

Προγραμματισμός Υπολογιστών & Υπολογιστική Φυσική

Θεόδωρος Σαμαράς

theosama@auth.gr

<http://users.auth.gr/voyatzis/CProg>
<http://users.auth.gr/theosama>

Τύπος πίνακα (array)

- 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
- Χαρακτηριστικά:
 - Μέγεθος πίνακα
 - Τύπος στοιχείων

Δήλωση πίνακα

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

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

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

- Αρχικοποίηση

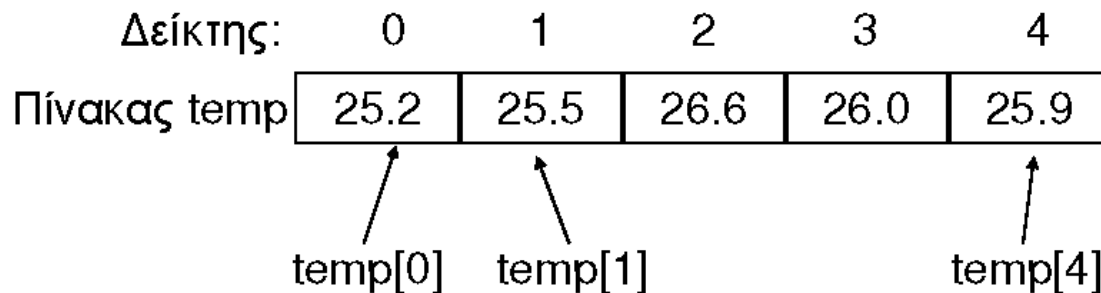
```
float temp[5]={25.2, 25.5, 26.6, 26, 25.9};
```

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

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

Αναφορά στα στοιχεία του πίνακα

- Ένα στοιχείο του πίνακα προσδιορίζεται από το όνομα του πίνακα και ένα αριθμό που προσδιορίζει τη σειρά του
- Το **πρώτο** στοιχείο του πίνακα έχει **σειρά 0** (αποθηκεύεται στη **θέση 0**)



Δήλωση και αρχικοποίηση πίνακα

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

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

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

Χρήση στοιχείων του πίνακα

- Τα στοιχεία του πίνακα τα χειριζόμαστε όπως και τις απλές μεταβλητές

```
x = temp[2]*10;
```

```
temp[0] = 25.4;
```

```
temp[i] = temp[i+1] + 2;
```

```
for (i=0; i<5; i++)
```

```
  if (temp[i]<25)
```

```
    temp[i]++;
```

```
  else temp[i]--;
```

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]);
```

Πολυδιάστατοι 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)

Δισδιάστατος 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}}
```

Αρχικοποίηση δισδιάστατου πίνακα

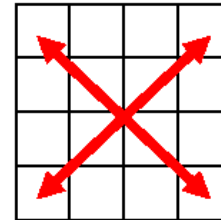
```
#define N 3
#define M 10

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

Επεξεργασία στοιχείων πίνακα (1)

- 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)
            sum1 += a[i][j];
        if ((i + j) == N-1)
            sum2 += a[i][j];
    }
```



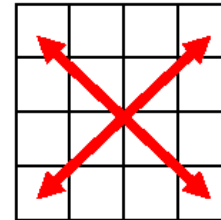
*/*κύρια διαγώνιος*/*

*/*δευτερεύουσα διαγώνιος*/*

Επεξεργασία στοιχείων πίνακα (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 (1D array)

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

Εύρεση μέσου όρου (2D array)

```
#include <stdio.h>
const int N=5, M=3;
void main() {
    int sum=0, i, j, x[N][M];
    float avg;

    for (i=0; i<N; i++)          /* Διάβαση πίνακα */
        for (j=0; j<M; j++) {
            printf("Dose to stoixeio x[%d][%d]: ", i, j);
            scanf("%d", &x[i][j]);
        }

    for (i=0; i<N; i++)          /* Υπολογισμός Μ.Ο. */
        for (j=0; j<M; j++)
            sum += x[i][j];
    avg = (float)sum / (N*M);
    printf("\nMesos oros = %.2f \n", avg);
}                                /* Ακρίβεια 2 δεκαδικών ψηφίων*/
```

Εύρεση μεγαλύτερου/μικρότερου και θέσης

```
#include <stdio.h>
const int N=5;
void main() {
    int max, maxp, min, minp, i, x[N];
    for (i=0; i<N; 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; 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 1^{ης} και τελευταίας γραμμής/στήλη

```
/* Δεδομένου πίνακα x[N][M] */
```

```
int i, temp;
```

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

```
for (i=0; i<N; 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];
/* Υπολογισμός γινομένου πίνακα */
for (i=0; i<X; i++) {
    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];
    }
}
/* Εκτύπωση αποτελέσματος */
for (i=0; i<X; i++) {
    for (j=0; j<Z; j++)
        printf("%.0f\t", c[i][j]);
    printf("\n");
}
```

Ταξινόμηση πίνακα

```
int a[] = {3, 2, 5, 7, 1, 8, 4, 9, 6};
```



```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

- Μέθοδοι ταξινόμησης:
 - Ταξινόμηση φυσαλίδας (bubble sort)
 - Ταξινόμηση επιλογής (selection sort)
 - Γρήγορη ταξινόμηση (quicksort)

Μέθοδος ταξινόμησης **Bubble Sort**

- Σύγκρινε το πρώτο με το δεύτερο στοιχείο
- Αν το πρώτο είναι μεγαλύτερο αντάλλαξε τις τιμές τους
- Συνέχισε με το δεύτερο και τρίτο στοιχείο
- ...
- Επανάλαβε τη διαδικασία ξεκινώντας πάλι από το πρώτο στοιχείο και συγκρίνοντας όλα τα στοιχεία εκτός του τελευταίου
- ...
- Επανάλαβε τη διαδικασία εφόσον υπάρχει έστω μία ανταλλαγή

Μέθοδος ταξινόμησης Bubble Sort

```
/* Δεδομένου πίνακα a[N] */
typedef unsigned char bool;

int i, n, tmp, a[N];
bool stop_taxinomisi;

n = N; /* Ο αριθμός στοιχείων που θα ταξινομηθούν */
do {
    stop_taxinomisi=TRUE;
    for (i=0; i<n-1; i++)
        if (a[i] > a[i+1]) {
            tmp = a[i];
            a[i] = a[i+1];
            a[i+1] = tmp;
            stop_taxinomisi=FALSE; /*έγινε ανταλλαγή*/
        }
    n=n-1; /*Μείωσε το εύρος των δυνατών συγκρίσεων */
} while(!stop_taxinomisi);
```