

Προγραμματισμός 1 –μάθημα 3^ο

Δομές επανάληψης.

Οι δομές επανάληψης χρησιμοποιούνται όταν πρέπει κάποιες εντολές –γραμμές κώδικα να επαναλαμβάνονται. Διακρίνονται στις παρακάτω:

1. **δομή for** πχ for (i=0;i<=4;i++)

```
{  
    εντολές  
}
```

Το πρόγραμμα μετράει από το 0 μέχρι το 4 και αυξάνεται κάθε φορά κατά ένα με αποτέλεσμα η μεταβλητή i να παίρνει διαδοχικά τις τιμές 0, 1, 2, 3 και 4. **Αν γράψουμε** for (i=4;i>=0;i--) , τότε ο βρόχος θα μετράει από το 4 μέχρι το 0, ελαττώνοντας τον μετρητή κατά ένα κάθε φορά. Η επανάληψη εδώ γίνεται όσο η συνθήκη είναι αληθής

2. **δομή while** πχ i=0; while (i<=4)

```
{  
    εντολές  
    i++;  
}
```

Ο βρόχος While επαναλαμβάνεται όσο η μεταβλητή i είναι μικρότερη ή ίση του 4, δηλ. η επανάληψη θα συμβεί για τις τιμές της μεταβλητής 0,1,2,3,4. Μόλις η μεταβλητή γίνει ίση με 5, θα βγούμε από τον βρόχο. Η επανάληψη εδώ γίνεται όσο η συνθήκη στη while είναι αληθής

3. **δομή do while()**

```
πχ i:=0  
do  
{  
    εντολές  
    i++;  
}  
while(i<=4);
```

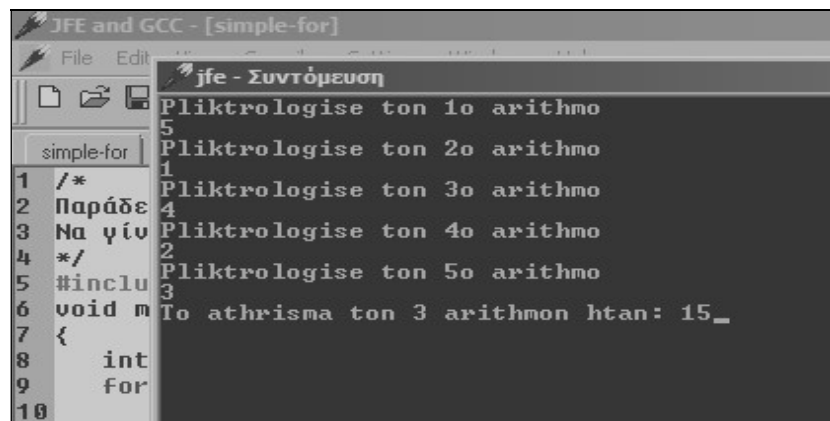
Και εδώ η επανάληψη γίνεται όσο η συνθήκη στη while είναι αληθής. Όμως η εντολή do while εκτελεί το κύριο σώμα του βρόχου τουλάχιστον μία φορά, ακόμα και αν δεν ισχύει η συνθήκη επανάληψης του βρόχου, καθώς η συνθήκη αυτή ελέγχεται στο τέλος του βρόχου και αφού έχουν εκτελεστεί οι εντολές του σώματος του βρόχου οπωσδήποτε μία φορά. Στο παράδειγμα εδώ του λέμε να κάνει επαναλήψεις όσο το i είναι μικρότερο ή ίσο του 4. Αν του δίναμε από την αρχή αρχική τιμή i=5 θα έκανε μία και μόνο μία φορά τις εντολές του μπλοκ της επανάληψης και μετά θα έβγαινε έξω από την επανάληψη.

Στη C όλες οι δομές επανάληψης χρησιμοποιούνται τόσο για γνωστό αριθμό επαναλήψεων όσο και για άγνωστο αριθμό επαναλήψεων.

Παράδειγμα με δομή επανάληψης for.

Να γίνει πρόγραμμα που να δέχεται 5 αριθμούς από το πληκτρολόγιο και να τους αθροίζει και να εμφανίζει το αποτέλεσμα.

```
#include<stdio.h>
void main()
{
    int x,i,sum=0;
    for(i=1;i<=5;i++)
    {
        printf("Pliktrologise ton %do arithmo\n",i);
        scanf("%d",&x);
        sum=sum+x;
    }
    printf("To athrisma ton %d arithmon htan: %d",i-1,sum);
}
```



Πώς θα γινόταν το παραπάνω πρόβλημα με δομή επανάληψης while?

```
i=1;
while(i<=5)
{
    εντολες....
    i=i+1;//i++
}

sum=0;
i=1;
while(i<=5)
{
    printf("Pliktrologise ton %do arithmo\n",i);
    scanf("%d",&x);
    sum=sum+x;
    i++;
}

printf("To athrisma ton %d arithmon htan: %d",--i,sum);
```

