

Δομημένος Προγραμματισμός

Τμήμα Επιχειρηματικού Σχεδιασμού και
Πληροφοριακών Συστημάτων

www.bpis.teicrete.gr

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

while

Άσκηση

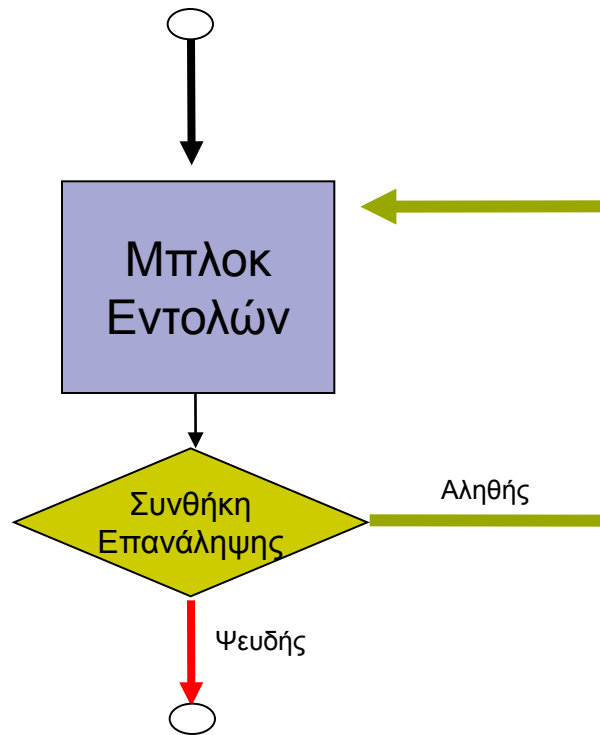
Γράψτε ένα πρόγραμμα που να διαβάζει αριθμούς από το πληκτρολόγιο και να τους εμφανίζει στην οθόνη, μέχρι ο χρήστης να εισάγει τον αριθμό 0
Χρησιμοποιήστε την εντολή `while`

```
int x;  
scanf( "%d", &x );  
while ( x != 0 )  
{  
    printf( "%d\n", x );  
    scanf( "%d", &x );  
}
```

Βρόχος dowhile

- Βρόχος με συνθήκη εξόδου
- Δυνατότητα καθοδήγησης με μετρητή ή γεγονός
- Διαφοροποιείται από την while στο σημείο ελέγχου της συνθήκης τερματισμού καθώς και στο ότι επιτρέπει την εκτέλεση της πρώτης επανάληψης πριν προχωρήσει στον έλεγχο της συνθήκης
- Δεν προαπαιτεί να είναι η συνθήκη αληθής πριν από την εκτέλεση του βρόχου.
- Πρέπει βέβαια να υπάρχει δυνατότητα μέσα στο μπλόκ εντολών να μετατρέπεται η συνθήκη σε ψευδής για να μπορεί να τερματισθεί ο βρόχος

do-while



Βρόχος dowhile στην C - Δήλωση & λειτουργία

```
int count = 30;  
int limit = 40;  
do  
{  
    count++;  
    printf( "count is %d\n", count );  
}  
while( count < limit );
```

