



Πανεπιστήμιο Δυτικής Μακεδονίας  
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

---

# Εισαγωγή στον δομημένο προγραμματισμό

Ενότητα 5<sup>η</sup>: Πίνακες

Αν. καθηγητής Στεργίου Κώστας

e-mail: [kstergiou@uowm.gr](mailto:kstergiou@uowm.gr)

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

---



Πανεπιστήμιο Δυτικής Μακεδονίας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# Άδειες Χρήσης

---

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Στόχοι της διάλεξης

---

- Περιγραφή της έννοιας του πίνακα ως μέσο χειρισμού μεγάλου πλήθους ομοειδών αντικειμένων στη C.
- Εξοικείωση με τη χρήση πινάκων και τα πιο συχνά λάθη της.
- Εισαγωγή σε πιο προχωρημένες έννοιες πινάκων όπως οι πίνακες πολλών διαστάσεων.



# Περιεχόμενα

---

- Πίνακες στη C.
  - Δήλωση.
  - Αρχικοποίηση.
  - Διαχείριση στοιχείων πίνακα.
- Παραδείγματα.
- Μέγεθος πίνακα και συχνά λάθη.
- Πολυδιάστατοι πίνακες.
  - Παραδείγματα.



# Πίνακες (arrays)

---

- Πολύ συχνά είναι απαραίτητο το πρόγραμμα μας να χειριστεί πολλά αντικείμενα του ίδιου τύπου.
  - Και για να τα χειριστεί πρέπει να τα αποθηκεύσει στη μνήμη.
  - Π.χ.
    - τα στοιχεία και του βαθμούς όλων των φοιτητών.
    - τις τιμές όλων των μοντέλων αυτοκινήτων.
    - τις θερμοκρασίες κάθε ημέρας του χρόνου.
    - τους μισθούς όλων των υπαλλήλων μιας εταιρίας.
    - κτλ.
- Για τον αποδοτικό χειρισμό τέτοιων απαιτήσεων, η C διαθέτει **πίνακες** .



# Τύπος Πίνακα (array)

---

- Είναι ένας σύνθετος τύπος δεδομένων.
- Αναπαριστά ένα σύνολο ομοειδών τιμών.
  - π.χ. βαθμοί μαθητών μιας τάξης, θερμοκρασίες ενός μήνα, κτλ.
- Τα στοιχεία ενός πίνακα αποθηκεύονται σε διαδοχικές θέσεις μνήμης.
  - Έχουν το ίδιο όνομα και τον ίδιο τύπο.
- Χαρακτηριστικά:
  - **Μέγεθος πίνακα.**
  - **Τύπος στοιχείων.**



# Δήλωση Πίνακα

<τύπος> <όνομα πίνακα>  
[<μέγεθος πίνακα>];

```
float temp[5];
```

```
int arr[10];
```

