

Π. Σταθοπούλου

psth@ece.upatras.gr ή
psth@upatras.gr

Ομάδα Α' (Φοιτητές με μονό αριθμό Μητρώου)

- Διδασκαλία : Παρασκευή 11πμ-13μμ ΗΛ7
- Φροντιστήριο : Δευτέρα 11πμ-12πμ ΗΛ4

Προηγούμενη Διάλεξη

Έννοια της μεταβλητής

- * διαχείριση, απεικόνιση, αριθμητικούς τύπους
- * τελεστής *sizeof()*
- * keyword *typedef*

Έννοια της σταθεράς

- * κυριολεκτικές & συμβολικές σταθερές (*#define, const*)

Προτάσεις, Εκφράσεις & Τελεστές

- * είδη προτάσεων, προτάσεις δομής συνθήκης (*if*)
- * προτάσεις επαναληπτικής δομής ελέγχου (*for*)
- * σχεσιακούς τελεστές & η σειρά προτεραιότητας αυτών

Σήμερα

Έννοια της σταθεράς

- * κυριολεκτικές & συμβολικές σταθερές
(`#define`, `const`)

Προτάσεις, Εκφράσεις & Τελεστές

- * είδη προτάσεων, προτάσεις δομής συνθήκης
(~~if~~, `switch-case`)
- * προτάσεις επαναληπτικής δομής ελέγχου
(~~for~~, `while`, `do while`)
- * μαθηματικούς τελεστές & η σειρά προτεραιότητας αυτών

Μεταβλητές - Σταθερές

Κυριολεκτική σταθερά είναι μία μεταβλητή που της αποδίδεται τιμή άμεσα στον κώδικα κάθε φορά που χρειάζεται

Παράδειγμα
Κυριολεκτικής σταθεράς



```
int count=20;  
float tax_rate=0.28;
```

Συμβολική σταθερά είναι μία μεταβλητή που της αποδίδεται καταρχήν ένα όνομα (σύμβολο) και στην συνέχεια η τιμή της.

Μεταβλητές - Σταθερές

Οι Συμβολικές σταθερές ορίζονται:

* με την εντολή προεπεξεργαστή **#define**

ή

* με την λέξη κλειδί **const**

συντάσσονται δε ως εξής:

➤ **#define constname value**

➤ **const datatype constname=value;**

Μεταβλητές - Σταθερές

Η διαφορά **#define** - **const** υφίσταται στην εμβέλεια των μεταβλητών και στους δείκτες

Παράδειγμα
Συμβολικής σταθεράς

```
#define PI 3.14  
circumference = PI*(2*radius);  
area = PI*(radius)*(radius);
```

```
const int count= 100;  
const int x=50, float rate=0.21;
```

Προτάσεις

Είδη Προτάσεων - (Παραδείγματα)

- ☞ Πρόταση δήλωσης
- ☞ Πρόταση ανάθεσης
- ☞ Σύνθετη πρόταση
- ☞ Πρόταση ελέγχου ροής
- ☞ Πρόταση κλήσης συνάρτησης

Είδη Προτάσεων - (Παραδείγματα)

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

int num1,num2,sum;
int add(int x, int y);

int main(int argc, char *argv[])
{
    printf("\n\t\t Give the First number:");
    scanf("%d",&num1);
    printf("\n\t\t Give the Second number:");
    scanf("%d",&num2);
    if(num1>num2)
    {
        printf("\n\t\t The First number is greater the Second");
        sum=add(num1,num2);
        printf("\n\t\t The SUM of two number is %d\n\n",sum);
    }
    else
        printf("\n\t\t The Second number is greater or equal the First \n\n");

    system("PAUSE");
    return 0;
}
```


Προτάσεις δήλωσης

Προτάσεις ελέγχου ροής

Προτάσεις ανάθεσης & κλήσης συνάρτησης

Σύνθετη πρόταση

```
int add(int x, int y)
{
    return (x+y);
}
```



Προτάσεις δομής συνθήκης

Η πρόταση *if*

απλή πρόταση → *if*
σύνθετη πρόταση → *if - else*
σύνθετη πρόταση → *if - else if*

Η πρόταση *switch-case*

Πρόταση συνθήκης *switch*

Η πρόταση ***switch-case*** είναι περισσότερο ευέλικτη πρόταση ελέγχου της C από την ***if***.

Η πρόταση ***switch-case*** καθοδηγεί το πρόγραμμα να εκτελεί διαφορετικές προτάσεις με βάση **μια έκφραση** που μπορεί να έχει περισσότερες από δύο τιμές.

Πρόταση συνθήκης **switch**

Η σύνταξη της πρότασης **switch-case** έχει ως εξής:
switch (έκφραση)

