



## 3η Σειρά Ασκήσεων - Δείκτες και Πίνακες

### Άσκηση 3.1

Να γράψετε μια συνάρτηση με όνομα `out_of_order` που θα δέχεται 2 παραμέτρους:

- ένα πίνακα με στοιχεία τύπου `float`
- το μέγεθος του πίνακα τύπου `int`

Η συνάρτηση θα ελέγχει αν ο πίνακας είναι ταξινομημένος σε αύξουσα διάταξη (δηλαδή  $a[0] \leq a[1] \leq a[2] \leq \dots$ ) και θα επιστρέφει:

- την τιμή `-1` αν όλα τα στοιχεία είναι ταξινομημένα
- τη θέση του πρώτου στοιχείου που είναι εκτός ταξινομημένης διάταξης.

Στη συνέχεια να γράψετε πρόγραμμα που να καλεί τη συνάρτηση `out_of_order` και να εμφανίζει το αποτέλεσμα.

### Άσκηση 3.2

Να γραφεί συνάρτηση που θα βρίσκει και θα επιστρέφει τα συμπληρωματικά στοιχεία ενός μονοδιάστατου πίνακα `n` θέσεων. Συμπληρωματικά είναι εκείνα τα στοιχεία του πίνακα που βρίσκονται στις θέσεις `0` και `n-1`, `1` και `n-2`, `2` και `n-3` κλπ. και το άθροισμά τους είναι ίσο του `99`.

Στη συνέχεια να γραφεί πρόγραμμα που θα γεμίζει έναν μονοδιάστατο πίνακα `100` θέσεων με ακέραιους θετικούς αριθμούς από `0-99` (θα γίνεται έλεγχος κατά την εισαγωγή, εφόσον χρησιμοποιηθεί η `scanf`), θα καλεί τη συνάρτηση και θα εμφανίζει τα αποτελέσματα.

### Άσκηση 3.3

Να γραφεί μια συνάρτηση με όνομα `addinew()` που θα δέχεται ως παραμέτρους 2 μονοδιάστατους πίνακες του ίδιου μεγέθους. Η συνάρτηση θα προσθέτει τα αντίστοιχα στοιχεία των δύο πινάκων και θα τα τοποθετεί σε τρίτο (νέο) πίνακα της ίδιας διάστασης.

Στη συνέχεια να γραφεί η συνάρτηση `main()` όπου:

- Θα καταχωρούνται τα στοιχεία σε 2 μονοδιάστατους πίνακες ακεραίων αριθμών `max` (`max=γνωστό`) σε πλήθος θέσεων, με χρήση τυχαίων αριθμών στο διάστημα `1-50`.
- Θα καλείται η συνάρτηση `addinew()`

### Άσκηση 3.4

Να γραφεί μια συνάρτηση με όνομα `final` και ορίσματα :

- ένα μονοδιάστατο πίνακα ακεραίων αριθμών `n` θέσεων
- έναν ακέραιο αριθμό `k`

Η συνάρτηση θα δημιουργεί έναν νέο πίνακα που θα περιέχει όλα τα στοιχεία του αρχικού πίνακα των οποίων το άθροισμα, ξεκινώντας από την αρχή του πίνακα, δεν υπερβαίνει τον αριθμό  $k$  π.χ. αν ο πίνακας περιέχει τα στοιχεία :

11	7	3	17	9	4	20	15
----	---	---	----	---	---	----	----

και  $k = 50$ ,

Τότε ο νέος πίνακας θα περιέχει τα στοιχεία:

11	7	3	17	9
----	---	---	----	---

(είναι  $11+7+3+17+9+4=51 > 50$  και το τελευταίο στοιχείο που ελέγχεται είναι το 4 αλλά δεν περιλαμβάνεται στο νέο πίνακα)

Στη συνέχεια να γραφεί ένα πρόγραμμα που :

- θα δημιουργεί ένα πίνακα θετικών ακεραίων τυχαίων διψήφιων αριθμών μεγέθους `max` (`max` = γνωστό)
- θα καλεί τη συνάρτηση `final` με παραμέτρους τον πίνακα των τυχαίων αριθμών μεγέθους `max` και μια τιμή για την παράμετρο `k`
- θα εμφανίζει τα αποτελέσματα που θα επιστρέψει η συνάρτηση

### Άσκηση 3.5

Να γραφεί μια συνάρτηση με όνομα `disparate` και ορίσματα :

- ένα μονοδιάστατο πίνακα ακεραίων αριθμών  $n$  θέσεων

Η συνάρτηση :

Θα δημιουργεί ένα νέο πίνακα που θα περιλαμβάνει όλα στοιχεία του αρχικού πίνακα αλλά μόνον μία φορά το καθένα, δηλ. δεν θα περιέχει 2η φορά το ίδιο στοιχείο. Π.χ. αν ο αρχικός πίνακας περιέχει τα στοιχεία `2, 4, 5, 9, 4, 3, 5, 2` ο νέος πίνακας θα περιέχει μόνο τα `2, 4, 5, 9, 3`.

