



## Προγραμματισμός Ι – 6<sup>η</sup> Σειρά Ασκήσεων

/\* Προθεσμία υποβολής μέσω του eclass: Δευτέρα, 24/11/25, 23:59 \*/  
/\* Επίδειξη στο εργαστήριο (ανάλογα με το Group που είστε γραμμένοι) \*/

### ΠΡΟΣΟΧΗ ΣΤΟΥΣ ΠΕΡΙΟΡΙΣΜΟΥΣ ΚΑΙ ΣΤΙΣ ΕΚΦΩΝΗΣΕΙΣ ΚΑΘΕ ΑΣΚΗΣΗΣ

1. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει τους βαθμούς 10 φοιτητών και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, το πρόγραμμα να διαβάζει δύο πραγματικούς αριθμούς (π.χ.  $a$  και  $b$ ) και να εμφανίζει πόσοι φοιτητές πήραν βαθμό στο διάστημα  $[a, b]$ . Σημειώστε ότι ο δεύτερος εισαγόμενος αριθμός πρέπει να είναι μεγαλύτερος ή ίσος από τον πρώτο (π.χ.  $b \geq a$ ), κάνοντας έλεγχο.

2. Να γραφεί ένα πρόγραμμα το οποίο να δηλώνει δύο πίνακες 5 ακεραίων. Το πρόγραμμα να ζητά από τον χρήστη να εισάγει 5 στοιχεία στον πρώτο πίνακα και στη συνέχεια να αντιγράφει στο αντίστοιχο στοιχείο του δεύτερου πίνακα μόνο τα στοιχεία που είναι άρτιοι αριθμοί, ενώ αν είναι περιττοί να αντιγράφει τη διπλάσια τιμή του στοιχείου, αντίστοιχα.

Για παράδειγμα, αν ο πίνακας  $x$  έχει στοιχεία με τιμές: 2, 4, 5, 6 και 7, τότε τα στοιχεία του πίνακα  $y$  θα έχουν τιμές: 2, 4, 10, 6 και 14, αντίστοιχα.

3. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει συνεχώς πραγματικούς αριθμούς τύπου **double** και να αποθηκεύει σε έναν πίνακα 100 θέσεων τους αριθμούς με τιμή μεγαλύτερη από 5. Αν ο χρήστης εισάγει την τιμή -1, η εισαγωγή των αριθμών να τερματίζει και το πρόγραμμα να εμφανίζει την ελάχιστη τιμή που αποθηκεύτηκε στον πίνακα.

4. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει συνεχώς ακεραίους και να τους αποθηκεύει σε έναν πίνακα 100 θέσεων μέχρι αυτός να «γεμίσει», με τον περιορισμό ένας εισαγόμενος αριθμός να αποθηκεύεται στον πίνακα μόνο αν είναι μικρότερος από τον αριθμό που εισήχθητε τελευταίος. Στο τέλος, το πρόγραμμα να εμφανίζει όλα τα στοιχεία του πίνακα.

5. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 100 ακεραίους (θεωρήστε ότι ο χρήστης θα εισάγει θετικούς ακεραίους) και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, να εμφανίζει το μεγαλύτερο στοιχείο του πίνακα, το οποίο να διαιρείται ακριβώς από όλα τα προηγούμενα στοιχεία. Αν δεν υπάρχει τέτοιο στοιχείο, το πρόγραμμα να εμφανίζει ανάλογο μήνυμα. Το πρώτο στοιχείο του πίνακα να μην ελεγχθεί, αφού δεν υπάρχουν στοιχεία αριστερά του. Για παράδειγμα, αν θεωρήσουμε ότι τα στοιχεία του πίνακα είναι πέντε και έχουν τις τιμές 1, 2, 4, 8, 12, το πρόγραμμα πρέπει να εμφανίζει την τιμή 8, ενώ αν τα στοιχεία έχουν τις τιμές 3, 5, 10, 15, 20, το πρόγραμμα πρέπει να εμφανίζει το μήνυμα ότι δεν υπάρχει κανένα τέτοιο στοιχείο.

6. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 100 ακεραίους και να τους αποθηκεύει σε έναν πίνακα. Το πρόγραμμα να εμφανίζει πόσα στοιχεία του πίνακα έχουν μεγαλύτερη τιμή από την τιμή του τελευταίου στοιχείου και πόσα στοιχεία έχουν μεγαλύτερη τιμή από τον μέσο όρο των στοιχείων.