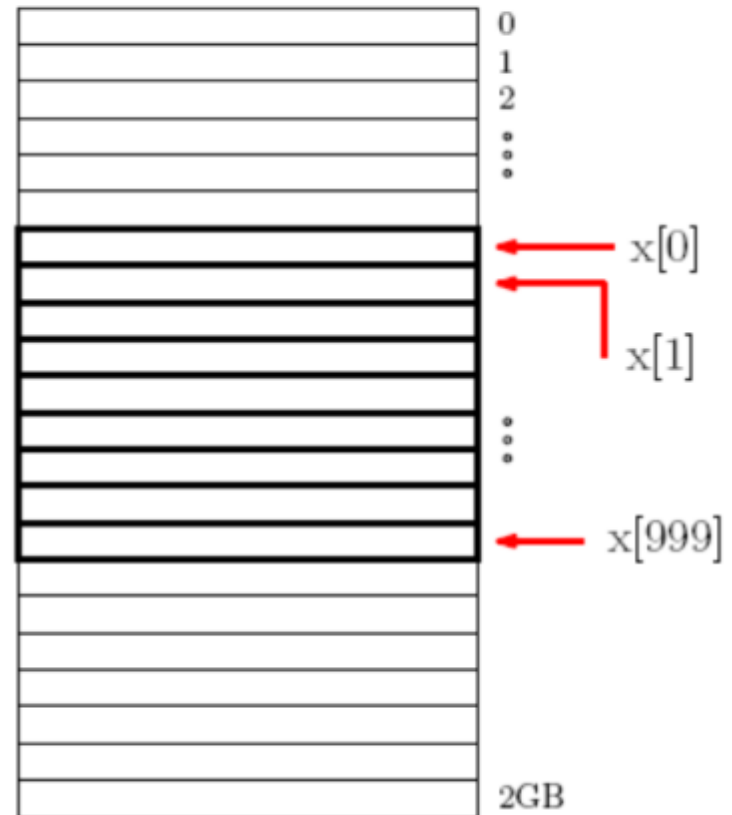
- Αρχικοποίηση.

```
float temp[5]={25.2,  
25.5, 26.6, 26,  
25.9};
```

```
float temp[]={25.2,  
25.5, 26.6, 26,  
25.9};
```

```
float temp[5]={25.2,  
25.5, 26.6};
```

```
int x[1000];
```





# Αναφορά στα Στοιχεία του Πίνακα

---

- Ένα στοιχείο του πίνακα προσδιορίζεται από το όνομα του πίνακα και ένα αριθμό που προσδιορίζει τη σειρά του:
  - Ο αριθμός που προσδιορίζει την σειρά ονομάζεται **δείκτης**.
  - Τα στοιχεία του πίνακα είναι συνεχόμενα στη μνήμη.
- Το πρώτο στοιχείο έχει πάντα σειρά 0.
- Το μέγεθος του πίνακα είναι πάντα ακέραιο.

Δείκτης:

0

1

2

3

4

25.2	25.5	26.6	26.0	25.9
------	------	------	------	------

temp[0]

temp[1]

temp[4]



# Παράδειγμα: Δήλωση και Αρχικοποίηση Πίνακα

```
#include <stdio.h>

#define N 100

void main()
{
    int a[N];
    int i;

    for ( i=0; i<N; i++ )
        a[i] = 0;
    ...
}
```

Τα στοιχεία ενός πίνακα με μέγεθος  $n$  είναι προσβάσιμα με ακέραιους δείκτες από το 0 ως το  $n-1$ .

0	a[0]
0	a[1]
0	a[2]
0	a[3]
0	a[4]
0	a[5]
0	a[6]
...	
0	a[98]
0	a[99]



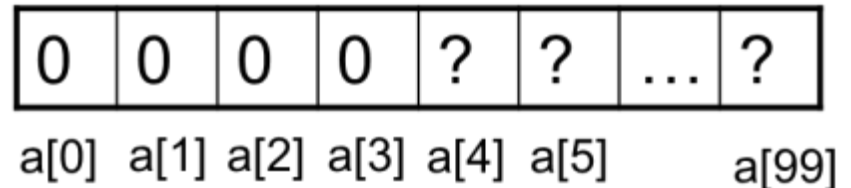
# Χρήση Στοιχείων του Πίνακα

- Τα στοιχεία του πίνακα τα χειριζόμαστε όπως και τις κοινές μεταβλητές.

```
int temp[100];  
...  
x = temp[2]*10;  
temp[0] = 25;  
temp[i] = temp[i+1] +  
    2;
```

```
for (i=0; i<5; i++)  
    if (temp[i]<25)  
        temp[i]++;  
    else temp[i]--;
```

```
#include <stdio.h>  
#define N 100  
  
void main()  
{  
    int a[N];  
    int i;  
  
    for ( i=0; i<4; i++ )  
        a[i] = 0;  
  
    ...  
}
```



# Παράδειγμα: Διαχείριση Στοιχείων Πίνακα

---

Αποθήκευση τιμών σε μονοδιάστατο πίνακα.

```
for (i=0; i<10; i++) scanf("%d", &a[i]);
```

