n = 6$:

Πίνακας a :

14	13	8	17	21	20	19	9
----	----	---	----	----	----	----	---

Πίνακας `b`:

7	5	22	11	16	25
---	---	----	----	----	----

Αποτελέσματα στη `main()`:

Ο Νέος Πίνακας είναι:

5	7	8	9	11	13	14	16	17	19	20	21	22	25
---	---	---	---	----	----	----	----	----	----	----	----	----	----

Πίνακας με τις τιμές που λείπουν στο διάστημα `[5, 25]`:

6	10	12	15	18	23	24
---	----	----	----	----	----	----