Υπόδειξη: Να εισάγετε πρώτα την τιμή του τελευταίου στοιχείου του πίνακα και στη συνέχεια των υπολοίπων (από την αρχή του πίνακα έως και το προ-τελευταίο στοιχείο).

7. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 100 ακεραίους και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, το πρόγραμμα να περιστρέφει τα στοιχεία του πίνακα μία θέση δεξιά. Για παράδειγμα, αν ο πίνακας ήταν 4 ακεραίων και τα στοιχεία του ήταν τα εξής: 1, -9, 5, 3 μετά την περιστροφή τα στοιχεία του θα πρέπει να είναι τα εξής: 3, 1, -9, 5.

**Προσοχή!** Δεν αρκεί να «εκτυπώσετε το αποτέλεσμα», δηλαδή να εμφανίζετε στην οθόνη χρησιμοποιώντας μία `printf()` πρώτα το τελευταίο στοιχείο του πίνακα και μετά όλα τα υπόλοιπα, αλλά **θα πρέπει να αλλάζουν οι τιμές των στοιχείων μέσα στον πίνακα** (το στοιχείο της τελευταίας θέσης να «μεταφερθεί» στην πρώτη θέση, το στοιχείο της πρώτης θέσης να «μεταφερθεί» στη δεύτερη, κ.ο.κ.).

**Σημείωση:** Απαγορεύεται να χρησιμοποιήσετε δεύτερο (βοηθητικό) πίνακα.

8. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 1821 ακεραίους, να τους αποθηκεύει σε έναν πίνακα και να εμφανίζει το συνολικό πλήθος αντιγράφων των ακεραίων που αποθηκεύτηκαν στον πίνακα. Για παράδειγμα, αν ο πίνακας ήταν 5 ακεραίων και τα στοιχεία του είναι {6, 6, 6, 6, 6}, το πρόγραμμα να εμφανίζει 4 (αφού ο αριθμός 6 επαναλαμβάνεται τέσσερις φορές), αν είναι {1, -3, 1, 50, -3} το πρόγραμμα να εμφανίζει 2 (αφού οι αριθμοί 1 και -3 επαναλαμβάνονται από μία φορά ο καθένας) ενώ αν είναι {3, -1, 22, 13, -7} το πρόγραμμα να εμφανίζει 0 (αφού δεν υπάρχει στοιχείο να επαναλαμβάνεται).

**Σημείωση:** Απαγορεύεται να χρησιμοποιήσετε δεύτερο (βοηθητικό) πίνακα.

9. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει τους βαθμούς 1000 φοιτητών και να εμφανίζει τη συχνότητα εμφάνισης του κάθε βαθμού, καθώς και ποιος βαθμός εμφανίστηκε τις περισσότερες φορές και πόσες είναι αυτές. Ελέγξτε και την περίπτωση που να είναι περισσότεροι από ένας οι βαθμοί με τον ίδιο μέγιστο αριθμό εμφανίσεων. Τέλος, θεωρήστε ότι οι βαθμοί είναι ακέραιοι στο διάστημα  $[0, 10]$  και το πρόγραμμα να κάνει έλεγχο εγκυρότητας (να μην δέχεται βαθμούς εκτός του διαστήματος  $[0, 10]$ ).

10. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει ακεραίους και να τους αποθηκεύει σε έναν τετραγωνικό πίνακα (π.χ.  $4 \times 4$ ). Στη συνέχεια, να ελέγχει αν ο πίνακας είναι άνω τριγωνικός, κάτω τριγωνικός ή και τα δύο και να εκτυπώνει κατάλληλο μήνυμα στην οθόνη. Ένας τετραγωνικός πίνακας είναι «άνω τριγωνικός» όταν όλα τα στοιχεία του κάτω από την κύρια διαγώνιο είναι ίσα με μηδέν ενώ είναι «κάτω τριγωνικός» όταν όλα τα στοιχεία του πάνω από την κύρια διαγώνιο είναι ίσα με μηδέν, αντίστοιχα.

**Σημείωση:** Η λύση σας να είναι «γενική» και να μπορεί να εφαρμοστεί ακόμα και αν ο αρχικός πίνακας ήταν  $N \times N$ , και όχι αναγκαστικά  $4 \times 4$ .