Άθροιση τιμών μονοδιάστατου πίνακα.

```
for (i=0; i<10; i++) sum += a[i];
```

Εμφάνιση τιμών μονοδιάστατου πίνακα.

```
for (i=0; i<10; i++)  
printf("ΗΜΕΡΑ %d - ΩΡΕΣ %d\n", i, a[i]);
```



# Διαχείριση Στοιχείων Πίνακα

Μπορούμε να κάνουμε πράξεις μέσα στους δείκτες ενός πίνακα.

```
#include <stdio.h>
#define N 10

void main()
{
    int a[N];
    int i,j;

    i = j = 1;
    a[i+j] = 9
    for (i=5; i<N; i++)
        a[i-j] = 0;
    ...
}
```

Η έκφραση ανάμεσα στα [ ] υπολογίζεται πριν γίνει προσπέλαση του στοιχείου του πίνακα.

?	a[0]
?	a[1]
9	a[2]
?	a[3]
0	a[4]
0	a[5]
0	a[6]
0	a[7]
0	a[8]
?	a[9]



# Προτεραιότητα τελεστών

---

Η προτεραιότητα υπολογισμού της έκφρασης [ ] είναι η υψηλότερη δυνατή.

- 1 παρενθέσεις:** `()`, `[]`  
Υπολογίζονται πρώτα, από τα αριστερά προς τα δεξιά. Εάν υπάρχουν ένθετες υπολογίζονται πρώτα οι εσωτερικές.
- 2 μοναδιαίοι τελεστές:** `+`, `-`, `++`, `--` και `!`  
Υπολογίζονται από δεξιά προς τα αριστερά.
- 3 πολλαπλασιασμός, διαίρεση και υπόλοιπο:** `*`, `/`, ή `%`  
Υπολογίζονται δεύτερα από αριστερά προς τα δεξιά.
- 4 πρόσθεση, αφαίρεση:** `+` ή `-`  
Εάν υπάρχουν πολλοί, υπολογίζονται από τα αριστερά προς τα δεξιά.
- 5 Σχεσιακοί:** `<`, `>`, `<=`, `>=`  
Υπολογίζονται από τα αριστερά προς τα δεξιά.
- 6 Ισότητας:** `==`, `!=`  
Υπολογίζονται από τα αριστερά προς τα δεξιά.
- 7 λογικό AND:** `&&` Από αριστερά προς τα δεξιά.
- 8 λογικό OR:** `||` Από αριστερά προς τα δεξιά.
- 9 εκχώρησης:** `=`, `+=`, `-=`, `*=`, `/=`, `%=`  
Από δεξιά προς τα αριστερά.



# Αρχικοποίηση πινάκων (1/3)

- Όπως οι μεταβλητές, και οι πίνακες χρειάζονται αρχικοποίηση.
  - Ένας πίνακας είναι ένα ομαδοποιημένο σύνολο μεταβλητών.

```
#include <stdio.h>
#define N 100

void main()
{
    int a[N];
    int i;

    for ( i=0; i<N; i++ )
        a[i] = 0;
    ...
}
```

0	a[0]
0	a[1]
0	a[2]
0	a[3]
0	a[4]
0	a[5]
0	a[6]
...	
0	a[98]
0	a[99]



# Αρχικοποίηση πινάκων (2/3)

Αν ο πίνακας έχει λίγα στοιχεία, μπορεί να αρχικοποιηθεί κατά τη δήλωση του.

```
#include <stdio.h>

void main()
{
    int a[10]={11,5,4,8,22,6,97,4,1,0};
    int i;

    for ( i=0; i<10; i++ )
        printf("Stoixeio %d is
%d\n",i,a[i];
    ...
}
```

11	a[0]
5	a[1]
4	a[2]
8	a[3]
22	a[4]
6	a[5]
97	a[6]
4	a[7]
1	a[8]
0	a[9]





# Αρχικοποίηση πινάκων (3/3)

- Ένας πίνακας δεν αρχικοποιείται αυτόματα.

```
void main()
{
    int a[10]={0};
    ...
}
```

- Αν όμως αρχικοποιηθούν λιγότερα στοιχεία από το μέγεθος του, τότε ο compiler αρχικοποιεί τα υπόλοιπα σε 0.