Το ίδιο πρόγραμμα με δομή επανάληψης do-while θα γινόταν ως εξής:

```
sum=0;
i=1;
do
{
    printf("Pliktrologise ton %do arithmo\n",i);
    scanf("%d",&x);
    sum=sum+x;
    i++;
}
while(i<=5);
printf("To athrisma ton %d arithmon htan: %d\n",--
i,sum);
}
```

Αν στο παραπάνω πρόγραμμα το i ξεκινούσε με αρχική τιμή όχι την τιμή 1, αλλά την τιμή 6 τί θα γινόταν;

Να γίνει πρόγραμμα που να δέχεται αριθμούς από το πληκτρολόγιο , να μετράει το πλήθος τους και να υπολογίζει το γινόμενό τους . Όταν το γινόμενο γίνει μεγαλύτερο ή ίσο με 1000 το πρόγραμμα να σταματάει και να εμφανίζει το πλήθος των αριθμών που πληκτρολογήθηκαν καθώς και το γινόμενο που υπολογίστηκε.

```
#include<stdio.h>
void main()
int number,ginomeno,plithos=0;
ginomeno=1;
do {
    printf("Pliktrologise arithmo\n");
    scanf("%d", &number);
    plithos++;
    ginomeno=ginomeno*number;
}
while (ginomeno<1000);
printf("Pliktrologises  %d arithmous me ginomeno
%d",plithos,ginomeno);*/

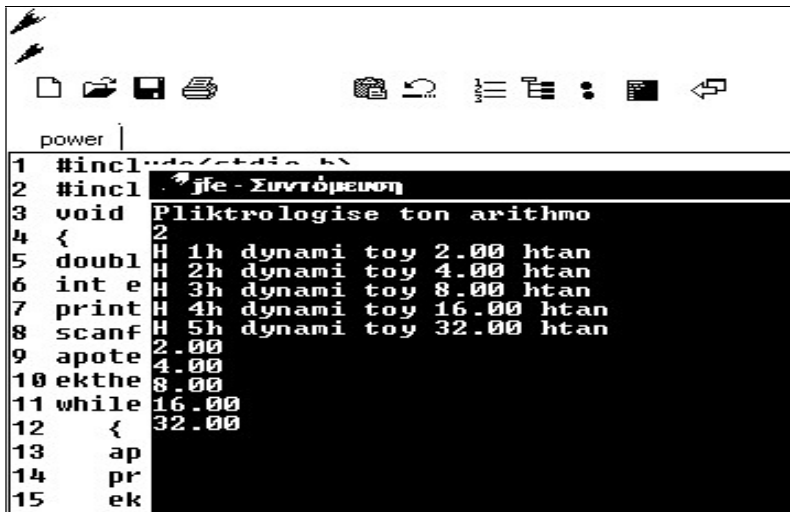
/*ΠΩΣ ΘΑ ΓΙΝΟΤΑΝ ΜΕ WHILE ΤΟ ΠΡΟΓΡΑΜΜΑ;*/
plithos=0;
ginomeno=1;
while (ginomeno<1000)
{
    printf("Pliktrologise arithmo\n");
    scanf("%d", &number);
    plithos++;
    ginomeno=ginomeno*number;
}
printf("Pliktrologises  %d arithmous me ginomeno
%d",plithos,ginomeno);
}
```

Τι θα κάνει το πρόγραμμα αν ο χρήστης πληκτρολογήσει από την αρχή την τιμή 1232 και γιατί;

.....

Να γίνει πρόγραμμα που να εμφανίζει τις 5 πρώτες δυνάμεις ενός αριθμού . Δηλαδή αν είναι x ο αριθμός να εμφανίσει x^1, x^2, x^3, x^4, x^5 .

```
#include<stdio.h>
#include<math.h>
void main(){
double x,apotelesma;
int ekthetis;
printf("Pliktrologise ton arithmo\n");
scanf("%lf",&x);
apotelesma=1;//an ekthetis =0
ekthetis=1;
while(ekthetis<=5)
{
    apotelesma=x*apotelesma;
    printf("H %dh dynami toy %.2lf htan \n",ekthetis,apotelesma);
    ekthetis++;
}
//enallaktika
ekthetis=1;
while(ekthetis<=5)
{
    printf("%.2lf\n",pow(x,ekthetis));
    ekthetis++;
}
}
```



```
power |
1 #include<stdio.h>
2 #include<math.h>
3 void main(){
4 {
5     double x,apotelesma;
6     int ekthetis;
7     printf("Pliktrologise ton arithmo\n");
8     scanf("%lf",&x);
9     apotelesma=1;
10    ekthetis=1;
11    while(ekthetis<=5)
12    {
13        apotelesma=x*apotelesma;
14        printf("H %dh dynami toy %.2lf htan \n",ekthetis,apotelesma);
15        ekthetis++;
16    }
17 }
```

Πως θα άλλαζε η συνθήκη επανάληψης για την αντίστοιχη εντολή do while;

.....

