



## 5η Σειρά Ασκήσεων - Πολυδιάστατοι Πίνακες

### Άσκηση 5.1

Να γράψετε κατάλληλες εντολές σε γλώσσα C, για να δείξετε ότι για τον πίνακα `int a[3][5]`, οι ακόλουθες 4 εκφράσεις, είναι ισοδύναμες με το `a[i][j]`:

1. `*(a[i] + j)`
2. `*(a + i)[j]`
3. `*((a + i) + j)`
4. `*(&a[0][0] + 5*i + j)`

### Άσκηση 5.2

Να γραφούν δύο συναρτήσεις σε γλώσσα C που να επιστρέφουν:

- Τον όγκο ενός κυλίνδρου με ακτίνα  $R$  και ύψος  $H$  ( $V = \pi R^2 H$ )
- Τον όγκο ενός ορθογωνίου ύψους  $H$  με βάση τετράγωνο πλευράς  $A$  ( $V = A^2 H$ )

Η συνάρτηση `main()` θέλουμε να :

1. γεμίζει ένα πίνακα `X[N][2]` με τυχαίους αριθμούς τύπου `double` με εύρος τιμών `[1.0-999.0]`. (Η τιμή του  $N$  ορίζεται ως σταθερά του προγράμματος).
2. καλεί για κάθε σειρά του πίνακα και τις δύο συναρτήσεις και θα εμφανίζει τις αντίστοιχες τιμές των συναρτήσεων (Η πρώτη στήλη του πίνακα αντιστοιχεί στα  $R$  ή  $A$  και η δεύτερη στήλη στο ύψος  $H$ ).

### Άσκηση 5.3

Να δημιουργήσετε μία συνάρτηση `displayMatrix` για την εμφάνιση ενός διδιάστατου πίνακα ακεραίων αριθμών.

Η συνάρτηση `main()` θέλουμε να :

1. γεμίζει ένα πίνακα `A[20][2]` με τυχαίους αριθμούς τύπου `int` με εύρος τιμών `[0-99]`.
2. δημιουργεί και να εμφανίζει ένας νέος πίνακας `B` που να περιέχει εκείνες τις γραμμές του πίνακα `A`, στις οποίες η διαφορά των τιμών στις 2 στήλες της κάθε γραμμής, είναι κατά μέγιστο 10.

### Άσκηση 5.4

Να γραφεί μια συνάρτηση `find_maxmin` που θα βρίσκει το μέγιστο και το ελάχιστο στοιχείο κάθε γραμμής ενός διδιάστατου πίνακα.

Η συνάρτηση `main()` θέλουμε να :

1. γεμίζει ένα πίνακα `A[10][5]` με τυχαίους αριθμούς τύπου `int` με εύρος τιμών `[10-99]`.
2. καλεί την `find_maxmin` και να εμφανίζει τα μέγιστα και ελάχιστα κάθε γραμμής μαζί με

τον πίνακα σε κατάλληλη θέση, όπως φαίνεται στο παρακάτω παράδειγμα:

19	45	77	43	34	77	19
55	79	65	27	45	79	27
27	13	16	37	88	88	13

### Άσκηση 5.5

Να γραφεί μια συνάρτηση με όνομα `addMatrix` που να προσθέτει όλα τα στοιχεία ενός πίνακα ακεραίων αριθμών, 2 διαστάσεων (γραμμές και στήλες).

Επίσης, να γραφεί μια συνάρτηση με όνομα `displayMatrix` που να εμφανίζει ένα πίνακα ακεραίων αριθμών, 2 διαστάσεων (γραμμές και στήλες).

Η συνάρτηση `main()` θέλουμε να :

1. γεμίζει έναν δισδιάστατο πίνακα ακεραίων με τυχαίους θετικούς αριθμούς τύπου `int` στο διάστημα `[0, 99]`.
2. καλεί τη συνάρτηση `displayMatrix`,
3. καλεί τη συνάρτηση `addMatrix` και να εμφανίζει το αποτέλεσμα που επιστρέφεται.

(ΥΠΟΔΕΙΞΗ: Οι διαστάσεις του πίνακα να ορίζονται κατά την εκτέλεση του προγράμματος. Ο πίνακας δεδομένων να δημιουργείται με χρήση της `malloc()`).

### Άσκηση 5.6

Να γραφεί μια συνάρτηση με όνομα `convert_to_1d` η οποία θα μετασχηματίζει ένα δισδιάστατο πίνακα ακεραίων θετικών αριθμών σε έναν μονοδιάστατο πίνακα. Η συνάρτηση θα επιστρέφει:

- έναν μονοδιάστατο πίνακα που θα περιλαμβάνει τα στοιχεία του αρχικού πίνακα δύο διαστάσεων, γραμμή προς γραμμή.
- έναν μονοδιάστατο πίνακα που θα περιλαμβάνει τα στοιχεία του αρχικού πίνακα δύο διαστάσεων, στήλη προς στήλη.

Στη συνάρτηση `main()`:

1. Να δημιουργηθεί ο δισδιάστατος πίνακας μέσω δυναμικής παραχώρησης μνήμης και μέσω προσδιορισμού των διαστάσεων του κατά τη διάρκεια εκτέλεσης του προγράμματος. Οι τιμές του πίνακα είναι τυχαίες ακέραιες στην περιοχή `[10, 99]`.
2. Να κληθεί η συνάρτηση `convert_to_1d`.
3. Να χρησιμοποιήσετε τις κατάλληλες συναρτήσεις `displayMatrix` και `display_array` για την εμφάνιση δεδομένων και αποτελεσμάτων.

