

Προγραμματισμός Η/Υ Ι (Χρήση της C)

4^η Θεωρία

ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ

Σκοπός του μαθήματος

Σκοπός του μαθήματος είναι να μάθετε να χειρίζεστε απλές δομές επανάληψης με χρήση των εντολών **while**, **do-while**, **for**, **continue**, **break** δημιουργώντας απλά προγράμματα στη γλώσσα προγραμματισμού C. Όλα τα προβλήματα που είναι της μορφής **(όσο η λογική παράσταση A είναι αληθής, να επαναλαμβάνεται ένα κομμάτι του προγράμματος πολλές φορές)**, επιλύονται με βάση τις παραπάνω εντολές. Έτσι, με τις εντολές αυτές επιτυγχάνουμε την επαναληπτική δομή μιας ακολουθίας από το πρόγραμμα.

Η εντολή **while**

Η εντολή **while** (ή βρόχος **while**) επιτυγχάνει την επανάληψη μιας πρότασης (απλής ή σύνθετης), όσο μια λογική παράσταση είναι αληθής. Όταν η συνθήκη ελέγχου γίνει ψευδής η επανάληψη διακόπτεται. Συντάσσεται ως εξής:

Απλή μορφή:

```
while (λογική_παράσταση)
    πρόταση;
```

Σύνθετη μορφή:

```
while (λογική_παράσταση)
{
    πρόταση;
    πρόταση;
}
```

Πρόταση: απλή (με ; στο τέλος) ή σύνθετη (κλεισμένη σε άγκιστρα).

Στο επόμενο παράδειγμα η επανάληψη δε θα σταματήσει ποτέ.

```
count = 3;
while (count < 8)
    printf ("Καλημέρα;\n");
```

Στο επόμενο παράδειγμα η επανάληψη δε θα αρχίσει ποτέ και η εντολή δε θα εκτελεστεί καμία φορά.

```
count = 10;  
while (count < 8)  
    printf ("Καλημέρα;\n");
```

ΠΡΟΣΟΧΗ: Ένας βρόχος while πρέπει να περιλαμβάνει κάτι που να αλλάζει την τιμή της έκφρασης, ώστε αυτή να γίνει ψευδής και ο βρόχος να λήξει, αλλιώς, αν ξεκινήσει η εκτέλεση του while, δεν θα τελειώσει ποτέ.

Παράδειγμα

```
main()  
{  
    int index;  
  
    index = 0;  
    while (index < 6)  
    {  
        printf("The value of index is %d\n", index);  
        index = index + 1;  
    }  
}
```

Στο παράδειγμα οι προτάσεις θα εκτελεστούν μέχρι που ο μετρητής **index** να παρει την τιμή 6. Εφ' όσον ο μετρητής αυξάνεται κατά 1 σε κάθε επανάληψη οι προτάσεις θα εκτελεστούν 6 φορές. Στη συνέχεια η επανάληψη θα τερματιστεί και η εκτέλεση του προγράμματος θα συνεχιστεί με την εκτέλεση της πρότασης αμέσως μετά το σώμα της επανάληψης. Έτσι το αποτέλεσμα στην οθόνη θα είναι

```
The value of index is 0  
The value of index is 1  
The value of index is 2  
The value of index is 3  
The value of index is 4  
The value of index is 5
```

1^η Άσκηση

Υπολογίστε και εμφανίστε το άθροισμα των φυσικών αριθμών (1+2+3+4+...) μέχρις ότου το άθροισμα γίνει μεγαλύτερο από 999.

2^η Άσκηση

Να γραφεί πρόγραμμα το οποίο θα εμφανίζει τους αριθμούς 1, 2, ..., 100 με χρήση της δομής επανάληψης while.

3^η Άσκηση

Να γραφεί πρόγραμμα το οποίο θα εμφανίζει τους αριθμούς 99, 97, 95, ..., 1 με τη χρήση της δομής επανάληψης while.

4^η Άσκηση

Να γραφεί πρόγραμμα το οποίο θα υπολογίζει το άθροισμα $1+3+5+\dots+99$ με χρήση της δομής επανάληψης while.

Η εντολή do-while

Ο βρόχος **do-while** κάνει την ίδια δουλειά με τη **while**, με τη διαφορά ότι ελέγχει τη λογική παράσταση στο τέλος και όχι στην αρχή του βρόχου. Συντάσσεται ως εξής:

Απλή μορφή:

```
do  
    πρόταση;  
while (λογική_παράσταση);
```

Σύνθετη μορφή:

```
do  
{  
    πρόταση;  
    πρόταση;  
    .....  
}  
while (λογική_παράσταση);
```

Παράδειγμα

```
main()
{
    int num, count=0;

    do
    {
        scanf ("%d", &num);
        count++;
    }
    while (num != 9);
}
```

Όταν δοθεί σαν είσοδος ο αριθμός 9 η επανάληψη θα σταματήσει.

ΠΡΟΣΟΧΗ : Ο έλεγχος της συνθήκης εξόδου γίνεται μετά από κάθε εκτέλεση, άρα ο βρόχος θα εκτελεστεί τουλάχιστον **μία** φορά. Η do-while χρειάζεται ελληνικό ερωτηματικό μετά από τη λογική παράσταση.