11. Με χρήση μόνο ενός διπλού **for** βρόχου να δημιουργήσετε έναν τετραγωνικό πίνακα  $5 \times 5$  στον οποίο τα στοιχεία της κύριας και της δευτερεύουσας διαγώνιου να είναι όλα ίσα με 1, στο «πάνω τρίγωνο» ίσα με 2, στο «δεξί τρίγωνο» ίσα με 3, στο «κάτω τρίγωνο» ίσα με 4 και στο «αριστερό τρίγωνο» ίσα με 5, όπως φαίνεται και στο παρακάτω σχήμα. Στη συνέχεια με χρήση ενός ακόμη διπλού **for** βρόχου να εμφανίσετε την αλγεβρική μορφή του  $5 \times 5$  πίνακα στην οθόνη, δηλαδή όπως φαίνεται στο σχήμα.

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 1 |
| 5 | 1 | 2 | 1 | 3 |
| 5 | 5 | 1 | 3 | 3 |
| 5 | 1 | 4 | 1 | 3 |
| 1 | 4 | 4 | 4 | 1 |

12. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει ακεραίους και να τους αποθηκεύει σε έναν πίνακα  $3 \times 4$ . Το πρόγραμμα να βρίσκει το μέγιστο στοιχείο κάθε γραμμής, να θέτει όλα τα στοιχεία πριν από αυτό (στην ίδια γραμμή) ίσα με αυτό και να εμφανίζει τον νέο πίνακα την οθόνη.

Για παράδειγμα, αν ο πίνακας ήταν ο:

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

να μετατρέπεται στον:

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

**Σημείωση:** Απαγορεύεται να χρησιμοποιήσετε δεύτερο (βοηθητικό) πίνακα.

**13.** Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 900 ακεραίους και να τους αποθηκεύει σε έναν τετραγωνικό πίνακα  $30 \times 30$ . Στη συνέχεια, να δημιουργεί τον ανάστροφό του, δηλαδή να δημιουργεί έναν πίνακα όπου οι γραμμές να γίνουν στήλες και αντίστροφα και να τον εμφανίζει.

Για παράδειγμα, αν είχαμε διαβάσει 9 ακεραίους και τους είχαμε αποθηκεύσει σε έναν τετραγωνικό

πίνακα  $3 \times 3$  και ήταν ο:

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

τότε ο ανάστροφός του είναι ο:

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

**Σημείωση:** Απαγορεύεται να χρησιμοποιήσετε δεύτερο (βοηθητικό) πίνακα. Δηλαδή, στο τέλος του προγράμματος, ο ανάστροφος πίνακας θα έχει αποθηκευτεί στον αρχικό σας πίνακα.

**14.** Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 25 ακεραίους και να τους αποθηκεύει σε έναν τετραγωνικό  $5 \times 5$  πίνακα. Στη συνέχεια, να διαβάζει έναν ακέραιο στο  $[1, 5]$  (π.χ.  $x$ ) και να αποθηκεύει σε έναν δεύτερο πίνακα τα στοιχεία του πρώτου, εκτός από τα στοιχεία της  $x$ -γραμμής και της  $x$ -στήλης (προφανώς, ο δεύτερος πίνακας θα είναι διάστασης  $4 \times 4$ ).

Π.χ. Αν ο αρχικός πίνακας ήταν ο:

|    |    |    |    |    |
|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  |
| 6  | 7  | 8  | 9  | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |

και ο χρήστης πληκτρολογήσει  $x=3$  τότε ο

δεύτερος πίνακας θα πρέπει να είναι ο:

|    |    |    |    |
|----|----|----|----|
| 1  | 2  | 4  | 5  |
| 6  | 7  | 9  | 10 |
| 16 | 17 | 19 | 20 |
| 21 | 22 | 24 | 25 |

(λείπει η 3<sup>η</sup> γραμμή και η 3<sup>η</sup> στήλη από τον αρχικό).

**Σημείωση:** Η λύση σας να είναι «γενική» και να μπορεί να εφαρμοστεί ακόμα και αν ο αρχικός πίνακας ήταν  $N \times N$ , και όχι αναγκαστικά  $5 \times 5$ . Επιτρέπεται να χρησιμοποιήσετε έως 2 τετραγωνικούς πίνακες συνολικά στο πρόγραμμά σας (έναν  $5 \times 5$  κι έναν  $4 \times 4$ ).