

Επαναλήψεις - Loops

Οι επαναλήψεις σε ένα πρόγραμμα μας επιτρέπουν μια ομάδα εντολών να εκτελείται για όσες φορές επιθυμούμε. Υπάρχουν τρεις τρόποι επανάληψης εντολών με τη χρήση:

- While loops
- For loops
- Do-while loops

Ο πιο κάτω πίνακας περιγράφει σε ποιες περιπτώσεις χρησιμοποιούμε τους τρεις πιο πάνω τρόπους:

| Kind | When used | C Implementation Structures |
|--------------------------|---|-----------------------------|
| Counting loop | We can determine before loop execution exactly how many loop repetitions will be needed to solve the problem. | while for |
| Sentinel-controlled loop | Input of a list of data of any length by a special value. | while for |
| Endfile-controlled loop | Input of a single list of data of any length from a data file. | while for |
| Input validation loop | Repeated interactive input of a data value until a value within a valid range is entered. | do-while |
| General conditional loop | Repeated processing of data until a desired condition is met. | while for |

While

Το while έχει την ακόλουθη σύνταξη:

```
while (συνθήκη)  
{  
    Εντολή 1;  
    Εντολή 2;  
    ....  
}
```

Όσο η συνθήκη είναι αληθής, οι εντολές μέσα στα { } εκτελούνται η μια μετά την άλλη.

Πιο κάτω, δίνεται ένα παράδειγμα που ανήκει στην κατηγορία **counting loop**:

```
/*Εμφάνιση N αστερίσκων*/  
count_star = 0;  
  
while (count_star < N )  
{  
    printf("*");  
    count_star = count_star + 1;  
}
```

Επεξήγηση σειράς εκτέλεσης επανάληψης:

Η συνθήκη ελέγχεται. Αν είναι αληθής, οι εντολές μέσα στο while εκτελούνται και η συνθήκη ελέγχεται και πάλι. Οι εντολές μέσα στο while εκτελούνται ενόσω η συνθήκη είναι αληθής. Όταν η συνθήκη γίνει ψευδής, το while τερματίζει, οι εντολές μέσα σε αυτό αγνοούνται και το πρόγραμμα συνεχίζει την εκτέλεση του μετά το while.

Ένα άλλο παράδειγμα που ανήκει στην κατηγορία **conditional loop** ακολουθεί:

```
int number=1  
  
/*To while stamata otan o xristis grapsei 0*/  
while (number)  
{  
    printf("\nEnter a number:");  
    scanf("%d", &number);  
}
```

Στο πιο πάνω παράδειγμα δε γνωρίζουμε εκ των προτέρων, τον αριθμό των επαναλήψεων που θα εκτελεστούν. Το while τερματίζει μόνο όταν ο χρήστης γράψει 0.

Μελέτη Περίπτωσης

Να γραφτεί ένα πρόγραμμα σε C στο οποίο ο χρήστης θα καταχωρεί τρεις αριθμούς που αντιπροσωπεύουν βαθμούς σε μαθήματα. Οι τρεις αριθμοί έχουν πεδίο τιμών από 1 μέχρι και 100. Ακολούθως, το πρόγραμμα θα υπολογίζει το μέσο όρο τους και θα τυπώνει το τελικό βαθμό με βάση τα πιο κάτω κριτήρια:

Αν ο μέσος όρος βρίσκεται μεταξύ:

| | |
|--------|------------------------|
| 86-100 | Τύπωσε το χαρακτήρα A. |
| 65-85 | Τύπωσε το χαρακτήρα B. |
| 50-64 | Τύπωσε το χαρακτήρα C. |

| | |
|-------|------------------------|
| 25-49 | Τύπωσε το χαρακτήρα D. |
| 1-24 | Τύπωσε το χαρακτήρα E. |

Το πρόγραμμα να κάνει επίσης έλεγχο για την ορθότητα των βαθμών που δίνονται από το χρήστη. Σε περίπτωση που ο χρήστης δώσει βαθμό έξω από το πεδίο τιμών 1 – 100, θα τυπώνεται μήνυμα λάθους και θα επιτρέπει στο χρήστη να δώσει νέο βαθμό.

Εντολή Επανάληψης : for

Η εντολή for

for (αρχική τιμή; συνθήκη; μετρητής)

```
{  
    εντολή1;  
    εντολή2;  
    .  
    .  
    εντολήN;  
}
```

Η επανάληψη for ξεκινά με τη λέξη κλειδί for ακολουθούμενη από ένα ζεύγος παρενθέσεων οι οποίες περιέχουν τρεις προτάσεις που χωρίζονται μεταξύ τους με ερωτηματικό.

