



Που είμαστε ;

Τύποι :

int

float

Εντολές :

if - else

for

Συναρτήσεις :

printf()

Που είμαστε ;

Τύποι :

int

float

Εντολές :

if - else

for

Συναρτήσεις :

printf()



Η εντολή for :

```
for ( i=0 ; i < points ; i=i+1 )  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }
```

```
for ( i=0 ; i < points ; i++ )  
    printf("The value of i is %d\n", i);
```

Η εντολή for :

```
float x;
```

```
for ( x=0.0 ; x <= 5.0 ; x += 1.0 )  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }
```


Ενθετες for :

```
float  x;
int    d;

for ( x=0.0 ; x <= 5.0 ; x += 1.0 )
{
    for ( d=0 ; d < 10 ; d++ )
    {
        ΕΝΤΟΛΕΣ . . .
    }
}
```


Ενθετες for :

```
float  x;  
int    d;  
  
for ( x=0.0 ; x <= 5.0 ; x += 1.0 )  
    {  
        for ( d=0 ; d < 10 ; d++ )  
            {  
                printf("%f %d\n", x, d );  
            }  
    }
```


Ενθετες for :

```
int    x;
int    d;

for ( x=0 ; x < 3 ; x++ )
{
    for ( d=x ; d < 3 ; d++ )
    {
        printf("%d %d\n", x, d );
    }
}
```


Ενθετες for :

```
#include <stdio.h>
```

```
#include <math.h>
```

```
main()
```

```
{
```

```
float x;
```

```
float y;
```

```
for ( x=0.0 ; x <= 5.0 ; x += 0.10 )
```

```
{
```

```
    for ( y=0.0 ; y <= 3.14 ; y += 0.010 )
```

```
        printf("%f %f %f\n", x, y, sqrt(x)*sin(y) );
```

```
    }
```

```
}
```


Ενθετες for :

```
#include <stdio.h>
```

```
#include <math.h>
```

```
main()
```

```
{
```

```
float x;
```

```
float y;
```

```
for ( x=5.0 ; x >= 0.0 ; x -= 0.10 )
```

```
{
```

```
    for ( y=0.0 ; y <= 3.14 ; y += 0.010 )
```

```
        printf("%f %f %f\n", x, y, sqrt(x)*sin(y) );
```

```
    }
```

```
}
```


Η συνάρτηση printf() :

```
printf(" Hello world\n");
```

Hello world

```
int a;
```

```
a = 3123;
```

```
printf("The value of a is %d \n", a );
```

The value of a is 3123

Η συνάρτηση printf() :

```
float x;  
x = 1.05670;  
printf("The value of x is %f \n", x );
```

The value of x is 1.0567

Η συνάρτηση printf() :

```
int    r;  
float x;  
x = 34.05;  
r = 95;  
printf("The values are %d and %f \n", r, x );
```

The values are 95 and 34.05

Η εντολή if :

```
if ( points < 2 )  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }
```


Το ζεύγος εντολών if-else :

```
if ( points < 2 )  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }  
else  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }
```


Συνδεδεμένες if-else :

EAN η τιμή της μεταβλητής i είναι ίση με 1 :

τύπωσε “Η τιμή της i είναι 1”, ΑΛΛΙΩΣ,

EAN η τιμή της μεταβλητής i είναι ίση με 2 :

τύπωσε “Η τιμή της i είναι 2”, ΑΛΛΙΩΣ,

EAN η τιμή της μεταβλητής i είναι ίση με 3 :

τύπωσε “Η τιμή της i είναι 3”, ΑΛΛΙΩΣ,

EAN η τιμή της μεταβλητής i είναι μικρότερη από 1 :

τύπωσε “Η τιμή της i είναι < 1 ”, ΑΛΛΙΩΣ,

τύπωσε “Η τιμή της i είναι μεγαλύτερη από 3”

Συνδεδεμένες if-else :

```
if ( i == 1 )  
    printf("The value of i is 1\n");  
else  
    if ( i == 2 )  
        printf("The value of i is 2\n");  
    else  
        if ( i == 3 )  
            printf("The value of i is 3\n");  
        else  
            if ( i < 1 )  
                printf("The value of i is less than 1\n");  
            else  
                printf("The value of i is greater than 3\n");
```


Συνδεδεμένες if-else :

```
if ( i == 1 )  
    printf("The value of i is 1\n");  
else  
    if ( i == 2 )  
        printf("The value of i is 2\n");  
    else  
        if ( i == 3 )  
            printf("The value of i is 3\n");  
        else  
            if ( i < 1 )  
                printf("The value of i is less than 1\n");  
            else  
                printf("The value of i is greater than 3\n");
```


Πολυκλαδικές αποφάσεις.

Συνδεδεμένες if-else και πώς να γίνουν όλα λάθος :

```
if ( a > b )  
    if ( b > c )  
        max = a;  
else  
    if ( b > c )  
        max = b;  
    else  
        max = c;
```


Συνδεδεμένες if-else και πώς να γίνουν όλα λάθος :

```
if ( a > b )  
    if ( b > c )  
        max = a;  
else  
    if ( b > c )  
        max = b;  
    else  
        max = c;
```



Απουσία αγκυλών, ένα else συνδέεται με την πλησιέστερη εντολή if (από αυτές που προηγούνται).