(ΥΠΟΔΕΙΞΗ : κάθε φορά που ένα στοιχείο πρόκειται να εισαχθεί στον νέο πίνακα να ελέγχετε αν υπάρχει ήδη σε αυτόν).

Στη συνέχεια να γραφεί ένα πρόγραμμα που :

- Θα δημιουργεί ένα πίνακα ακεραίων μονοψήφιων τυχαίων αριθμών μεγέθους `max` (`max` = γνωστό)
- Θα καλεί τη συνάρτηση `disparate` με παράμετρο εισόδου τον πίνακα των τυχαίων αριθμών
- Θα εμφανίζει τα αποτελέσματα που θα επιστρέψει η συνάρτηση.

### Άσκηση 3.6

Να γραφεί μια συνάρτηση με όνομα `overweight` και ορίσματα :

- ένα μονοδιάστατο πίνακα ακεραίων αριθμών  $n$  θέσεων

- Έναν ακέραιο αριθμό  $m$

Σε κάθε θέση του πίνακα, αντιστοιχεί ένα βάρος που είναι αντιστρόφως ανάλογο με τον αριθμό της θέσης στον πίνακα. Δηλαδή η 1η θέση έχει βάρος  $n$ , η 2η θέση βάρος  $n-1$  ... και η  $n$ -οστή θέση βάρος 1.

Για παράδειγμα αν  $n=7$  ο πίνακας θα έχει την παρακάτω δομή:

θέση	1η	2η	3η	4η	5η	6η	7η
βάρος θέσης	7	6	5	4	3	2	1
στοιχείο	5	3	7	6	1	4	9

Η συνάρτηση θα υπολογίζει για κάθε θέση την τιμή: στοιχείο \* βάρος θέσης και θα επιστρέφει:

- Τη θέση του πίνακα, έστω  $k$ , για την οποία το άθροισμα:  
 $\text{στοιχείο}_1 * \text{βάρος θέσης}_1 + \text{στοιχείο}_2 * \text{βάρος θέσης}_2 + \dots + \text{στοιχείο}_k * \text{βάρος θέσης}_k$   
 γίνεται μεγαλύτερο της τιμής  $m$ .
- την τιμή -1 αν δεν βρεθεί τέτοια θέση στον πίνακα

π.χ για το παραπάνω παράδειγμα, και με τιμή  $m=100$

το άθροισμα:  $5 * 7 + 3 * 6 + 7 * 5 + 6 * 4 = 102$  υπερβαίνει την τιμή  $m=100$  στην 4η θέση, άρα η συνάρτηση θα επιστρέψει την τιμή 4.

Στη συνέχεια να γραφεί πρόγραμμα που να καλεί τη συνάρτηση και να εμφανίζει τα αποτελέσματα.

### Άσκηση 3.7

Μια μονάδα παράγει ισομεγέθεις μεταλλικές ράβδους μήκους  $d$  και μετρά τις αποκλίσεις των παραγομένων ράβδων από την επιθυμητή διάσταση  $d$ . Θεωρείστε μια σειρά ακεραίων αριθμών, πλήθους  $N$  ( $N = \text{γνωστό}$ ) που αντιστοιχεί στις αποκλίσεις που μετρήθηκαν σε ένα δείγμα  $N$  ράβδων.

π.χ. για  $N = 20$ , η σειρά μπορεί να είναι:

2, 6, 7, 0, 4, 0, 1, 3, 8, 0, 0, 2, 6, 9, 4, 5, 0, 1, 1, 3

Να γράψετε μια συνάρτηση με όνομα `bar_count` που θα δέχεται ως είσοδο ένα μονοδιάστατο πίνακα  $N$  θέσεων που αντιστοιχεί στις αποκλίσεις  $N$  ράβδων από την επιθυμητή διάσταση και θα επιστρέφει στη `main()`, υπό μορφή πίνακα, το πλήθος των ράβδων που μεσολαβούν ανάμεσα στις ράβδους που έχουν μηδενικές αποκλίσεις. Η συνάρτηση θα καλείται από τη `main()`.