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

dowhile

Άσκηση

Να βρεθεί το άθροισμα

1+2+3 100

με χρήση της εντολής do-while

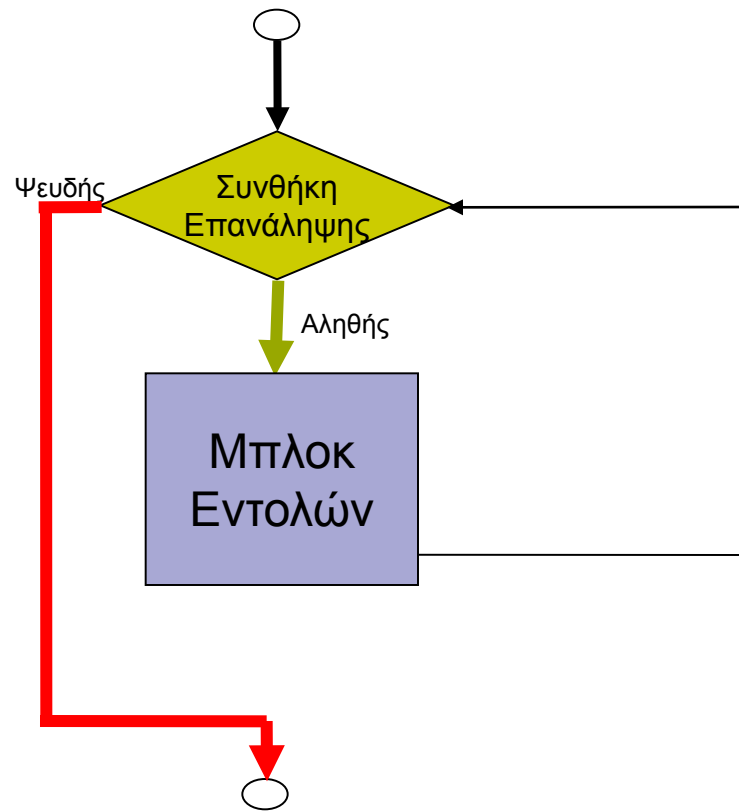
```
int i = 1;
int res = 0;
do
{
    res += i;
    i++;
} while( i <= 101 );

printf("\nΤο αθροισμα είναι : %d",res);
```

Βρόχος οδηγούμενος από μετρητή

- Χρησιμοποιεί έναν μετρητή
- Έχει αρχική συνθήκη
- Οδηγείται από μετρητή
(ο μετρητής μέλος στην συνθήκη)
- Έχει Δυνατότητα ρύθμισης προσαύξησης θετική – αρνητική
- Παρέχει την δυνατότητα διαβάθμισης προσαύξησης (βήμα)

for



Βρόχος οδηγούμενος από μετρητή στην C

```
for( αρχική τιμή μετρητή; συνθήκη; βήμα μετρητή )  
{  
    προτάσεις;  
}
```

Όταν οι προτάσεις ή εκφράσεις ή εντολές είναι μόνο μία τα άγκιστρα μπορούν να παραλειφθούν

Εντολή for παράδειγμα

```
for (n=0; n<10; n=n+2)
{
    Ομάδα_εντολών
}
printf("Τελος epanalipshs");
```

Η μεταβλητή παίρνει μια αρχική τιμή ($n=0$).

Ελέγχεται η συνθήκη ($n<10$).

Αν η συνθήκη είναι αληθής τότε εκτελείται η ομάδα εντολών ανάμεσα στα άγκιστρα. Αυξάνεται η τιμή του μετρητή κατά το βήμα (κατά 2: $n=n+2$).

Ελέγχεται και πάλι η συνθήκη κοκ

Η διαδικασία αυτή πραγματοποιείται μέχρι η συνθήκη να γίνει ψευδής.

Μόλις η συνθήκη γίνει ψευδής οι εντολές μέσα στα άγκιστρα αγνοούνται και εκτελείται η επόμενη εντολή (printf).

Εντολή for παραδείγματα


```
for( n=0; n<10; n++ )  
    printf(“\n=%d\n”,n);
```

```
for( n=10; n>0; n-- )  
    printf(“n=%d\n”,n);
```

```
for( n=0; n<60; n=n+13 )  
    printf(“n=%d\n”,n);
```

```
for( n='a'; n<'z'; n++ )  
    printf(“ n=%d, the ASCII value of %c is %d\n”, n, n, n );
```

```
for( n=0; n<60.0; n=1.2*n )  
    printf(“n=%f\n”,n);
```



Εντολή for παραδείγματα

```
#include <stdio.h>
main()
{
    int count = 30;
    int max_count = 40;
    for( count=0; count < max_count; count++ )
    {
        printf( "count is %d\n", count );
        <άλλες προτάσεις>
    }
    ...
}
```

