

Μέτρηση

Για να μετρήσουμε χρησιμοποιούμε μια **ακέραια** μεταβλητή (μετρητή). Τη μηδενίζουμε (εξωτερικά από ένα επαναληπτικό σχήμα) και όποτε συμβαίνει αυτό που θέλουμε την αυξάνουμε κατά 1 (εσωτερικά σε ένα επαναληπτικό σχήμα ή σε μία δομή επιλογής).

Π.χ.
 /* Πλήθος θετικών μεταξύ 10 αριθμών */
 #include <stdio.h>
 int i,count;
 float a;
 void main() {
 count=0;
 for (i=1;i<=10;i++) {
 printf("DOSTE ARITHMO : ");
 scanf("%f",&a);
 if (a > 0.0)
 count ++;
 } /* for */
 printf("8ETIKA : %d",count);
 } /* main */

Υπολογισμός αθροίσματος N αριθμών

Για να αθροίσουμε χρησιμοποιούμε μια μεταβλητή (αθροιστή). Τη μηδενίζουμε (εξωτερικά από ένα επαναληπτικό σχήμα) και προσθέτουμε σε αυτήν αυτά που θέλουμε να αθροίσουμε (εσωτερικά σε ένα επαναληπτικό σχήμα ή σε μία δομή επιλογής).

Π.χ.
 #include <stdio.h>
 /* Υπολογισμός αθροίσματος N=10 */
 int i;
 float a,sum;
 void main() {
 sum=0.0;
 for (i=1;i<=10;i++) {
 printf("DOSTE ARITHMO : ");
 scanf("%f",&a);
 sum += a;
 } /* for */
 printf("A8ROISMA = %f",sum);
 } /* main */

Υπολογισμός γινομένου N αριθμών

Για να υπολογίσουμε γινόμενο χρησιμοποιούμε μια μεταβλητή. Τη θέτουμε ίση με 1 και πολλαπλασιάζουμε με αυτήν αυτά που θέλουμε να συμπεριλάβουμε στο γινόμενο.

Π.χ.
 #include <stdio.h>

 /* Υπολογισμός γινομένου N=10*/
 int i;
 float a,prod;
 void main()
 {
 prod=1.0;
 for (i=1;i<=10;i++) {
 printf("DOSTE ARITHMO : ");
 scanf("%f",&a);
 prod *= a;
 } /* for */
 printf("GINOMENO = %f",prod);
 }

Μέσος όρος N Αριθμών

Ο μέσος όρος είναι άθροιση όλων και διαίρεση δια του πλήθους. Εδώ το πλήθος των αριθμών, άρα και των επαναλήψεων είναι γνωστό, οπότε χρησιμοποιούμε το επαναληπτικό σχήμα **for**.

Π.χ.
 #include <stdio.h>

 int i;
 float a,sum,mo;

 void main() {
 sum=0.0;
 for (i=1;i<=10;i++) {
 printf("DOSTE ARITHMO : ");
 scanf("%f",&a);
 sum += a;
 } /* for */
 mo = sum/10;
 printf("MESOS OROS = %f",mo);
 } /* main */

Μέσος όρος άγνωστου πλήθους αριθμών

Εδώ το πλήθος των αριθμών, άρα και των επαναλήψεων δεν είναι γνωστό, οπότε χρησιμοποιούμε **while**. Παράλληλα πρέπει να μετράμε και το πλήθος των αριθμών.

Π.χ.
 #include <stdio.h>

 float a,sum,mo;
 int count;

 void main()
 {
 sum=0.0;
 count=0;
 /* Για τιμή -10000.0 τερματίζει */
 printf ("dose arithmo:");
 scanf ("%f",&a);
 while (a!=-10000.0) {
 count++;
 sum+=a;
 printf ("dose arithmo:");
 scanf ("%f",&a);
 }
 if(count==0)
 mo = 0.0;
 else
 mo = sum/count;
 printf("mo = %f\n",mo);
 }

Επανάληψη μέχρι να δοθεί σωστή τιμή

Για να εξασφαλίσουμε ότι ο χρήστης δεν δίνει τιμή εκτός ορίων, χρησιμοποιούμε **do – while**.