Συνδεδεμένες if-else και πώς να γίνουν όλα λάθος :

```
if ( a > b )  
    {  
        if ( b > c )  
            max = a;  
    }  
  
else  
    {  
        if ( b > c )  
            max = b;  
        else  
            max = c;  
    }
```


Συνδεδεμένες if-else και πώς να γίνουν όλα λάθος :

```
if ( a > b )  
    {  
        if ( b > c )  
            max = a;  
    }  
  
else  
    {  
        if ( b > c )  
            max = b;  
        else  
            max = c;  
    }
```

Χρησιμοποιείτε αγκύλες :
Κάνουν το πρόγραμμα (και τις προθέσεις σας) πολύ πιο ευανάγνωστες.

if και λογικοί τελεστές :

Εάν : $a > b$ και $b > c$ τότε ...

```
if ( a > b  &&  b > c )  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }
```

Εάν : $a > b$ ή $a > c$ τότε ...

```
if ( a > b  ||  a > c )  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }
```


if και λογικοί τελεστές :

Εάν : (a ίσο με b) και $b > c$ τότε ...

```
if ( (a == b)  &&  (b > c) )  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }
```

Εάν : $a > b$ ή (a διάφορο του c) τότε ...

```
if ( (a > b)  ||  (a != c) )  
    {  
        ΕΝΤΟΛΕΣ . . .  
    }
```


if και λογικοί τελεστές :

Εάν το : $(a \text{ ίσο με } b) \text{ και } b > c$ είναι ψευδές τότε ...

```
if ( !(a == b && b > c) )  
{  
    ΕΝΤΟΛΕΣ . . .  
}
```


if και λογικοί τελεστές :

Εάν το : $(a \text{ ίσο με } b) \text{ και } b > c$ είναι ψευδές τότε ...

```
if ( ! ( (a == b) && (b > c) ) )  
{  
    ΕΝΤΟΛΕΣ . . .  
}
```


Επανάληψη τελεστών :

<code>d++ ;</code>	Αύξησε κατά μια μονάδα την τιμή της μεταβλητής d
<code>f += 5.0 ;</code>	Αύξησε κατά 5.0 την τιμή της μεταβλητής f
<code>f += q ;</code>	Αύξησε την τιμή της μεταβλητής f κατά q
<code>f -= q ;</code>	Μείωσε την τιμή της μεταβλητής f κατά q
<code>f /= q ;</code>	Η καινούργια τιμή της f είναι η παλιά τιμή της f διαιρεμένη με το q
<code>f *= q ;</code>	Η καινούργια τιμή της f είναι η παλιά τιμή της f πολλαπλασιασμένη με το q
<code>&&</code>	Λογικό ΚΑΙ (AND)
<code> </code>	Λογικό Ή (OR)
<code>!</code>	Λογικό ΟΧΙ (NOT)
<code>==</code>	Έλεγχος ισότητας (ενώ το απλό = για ανάθεση τιμής)
<code>!=</code>	Έλεγχος ανισότητας


```

#include <stdio.h>
#include <math.h>

main()
{
    float  x[5000];
    float  y[5000];
    int     points;
    float  val1, val2;
    float  Sxy, Sx, Sy, Sx2;
    float  a, b;
    int     i;

    points = 0;
    while ( scanf("%f %f", &val1, &val2) == 2 )
    {
        x[points] = val1;
        y[points] = val2;
        points++;
    }

    Sxy = 0.0;
    Sx  = 0.0;
    Sy  = 0.0;
    Sx2 = 0.0;
    for ( i=0 ; i < points ; i++ )
    {
        Sx = Sx + x[i];
        Sy = Sy + y[i];
        Sx2 += x[i]*x[i];
        Sxy += x[i]*y[i];
    }

    a = (points * Sxy - Sx * Sy) / (points * Sx2 - Sx * Sx);
    b = ((Sy * Sx2) - (Sxy * Sx)) / ((points * Sx2) - (Sx * Sx));

    printf(" %f %f\n", a, b );
}

```

Τι μας είχε μείνει από το πρόγραμμα για τη μέθοδο Bradford ?

Η εντολή while :

```
while ( i <= 10 )  
{  
    ΕΝΤΟΛΕΣ . . .  
}
```

- Εάν η τιμή της μεταβλητής *i* είναι μεγαλύτερη του 10, *αγνόησε (μην εκτελείς)* όσες εντολές είναι ανάμεσα στις αγκύλες που ακολουθούν την εντολή `while`. **Αλλιώς ...**
- Όσο η τιμή της μεταβλητής *i* είναι μικρότερη ή ίση του 10, συνέχισε να εκτελείς επαναληπτικά τις εντολές που περιέχονται ανάμεσα στο ζεύγος των αγκυλών (μέχρις ότου η συνθήκη `i <= 10` να είναι ψευδής).