### Άσκηση 5.7

Να γραφεί μία συνάρτηση με όνομα `fill_2d_array` η οποία θα γεμίζει έναν δισδιάστατο τετραγωνικό πίνακα ακεραίων αριθμών, διαστάσεων  $N \times N$ , τύπου `short int`, ως εξής:

- Όλα τα στοιχεία της κύριας διαγωνίου θα είναι ίσα με `0`.
- Ο κάτω και αριστερά τριγωνικός πίνακας θα έχει όλα τα στοιχεία του ίσα με `-1`.
- Ο επάνω και δεξιά άνω τριγωνικός πίνακας θα έχει όλα τα στοιχεία του ίσα με `1`.

Στη συνάρτηση `main()`:

1. Να δημιουργηθεί ο δισδιάστατος πίνακας μέσω δυναμικής παραχώρησης μνήμης και μέσω προσδιορισμού των διαστάσεων του κατά τη διάρκεια εκτέλεσης του προγράμματος.
2. Να κληθεί η συνάρτηση `fill_2d_array`.
3. Να χρησιμοποιηθεί η συνάρτηση `displayMatrix`, για την εμφάνιση του πίνακα.

## Άσκηση 5.8

Να γραφεί μία συνάρτηση με όνομα `row_column_average` η οποία θα υπολογίζει τη μέση τιμή κάθε γραμμής και κάθε στήλης ενός πίνακα δύο διαστάσεων  $N \times M$ . Η συνάρτηση θα επιστρέφει:

- Έναν μονοδιάστατο πίνακα που θα περιέχει τη μέση τιμή κάθε γραμμής του πίνακα
- Έναν μονοδιάστατο πίνακα που θα περιέχει τη μέση τιμή κάθε στήλης του πίνακα

Στη συνάρτηση `main()`:

1. Να δημιουργηθεί ο δισδιάστατος πίνακας : μέσω προσδιορισμού των διαστάσεων του κατά τη διάρκεια εκτέλεσης του προγράμματος και στη συνέχεια μέσω δυναμικής παραχώρησης μνήμης.
2. Να κληθεί η συνάρτηση `row_column_average`.
3. Να χρησιμοποιηθούν οι συναρτήσεις `displayMatrix` και `display_array` για την εμφάνιση δεδομένων και αποτελεσμάτων.

## Άσκηση 5.9

Να γραφεί μια συνάρτηση με όνομα `is_magic` που θα ελέγχει εάν ένας τετραγωνικός πίνακας  $N \times N$  με στοιχεία τους ακεραίους θετικούς αριθμούς  $1, 2, 3, \dots, N^2$  είναι «μαγικός». Εάν δηλαδή το άθροισμα των στηλών, το άθροισμα των γραμμών, το άθροισμα των στοιχείων της κυρίας διαγωνίου και το άθροισμα των στοιχείων της δευτερεύουσας διαγωνίου είναι ίσα μεταξύ τους.

Η συνάρτηση πρέπει καταρχήν να καλεί μια άλλη συνάρτηση `test_numbers` που θα ελέγχει εάν υπάρχουν όλες οι τιμές  $1, 2, 3, \dots, N^2$ , από μία μόνο φορά, στον τετραγωνικό πίνακα  $N \times N$ .

Παράδειγμα:

6	1	8
7	5	3
2	9	4

A)  $6+1+8=7+5+3=2+9+4=15$

B)  $6+7+2=1+5+9=8+3+4=15$

Γ)  $6+5+4=15$

Δ)  $2+5+8=15$

Η συνάρτηση `main()`:

1. Θα δέχεται ως είσοδο μία τιμή για το  $N$ , κατά τη διάρκεια εκτέλεσης του προγράμματος (πρέπει να ισχύει  $2 < N < 20$ ),

2. Θα καλεί διαδοχικά τις συναρτήσεις `test_numbers` και `is_magic`
3. Θα εμφανίζει τον πίνακα, μέσω της συνάρτησης `displayMatrix`, με το κατάλληλο μήνυμα εφόσον ο πίνακας είναι «μαγικός».

### Άσκηση 5.10

Ένας πίνακας  $M$  με  $i$  γραμμές και  $j$  στήλες μπορεί να αναστραφεί σε έναν πίνακα  $N$  (transposed matrix) με  $j$  γραμμές και  $i$  στήλες θέτοντας την τιμή  $N_{a,b}$  ίση με την τιμή  $M_{b,a}$  για όλες τις σχετικές τιμές των  $a$  και  $b$ .

Να γράψετε μία συνάρτηση με όνομα `transposeMatrix` που να αντιμεταθέτει τον πίνακα  $4 \times 5$  στον πίνακα  $5 \times 4$ . Στη συνέχεια να γράψετε το κατάλληλο πρόγραμμα (συνάρτηση `main()`) για να ελέγξετε τη συνάρτηση, εμφανίζοντας τους δύο πίνακες μέσω της συνάρτησης `displayMatrix`.