

Εισαγωγή στη γλώσσα C

Συναρτήσεις

Πρόχειρη παρουσίαση με πρακτικά
παραδείγματα για χρήση στο εργαστήριο

Παναγιώτης Παπάζογλου
Αν. Καθηγητής

Συνάρτηση

- Η γλώσσα C βασίζεται σε συναρτήσεις
- Η `main` είναι συνάρτηση
- Μια συνάρτηση αποτελεί τμήμα ή ολοκληρωμένο πρόγραμμα
- Εκτελείται όταν κληθεί
- Μεταβιβάζονται δεδομένα στη συνάρτηση η οποία επιστρέφει κάποιο αποτέλεσμα
- Υπάρχουν «συναρτήσεις» που δεν δέχονται δεδομένα ή δεν επιστρέφουν αποτελέσματα ή και τα δύο

Μορφή συνάρτησης

```
Τύπος_αποτελέσματος ΟΝΟΜΑ_ΣΥΝΑΡΤΗΣΗΣ (παράμετροι)
{
    /* Κυρίως σώμα συνάρτησης*/
}
```

Παράδειγμα

Τύπος
αποτελέσματος

Πρώτο όρισμα

Δεύτερο όρισμα

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

Εντολή επιστροφής
αποτελέσματος και
εξόδου από τη
συνάρτηση

Παράδειγμα κλήσης συνάρτησης

```
#include <stdio.h>
int add(int x, int y);
main()
{
  int a,b,c;
  printf("a:"); scanf("%d",&a);
  printf("b:"); scanf("%d",&b);
```

Δήλωση συνάρτησης

```
c=add(a,b);
```

```
printf("a+b=%d",c);
```

```
}
```

```
int add(int x, int y)
```

```
{
```

```
return x+y;
```

```
}
```

Ορισμός συνάρτησης

Η λειτουργία return

```
#include <stdio.h>
int add(int x, int y);
main()
{
    int a,b,c;
    printf("a:"); scanf("%d",&a);
    printf("b:"); scanf("%d",&b);

    c=add(a,b);

    printf("a+b=%d",c);
}
```

```
int add(int x, int y)
{

    return x+y;
    printf("Hello");
}
```

Η εντολή **printf("Hello")** δεν θα εκτελεστεί ποτέ γιατί η εντολή **return** προκαλεί την έξοδο από τη συνάρτηση επιστρέφοντας δεδομένα στο σημείο κλήσης

«Συνάρτηση» που δεν επιστρέφει αποτέλεσμα

```
#include <stdio.h>
void add(int x, int y);
main()
{
    int a,b;
    printf("a:"); scanf("%d",&a);
    printf("b:"); scanf("%d",&b);

    add(a,b);
}
```

Η κλήση γίνεται απλά με το όνομά της και τις παραμέτρους.

```
void add(int x, int y)
{
    printf("%d", x+y);
}
```

Αντί για τύπο αποτελέσματος γράφουμε τη λέξη **void**, ενώ επίσης δεν υπάρχει εντολή **return**.

«Συνάρτηση» που δεν επιστρέφει αποτέλεσμα και δεν δέχεται ορίσματα

```
#include <stdio.h>
void message();
main()
{
    message();
}
```

Κλήση
«συνάρτησης»

```
void message()
{
    printf("Hello");
}
```

ή

```
void message(void)
{
    printf("Hello");
}
```


Καθολικές και τοπικές μεταβλητές (1)

Γενικός κανόνας

- Οι μεταβλητές που δηλώνονται στο σώμα μιας συνάρτησης είναι «ορατές» μόνο σε αυτές
- Οι μεταβλητές που δηλώνονται εκτός όλων των συναρτήσεων είναι «ορατές» σε οποιοδήποτε σημείο

```
#include <stdio.h>
void message(void);
main()
{
    int a=0;
    b=1;
    message();
}

void message(void)
{
    int b=0;
    a=1;
    printf("Hello");
}
```

Η μεταβλητή **a** δηλώνεται εντός της **main**. Έτσι, μόνο η **main** την αναγνωρίζει.

ΣΦΑΛΜΑ: Η μεταβλητή **b** δεν αναγνωρίζεται από τη **main**.

Η μεταβλητή **b** δηλώνεται εντός της **message**. Έτσι, μόνο η **message** την αναγνωρίζει.

ΣΦΑΛΜΑ: Η μεταβλητή **a** δεν αναγνωρίζεται από τη **message**.

Καθολικές και τοπικές μεταβλητές (2)

```
#include <stdio.h>
int a;
void message(void);
main()
{
  a=1;
  message();
}

void message(void)
{
  a=2;
  printf("Hello");
}
```

Η μεταβλητή **a** δηλώνεται εκτός των συναρτήσεων **main** και **message**.

Η συνάρτηση **main** αναγνωρίζει τη μεταβλητή **a**.

Η συνάρτηση **message** αναγνωρίζει τη μεταβλητή **a**.

Άσκηση 1

Πρόγραμμα που διαβάζει τρεις αριθμούς και εμφανίζει το άθροισμά τους. Χρήση μιας συνάρτησης που εμφανίζει το μήνυμα εισαγωγής (για κάθε αριθμό) και επιστρέφει τον αριθμό που διαβάστηκε.

```
#include <stdio.h>
int read (int x);
main()
{
    int a,b,c;
    a=read(1);
    b=read(2);
    c=read(3);

    printf("sum=%d",a+b+c);
}

int read(int x)
{
    int number;
    printf("Enter number %d:",x);
    scanf("%d",&number);
    return number;
}
```

Άσκηση 2

Υλοποίηση και χρήση συνάρτησης υπολογισμού δύναμης

```
#include <stdio.h>
int power(int a, int x);
main()
{
    int n1,n2;
    printf("a^x\na=");
    scanf("%d",&n1);
    printf("x=");
    scanf("%d",&n2);
    printf("%d^%d=%d",n1,n2,power(n1,n2));
}

int power(int a, int x)
{
    int p=1;
    for(int i=1;i<=x;i++)
        p*=a;

    return p;
}
```