



## Προγραμματισμός Ι – 9<sup>η</sup> Σειρά Ασκήσεων (παράδοση και επίδειξη στο εργαστήριο: 8/1/2018)

1. Δημιουργήστε μία συνάρτηση που να δέχεται σαν παράμετρο έναν πίνακα και να ελέγχει αν είναι ταξινομημένος. Αν είναι ταξινομημένος σε αύξουσα σειρά να επιστρέφει 1, αν είναι σε φθίνουσα να επιστρέφει 2, αλλιώς κάποια άλλη αθέσιμη τιμή. Να γραφεί ένα πρόγραμμα το οποίο με χρήση της συνάρτησης `rand()` να δημιουργεί 10 τυχαίες ακέραιες τιμές στο `[5, 20]`, να τις αποθηκεύει σε έναν πίνακα και με χρήση της συνάρτησης να εμφανίζει μήνυμα για το αν είναι ταξινομημένος ή όχι.

2. Δημιουργήστε μία `void` συνάρτηση που να δέχεται σαν παραμέτρους δύο πίνακες και τον αριθμό (π.χ. `N`) των στοιχείων που επιθυμούμε να συγκριθούν. Αν τα `N` πρώτα στοιχεία των δύο πινάκων έχουν τις ίδιες τιμές, η συνάρτηση να επιστρέφει 1, αλλιώς 0. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 200 πραγματικούς αριθμούς τύπου `double` και να τους αποθηκεύει σε δύο πίνακες 100 θέσεων. Στη συνέχεια, το πρόγραμμα να διαβάζει τον αριθμό των στοιχείων που θα συγκριθούν και να χρησιμοποιεί τη συνάρτηση για να τους συγκρίνει και να εμφανίζει το αποτέλεσμα της σύγκρισης.

Υπόδειξη: Αφού ο τύπος επιστροφής της συνάρτησης είναι `void` να προσθέσετε μία παράμετρο δείκτη, ώστε να επιστρέψετε την τιμή επιστροφής.

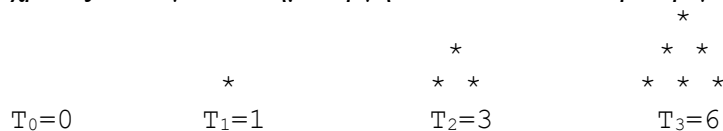
3. Δημιουργήστε μία συνάρτηση που να δέχεται σαν παράμετρο έναν πίνακα και να ελέγχει αν περιέχει τιμές που να επαναλαμβάνονται. Αν ναι, η συνάρτηση να επιστρέφει έναν δείκτη στο στοιχείο που επαναλαμβάνεται τις περισσότερες φορές, αλλιώς να επιστρέφει `NULL`. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 100 ακραίους, να τους αποθηκεύει σε έναν πίνακα, να καλεί τη συνάρτηση και να χρησιμοποιεί την τιμή επιστροφής της για να εμφανίσει την τιμή του στοιχείου με τις περισσότερες εμφανίσεις. (Σημ. Αν υπάρχουν παραπάνω από ένα στοιχεία με τον ίδιο αριθμό μέγιστων εμφανίσεων, το πρόγραμμα να εμφανίζει το πρώτο από αυτά).

4. Δημιουργήστε μία συνάρτηση που να δέχεται σαν παραμέτρους τρεις  $2 \times 4$  διδιάστατους πίνακες, να υπολογίζει το άθροισμα των δύο πρώτων και να το αποθηκεύει στον τρίτο. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 16 ακραίους, να τους αποθηκεύει σε δύο διδιάστατους πίνακες  $2 \times 4$  και να εμφανίζει το άθροισμά τους με χρήση της συνάρτησης.

5. Δημιουργήστε μία `void` συνάρτηση που να δέχεται σαν παραμέτρους έναν  $3 \times 4$  διδιάστατο πίνακα, τον αριθμό μίας γραμμής του και τον αριθμό μίας στήλης του και να επιστρέφει μέσω κατάλληλων παραμέτρων το μεγαλύτερο στοιχείο της γραμμής και το μικρότερο στοιχείο της στήλης. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 12 ακραίους και να τους αποθηκεύει σε έναν διδιάστατο πίνακα  $3 \times 4$ . Στη συνέχεια, να διαβάζει τον αριθμό μίας γραμμής και μίας στήλης του και να εμφανίζει το μεγαλύτερο στοιχείο της γραμμής και το μικρότερο στοιχείο της στήλης με χρήση της συνάρτησης.

6. Τροποποιήστε τη συνάρτηση `power()` της άσκησης K.11.11 του βιβλίου και χρησιμοποιήστε τον τύπο  $m^n = m \times m^{n-1}$  για να υπολογίσετε το  $m^n$  με αναδρομικό τρόπο.

7. Στα μαθηματικά, ένας τριαδικός αριθμός συμβολίζει τον αριθμό των αντικείμενων που χρειάζονται για να δημιουργήσουν ένα ισόπλευρο τρίγωνο, όπως φαίνεται παρακάτω.



Ο  $T(n)$  τριαδικός αριθμός δηλώνει τον αριθμό των αντικειμένων από τα οποία αποτελείται το ισόπλευρο τρίγωνο και είναι ίσος με το άθροισμα των  $n$  αριθμών από το 1 μέχρι το  $n$ . Επομένως, ο  $T(n)$  τριαδικός αριθμός εκφράζεται ως εξής:  $T(n) = 1 + 2 + 3 + \dots + (n-1) + n$

και σε αναδρομική μορφή: 
$$T(n) = \begin{cases} n, & \text{για } n = 0 \text{ ή } n = 1 \\ n + T(n-1), & \text{για } n > 1 \end{cases}$$

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έναν θετικό ακέραιο (π.χ.  $n$ ) μέχρι το 20 και να χρησιμοποιεί μία αναδρομική συνάρτηση για να εμφανίσει τον  $T(n)$  τριαδικό αριθμό.