(Δηλαδή για το παράδειγμα θα επιστρέφονται οι τιμές: 3, 1, 3, 0, 5, 3).

### Άσκηση 3.8

Οι  $N$  φάσεις κατεργασίας που εκτελούνται στις εργαλειομηχανές μιας παραγωγικής μονάδας για την παραγωγή ενός προϊόντος ( $N = \text{γνωστό}$ ) εφοδιάζονται με έναν αριθμό ταυτοποίησης ( $id$ , ακέραιος αριθμός από  $1 - 99$ ) και την κατάσταση ( $status$ ) με τις εξής τιμές:

- 0: φάση κατεργασίας σε παύση εκτέλεσης,
- 1: φάση κατεργασίας σε εξέλιξη,
- 2: η φάση κατεργασίας απέτυχε,
- 3: η φάση κατεργασίας ολοκληρώθηκε με επιτυχία.

Ένας πίνακας  $a[2*N]$  περιέχει ζεύγη τιμών ( $id$ ,  $status$ ).

Να γραφεί μια συνάρτηση με όνομα `find_status` και ορίσματα εισόδου:

- α) ένα μονοδιάστατο πίνακα ακεραίων θετικών αριθμών  $2*N$  θέσεων,  $a[2*N]$
- β) έναν ακέραιο αριθμό, έστω  $flag$ , με επιτρεπτές τιμές 0, 1, 2 ή 3.

Η συνάρτηση `find_status` θα επιστρέφει στη συνάρτηση `main()` έναν νέο πίνακα, που θα περιλαμβάνει μόνο εκείνους τους αριθμούς ταυτοποίησης (δηλ. τις φάσεις κατεργασίας  $id$ ) των οποίων το  $status$  είναι ίσο με το  $flag$ , σε αύξουσα σειρά του  $id$ , ΧΩΡΙΣ να χρησιμοποιηθεί διαδικασία ταξινόμησης.

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

Στη συνέχεια να γραφεί ένα πρόγραμμα που :

1. θα γεμίζει τον πίνακα  $a$ , με χρήση της συνάρτησης δημιουργίας τυχαίων αριθμών `rand()`. Στην περίπτωση που μια φάση κατεργασίας υπάρχει περισσότερες από μία φορές δεν δημιουργείται πρόβλημα (η φάση αυτή εκτελείται περισσότερες από μία φορές).
2. θα καλεί διαδοχικά τη συνάρτηση `find_status`, για όλες τις πιθανές τιμές της  $flag$ , και θα εμφανίζει τα αποτελέσματα που θα επιστρέφει η συνάρτηση.

ΠΑΡΑΔΕΙΓΜΑ με  $N=13$

Πίνακας με αριθμό φάσης κατεργασίας και κατάσταση, ανά ζεύγη (δηλ. 13 ζεύγη):

7	3	95	1	14	0	27	1	74	2	88	0	55	3	19	1	4	0	90	2	22	0	35	1	16	3
---	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	---	---	----	---	----	---	----	---	----	---

Επιστρεφόμενοι πίνακες στη συνάρτηση `main()`:

status	Φάσεις κατεργασίας				
0	<table border="1"><tr><td>4</td><td>14</td><td>22</td><td>88</td></tr></table>	4	14	22	88
4	14	22	88		
1	<table border="1"><tr><td>19</td><td>27</td><td>35</td><td>95</td></tr></table>	19	27	35	95
19	27	35	95		
2	<table border="1"><tr><td>74</td><td>90</td></tr></table>	74	90		
74	90				
3	<table border="1"><tr><td>7</td><td>16</td><td>55</td></tr></table>	7	16	55	
7	16	55			

### Άσκηση 3.9

Να γραφεί συνάρτηση με όνομα `sorting`, που να δέχεται ως ορίσματα τα παρακάτω:

- ένα μονοδιάστατο πίνακα ακεραίων αριθμών  $n$  θέσεων, (θεωρούμε το  $n$  γνωστό)
- μια ακέραια μεταβλητή  $m$  που μπορεί να λάβει τις τιμές  $0$  ή  $1$ .