```
{  
  case πρότυπο_1: πρόταση/προτάσεις;  
  case πρότυπο_2: πρόταση/προτάσεις;  
  .....  
  case πρότυπο_n: πρόταση/προτάσεις;  
  default: πρόταση/προτάσεις;  
}
```

Πρόταση συνθήκης **switch**

Στην πρόταση **switch** (έκφραση)

- η **έκφραση** είναι ο έλεγχος
- υπολογίζεται σε τιμή τύπου **long**, **int**, ή **char**
- Η πρόταση **switch** εκτιμά την έκφραση & συγκρίνει την τιμή ως προς τα **πρότυπα** που ακολουθούν κάθε ετικέτα **case**.
- τότε ακολουθούν τα παρακάτω βήματα:

Πρόταση συνθήκης **switch**

- Εάν η έκφραση ικανοποιεί ένα από τα πρότυπα τότε η εκτέλεση του προγράμματος μεταφέρεται στην πρόταση που έπεται της ετικέτας **case**.
- Εάν δεν υπάρχει αντιστοιχία μεταφέρεται στην πρόταση της προαιρετικής ετικέτας **default**.
- Εάν δεν υπάρχει αντιστοιχία & δεν υπάρχει η προαιρετική ετικέτα **default** τότε η εκτέλεση περνά στην 1^η πρόταση που έπεται του άγκιστρου κλεισίματος της πρότασης **switch**.

Παράδειγμα της πρότασης **switch**

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

int main(int argc, char *argv[])
{
    int reply;
    puts("Enter a number between 1 and 5:");
    scanf("%d",&reply);
    switch (reply)
    {
        case 1:
            puts("You entered 1.\n");
        case 2:
            puts("You entered 2.\n");
        case 3:
            puts("You entered 3.\n");
        case 4:
            puts("You entered 4.\n");
        case 5:
            puts("You entered 5.\n");
        default:
            puts("Out of range, try again.\n");
    }
    system("PAUSE");
    return 0;
}
```

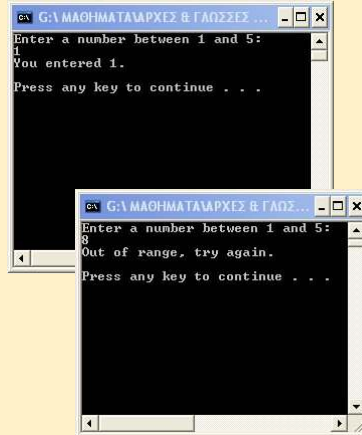


```
DEV
G:\ΜΑΘΗΜΑΤΑ\ΑΡΧΕΣ & ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤ
Enter a number between 1 and 5:
1
You entered 1.
You entered 2.
You entered 3.
You entered 4.
You entered 5.
Out of range, try again.
Press any key to continue . . .
```

Παράδειγμα της πρότασης *switch*

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

int main(int argc, char *argv[])
{
    int reply;
    puts("Enter a number between 1 and 5:");
    scanf("%d",&reply);
    switch (reply)
    {
        case 1:
        {
            puts("You entered 1.\n");
            break;
        }
        case 2:
        {
            .....
        }
        default:
        {
            puts("Out of range, try again.\n");
        }
    }
    system("PAUSE");
    return 0;
}
```



Προτάσεις επαναληπτικής δομής ελέγχου

Η πρόταση *for*

Η πρόταση *while*

Η πρόταση *do ...while*

Είναι δομές προγραμματισμού της C που εκτελούν ένα μπλοκ μιας ή περισσότερων προτάσεων για ορισμένες φορές

Πρόταση **for**

Εκφράσεις ελέγχου πολύπλοκες που τερματίζουν τον βρόχο