8. Να γραφεί ένα πρόγραμμα το οποίο με χρήση της συνάρτησης `rand()` να δημιουργεί 10 ακέραιους που ανήκουν στο  $[0, 2]$  και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, το πρόγραμμα να καλεί μία συνάρτηση που να ελέγχει με αναδρομικό τρόπο αν ο πίνακας είναι συμμετρικός, δηλαδή αν το πρώτο του στοιχείο είναι ίσο με το τελευταίο, το δεύτερο με το προτελευταίο, κ.ο.κ. Αν ο πίνακας δεν είναι συμμετρικός, η συνάρτηση να επιστρέφει τη θέση του πρώτου στοιχείου που «χαλάει» τη συμμετρία.

9. Δημιουργήστε μία συνάρτηση που να δέχεται σαν παράμετρο έναν διδιάστατο πίνακα και έναν ακέραιο και να επιστρέφει έναν δείκτη στη γραμμή όπου αυτός ο αριθμός εμφανίζεται τις περισσότερες φορές. Αν ο αριθμός δεν υπάρχει στον πίνακα, η συνάρτηση να επιστρέφει `NULL`. Να γραφεί ένα πρόγραμμα το οποίο να θέτει τυχαίες τιμές σε έναν  $5 \times 5$  πίνακα ακεραίων, να διαβάζει έναν ακέραιο, να καλεί τη συνάρτηση και να χρησιμοποιεί την τιμή επιστροφής της για να εμφανίσει τα στοιχεία της γραμμής στην οποία περιέχεται τις περισσότερες φορές.

(Σημ: Προσοχή στην εκφώνηση!!!)

10. Να γραφεί ένα πρόγραμμα το οποίο, με χρήση της συνάρτησης `rand()`, να θέτει τυχαίες τιμές στο  $[1000, 2000]$  στα στοιχεία της κυρίας διαγωνίου ενός πίνακα ακεραίων  $5 \times 5$ . Στα υπόλοιπα στοιχεία να ανατεθούν τυχαίες τιμές, με τον περιορισμό οι τιμές των στοιχείων κάτω από την κύρια διαγώνιο να είναι μικρότερες από τις τιμές των στοιχείων της κυρίας διαγωνίου, καθώς και οι τιμές των στοιχείων πάνω από την κύρια διαγώνιο να είναι μεγαλύτερες από τις τιμές των στοιχείων της κυρίας διαγωνίου.

11. Να γραφεί ένα πρόγραμμα το οποίο, με χρήση της συνάρτησης `rand()`, να θέτει τυχαίες τιμές στο  $[0, 20]$  σε όλα τα στοιχεία ενός πίνακα ακεραίων  $3 \times 5$ , εκτός από αυτά της τελευταίας γραμμής. Σε αυτά τα στοιχεία πρέπει να τοποθετούνται κατάλληλες τιμές, ώστε το άθροισμα των στοιχείων κάθε στήλης του πίνακα να είναι ίσο με 50. Επίσης, το πρόγραμμα να τοποθετεί σε μονοδιάστατο πίνακα την ελάχιστη τιμή κάθε στήλης του διδιάστατου πίνακα και στο τέλος να τον εμφανίζει.

**12.** Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έως δέκα (το πολύ) θετικούς ακέραιους και να τους αποθηκεύει σε έναν πίνακα ακεραίων (έστω `a[10]`). Η εισαγωγή των ακεραίων να σταματάει, όταν ο χρήστης εισάγει κάποια αρνητική τιμή. Στη συνέχεια, το πρόγραμμα, με χρήση της συνάρτησης `rand()`, να «γεμίζει» με τυχαίους διψήφιους ακέραιους αριθμούς που ανήκουν στο `[10,99]` έναν διδιάστατο  $4 \times 4$  πίνακα (έστω `b[4][4]`). Τέλος, το πρόγραμμα να επιλέγει τυχαία (με χρήση της συνάρτησης `rand()`) μία από τις τιμές που έχουν αποθηκευτεί στον πίνακα `a` και να ελέγχει αν η τιμή αυτή υπάρχει σε κάποιο στοιχείο της κύριας ή της δευτερεύουσας διαγωνίου του πίνακα `b`. Αν υπάρχει, να εμφανίζει ένα μήνυμα με τη θέση του πίνακα `b` στην οποία βρέθηκε, ειδικά, να εμφανίζει κατάλληλο μήνυμα ότι δεν βρέθηκε η συγκεκριμένη τιμή.

Π.χ. Αν οι πίνακες `a` και `b` είχαν τιμές όπως φαίνονται παρακάτω τότε:

i) αν από τον πίνακα `a` επιλεγόταν τυχαία η τιμή 11, τότε η έξοδος του προγράμματος θα έπρεπε να ήταν: *Number 11 found in array b (row:2 column:3)*

ii) αν από τον πίνακα `a` επιλεγόταν τυχαία η τιμή 87, τότε η έξοδος του προγράμματος θα έπρεπε να ήταν: *Number 87 didn't find in array b*

`a = [1, 51, 11, 661, 100, 87]`

`b =`

14	41	20	94
39	30	11	12
75	90	13	18
55	85	15	82