- Ο compiler μπορεί να βρει το μέγεθος του πίνακα από τη λίστα αρχικοποίησης.

```
void main()
{
    int a[]={0,1,2,3,4,5,6,7};
    ...
}
```

- Είναι το ίδιο σαν να είχαμε δηλώσει  
`int a[8]={0,1,2,3,4,5,6,7};`



# Μέγεθος πινάκων

- Για τους πίνακες που έχουμε δει μέχρι τώρα, πρέπει το μέγεθος τους να είναι γνωστό κατά τη διάρκεια της μεταγλώττισης.
  - Μπορεί να είναι μια κυριολεκτική σταθερά.
    - `int X[100];`
  - Μπορεί να έχει δηλωθεί ως σταθερά του προ-επεξεργαστή με το `#define`.
    - `#define SIZE 100`
    - ...
    - `int X[SIZE];`
  - Μπορεί να βρεθεί από τη λίστα αρχικοποίησης.
  - Σε σύγχρονους compilers μπορεί να είναι μια αρχικοποιημένη μεταβλητή ή σταθερά ορισμένη με `const`.
    - `int size = 100`
    - `int X[size];`
- Οι πίνακες αυτοί ονομάζονται **στατικοί**.
  - Αργότερα θα δούμε πως μπορούμε να χρησιμοποιήσουμε **δυναμικούς πίνακες** των οποίων το μέγεθος προσδιορίζεται κατά τη διάρκεια εκτέλεσης.

← Καλό είναι να το αποφεύγουμε αυτό



# Σφάλμα εκτός ορίων πίνακα (1/2)

- Τι θα συμβεί αν έχουμε δηλώσει έναν πίνακα N στοιχείων `arr[N]` και προσπαθήσουμε μέσα στον κώδικα να προσπελάσουμε το στοιχείο `arr[N]`;
  - Ο πίνακας ξεκινάει από το στοιχείο `arr[0]` οπότε το `arr[N]` είναι εκτός των ορίων του.

```
#include <stdio.h>

void main()
{
    int
    x[11]={11,5,4,8,22,6,97,4,1,0,4};
    int i;

    for ( i=0; i<=11; i++ )
        printf("Stoixeio %d is
        %d\n",i,x[i];
    ...
}
```

## Σφάλμα χρόνου εκτέλεσης (RUN TIME ERROR)

- Το στοιχείο `x[11]` αντιστοιχεί σε μια θέση μνήμης όπου δεν ξέρουμε τι υπάρχει.

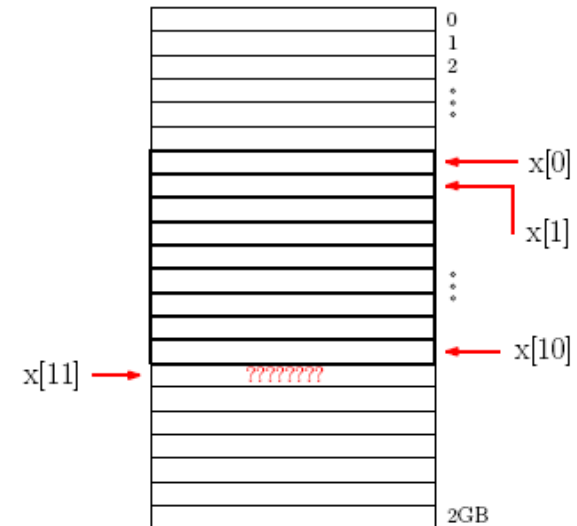


# Σφάλμα εκτός ορίων πίνακα (2/2)

```
#include <stdio.h>

void main()
{
    int x[11]={11,5,4,8,22,6,97,4,1,0,4};
    int i;

    for ( i=0; i<=11; i++ )
        printf("Stoixeio %d is %d\n",i,x[i]);
    ...
}
```



## Σφάλμα χρόνου εκτέλεσης (RUN TIME ERROR)

- Το στοιχείο x[11] αντιστοιχεί σε μια θέση μνήμης όπου δεν ξέρουμε τι υπάρχει .