Π.χ.
 #include <stdio.h>
 /* zhteitai poso apo 1 os 999 */
 int poso;

 main()
 {
 do {
 printf("doste poso:");
 scanf ("%d",&poso);
 }

```
while ((poso<1) || (poso>999));  
printf ("poso sosto");  
}
```

Μέγιστος και Ελάχιστος N Αριθμών

Για το μέγιστο χρησιμοποιούμε μια μεταβλητή που τη θέτουμε ίση με μια πολύ μικρή τιμή. Μετά αν ο αριθμός που βλέπουμε είναι μεγαλύτερος την αντικαθιστούμε με αυτόν. Για το ελάχιστο χρησιμοποιούμε μια μεταβλητή που τη θέτουμε ίση με μια πολύ μεγάλη τιμή. Μετά αν ο αριθμός που βλέπουμε είναι μικρότερος την αντικαθιστούμε με αυτόν.

Π.χ. /* Eyresi megistoy-elaxistoy */
 #include <stdio.h>

 int min,max,a,i;

 main() {
 min=32767;
 max=-32768;
 for(i=1;i<10;i++){
 printf("DOSE ARITHMO:");
 scanf("%d",&a);
 if(a>max)
 max=a;
 if(a<min)
 min=a;
 }
 printf("MEGALYTEROS=%d\n",max);
 printf("MIKROTEROS=%d\n",min);
 }

Πίνακες

Πίνακες Μιας Διάστασης

Δήλωση Μονοδιάστατου Πίνακα

Τύπος δεδομένων όνομα[διάσταση];

Π.χ.
 int a[10];

Χρήση των στοιχείων του Πίνακα

Π.χ.
 a[0] = 5;
 a[1] = a[0] + 2;
 if (a[i] == 0) {
 ...
 }

Αποθήκευση τιμών σε μονοδιάστατο πίνακα

Π.χ.
 for (i=0; i<=9; i++)
 scanf("%d", &a[i]);

Άθροιση τιμών σε μονοδιάστατο πίνακα

Π.χ.
 sum=0;
 for (i=0; i<=9; i++)
 sum += a[i];

Εμφάνιση (εκτύπωση) τιμών μονοδιάστατου πίνακα

Π.χ.
 for (i=0; i<=9; i++)
 printf("%d ",a[i]);

Μέσος Όρος N στοιχείων σε Πίνακα

Τα στοιχεία διαβάζονται ένα-ένα και αποθηκεύονται στον πίνακα. Εδώ το πλήθος των αριθμών, άρα και των επαναλήψεων είναι γνωστό, οπότε χρησιμοποιούμε **for**.

```
Π.χ.
#include <stdio.h>
/* mo - pinaka - 1 x M */
#define M 11
int p[M];
int n,x,i;
float mo;

main() {
scanf("%d",&n);
x=0;
for (i=1;i<=n;i++) {
printf("%d",p[i]);
x += p[i];
} /* for */
mo= (float)x/n;
printf("%f",mo);
}
```

Ταξινόμηση αριθμών σε πίνακα

Για την ταξινόμηση (διάταξη) αριθμών σε πίνακα εφαρμόζουμε τον πιο εύκολο (όχι όμως και πιο γρήγορο) αλγόριθμο *Bubble Sort* (Διάταξη Φυσαλίδας).

Ο αλγόριθμος κάνει από 1 ως N περάσματα (L). Σε κάθε πέραςμα ελέγχει το κάθε στοιχείο με το διπλάνο του. Αν βρει ότι δεν βρίσκονται στη σωστή σειρά τα ανταλλάσσει μεταξύ τους. Επειδή η πλήρης διάταξη μπορεί να έχει επιτευχθεί με λιγότερα από N περάσματα, χρησιμοποιούμε τη λογική μεταβλητή DONE.

Αν έχει κάνει ένα πέραςμα χωρίς να κάνει αλλαγή, σημαίνει ότι η πλήρης διάταξη έχει γίνει, οπότε τα περάσματα σταματούν.

```
Π.χ.
#include <stdio.h>
#define N 10
#define TRUE 1
#define FALSE 0
int a[N];
int j, l, t, done;

main() {
l = N;
do
{
done = TRUE;
l--;
for(j=0; j<l; j++)
if(a[j] > a[j+1]) {
done = FALSE;
t = a[j];
a[j] = a[j+1];
a[j+1] = t;
}
} while(!done);
}
```

Αναζήτηση τιμής σε πίνακα

Γραμμική Αναζήτηση (Linear Search)

Χρησιμοποιούμε τη γραμμική αναζήτηση για να βρούμε κάποια συγκεκριμένη τιμή σε πίνακα, **αν τα στοιχεία του πίνακα ΔΕΝ ΕΙΝΑΙ ταξινομημένα**.