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

for

Άσκηση

Γράψτε ένα πρόγραμμα που να τυπώνει του αριθμούς από το **1 έως και το 10** χρησιμοποιώντας την δομή ελέγχου **for**

```
int i;  
for ( i=1; i<=10; i++ )  
{  
    printf ( “%d\n”, i );  
}
```

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

for2

Άσκηση

Γράψτε ένα πρόγραμμα που να προσθέτει τις
δεκάδες ($10+20+30+\dots+90+100$)
και να τυπώνει το τελικό άθροισμα
χρησιμοποιώντας την δομή ελέγχου **for**

```
int i;  
int sum = 0;  
for ( i=10; i<=100; i=i+10 )  
{  
    sum = sum + i;  
}  
printf ( "Το α8ροισμα einai : %d\n", sum );
```

Τελεστής , και χρήση σε loop

- Επιτρέπει την παράθεση περισσότερων της μίας έκφρασης σε θέσεις όπου επιτρέπεται μία έκφραση.
- Έχει περιορισμένη εφαρμογή
- Στην for χρησιμοποιείται ως συνθετικό των εκφράσεων αρχικοποίησης και ανανέωσης

```
int i, j;  
for ( i=0, j=10; i<8 ; i++, j++)  
    printf(" \n i=%d \t j=%d ", i, j );
```

Εντολή for παραδείγματα

```
#include <stdio.h>
main()
{
    int x,y;
    for( x=0,y=0; x+y<100; x=x+20, y=y+10 )
        printf("x=%d\ty=%d\tx+y=%d\n",x,y,x+y);

    printf("x=%d\ty=%d",x,y);
}
```

Μετασχηματισμός βρόχων for while

```
for( αρχική τιμή μετρητή ; συνθήκη ; βήμα μετρητή )  
{  
    Προτάσεις;  
}
```



```
αρχική τιμή μετρητή;  
  
while( συνθήκη )  
{  
    προτάσεις;  
    βήμα μετρητή;  
}
```

Μετασχηματισμός βρόχων for σε do-while

```
for(n=4; n>1; n--)  
    printf( “ %d ”, n);
```



```
n=4;  
do  
{  
    printf( “ %d ”, n);  
    n--;  
} while(n>1);
```

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

for-while

Άσκηση

Γράψτε ένα πρόγραμμα που να προσθέτει τις
δεκάδες ($10+20+30+\dots+90+100$)
και να τυπώνει το τελικό άθροισμα
χρησιμοποιώντας την δομή ελέγχου **while**

```
int i=0;
int sum=0;

while ( i<=100 )
{
    sum = sum + i;
    i = i + 10;
}
printf ( "Το αθροισμα είναι : %d\n", sum );
```

Άσκηση

Τροποποιήστε το πρόγραμμά σας χρησιμοποιώντας την δομή ελέγχου **do-while**

```
int i=0;
int sum=0;

do
{
    i = i + 10;
    sum = sum + i;
} while ( i<100 );

printf ( "Το αδροισμα είναι : %d\n", sum );
```

Διακοπτόμενοι βρόχοι στη C

- Ονομάζεται ο βρόχος στον οποίο η επανάληψη διακόπτεται για μεταφορά του ελέγχου
 - στην αρχή του βρόχου και έλεγχο της συνθήκης
 - εκτός βρόχου
- Υλοποίηση με την χρήση
 - Κωδικής λέξης **break**
 - Πρόταση **continue**
 - Πρόταση **goto**

Break

- Στους βρόχους η κωδική λέξη `break` έχει την σημασία του τερματισμού εκτέλεσης ενός βρόχου, μεταβιβάζοντας τον έλεγχο του προγράμματος στην εντολή που βρίσκεται αμέσως μετά το βρόχο

```
int t;
for( t=0; t<100; t++ )
{
    printf( “ \n %d ”, t );
    if ( t==10 )
        break;
}
printf( “ \nt = %d ”, t );
```