Η εντολή while :

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
int    i;
```

```
i = 0;
```

```
while ( i <= 10 )
```

```
{
```

```
    printf("The value of i is %d\n", i );
```

```
    i++;
```

```
}
```

```
}
```


Η συνάρτηση scanf() :

Είναι το ανάλογο της printf() αλλά αντί να μετατρέπει και να τυπώνει στην καθιερωμένη έξοδο, διαβάζει και μετατρέπει από την καθιερωμένη είσοδο.

Η συνάρτηση scanf() :

```
scanf ("%f %f %d", &x, &y, &i) ;
```

Διάβασε χαρακτήρες από την καθιερωμένη είσοδο και μετέτρεψε την ακολουθία χαρακτήρων σε τρεις αριθμούς : τους δυο πρώτους σε τύπο float, τον τελευταίο σε τύπο int. Τοποθέτησε τις τιμές αυτές στις μεταβλητές με τα ονόματα x, y και i (οι οποίες θα πρέπει να είχαν δηλωθεί ως float, float και int).

Η συνάρτηση scanf() :

```
scanf("%f %f %d", &x, &y, &i);
```



Η συνάρτηση scanf() :

Η scanf() επιστρέφει (στο πρόγραμμα που την κάλεσε) μια τιμή η οποία είναι ίση με τον αριθμό των στοιχείων εισόδου που μετέτρεψε και αποθήκευσε με επιτυχία ή μια ειδική τιμή (που συμβολίζεται με το **EOF**) εάν έφτασε στο τέλος του αρχείου (End Of File).

Η συνάρτηση scanf() :

Η scanf() επιστρέφει (στο πρόγραμμα που την κάλεσε) μια τιμή η οποία είναι ίση με τον αριθμό των στοιχείων εισόδου που μετέτρεψε και αποθήκευσε με επιτυχία ή μια ειδική τιμή (που συμβολίζεται με το EOF) εάν έφτασε στο τέλος του αρχείου (End Of File).

Και εάν δεν είναι αρχείο αλλά το τερματικό ;



Η συνάρτηση scanf() :

```
scanf("%f %f %d", &x, &y, &i);
```

Η παρουσία των «&» πριν από τα ονόματα των μεταβλητών στις οποίες θέλουμε να αναθέσουμε τιμές είναι αναγκαία (και η απουσία τους, ένα από τα συχνότερα συντακτικά σφάλματα).

Η συνάρτηση scanf() :

```
#include <stdio.h>
```

```
#include <math.h>
```

```
main()
```

```
{
```

```
    float        sum;
```

```
    float        val;
```

```
    int          how_many;
```

```
    how_many = 0;
```

```
    sum = 0.0;
```

```
    while( scanf("%f", &val) != EOF )
```

```
    {
```

```
        sum += val;
```

```
        how_many++;
```

```
    }
```

```
    printf("The average value is %f\n", (sum / how_many));
```

```
}
```


Η συνάρτηση scanf() :

```
#include <stdio.h>
```

```
#include <math.h>
```

```
main()
```

```
{
```

```
    float        sum;
```

```
    float        val;
```

```
    int          how_many;
```

```
    how_many = 0;
```

```
    sum = 0.0;
```

```
    while( scanf("%f", &val) != EOF )
```

```
    {
```

```
        sum += val;
```

```
        how_many++;
```

```
    }
```

```
    printf("The average value is %f\n", (sum / how_many));
```

```
}
```

Με αυτή τη μορφή της while επιτυγχάνονται ταυτόχρονα

1. Η κλήση της scanf() για την ανάγνωση του επόμενου αριθμού.
2. Ο έλεγχος της επιστρεφόμενης [από την scanf()] τιμής.

Η συνάρτηση scanf() :

```
how_many = 0;
sum = 0.0;
from_scanf = scanf("%f", &val);
while( from_scanf != EOF )
{
    sum += val;
    how_many++;
    from_scanf = scanf("%f", &val);
}
```


Η συνάρτηση scanf() :

```
#include <stdio.h>
```

```
#include <math.h>
```

```
main()
```

```
{
```

```
    float        sum;
```

```
    float        val;
```

```
    int  how_many;
```

```
    how_many = 0;
```

```
    sum = 0.0;
```

```
    while( scanf("%f", &val) == 1 )
```

```
    {
```

```
        sum += val;
```

```
        how_many++;
```

```
    }
```

```
    printf("The average value is %f\n", (sum / how_many));
```

```
}
```


Η συνάρτηση scanf() :

```
#include <stdio.h>
```

```
#include <math.h>
```

```
main()
```

```
{
```

```
    float        sum;
```

```
    float        val;
```

```
    int  how_many;
```

```
    how_many = 0;
```

```
    sum = 0.0;
```

```
    while( scanf("%f", &val) == 1 )
```

```
    {
```

```
        sum += val;
```


```
        how_many++;
```

```
    }
```

```
    printf("The average value is %f\n", (sum / how_many));
```

```
}
```

Σημειώστε ότι το πρόγραμμα σε αυτή τη μορφή είναι καθ' όλα έτοιμο για μια παταγώδη αποτυχία εάν η είσοδος δεν μας δώσει ούτε ένα αριθμό ...



Το πρόγραμμα για τη μέθοδο Bradford


```

#include <stdio.h>
#include <math.h>

main()
{
    float x[5000];
    float y[5000];
    int points;
    float val1, val2;
    float Sxy, Sx, Sy, Sx2;
    float a, b;
    int i;

    points = 0;
    while ( scanf("%f %f", &val1, &val2) == 2 )
    {
        x[points] = val1;
        y[points] = val2;
        points++;
    }