Στην περίπτωση που  $m=0$  η συνάρτηση θέλουμε να ταξινομή τον πίνακα κατά αύξουσα σειρά, ενώ στην περίπτωση που  $m=1$  κατά φθίνουσα, εφαρμόζοντας τον αλγόριθμο της φυσαλίδας.

Στη συνέχεια, να γραφεί πρόγραμμα που:

1. θα γεμίζει τον πίνακα με τυχαίους ακέραιους αριθμούς στο διάστημα  $[0-22]$ ,
2. θα ταξινομή τον πίνακα κατά αύξουσα σειρά,
3. θα εμφανίζει τον ταξινομημένο πίνακα,
4. θα εμφανίζει τη μεγαλύτερη ακολουθία διαδοχικών τιμών του πίνακα (σε περίπτωση που υπάρχουν πολλές με ίδιο μέγιστο πλήθος, να εμφανίζει την 1η που θα εντοπίσει).

**Σημείωση:** για το ερώτημα 4, προτείνεται να χρησιμοποιήσετε κατάλληλη συνάρτηση που θα δέχεται ως όρισμα τον ταξινομημένο πίνακα και θα επιστρέφει 2 ακέραιες τιμές με τον αρχικό και τελικό δείκτη της μεγαλύτερης διαδοχικής ακολουθίας. Η συνάρτηση ΔΕΝ θα πρέπει να περιέχει εντολές `printf`.

Παράδειγμα:

Αν ο ταξινομημένος πίνακας περιέχει τις τιμές:

7	8	9	11	13	14	16	18	19	20	21	22
---	---	---	----	----	----	----	----	----	----	----	----

Η μεγαλύτερη διαδοχική ακολουθία τιμών του πίνακα θα είναι:

18	19	20	21	22
----	----	----	----	----

### Άσκηση 3.10

Δύο μονοδιάστατοι αριθμητικοί πίνακες θετικών ακεραίων αριθμών, έστω  $a$  και  $b$ , με πλήθος στοιχείων  $m$ ,  $n$  αντίστοιχα, περιέχουν θετικούς ακέραιους αριθμούς στο διάστημα  $[0, 99]$ .

Να θεωρήσετε ότι οι δύο πίνακες περιέχουν κοινά στοιχεία.

Να γραφεί μια συνάρτηση με όνομα `check_chain` με ορίσματα εισόδου τους πίνακες  $a$  και  $b$ .

Η συνάρτηση `check_chain` θα επιστρέφει στη συνάρτηση `main()` :

- Έναν νέο πίνακα που θα περιέχει σε αύξουσα διάταξη, ΧΩΡΙΣ να χρησιμοποιηθεί διαδικασία ταξινόμησης, όλα τα στοιχεία των δύο πινάκων  $a$  και  $b$ , υπό την προϋπόθεση ότι κάθε στοιχείο στον νέο πίνακα θα εμφανίζεται ΜΟΝΟ ΜΙΑ ΦΟΡΑ.
- Τη μεγαλύτερη ακολουθία διαδοχικών τιμών στον νέο πίνακα. Η ακολουθία θα δημιουργεί έναν κατάλληλο αριθμητικό πίνακα. Να θεωρήσετε ότι υπάρχει μόνον μία τέτοια ακολουθία (δηλ. δεν υπάρχει δεύτερη ακολουθία με το ίδιο μήκος).

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

Στη συνέχεια να γραφεί ένα πρόγραμμα που :

1. Θα γεμίζει τους πίνακες `a` και `b`, με κατάλληλη χρήση της συνάρτησης δημιουργίας τυχαίων αριθμών `rand()` ώστε οι τιμές να ανήκουν στο διάστημα `[0, 99]`.
2. Θα καλεί τη συνάρτηση `check_chain` και θα εμφανίζει τα σχετικά αποτελέσματα.

Παράδειγμα με `m=8` και `n=6`:

Πίνακας `a`:

14	13	9	8	7	21	20	19	9
----	----	---	---	---	----	----	----	---

Πίνακας `b`:

7	8	22	11	18	16
---	---	----	----	----	----

Ο νέος πίνακας στη `main()`:

7	8	9	11	13	14	16	18	19	20	21	22
---	---	---	----	----	----	----	----	----	----	----	----

Η μεγαλύτερη διαδοχική ακολουθία τιμών του πίνακα θα είναι:

18	19	20	21	22
----	----	----	----	----