Άσκηση

Τροποποιήστε το πρόγραμμά σας
ώστε να προκύψει το άθροισμα μέχρι και το 50

χρησιμοποιήστε κατάλληλα την εντολή **break**

```
int i=0;
int sum=0;

do
{
    i = i + 10;
    sum = sum + i;
    if ( i == 50)
        break;

}
while ( i<100 );
printf ( "To a8roisma einai : %d\n", sum );
```

Continue

- Μεταφέρει τον έλεγχο της ροής στην αρχή του βρόχου, παραλείποντας την εκτέλεση του υπολοίπου τμήματος του σώματος του βρόχου και προχωρώντας στην επόμενη επανάληψη

```
int x;
for (x=0; x<100; x++)
{
    if ( x%2==0 )
        continue;
    printf( " \t %d ", x );
}
```

Άσκηση

Τροποποιήστε το πρόγραμμά σας
ώστε να προκύψει το άθροισμα μόνο των μονών
δεκάδων μέχρι και το 90
(10+30+50+...+70+90)
χρησιμοποιήστε κατάλληλα την εντολή **continue**

```
int i=0;
int sum=0;

do
{
    i = i + 10;
    if ( i % 2 == 0 )
        continue;
    sum = sum + i;
}
while ( i<100 );

printf ( "To a8roisma einai : %d\n", sum );
```

Goto

- Είναι η πρόταση ρητής διακλάδωσης
 - goto <ετικέτα>
- Μεταφέρει τον έλεγχο στην πρόταση που σημειώνεται με την ετικέτα ως
 - <ετικέτα>:πρόταση
- Αποφυγή χρήσης διότι διασπά τις αρχές του δομημένου προγραμματισμού δημιουργώντας επιπλοκές στην λογική ροή του προγράμματος
- Συνήθως χρησιμοποιείται όταν θέλουμε να διασπάσουμε το τμήμα του προγράμματος που εκτελείται μέσα σε πολλές ένθετες επαναλήψεις (βρόχους)

Παράδειγμα goto

```
loop:
```

```
    i++;  
    printf("%d ",i);
```

```
    if (i==10)  
        goto endloop;  
    goto loop;
```

```
endloop:
```



```
for (i=1;i<=10;i++)  
    printf("%d ",i);
```

Κανόνες χρήσης προτάσεων επανάληψης

- Σωστή χρήση στηλοθεσίας για εύκολη αναγνωσιμότητα κώδικα
- Χρησιμοποιείται { } για να περικλείετε περισσότερες της μίας πρότασης
- **Αποφεύγετε τη χρήση της πρότασης διακλάδωσης goto**
- Προτιμήστε το βρόχο επανάληψης με συνθήκη εισόδου (while) από τον αντίστοιχο με συνθήκη εξόδου (dowhile)

Κανόνες χρήσης προτάσεων επανάληψης

- Αποφεύγετε κατά το δυνατόν τη χρήση των **break** και **continue** σε βρόχους επανάληψης
- Εκτελέστε λεπτομερή έλεγχο και βεβαιωθείτε ότι κάθε συνθήκη βρόχου επανάληψης οδηγεί στην έξοδο μετά από πεπερασμένες επαναλήψεις, έτσι ώστε να μη δημιουργούνται ατέρμονοι βρόχοι

Ένθετοι βρόχοι (nested loops)

- Ονομάζεται ο βρόχος που περικλείεται σε έναν άλλο
- Ο εσωτερικός βρόχος λογίζεται ως μια πρόταση του εξωτερικού – δηλαδή πρώτα θα εκτελείται ο εσωτερικός βρόχος και μετά η επόμενη επανάληψη του εξωτερικού
- Η C επιτρέπει πολλαπλή ένθεση
- Ο συνολικός αριθμός επαναλήψεων σε έναν πολλαπλό βρόχο είναι το γινόμενο του αριθμού των επαναλήψεων όλων των επιμέρους βρόχων

Ένθετοι βρόχοι (nested loops)

```
for( i=0; i<4; i++)  
{  
    for( j=0; j<3; j++)  
    {  
        printf( “ (%d.%d) ”, i , j );  
    }  
}
```