    Sxy = 0.0;
    Sx = 0.0;
    Sy = 0.0;
    Sx2 = 0.0;
    for ( i=0 ; i < points ; i++ )
    {
        Sx = Sx + x[i];
        Sy = Sy + y[i];
        Sx2 += x[i]*x[i];
        Sxy += x[i]*y[i];
    }

    a = (points * Sxy - Sx * Sy) / (points * Sx2 - Sx * Sx);
    b = ((Sy * Sx2) - (Sxy * Sx)) / ((points * Sx2) - (Sx * Sx));

    printf(" %f %f\n", a, b );
}

```


Επικεφαλίδα και δηλώσεις :

```
#include <stdio.h>
#include <math.h>

main()
{
    float      x[5000];
    float      y[5000];
    int         points;
    float       val1, val2;
    float       Sxy, Sx, Sy, Sx2;
    float       a, b;
    int         i;
```


Εισαγωγή και αποθήκευση των δεδομένων :

```
points = 0;
while ( scanf("%f %f", &val1, &val2) == 2 )
{
    x[points] = val1;
    y[points] = val2;
    points++;
}
```


Επεξεργασία και έξοδος :

```
Sxy = 0.0;
Sx  = 0.0;
Sy  = 0.0;
Sx2 = 0.0;
for ( i=0 ; i < points ; i++ )
{
    Sx = Sx + x[i];
    Sy = Sy + y[i];
    Sx2 += x[i]*x[i];
    Sxy += x[i]*y[i];
}

a = (points * Sxy - Sx * Sy) / (points * Sx2 - Sx * Sx);
b = ((Sy * Sx2) - (Sxy * Sx)) / ((points * Sx2) - (Sx * Sx));

printf(" %f %f\n", a, b );
}
```


Άλλα παραδείγματα ...

Διάβασε τους συντελεστές a και b
της εξίσωσης $f(x)=ax+b$ και
υπολόγισε και τύπωσε την τιμή
της $f(x)$ για όλες τις τιμές του x
από 0.0 έως $+150.0$


```
#include <stdio.h>
#include <math.h>

main()
{
    float        a, b;
    float        x, y;
    int          value_from_scanf;

    value_from_scanf = scanf("%f %f", &a, &b);
    if ( value_from_scanf !=2 )
    {
        printf("Error. Expecting two numbers ...\n");
        exit(1);
    }

    for ( x=0.0 ; x < 150.0 ; x += 1.0 )
    {
        y = a*x + b;
        printf(" %f %f\n", x, y);
    }
}
```



```
#include <stdio.h>
```

```
#include <math.h>
```

```
main()
```

```
{
```

```
    float        a, b;
```

```
    float        x;
```

```
    if ( scanf("%f %f", &a, &b) !=2 )
```

```
    {
```

```
        printf("Error. Expecting two numbers ...\n");
```

```
        exit(1);
```

```
    }
```

```
    for ( x=0.0 ; x < 150.0 ; x += 1.0 )
```

```
        printf(" %+15.5f %+15.5f\n", x, (a*x+b));
```

```
}
```



```
#include <stdio.h>
#include <math.h>
```

```
main()
{
```

```
    float    a, b;
    float    x;
```


```
    if ( scanf("%f %f", &a, &b) !=2 )
    {
```

```
        printf("Error. Expecting two numbers ...\n");
        exit(1);
    }
```

```
    for ( x=0.0 ; x < 150.0 ; x += 1.0 )
        printf(" %+15.5f %+15.5f\n", x, (a*x+b));
```

```
}
```

```
value_from_scanf = scanf("%f %f", &a, &b);
if ( value_from_scanf !=2 )
```




```
#include <stdio.h>
#include <math.h>
```

```
main()
{
```

```
    float    a, b;
    float    x;
```

```
    if ( scanf("%f %f", &a, &b) !=2 )
    {
```

```
        printf("Error. Expecting two numbers ...\n");
        exit(1);
    }
```

```
y = a*x + b;
printf(" %f %f\n", x, y);
```

```
    for ( x=0.0 ; x < 150.0 ; x += 1.0 )
        printf(" %+15.5f %+15.5f\n", x, (a*x+b));
```

```
}
```



```
#include <stdio.h>
#include <math.h>

main()
{

    float        a, b;
    float        x;

    if ( scanf("%f %f", &a, &b) !=2 )
    {
        printf("Error. Expecting two numbers ...\n");
        exit(1);
    }

    for ( x=0.0 ; x < 150.0 ; x += 1.0 )
        printf("  %+15.5f  %+15.5f\n", x, (a*x+b));
}
```

Τι είναι αυτό το +15.5 ανάμεσα στο % και το f ;


```
printf(" %f %f\n", x, y);
```

```
-7.500000 -54453.906250  
-7.000000 -37477.000000  
-6.500000 -24969.718750  
-6.000000 -15994.000000  
-5.500000 -9752.031250  
-5.000000 -5575.000000  
-4.500000 -2911.843750  
-4.000000 -1318.000000  
-3.500000 -444.156250  
-3.000000 -25.000000  
-2.500000 132.031250  
-2.000000 158.000000  
-1.500000 133.718750  
-1.000000 101.000000  
-0.500000 73.906250  
0.000000 50.000000  
0.500000 21.593750  
1.000000 -13.000000  
1.500000 -38.218750  
2.000000 -10.000000  
2.500000 155.468750  
3.000000 593.000000  
3.500000 1499.656250  
4.000000 3146.000000  
4.500000 5887.343750  
5.000000 10175.000000  
5.500000 16567.531250  
6.000000 25742.000000
```