Ο αλγόριθμος ελέγχει όλα τα στοιχεία με τη σειρά. Αν βρει το στοιχείο σταματάει την αναζήτηση.

```
Π.χ.
#include <stdio.h>

#define N 10
#define TRUE 1
#define FALSE 0

int a[N];
int i,found;
int timi;

main() {
found = FALSE;
scanf("%d", &timi);
i = 0;
while ( (!found) && (i < N) )
if ( a[i]==timi )
found = TRUE;
else i++;
if ( found ) printf("ΒΡΕΘΗΚΕ ΤΟ %d ΣΤΗ ΘΕΣΗ %d\n", a[i],i);
else printf("ΔΕΝ ΒΡΕΘΗΚΕ\n");
}
```

Διαδική Αναζήτηση (Binary Search)

Χρησιμοποιούμε τη δυαδική αναζήτηση για να βρούμε κάποια συγκεκριμένη τιμή σε πίνακα, **αν τα στοιχεία του ΕΙΝΑΙ ταξινομημένα**.

Ο αλγόριθμος ελέγχει το μεσαίο στοιχείο. Αν η τιμή που αναζητά είναι μικρότερη από το μεσαίο, περιορίζει το διάστημα αναζήτησης, από την αρχή έως το πριν από το μεσαίο και δεν ξαναελέγχει τα υπόλοιπα. Αν είναι μεγαλύτερη, τότε περιορίζει το διάστημα από το μετά το μεσαίο έως το τέλος. Αν βρει το στοιχείο σταματάει την αναζήτηση.

```
Π.χ.
#include <stdio.h>

#define N 10
#define TRUE 1
#define FALSE 0

int a[N];
int i,found;
int timi;
int top,bot,mid;

main()
{
top = 0;
bot = N-1;
found = FALSE;
scanf("%d", &timi);
while((!found) && (top <= bot))
{
mid = (top+bot)/2;
if(a[mid] == timi)
found = TRUE;
else if(a[mid] > timi)
bot = mid-1;
else top = mid+1;
}
if (found)
printf("ΒΡΕΘΗΚΕ ΤΟ %d ΣΤΗ ΘΕΣΗ %d\n", a[mid],mid);
else printf("ΔΕΝ ΒΡΕΘΗΚΕ\n");
}
```

Πίνακες δύο διαστάσεων

Δήλωση διδιάστατου πίνακα (και βοηθητικών μεταβλητών)

```
Π.χ.
#define N 3
#define M 10

double f[N][M];
int i,j;
double a[N][N];
double sum1,sum2;
```

Αρχικοποίηση διδιάστατου πίνακα

(μηδενισμός όλων των στοιχείων του πίνακα)

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

Η χρήση των δύο **for** δημιουργεί όλους τους συνδυασμούς i, j έτσι ώστε η πράξη να γίνει για όλα τα στοιχεία του πίνακα.

Επεξεργασία στοιχείων

Αύξηση όλων των στοιχείων της 2^{ης} στήλης κατά 1

```
for(i=0; i < N; i++)
f[i][1] = f[i][1]+ 1;
```

Άθροιση κατά γραμμές (π.χ. 3^η = 1^η + 2^η)

```
for(j=0; j < M; j++)
f[2][j]=f[0][j]+f[1][j];
```

Εκτύπωση των στοιχείων

```
for ( i=0; i < N; i++ ){
for ( j=0; j < M; j++ )
printf("%1f",f[i][j] );
printf("\n");
}
```

Άθροιση των στοιχείων των διαγωνίων σε τετραγωνικό πίνακα

Τα στοιχεία της πρώτης διαγωνίου έχουν όλα αριθμό γραμμής = αριθμό στήλης δηλ. i = j. Τα στοιχεία της δεύτερης διαγωνίου έχουν όλα αριθμό γραμμής+αριθμό στήλης=διάσταση -1 δηλαδή i + j = N-1.

```
sum1 = sum2 = 0.0;
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];
}
```

Πολλαπλασιασμός πινάκων

```
#define M 2
#define L 2
#define N 2

main(){
int s,i,j,k;
int A[M][L]={ {1,2}, {3,4} };
int B[L][N]={ {5,6}, {7,8} },C[M][N];

for( i=0; i < M ; i++ )
for( j=0; j < N; j++ ){
for( k=0, s=0; k < L; k++ )
s+=A[i][k]*B[k][j];
C[i][j]=s;
}
for( i=0; i < M; i++ )
for( j=0; j < N; j++ )
printf("C[%d][%d] = %3d\n",i,j,C[i][j]);
}
```