- Στην πράξη αυτό που συμβεί είναι:
  - το πρόγραμμα μας θα τερματίσει απότομα (θα “κρασάρει” - crash).
  - θα πάρουμε “σκουπίδια” (garbage) όταν προσπελάσουμε το x[11].



# Εύρεση Αθροίσματος Στοιχείων Μονοδιάστατου Πίνακα

```
#include <stdio.h>

const int N=5;

void main()
{
    int sum=0,i,x[N];
        for (i=0; i<N; i++) { /* Διάβασμα πίνακα */
            printf("Dose to %d stoixeio: ",i+1);
            scanf("%d",&x[i]);
        }
        for (i=0; i<=N-1; i++)/* Υπολογισμός αθροίσματος */
            sum += x[i];
        printf("\nTo athroisma einai %d \n",sum);
    }
        /* Εκτύπωση αθροίσματος*/
```



# Εύρεση Μέσου Όρου Στοιχείων Μονοδιάστατου Πίνακα (1/2)

---

```
#include <stdio.h>
const int N=5;
void main()
{   int sum=0,i,x[N];
    float av;
    for (i=0;i<=N-1;i++) { /* Διάβασμα πίνακα */
        printf("Dose to %d stoixeio: ",i+1);
        scanf("%d",&x[i]); }
    for (i=0;i<=N-1;i++) /* Υπολογισμός Μ.Ο. */
        sum+=x[i];
    av=(float) sum/N;
    printf("\nMesos oros = %.2f \n",av);
}   /* Ακρίβεια 2 δεκαδικών ψηφίων*/
```



# Πολυδιάστατοι Πίνακες

---

- Στη C μπορούμε να ορίσουμε και πολυδιάστατους πίνακες.
- Ένας πίνακας μπορεί να έχει στοιχεία που είναι πίνακες.

```
int array[4][12]; /* 4 γραμμές, 12 στήλες
*/
array[0][1]; /* το στοιχείο της πρώτης
               γραμμής και δεύτερης
στήλης */
array[0,1]; /* ΛΑΘΟΣ */
```

- Αποθήκευση στη μνήμη σε γραμμές (ως μονοδιάστατος).



# Παράδειγμα: Δυσδιάστατος Πίνακας

- Βαθμοί 5 φοιτητών σε 3 μαθήματα.
- `int bathmoi[5][3];`

		Μαθήματα		
		0	1	2
Φοιτητές	0	7	8	6
	1	3	7	4
	2	3	5	5
	3	7	9	8
	4	4	4	6

- Αρχικοποίηση:
- `bathmoi[5][3] =`
- `{ {7, 8, 6},`
- `{3, 7, 4},`
- `{3, 5, 5},`
- `{7, 9, 8},`
- `{4, 4, 6} }`





# Παράδειγμα: Αρχικοποίηση δυσδιάστατου πίνακα

---

```
#define N 3
#define M 10

double f[N][M];
int i, j;
for (i = 0; i < N; i++)
    for (j = 0; j < M; j++)
        f[i][j] = 0.0;
```

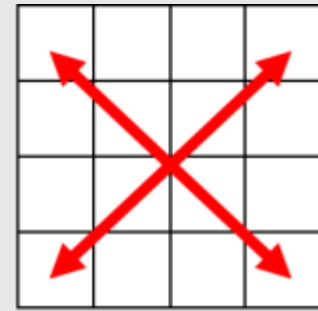


# Παράδειγμα:

## Επεξεργασία Στοιχείων Πίνακα (1/2)

- Άθροιση των στοιχείων των διαγωνίων σε τετραγωνικό πίνακα.

```
int i, j;
int sum1=0;
int sum2=0;
for ( i=0; i < N; i++ )
    for ( j=0; j < N; j++ )
    {
        if ( i == j)          /*κύρια διαγώνιος*/
            sum1 += a[i][j];
        if ((i + j) == N-1) /*δευτερεύουσα διαγώνιος*/
            sum2 += a[i][j];
    }
```