```
printf(" %+15.5f %+15.5f\n", x, y);
```

-12.50000	-151885.78125
-10.00000	-113220.00000
-7.50000	-82878.59375
-7.00000	-59436.00000
-6.50000	-41639.90625
-6.00000	-28400.00000
-5.50000	-18776.71875
-5.00000	-11970.00000
-4.50000	-7308.03125
-4.00000	-4236.00000
-3.50000	-2304.84375
-3.00000	-1160.00000
-2.50000	-530.15625
-2.00000	-216.00000
-1.50000	-78.96875
-1.00000	-30.00000
-0.50000	-18.28125
+0.00000	-20.00000
+0.50000	-27.09375
+1.00000	-36.00000
+1.50000	-36.40625
+2.00000	+0.00000
+2.50000	+130.78125
+3.00000	+454.00000
+3.50000	+1119.46875
+4.00000	+2340.00000
+4.50000	+4402.65625
+5.00000	+7680.00000


```
printf(" %+15.5f %+15.5f\n", x, y);
```

-12.50000	-151885.78125
-10.00000	-113220.00000
-7.50000	-82878.59375
-7.00000	-59436.00000
-6.50000	-41639.90625
-6.00000	-28400.00000
-5.50000	-18776.71875
-5.00000	-11970.00000
-4.50000	-7308.03125
-4.00000	-4236.00000
-3.50000	-2304.84375
-3.00000	-1160.00000
-2.50000	-530.15625
-2.00000	-216.00000
-1.50000	-78.96875
-1.00000	-30.00000
-0.50000	-18.28125
+0.00000	-20.00000
+0.50000	-27.09375
+1.00000	-36.00000
+1.50000	-36.40625
+2.00000	+0.00000
+2.50000	+130.78125
+3.00000	+454.00000
+3.50000	+1119.46875
+4.00000	+2340.00000
+4.50000	+4402.65625
+5.00000	+7680.00000

+1540133.34567

-32.00030

----- . -----

5

15

Διάβασε τους συντελεστές μιας
εξίσωσης 5ου βαθμού και
υπολόγισε και τύπωσε την τιμή
της $f(x)$ για όλες τις τιμές του x
από -50.0 έως $+50.0$


```

#include <stdio.h>
#include <math.h>

main()
{

    float          a, b, c, d, e, f;
    float          x, y;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
    {
        printf("Error. Expecting six numbers ...\n");
        exit(1);
    }

    for ( x= -50.0 ; x <= 50.0 ; x += 0.50 )
    {
        y = a*x*x*x*x*x + b*x*x*x*x + c*x*x*x + d*x*x + e*x +f;
        printf(" %+15.5f %+15.5f\n", x, y);
    }
}

```



```
#include <stdio.h>
#include <math.h>

main()
{

    float          a, b, c, d, e, f;
    float          x, y;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
    {
        printf("Error. Expecting six numbers ...\n");
        exit(1);
    }

    for ( x= -50.0 ; x <= 50.0 ; x += 0.50 )
    {
        y = a*x*x*x*x*x + b*x*x*x*x + c*x*x*x + d*x*x + e*x +f;
        printf(" %+15.5f %+15.5f\n", x, y);
    }
}
```



```

#include <stdio.h>
#include <math.h>

#define LIMIT    50.0
#define STEP     0.50

main()
{

    float  a, b, c, d, e, f;
    float  x, y;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
    {
        printf("Error. Expecting six numbers ...\n");
        exit(1);
    }

    for ( x= -LIMIT ; x <= LIMIT ; x += STEP )
    {
        y = a*x*x*x*x*x + b*x*x*x*x + c*x*x*x + d*x*x + e*x +f;
        printf(" %15.5f %15.5f\n", x, y);
    }
}

```



```

#include <stdio.h>
#include <math.h>

#define LIMIT    50.0
#define STEP     0.50

main()
{

    float  a, b, c, d, e, f;
    float  x, y;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
    {
        printf("Error. Expecting six numbers ...\n");
        exit(1);
    }

    for ( x= -LIMIT ; x <= LIMIT ; x += STEP )
    {
        y = a*x*x*x*x*x + b*x*x*x*x + c*x*x*x + d*x*x + e*x + f;
        printf(" %+15.5f %+15.5f\n", x, y);
    }
}

```

Τα LIMIT και STEP *δεν είναι* μεταβλητές του προγράμματος. Είναι συμβολικά ονόματα για αριθμητικές σταθερές που χρησιμοποιούνται από το πρόγραμμα.

Που είμαστε ;

Τύποι :

int

float

Εντολές :

if - else

for

while

Συναρτήσεις :

printf()

scanf()

Τι μας λείπει ;

Εντολές	<code>for</code> <code>while</code> <code>if</code> και <code>if-else</code>
Συναρτήσεις εισόδου/εξόδου	<code>printf()</code> <code>scanf()</code>
Τύποι μεταβλητών	<code>int</code> <code>float</code> <code>char</code>

Αλλά πριν από αυτό ...

Συναρτήσεις :

Οι συναρτήσεις είναι (για τα πλαίσια αυτού του μαθήματος) αυτόνομα και αυτοδύναμα κομμάτια κώδικα τα οποία εκτελούν μια καλά καθορισμένη διαδικασία (υπολογισμό). Για την ενεργοποίηση των συναρτήσεων (δηλ. για την εκτέλεση των υπολογισμών που κωδικοποιούν) αρκεί η αναφορά του ονόματος τους μαζί με τις τυχόν παραμέτρους που απαιτούνται.

Συναρτήσεις :

Παραδείγματα συναρτήσεων που έχετε ήδη συναντήσει :

printf() : Μορφοποίηση και εκτύπωση στην καθιερωμένη έξοδο.

sqrt() : Υπολογισμός της τετραγωνικής ρίζας.

scanf() : Ανάγνωση και μετατροπή της καθιερωμένης εισόδου.

Παράδειγμα για την χρήση
συναρτήσεων : Διάβασε τους
συντελεστές μιας εξίσωσης 5ου
βαθμού και για κάθε τιμή x που
ορίζει ο χρήστης υπολόγισε και
τύπωσε την τιμή της $f(x)$.

Enter values for a,b,c,d,e,f : 1.0 2.0 -3 -4 -5 -6

Now enter values for x. Finish with <CTRL-D>

x=1

y= -15.0000

x=1.1

y= -15.7943

x=0

y= -6.0000

x=12.2

y= 308467.5938

. . . .