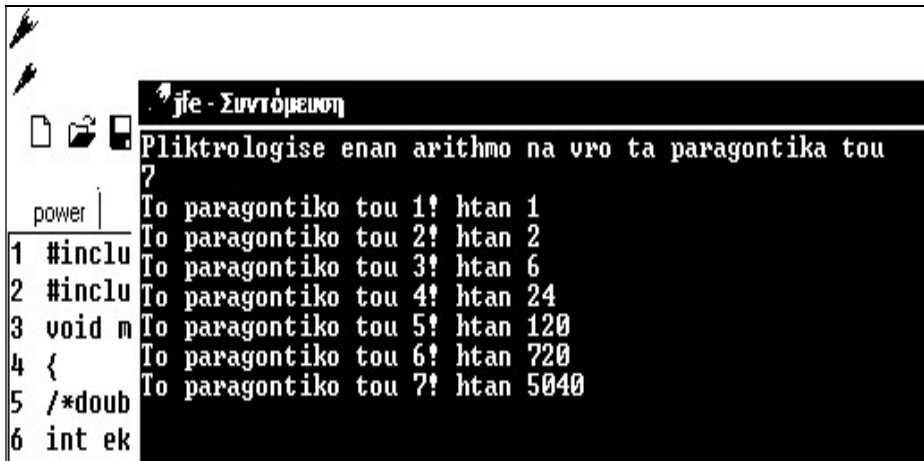
Να γίνει πρόγραμμα που να δέχεται έναν αριθμό και να εμφανίζει τα παραγοντικά του.

```
#include<stdio.h>
#include<math.h>
void main() {

int paragontiko,arithmos,fores;
paragontiko=1;
printf("Pliktrologise enan arithmo na vro ta paragontika tou\n");
scanf("%d",&arithmos);
for(fores=1;fores<=arithmos;fores++)
{
    paragontiko=paragontiko*fores;
    printf("To paragontiko tou %d! htan %d\n",fores ,paragontiko);

}

}
```



```
je - Συντόμηση
Pliktrologise enan arithmo na vro ta paragontika tou
?
To paragontiko tou 1! htan 1
To paragontiko tou 2! htan 2
To paragontiko tou 3! htan 6
To paragontiko tou 4! htan 24
To paragontiko tou 5! htan 120
To paragontiko tou 6! htan 720
To paragontiko tou 7! htan 5040
```

Πώς θα διορθώναμε το παραπάνω πρόγραμμα να μην δέχεται εύρεση παραγοντικού για αριθμό αρνητικό;

Να γίνει πρόγραμμα που να βρίσκει και να εμφανίζει τους ν πρώτους αριθμούς που βρίσκονται στο διάστημα 1 έως 25. Υπενθυμίζεται ότι ένας αριθμός λέγεται πρώτος όταν διαιρείται ακριβώς με τον εαυτό του και τη μονάδα και με κανέναν άλλον μικρότερο του. (Πχ Ο 1 ναι, ο 2 ναι, ο 3 ναι, ο 4 όχι, ο 5 ναι, ο 6 όχι γιατί διαιρείται με τον 2 και τον 3 ακριβώς, ο 7 ναι, ο 8 όχι, ο 9 όχι γιατί διαιρείται με το 3 κτλ)

```
#include<stdio.h>

void main() {

int arithmos,protos,arxi,plithos=1;
do
    {
        protos=1;
        arithmos=plithos;
        for(arxi=2;arxi<arithmos;arxi++)
            {
                if (arithmos%arxi==0)
                {
                    protos=0;
                    break;
                }
            }
        if (protos==1)
        {
            printf("Ο αριθμος %d είναι protos\n",arithmos);
        }
        else
        {
            printf("Ο αριθμος %d ΔΕΝ είναι
protos\n",arithmos);
        }
        plithos++;
    }
while(plithos<=25);

}
```



```
power |
27 scanf(
28 for(fo
29 {
30 par
31 pri
32 }
33 */
34 //p
35 int ar
36 do
37 {
38 //prin
39 //scan
40 protos
41 arithm
42 for(ar
43 {
44
45
46
47
48
49
```

jfe - Συντόμηση

0	arithmos	1	einai	protos	
0	arithmos	2	einai	protos	
0	arithmos	3	einai	protos	
0	arithmos	4	DEN	einai	protos
0	arithmos	5	einai	protos	
0	arithmos	6	DEN	einai	protos
0	arithmos	7	einai	protos	
0	arithmos	8	DEN	einai	protos
0	arithmos	9	DEN	einai	protos
0	arithmos	10	DEN	einai	protos
0	arithmos	11	einai	protos	
0	arithmos	12	DEN	einai	protos
0	arithmos	13	einai	protos	
0	arithmos	14	DEN	einai	protos
0	arithmos	15	DEN	einai	protos
0	arithmos	16	DEN	einai	protos
0	arithmos	17	einai	protos	
0	arithmos	18	DEN	einai	protos
0	arithmos	19	einai	protos	
0	arithmos	20	DEN	einai	protos
0	arithmos	21	DEN	einai	protos
0	arithmos	22	DEN	einai	protos
0	arithmos	23	einai	protos	
0	arithmos	24	DEN	einai	protos
0	arithmos	25	DEN	einai	protos