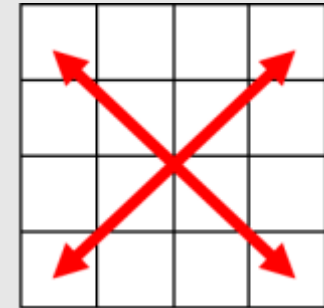


# Παράδειγμα:

## Επεξεργασία Στοιχείων Πίνακα (2/2)

- Άθροιση των στοιχείων των διαγωνίων σε τετραγωνικό πίνακα.

```
int i;
int sum1=0;
int sum2=0;
for (i=0; i < N; i++)
{
    sum1 +=a[i][i]; /*κύρια διαγώνιος*/
    sum2 += a[i][N-1-i];/*δευτερεύουσα
διαγώνιος*/
}
```



# Διάβασμα τιμών 2 πινάκων A και B Σύγκριση των στοιχείων και αποθήκευση του μεγαλύτερου στον πίνακα C

```
#include <stdio.h>

#define N 15 /* Μέγεθος των πινάκων */
int A[N], B[N], C[N];
main()
{
    int i;
    for (i=0; i<N; i++)
    {
        printf("Δώστε τιμή για το στοιχείο A[%d]: ", i);
        scanf("%d",&A[i]);
        printf("Δώστε τιμή για το στοιχείο B[%d]: ", i);
        scanf("%d",&B[i]);

        C[i] = (A[i]>B[i]) ? A[i] : B[i];
        /*Ισοδύναμο με: if (A[i]>B[i]) C[i] = A[i];
        else C[i] = B[i];*/
    }
}
```



# Εύρεση Μέσου Όρου Στοιχείων Μονοδιάστατου Πίνακα (2/2)

---

```
#include <stdio.h>
const int N=5;
void main()
{
    int sum=0,i,x[N];
    float av;
    for (i=0;i<=N-1;i++) { /* Διάβαση πίνακα */
        printf("Dose to %d stoixeio: ",i+1);
        scanf("%d",&x[i]); }
    for (i=0;i<=N-1;i++) /* Υπολογισμός Μ.Ο. */
        sum+=x[i];
    av=(float) sum/N;
    printf("\nMesos oros = %.2f \n",av);
} /* Ακρίβεια 2 δεκαδικών ψηφίων*/
```



# Εύρεση Μέσου Όρου Στοιχείων Δυσδιάστατου Πίνακα

```
#include <stdio.h>
const int N=5,M=3;
void main()
{
    int sum=0,i,j,x[N][M];
    float av;
    for (i=0;i<=N-1;i++)                /* Διάβασμα πίνακα */
        for (j=0;j<=M-1;j++) {
            printf("Dose to stoixeio x[%d][%d]: ",i,j);
            scanf("%d",&x[i][j]); }
    for (i=0;i<=N-1;i++)                /* Υπολογισμός Μ.Ο. */
        for (j=0;j<=M-1;j++)
            sum+=x[i][j];
    av=(float)sum/(N*M);
    printf("\nMesos oros = %.2f \n",av);
}                                       /* Ακρίβεια 2 δεκαδικών ψηφίων*/
```



# Εύρεση Μεγαλύτερου και Μικρότερου Στοιχείου Πίνακα καθώς και των Θέσεών τους

```
#include <stdio.h>
const int N=5;
void main()
{   int max,maxp,min,minp,i,x[N];
    for (i=0;i<=N-1;i++)           /* Διάβασμα Πίνακα */
        {   printf("Dose to %d stoixeio: ",i+1);
            scanf("%d",&x[i]);}
    max=min=x[0]; maxp=minp=0;
    for (i=0;i<=N-1;i++)           /* Εύρεση Μεγαλ./Μικρ. */
        {   if (x[i]>max) {
                max=x[i];
                maxp=i; }
            if (x[i]<min) {
                min=x[i];
                minp=i; } }
    printf("Megisto = %d sti thesi %d \n",max,maxp);
    printf("Elaxisto = %d sti thesi %d \n",min,minp);
}
```



