



## Προγραμματισμός Ι – 9<sup>η</sup> Σειρά Ασκήσεων

/\* Προθεσμία υποβολής μέσω του eclass: Δευτέρα 18/12/23, 23:59:59 \*/

/\* Επίδειξη στο εργαστήριο (ανάλογα με το Group που είστε γραμμένοι) \*/

1. Να γραφεί μία συνάρτηση, η οποία θα δέχεται δύο παραμέτρους τύπου **double** και θα επιστρέφει έναν δείκτη σε **double**. Ο δείκτης αυτός θα πρέπει να δείχνει σε μία τοπική μεταβλητή (τύπου **double** προφανώς) της ίδιας της συνάρτησης (ΠΡΟΣΟΧΗ ΛΟΙΠΟΝ!!), στην οποία μεταβλητή θα αποθηκεύεται η τιμή της μεγαλύτερης εκ των δύο παραμέτρων. Σε περίπτωση που οι δύο παράμετροι είναι ίσες, ο δείκτης θα πρέπει να παίρνει την τιμή **NULL**. Στη συνέχεια, να γραφεί ένα πρόγραμμα το οποίο να διαβάζει συνεχώς δύο πραγματικούς αριθμούς και με χρήση της συνάρτησης να εμφανίζει τον μεγαλύτερο εκ των δύο ή να ενημερώνει τον χρήστη ότι οι δύο αριθμοί είναι ίσοι (όταν αυτό ισχύει). Το πρόγραμμα να σταματάει μόνο αν ο χρήστης εισάγει δύο αρνητικές τιμές προς σύγκριση.

2. 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 ακεραία τιμή. Να γραφεί ένα πρόγραμμα το οποίο με χρήση της συνάρτησης `rand()` να δημιουργεί 3 τυχαίες ακεραίες τιμές στο  $[5, 20]$ , να τις αποθηκεύει σε έναν πίνακα ακεραίων τριών θέσεων και με χρήση της συνάρτησης να εμφανίζει μήνυμα για το αν είναι ταξινομημένος ή όχι.

3. Δημιουργήστε μία συνάρτηση που να επιστρέφει την τιμή του πολωνύμου:

$$a_0 + a_1x + a_2x^2 + \dots + a_nx^n \text{ για μία δοσμένη τιμή του } x.$$

Το πρωτότυπο της συνάρτησης να είναι:

```
double poly(const double a[], double x, int n);
```

όπου στον πίνακα `a` βρίσκονται οι συντελεστές του πολωνύμου και το `n` είναι ο βαθμός του πολωνύμου. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει τον βαθμό του πολωνύμου, τους συντελεστές του, μία τιμή και να εμφανίζει την τιμή του πολωνύμου με χρήση της συνάρτησης. Θεωρήστε ότι ο μέγιστος βαθμός του πολωνύμου είναι 100. Για να βρείτε τις δυνάμεις των συντελεστών χρησιμοποιήστε τη συνάρτηση `pow()`, της βιβλιοθήκης `math.h`, όπως φαίνεται στο Παράρτημα Γ του βιβλίου.

4. 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 «παραμέτρο δείκτη» στις παραμέτρους της συνάρτησης για την τιμή επιστροφής.

5. 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 ακεραίους, 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).

6. 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 ακεραίους, να τους αποθηκεύει σε δύο διδιάστατους πίνακες  $2 \times 4$  και να εμφανίζει το άθροισμά τους με χρήση της συνάρτησης.

7. 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 ακεραίους 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.  
Υπόδειξη: Αφού ο τύπος επιστροφής της συνάρτησης είναι **void** να προσθέσετε δύο «παραμέτρους δείκτες» στις παραμέτρους της συνάρτησης για τις τιμές επιστροφής.

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

		*		* *		* * *
	*		* *		* * *	
$T_0=0$	$T_1=1$	$T_2=3$		$T_3=6$		

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.

9. Να γραφεί ένα πρόγραμμα το οποίο με χρήση της συνάρτησης `rand()` να δημιουργεί 10 ακεραίους 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 «χαλαί» τη συμμετρία.

10. 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.