```
for (count=0, j=100; count<=120; count++, j--)  
    b[j] =a[count];
```



Ποιες είναι οι τιμές του **j** ?

Η έκφραση αυτή είναι σωστή ?

Πρόταση **while**

Η πρόταση **while** (ή βρόχος **while**) εκτελεί ένα μπλοκ προτάσεων εφόσον μία συνθήκη είναι αληθής

Συντάσσεται ως εξής:

**while (συνθήκη)
 πρόταση_1**

Η συνθήκη μπορεί να είναι **μία απλή πρόταση**
ή **συνδυαστική πρόταση**

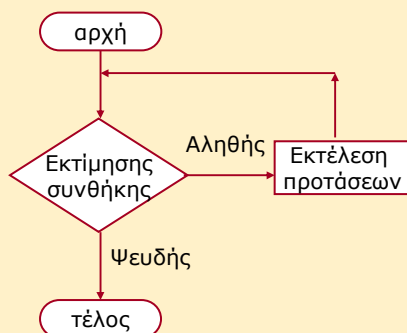
Πρόταση **while**

Στην πρόταση **while** λαμβάνουν χώρα τα παρακάτω βήματα:

1. Υπολογίζεται η έκφραση συνθήκη
2. Εάν **συνθήκη** ψευδής η πρόταση **while** τερματίζει & το πρόγραμμα μεταπηδά στην πρόταση που έπεται της πρόταση_1
3. Εάν **συνθήκη** αληθής η πρόταση **while** εκτελεί την πρόταση_1
4. Η εκτέλεση επιστρέφει στο βήμα 1

Πρόταση **while**

Λειτουργία της πρότασης **while**



```
#include <stdio.h>
#include <stdlib.h>
int count;

int main(int argc, char
*argv[])
{
    count=1;
    while (count<=20)
    {
        printf("%d\n",count);
        count++;
    }
    system("PAUSE");
    return 0;
}
```

Προτάσεις *while* & *for*

Η πρόταση *while* ουσιαστικά είναι μία πρόταση *for*

for(;συνθήκη;)
while(συνθήκη)

παραδείγματα

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

int value[10];
int ctr=0, nbr;

int main(int argc, char *argv[]) AND
{
    while(ctr<10 && nbr!=99)
    {
        printf("%d Enter a number, 99 to quit ",ctr);
        scanf("%d",&nbr);
        value[ctr]=nbr;
        ctr++;
    }
    system("PAUSE");
    return 0;
}
```

Ένθεση προτάσεων *while*

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

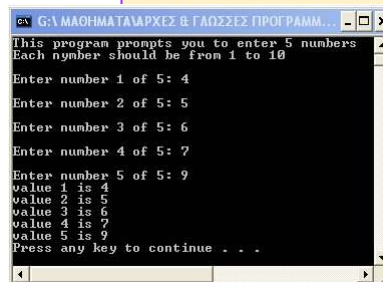
int array[5];

int main(int argc, char *argv[])
{
    int ctr=0, nbr=0;

    printf("This program prompts you to enter 5 numbers\n");
    printf("Each number should be from 1 to 10\n");

    while(ctr<5)
    {
        nbr=0;
        while (nbr<1||nbr>10)
        {
            printf("\nEnter number %d of 5: ",ctr+1);
            scanf("%d",&nbr);
        }
        array[ctr]=nbr;
        ctr++;
    }
    for (ctr=0;ctr<5;ctr++)
        printf("value %d is %d\n",ctr+1,array[ctr]);

    system("PAUSE");
    return 0;
}
```



Πρόταση **do***while*

Η πρόταση **do... while** (ή βρόχος **do ...while**) εκτελεί ένα μπλοκ προτάσεων εφόσον μία συνθήκη είναι αληθής

Ο βρόχος **do ...while** ελέγχει την συνθήκη στο τέλος του βρόχου αντί στην αρχή.

Συντάσσεται ως εξής:

```
do  
  πρόταση_1  
while(συνθήκη)
```

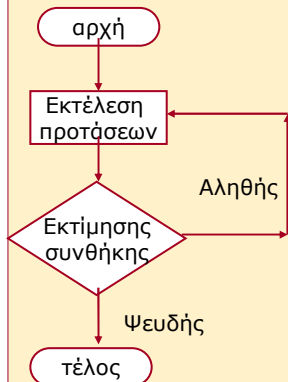
Πρόταση **do***while*

Στην πρόταση **do ... while** λαμβάνουν χώρα τα παρακάτω βήματα:

1. Εκτελούνται οι προτάσεις **στην πρόταση_1**
2. Υπολογίζεται η συνθήκη. Εάν είναι αληθής η εκτέλεση επιστρέφει στο βήμα 1. Εάν είναι ψευδής η πρόταση τερματίζεται.

Πρόταση **dowhile**

Λειτουργία της πρότασης **while**



```
do  
{  
printf("\n");  
printf("\n1 - Add a Record");  
printf("\n2 - Delete a Record");  
printf("\n3 - Change a Record");  
printf("\n4 - Quit");  
printf("\n");  
printf("\nEnter a selection: ");  
scanf("%d",&selection);  
} while (selection <1||selection >4);
```