```

#include <stdio.h>
#include <math.h>

float    poly_5th();

main()
{
    float    a, b, c, d, e, f;
    float    x;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
        {
            printf("Error. Expecting six numbers ...\n");
            exit(1);
        }

    printf("Now enter values for x. Finish with <CTRL-D>\n");
    printf("x=");
    while( scanf("%f", &x) == 1 )
        {
            printf("y=%17.4f\n", poly_5th(x,a,b,c,d,e,f) );
            printf("x=");
        }
}

float    poly_5th( x, A, B, C, D, E, F)
float    x, A, B, C, D, E, F;
{
    float result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
    return( result );
}

```



```
#include <stdio.h>
```

```
#include <math.h>
```

```
float poly_5th();
```

```
main()
```

```
{
```

```
    float a, b, c, d, e, f;
```

```
    float x;
```

```
    printf("Enter values for a,b,c,d,e,f : ");
```

```
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
```

```
        {          printf("Error. Expecting six numbers ...\n");
```

```
            exit(1);
```

```
        }
```

```
    printf("Now enter values for x. Finish with <CTRL-D>\n");
```

```
    printf("x=");
```

```
    while( scanf("%f", &x) == 1 )
```

```
        {          printf("y=%17.4f\n", poly_5th(x,a,b,c,d,e,f) );
```

```
            printf("x=");
```

```
        }
```

```
}
```

```
float poly_5th( x, A, B, C, D, E, F)
```

```
float x, A, B, C, D, E, F;
```

```
{
```

```
    float result;
```

```
    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
```

```
    return( result );
```

```
}
```

Δήλωση του ονόματος της συνάρτησης και του τύπου (float) της τιμής που επιστρέφει.


```

#include <stdio.h>
#include <math.h>

float    poly_5th();

main()
{
    float  a, b, c, d, e, f;
    float  x;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
        {
            printf("Error. Expecting six numbers ...\n");
            exit(1);
        }


    printf("Now enter values for x. Finish with <CTRL-D>\n");
    printf("x=");
    while( scanf("%f", &x) == 1 )
        {
            printf("y=%17.4f\n", poly_5th(x,a,b,c,d,e,f) );
            printf("x=");
        }
}

float    poly_5th( x, A, B, C, D, E, F)
float    x, A, B, C, D, E, F;
{
    float result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
    return( result );
}

```

Κλήση της συνάρτησης με
παραμέτρους τις τιμές των
x, a, b, c, d, e, f.




```

#include <stdio.h>
#include <math.h>

float    poly_5th();

main()
{
    float  a, b, c, d, e, f;
    float  x;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
        {
            printf("Error. Expecting six numbers ...\n");
            exit(1);
        }

    printf("Now enter values for x. Finish with <CTRL-D>\n");
    printf("x=");
    while( scanf("%f", &x) == 1 )
        {
            printf("y=%17.4f\n", poly_5th(x,a,b,c,d,e,f) );
            printf("x=");
        }
}

float    poly_5th( x, A, B, C, D, E, F)
float    x, A, B, C, D, E, F;
{
    float result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
    return( result );
}

```

Αυτή καθ' αυτή η συνάρτηση ...

... η οποία είναι εκτός των ορίων της main()

```
#include <stdio.h>
#include <math.h>

float    poly_5th();

main()
{
    float  a, b, c, d, e, f;
    float  x;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
        {
            printf("Error. Expecting six numbers ...\n");
            exit(1);
        }

    printf("Now enter values for x. Finish with <CTRL-D>\n");
    printf("x=");
    while( scanf("%f", &x) == 1 )
        {
            printf("y=%17.4f\n", poly_5th(x,a,b,c,d,e,f) );
            printf("x=");
        }
}

float    poly_5th( x, A, B, C, D, E, F)
float    x, A, B, C, D, E, F;
{
    float result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
    return( result );
}
```


Το όνομα της συνάρτησης

```
float poly_5th( x, A, B, C, D, E, F)
```

```
float x, A, B, C, D, E, F;
```

```
{
```

```
    float    result;
```

```
    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x + F;
```

```
    return( result );
```

```
}
```


Ο τύπος της αριθμητικής τιμής που επιστρέφει.