5^η Άσκηση

Να γραφεί πρόγραμμα το οποίο θα διαβάζει αριθμούς από το πληκτρολόγιο μέχρι να δοθεί σαν είσοδος ο αριθμός 0. Για κάθε αριθμό που θα εισάγεται θα εμφανίζεται ο διπλάσιος του.

6^η Άσκηση

Να γραφεί πρόγραμμα το οποίο θα διαβάζει αριθμούς από το πληκτρολόγιο μέχρι να δοθεί σαν είσοδος ο αριθμός 0 και θα υπολογίζει το πλήθος, το άθροισμα και το γινόμενο των μη μηδενικών αριθμών.

7^η Άσκηση

Να γραφεί πρόγραμμα το οποίο θα διαβάζει αριθμούς από το πληκτρολόγιο μέχρι να δοθεί σαν είσοδος ο αριθμός 0. Ο αλγόριθμος θα υπολογίζει και θα εκτυπώνει το πλήθος των θετικών και των αρνητικών αριθμών καθώς και τα ποσοστά τους.

8^η Άσκηση

Να γραφεί πρόγραμμα το οποίο θα διαβάζει επαναληπτικά αριθμούς μέχρι το άθροισμα τους να γίνει μεγαλύτερο ή ίσο του 100. Στο τέλος να εμφανίζει το άθροισμα και το πλήθος των αριθμών που διαβάστηκαν.

Η εντολή for

Χρησιμοποιεί ένα μετρητή επαναλήψεων. Συντάσσεται ως εξής:

Απλή μορφή:

```
for (εκτελέσιμη_πρόταση1;λογική παράσταση;εκτελέσιμη πρόταση2)  
πρόταση ;
```

Σύνθετη μορφή:

```
for (εκτελέσιμη_πρόταση1;λογική παράσταση;εκτελέσιμη πρόταση2)  
{  
    πρόταση ;  
    πρόταση ;  
    .....  
}
```

Παράδειγμα

```
for (count=1; count<=15; count++)  
    printf ("Καλημέρα\n");
```

Παρατηρήσεις:

- | | |
|-------------|---|
| count = 1 | Απόδοση αρχικής τιμής στον μετρητή count. Εκτελείται μια φορά στην αρχή. |
| count <= 15 | Έκφραση ελέγχου. Υπολογίζεται πριν από κάθε εκτέλεση του βρόχου. Μόλις γίνει ψευδής, ο βρόχος τερματίζεται. |
| count ++ | Τρόπος μεταβολής της τιμής του μετρητή (count). Υπολογίζεται στο τέλος κάθε επανάληψης. |

Παράδειγμα

Υπολογίστε και εμφανίστε το άθροισμα των 10 πρώτων φυσικών αριθμών (1+2+3+4+...+10).

Απάντηση

```
main()
{
    int num;
    int sum;
    sum=0;
    for(num=1;sum<=10;num++)
    {
        sum+= num;
    }
    printf("to athroisma einai %10.3f \n", sum);
}
```

Παράδειγμα

Να υπολογιστεί και εκτυπωθεί ο πίνακας πολλαπλασιασμού από το 1 έως και το 10

Απάντηση

```
#include <stdio.h>
#include <conio.h>
main()
{
    int line, column;
    float product;
    product = 1.0;
    for(line=1;line<=10;line++)
    {
        printf("\n Ta pollaplasia tou %2d \n",line);
        for(column=1;column<=10;column++)
        {
            product= line*column;
            printf("%7.0f",product);
        }
    }
    printf("\nnow line and column are %d %d \n",line, column);
}
```

Παρατηρήστε την επανάληψη μέσα σε άλλη επανάληψη στο παραπάνω παράδειγμα.

9^η Άσκηση

Να γράψετε πρόγραμμα που να υπολογίζει το γινόμενο όλων των περιττών αριθμών από το 1 έως το 20.

10^η Άσκηση

Να γράψετε πρόγραμμα που να υπολογίζει το γινόμενο όλων των άρτιων αριθμών από το 1 έως το 20.

11^η Άσκηση

Να γράψετε πρόγραμμα που να διαβάζει 20 ακέραιους αριθμούς και στο τέλος να εμφανίζει το μικρότερο από αυτούς.

12^η Άσκηση

Να γράψετε πρόγραμμα που να διαβάζει 100 ακέραιους αριθμούς και στο τέλος να εμφανίζει το μεγαλύτερο από αυτούς καθώς και το πλήθος των εμφανίσεων αυτού του αριθμού.

13^η Άσκηση

Να γράψετε πρόγραμμα που να διαβάζει συνεχώς αριθμούς μέχρι να διαβαστεί κάποιος αρνητικός αριθμός. Στο τέλος να εμφανίζει το μέσο όρο όλων αυτών των αριθμών εκτός του αρνητικού. Αν ο πρώτος αριθμός ήταν αρνητικός τότε να εμφανίζει το μήνυμα «Ελλιπή στοιχεία».