# Εναλλαγή Πρώτης και Τελευταίας Γραμμής/Στήλης

```
/* Δεδομένου πίνακα x[N][M] */  
int i, temp;  
  
for (i=0; i<=M-1; i++)           /* Εναλλαγή γραμμής */  
{  
    temp=x[0][i];  
    x[0][i]=x[N-1][i];  
    x[N-1][i]=temp; }  
  
for (i=0; i<=N-1; i++)          /* Εναλλαγή στήλης */  
{  
    temp=x[i][0];  
    x[i][0]=x[i][M-1];  
    x[i][M-1]=temp; }
```





# Πολλαπλασιασμός Πινάκων

```
/* Δεδομένων Πινάκων a[X][Y] και b[Y][Z] */  
float c[X][Z];  
for (i=0;i<X;i++) /* Υπολογισμός γινομένου πίνακα */  
{  
    for (j=0;j<Z;j++)  
    {  
        c[i][j]=0;  
        for (k=0;k<Y;k++)  
            c[i][j]+=a[i][k]*b[k][j];  
    }  
}  
for (i=0;i<X;i++)          /* Εκτύπωση αποτελέσματος */  
{    for (j=0;j<Z;j++)  
        printf("%.0f\t",c[i][j]); printf("\n");  
}
```



# Άσκηση: Τι περιέχει το a[8];

---

```
int i, a[10];
for (i = 0; i < 10; i++)
    a[i] = 9 - i;
for (i = 0; i < 10; i++)
    a[i] = a[a[i]];
```

## ΛΥΣΗ:

	i:	0	1	2	3	4	5	6	7	8	9
a[i] after 1st loop:		9	8	7	6	5	4	3	2	1	0
a[i] after 2nd loop:		0	1	2	3	4	4	3	2	1	0



# Άσκηση

Σε ένα πίνακα 10 θέσεων αποθηκεύουμε συχνότητες εμφάνισης 10 αντικειμένων (20 εμφανίσεις) και βρίσκουμε το πιο συχνό αντικείμενο. Εντοπίστε πιθανά λάθη.

```
main()
{
  int num, i, j, maxi,
    a[9];
  for (j=0; j<20; j++)
    {scanf("%d", num);
     a[num-1]++;}
  for (i = 0; i < 10; i++);
    if (a[i] > a[maxi])
      maxi = i;
  printf("max = %d\n",
    maxi);
}
```



# Λάθη

```
main()
{
  int num, i, j, maxi,
  a[9];
  for (j=0; j<20; j++)
    {scanf("%d", num);
      a[num-1]++;}
  for (i = 0; i < 10; i++);
    if (a[i] > a[maxi])
      maxi = i;
  printf("max = %d\n",
  maxi);
}
```

- **#include <stdio.h>**
- **int a[10]**
- **scanf("%d", &num)**
- **Ο a[] πρέπει να αρχικοποιηθεί**
- **Το maxi πρέπει να αρχικοποιηθεί.**
- **Λάθος ; μετά τη 2η for.**
- **Μπορεί να εισαχθεί τιμή <=0 ή >10.**



---

# Τέλος Ενότητας



# Σημείωμα Αναφοράς

---

- Copyright Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, Στεργίου Κωνσταντίνος. «Εισαγωγή στον Δομημένο Προγραμματισμό». Έκδοση: 1.0. Κοζάνη 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.uowm.gr/courses/ICTE258/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Όχι Παράγωγα Έργα Μη Εμπορική Χρήση 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Ως Μη Εμπορική ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό



# Διατήρηση Σημειωμάτων

---

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους  
υπερσυνδέσμους.