```
float poly_5th( x, A, B, C, D, E, F)
```

```
float x, A, B, C, D, E, F;
```

```
{
```

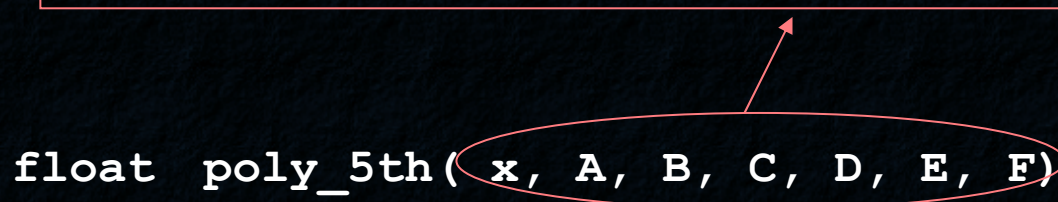
```
    float    result;
```

```
    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
```

```
    return( result );
```

```
}
```


Τα ονόματα των μεταβλητών μέσω των οποίων η συνάρτηση θα μπορεί να χρησιμοποιεί τις τιμές που της δόθηκαν [κατά την κλήση της από το main()].



```
float poly_5th( x, A, B, C, D, E, F)
float x, A, B, C, D, E, F;
{
    float      result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
    return( result );
}
```


Τα ονόματα των μεταβλητών μέσω των οποίων η συνάρτηση θα μπορεί να χρησιμοποιεί τις τιμές που της δόθηκαν [κατά την κλήση της από το main()].

```
float poly_5th( x, A, B, C, D, E, F)
float x, A, B, C, D, E, F;
{
    float    result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
    return( result );
}
```

Οι μεταβλητές αυτές είναι γνωστές μόνο στη συνάρτηση (και όχι στο υπόλοιπο πρόγραμμα ή άλλες συναρτήσεις). Τα ίδια ονόματα μεταβλητών μπορούν να χρησιμοποιούνται και αλλού χωρίς να υπάρχει μεταξύ τους αλληλεπίδραση.

Τα ονόματα των μεταβλητών μέσω των οποίων η συνάρτηση θα μπορεί να χρησιμοποιεί τις τιμές που της δόθηκαν [κατά την κλήση της από το `main()`].

```
float poly_5th( x, A, B, C, D, E, F)
float x, A, B, C, D, E, F;
{
    float    result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x + F;
    return( result );
}
```

Αυτό που δίδεται στη συνάρτηση (κατά την κλήση της) είναι ένα αντίγραφο των τιμών των μεταβλητών και όχι οι μεταβλητές αυτές καθ' αυτές. Για αυτό το λόγο, η συνάρτηση δεν μπορεί να τροποποιήσει τις τιμές των μεταβλητών που χρησιμοποιεί το τμήμα του προγράμματος που την κάλεσε.

Τα ονόματα των μεταβλητών μέσω των οποίων η συνάρτηση θα μπορεί να χρησιμοποιεί τις τιμές που της δόθηκαν [κατά την κλήση της από το main()].

```
float poly_5th(x, A, B, C, D, E, F)
float x, A, B, C, D, E, F;
{
    float    result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x + F;
    return( result );
}
```

Αυτό που δίδεται στη συνάρτηση (κατά την κλήση της) είναι ένα αντίγραφο των τιμών των μεταβλητών και όχι οι μεταβλητές αυτές καθ' αυτές. Για αυτό το λόγο, η συνάρτηση δεν μπορεί να τροποποιήσει τις τιμές των μεταβλητών που χρησιμοποιεί το τμήμα του προγράμματος που την κάλεσε. **ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ scanf() ΚΑΙ ΤΟ ΝΟΗΜΑ ΤΩΝ «&»**

Οι τύποι των τιμών των μεταβλητών που δόθηκαν στη συνάρτηση κατά την κλήση της.

```
float poly_5th( x, A, B, C, D, E, F)
```

```
float x, A, B, C, D, E, F;
```

```
{
```

```
    float    result;
```

```
    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x + F;
```

```
    return( result );
```

```
}
```


Δήλωση μιας μεταβλητής (με το όνομα result), τύπου κινητής υποδιαστολής. Και αυτή η δήλωση έχει μόνο τοπική εμβέλεια (υφίσταται μόνο για αυτή τη συνάρτηση).

```
float poly_5th( x, A, B, C, D, E, F)
float x, A, B, C, D, E, F;
{
```

```
    float    result;
```

```
    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x + F;
    return( result );
}
```


Ανάθεση τιμής στη μεταβλητή `result` χρησιμοποιώντας τις τιμές των μεταβλητών που δόθηκαν στη συνάρτηση.

```
float poly_5th( x, A, B, C, D, E, F)
```

```
float x, A, B, C, D, E, F;
```

```
{
```

```
    float    result;
```

```
    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x + F;
```

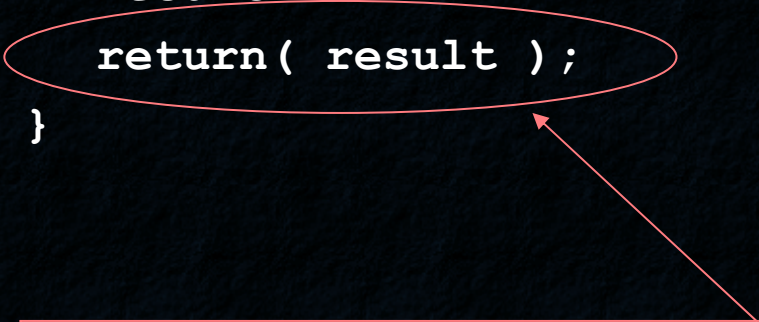
```
    return( result );
```

```
}
```



```
float poly_5th( x, A, B, C, D, E, F)
float x, A, B, C, D, E, F;
{
    float    result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
    return( result );
}
```



Τέλος της κλήσης της συνάρτησης. Η επιστρεφόμενη τιμή είναι ίση με την τιμή της μεταβλητής result.