14^η Άσκηση

Να διαβαστούν οι βαθμοί 10 μαθητών και να υπολογιστεί ο μέσος όρος των βαθμών. Αν ένας βαθμός είναι μικρότερος του 0 ή μεγαλύτερος του 20 δε λαμβάνεται υπόψιν.

Οι εντολές break-continue

Εκτός από τη χρήση της εντολής **break** που είδαμε στο προηγούμενο εργαστήριο (τερματισμός της πρότασης **case** στην εντολή **switch**), υπάρχει και η χρήση της **break** ως εξής:

Αναγκάζει άμεσο τερματισμό ενός βρόχου **while**, **do-while** ή **for**. Γενικά, οι προτάσεις **break** και **continue** χρησιμοποιούνται για την παράκαμψη δομών ελέγχου. Η πρόταση **break** εφαρμόζεται σε δομές επαναλήψεων και αποφάσεων και έχει ως αποτέλεσμα την συνέχεια της εκτέλεσης του προγράμματος στη πρώτη πρόταση μετά το τέλος της δομής. Η πρόταση **continue** εφαρμόζεται σε δομές επαναλήψεων και έχει ως αποτέλεσμα την συνέχεια της εκτέλεσης από την επόμενη επανάληψη. Η λειτουργία των εντολών αυτών φαίνεται αναλυτικά στο παρακάτω παράδειγμα:

Παράδειγμα


```
main()
{
    int x;

    for(x = 5; x < 15; x = x + 1){
        if (x == 8)
            break;
        printf("In the break loop, x is now %d\n", x);
    }

    for(x = 5; x < 15; x = x + 1){
        if (x == 8)
            continue;
        printf("In the continue loop, x is now %d\n", x);
    }
}
```

Επεξήγηση παραδείγματος

Το πρόγραμμα είναι μία ακολουθία δύο επαναλήψεων **for** με την κάθε επανάληψη να εκτελείται κανονικά εννέα φορές (για τις τιμές του μετρητή x 5, 6, ..., 14). Στη πρώτη επανάληψη υπάρχει μία πρόταση **break** η οποία ενεργοποιείται όταν ο μετρητής x πάρει την τιμή 8. Η εκτέλεση του **break** τερματίζει την πρώτη επανάληψη **for** και οδηγεί στην δεύτερη. Έτσι η πρώτη δομή εκτελείται ολόκληρη μόνο τρεις φορές αντί για εννέα. Κατά την τέταρτη επανάληψη (για x ίσο με 8) θα εκτελεστούν μόνο οι προτάσεις του σώματος που προηγούνται του **break**. Στη δεύτερη επανάληψη υπάρχει μία πρόταση **continue** στη ίδια ακριβώς θέση που είχε προηγουμένα η πρόταση **break**. Όταν ο μετρητής x πάρει την τιμή 8 η πρόταση **continue** εκτελείται αλλά δεν οδηγεί σε τερματισμό της επαναληψης. Απλά οδηγεί στην παράκαμψη των υπολοίπων προτάσεων του σώματος της επανάληψης, έτσι ώστε ο έλεγχος του προγράμματος μετατίθεται στην πρόταση αύξησης του μετρητή. Οι προηγούμενες και οι επόμενες επαναλήψεις εκτελούνται κανονικά. Έτσι έχουμε την εκτέλεση οκτώ πλήρων επαναλήψεων και μιάς διακεκομένης, όπου εκτελούνται

μόνον οι εντολές πριν από το **continue**. Η εκτέλεση του προγράμματος φαίνεται παρακάτω:

In the break loop, x is now 5
In the break loop, x is now 6
In the break loop, x is now 7
In the continue loop, x is now 5
In the continue loop, x is now 6
In the continue loop, x is now 7
In the continue loop, x is now 9
In the continue loop, x is now 10
In the continue loop, x is now 11
In the continue loop, x is now 12
In the continue loop, x is now 13
In the continue loop, x is now 14

15^η Άσκηση

Να γραφεί πρόγραμμα που να εμφανίζει τους 5 πρώτους μεγαλύτερους του μηδενός ακέραιους αριθμούς.

16^η Άσκηση

Να γραφεί πρόγραμμα που να υπολογίζει και να εμφανίζει στην οθόνη το γινόμενο $N! = 1 * 2 * 3 * \dots * N$ (N γνωστό).

17^η Άσκηση

Ο σημερινός πληθυσμός της Ελλάδος είναι 9.800.000 άτομα. Αν αυτός αυξάνεται κατά 3,5% το χρόνο να γραφεί πρόγραμμα το οποίο υπολογίζει και εμφανίζει στη οθόνη τον πληθυσμό της Ελλάδας μετά την πάροδο 8 ετών.

18^η Άσκηση

Να γραφεί πρόγραμμα το οποίο θα ζητάει έναν ακέραιο θετικό αριθμό και θα εμφανίζει το άθροισμα των ψηφίων του. Για παράδειγμα, αν δοθεί σαν είσοδος ο αριθμός 4567 θα εμφανίζει τον αριθμό 22.