- Ο βρόχος που ορίζει η for εκτελείται ως εξής:
 1. Υπολογίζεται η αρχική τιμή
 2. Ελέγχεται η συνθήκη και εάν είναι αληθής τότε εκτελούνται οι εντολές.
 3. Υπολογίζεται ο μετρητής.
 4. Ο βρόχος συνεχίζει από το βήμα 2.

Σημείωση :

Η **πρόταση αρχικοποίησης** μπορεί να είναι όσο σύνθετη θέλουμε, χωρίζοντας διαδοχικές προτάσεις με κόμμα, καλό είναι όμως να μη το παρακάνουμε. Το δεύτερο μέρος είναι η **συνθήκη ελέγχου**, η οποία συνθήκη εξετάζεται κατά την

είσοδο κάθε επανάληψης. Η συνθήκη μπορεί να είναι οσοδήποτε περίπλοκη αρκεί να καταλήγει σε μία τιμή (αληθή ή ψευδή). Το τρίτο τμήμα περιέχει συνήθως τη **πρόταση αύξησης ενός μετρητή** και εκτελείται στο τέλος κάθε επανάληψης, δηλαδή μετά την εκτέλεση των προτάσεων του σώματος. Μπορεί, όπως και το πρώτο πεδίο της `for`, να περιέχει αρκετές προτάσεις χωρισμένες με κόμμα.

- **Παράδειγμα** (τυπώνει τους αριθμούς 0-9):
- `#include <stdio.h>`
-
- `main()`
- `{`
- `int i;`
-
- `for (i = 0; i < 10; i++)`
- `printf("%d\n", i);`
- `}`

Η ακριβής συσχέτιση των επαναλήψεων `for` και `while` δίνεται παρακάτω.

| | |
|------------------------|--|
| αρχικοποίηση; | for (αρχικοποίηση; συνθήκη; αύξηση) |
| | { |
| while (συνθήκη) | εντολή; |
| { | ... |
| εντολή; | εντολή; |
| ... | } |
| εντολή; | |
| αύξηση; | |
| } | |

Η βασική χρήση της επανάληψης `for` είναι σε περιπτώσεις που ο αριθμός των επαναλήψεων εκφράζεται με βάση κάποια μαθηματικού τύπου έκφραση και την αύξηση ή μείωση κάποιου μετρητή (π.χ πράξεις πινακοποιημένων δεδομένων), ενώ η χρήση της επανάληψης `while` προορίζεται για περιπτώσεις που ο αριθμός των επαναλήψεων είναι γενικά απροσδιόριστος και η συνθήκη ελέγχου είναι μία λογική συνθήκη η οποία στηρίζεται στα εκάστοτε δεδομένα (π.χ. επεξεργασία αρχείων).

Άλλοι τρόποι γραφής της δομής for:

(a)

```
for (i=1, sum=0; i<=10; ++i)
{
    sum+= i;          /* sum = sum + i ; */
}
```

(b)

```
for (i=1, sum=0; i<=10; sum+=i, ++i)
```

```
/* empty statements */
```

Ασκήσεις για την τάξη

Ασκηση 1

Γράψετε ένα πρόγραμμα που να τυπώνει στην οθόνη τους αριθμούς από το 5 μέχρι το 1

Output :

```
5
4
3
2
1
```

Ασκηση 2

Γράψετε ένα πρόγραμμα το οποίο διαβάζει από το πληκτρολόγιο 10 ακέραιους αριθμούς με επανάληψη ένα κάθε φορά και στο τέλος τυπώνει τον μέσο όρο των αριθμών αυτών στην οθόνη.

Output :

```
Enter a number: 2
Enter a number: 4
Enter a number: 3
.
.
.
```

The Average is : 6

Άσκηση 3

Γράψετε ένα πρόγραμμα το οποίο να τυπώνει 20 φορές ($i = 1$ μέχρι $i < 20$) τις τιμές που παίρνει κάθε φορά η μεταβλητή i , η μεταβλητή j αν υψώσουμε το i στο τετράγωνο και η μεταβλητή k αν υψώσουμε το i στην τρίτη δύναμη.

Σημείωση :

Χρήση του `\t`

Output

| i | j | k |
|-----|-----|------|
| 1 | 1 | 1 |
| 2 | 4 | 8 |
| 3 | 9 | 27 |
| 4 | 16 | 64 |
| 5 | 25 | 125 |
| . | . | . |
| . | . | . |
| . | . | . |
| . | . | . |
| . | . | . |
| 20 | 400 | 8000 |