```

#include <stdio.h>
#include <math.h>

float    poly_5th();

main()
{
    float    a, b, c, d, e, f;
    float    x;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
        {
            printf("Error. Expecting six numbers ...\n");
            exit(1);
        }

    printf("Now enter values for x. Finish with <CTRL-D>\n");
    printf("x=");
    while( scanf("%f", &x) == 1 )
        {
            printf("y=%17.4f\n", poly_5th(x,a,b,c,d,e,f) );
            printf("x=");
        }
}

float    poly_5th( x, A, B, C, D, E, F)
float    x, A, B, C, D, E, F;
{
    float result;

    result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F;
    return( result );
}

```

Μπορούμε να κάνουμε το πρόγραμμα πιο συμπαγές ;


```

#include <stdio.h>
#include <math.h>

float    poly_5th();

main()
{
    float    a, b, c, d, e, f;
    float    x;

    printf("Enter values for a,b,c,d,e,f : ");
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
        {
            printf("Error. Expecting six numbers ...\n");
            exit(1);
        }

    printf("Now enter values for x. Finish with <CTRL-D>\n");
    printf("x=");
    while( scanf("%f", &x) == 1 )
        printf("y=%17.4f\nx=", poly_5th(x,a,b,c,d,e,f) );
}

float    poly_5th( x, A, B, C, D, E, F)
float    x, A, B, C, D, E, F;
{
    return( A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F);
}

```



```
#include <stdio.h>
```

```
#include <math.h>
```

```
float    poly_5th();
```

```
main()
```

```
{
```

```
    float  a, b, c, d, e, f;
```

```
    float  x;
```

```
    printf("Enter values for a,b,c,d,e,f : ");
```

```
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
```

```
        {          printf("Error. Expecting six numbers ...\n");
```

```
            exit(1);
```

```
        }
```

```
    printf("Now enter values for x. Finish with <CTRL-D>\n");
```

```
    printf("x=");
```

```
    while( scanf("%f", &x) == 1 )
```

```
        printf("y=%17.4f\nx=", poly_5th(x,a,b,c,d,e,f) );
```

```
}
```

```
float    poly_5th( x, A, B, C, D, E, F)
```

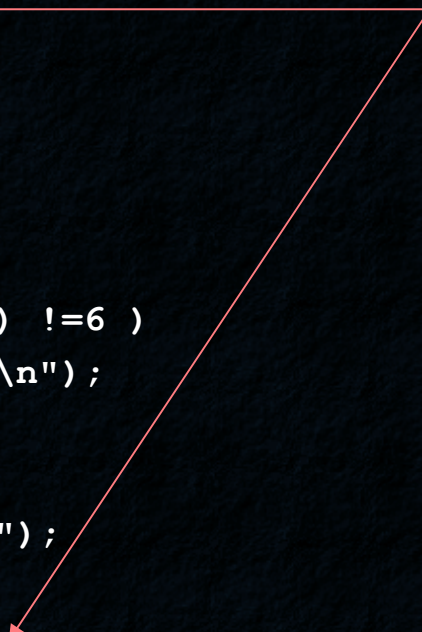
```
float    x, A, B, C, D, E, F;
```

```
{
```

```
    return( A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x +F);
```

```
}
```

```
{  
    printf("y=%17.4f\n", poly_5th(x,a,b,c,d,e,f) );  
    printf("x=");  
}
```




```
#include <stdio.h>
#include <math.h>
```

```
float    poly_5th();
```

```
float    result;
result = A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x + F;
return( result );
```

```
main()
```

```
{
```

```
    float    a, b, c, d, e, f;
```

```
    float    x;
```

```
    printf("Enter values for a,b,c,d,e,f : ");
```

```
    if ( scanf("%f %f %f %f %f %f", &a, &b, &c, &d, &e, &f) !=6 )
```

```
        {          printf("Error. Expecting six numbers ...\n");
```

```
            exit(1);
```

```
        }
```

```
    printf("Now enter values for x. Finish with <CTRL-D>\n");
```

```
    printf("x=");
```

```
    while( scanf("%f", &x) == 1 )
```

```
        printf("y=%17.4f\nx=", poly_5th(x,a,b,c,d,e,f) );
```

```
}
```

```
float    poly_5th( x, A, B, C, D, E, F)
```

```
float    x, A, B, C, D, E, F;
```

```
{
```

```
    return( A*x*x*x*x*x + B*x*x*x*x + C*x*x*x + D*x*x + E*x + F);
```

```
}
```


Τι μας λείπει ;

Εντολές	<code>for</code> <code>while</code> <code>if</code> και <code>if-else</code>
Συναρτήσεις εισόδου/εξόδου	<code>printf()</code> <code>scanf()</code>
Τύποι μεταβλητών	<code>int</code> <code>float</code> <code>char</code>