

Π. Σταθοπούλου

pstath@ece.upatras.gr ή
pstath@upatras.gr

Ομάδα Α' (Φοιτητές με μονό αριθμό Μητρώου)

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

Προηγούμενη Διάλεξη

Μαθηματικοί Τελεστές

* *Είδη τελεστών & προτεραιότητα*

Έννοια της συνάρτησης

* *Ορισμός, σύνταξη & λειτουργία*

Σήμερα

Συνέχεια μελέτη Συναρτήσεων

- * **Είδη Συναρτήσεων**
 - με εσωτερικές δομές ελέγχου
 - που δεν επιστρέφουν τιμές
 - που επιστρέφουν τιμές
 - που επιστρέφουν μη αριθμητικές τιμές
- * **Μεταβίβαση παραμέτρων**
 - Ορίσματα
 - Τιμές επιστροφής
- * **Κλήση συναρτήσεων**
 - με απλή κλήση του ονόματος
 - με πρόταση εκχώρησης
- * **Θέση Συναρτήσεων**
 - Στο κυρίως πρόγραμμα που καλείται
 - σε ξεχωριστό αρχείο

Χαρακτηριστικά Συνάρτησης

- `type name (type var1, type var2,... type varn);`
1. **Δήλωση αρχέτυπης/πρότυπου συνάρτησης**
 2. **Ορισμός συνάρτησης**
 - `type name (type var1,... type varn)`
 - `{`
 - `.....`
 - `return;`
 - `}`
 3. **Κλήση συνάρτησης**
 - `name (value1, value2,... valuen);`
 - ή**
 - `K = name (value1, value2,... valuen);`

Είδη Συναρτήσεων

Με εσωτερικές δομές ελέγχου

Πρόγραμμα που εισάγει ένα αριθμό και υπολογίζει την απόλυτη τιμή του

Η δομή ελέγχου μπορεί να είναι:

if

switch - case

Είδη Συναρτήσεων

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

```
int abs(int n);
```

```
int main(int argc, char *argv[])
{
    int n, result;
    printf("Press an Integer Number,Please\n");
    scanf("%d",&n);
    result=abs(n);
    printf("\nthe absolute value of %d is the %d\n",n,result);
    system("PAUSE");
    return 0;
}
```

```
int abs(int n)
{
    if (n<0)
        return(-n);
    else
        return (n);
}
```

Δήλωση
συνάρτησης

Εκχώρηση
τιμής

Ορισμός
συνάρτησης

Επιστρέφει 2
διαφορετικές τιμές

Είδη Συναρτήσεων

Συναρτήσεις που δεν επιστρέφουν τιμές

Πρέπει να συντάσσονται ως εξής:

void name (type var₁, type var₂,... type var_n);

Η λέξη **void** υποδεικνύει στον μεταγλωττιστή ότι η συνάρτηση δεν επιστρέφει τιμή

display(x,y)

```
int x, y, sum;
{
  If (x==y) return;
  sum=x+y;
  printf("the sum is %d", sum);
}
```

void display(x,y)

Προστασία

Δεν επιτρέπει Προτάσεις εκχώρησης
k=display(5, 12);

Είδη Συναρτήσεων

```
void Selec_Type_Lab(int x)
{
  int choice;
  choice=x;
  switch(choice)
  {
    case 1:
      labLoadLabyrinth("labyrinth5.lab");
      break;
    case 2:
      labLoadLabyrinth("labyrinth6.lab");
      break;
    case 3:
      labLoadLabyrinth("labyrinth10.lab");
      break;
    case 4:
      labLoadLabyrinth("labyrinth12.lab");
      break;
    case 5:
      return;
      break;
    default:
      printf("You are selected wrong type Labyrinth");
  }
  return;
}
```

δήλωση

void Selec_Type_Lab(int x);

Κλήση

k=a+b;
Selec_Type_Lab(k);

Είδη Συναρτήσεων

Συναρτήσεις που επιστρέφουν τιμές

Πρέπει να συντάσσονται ως εξής:

type name (type var₁, type var₂,... type var_n);

Ο τύπος επιστροφής συνάρτησης μπορεί να είναι:

char, int, long, float ή double

Π.χ.

```
char function1(.....),      long function3(.....),  
int function2(.....),      float function4(.....),  
double function5(.....)
```

Είδη Συναρτήσεων

Δέχεται ένα αριθμό



τον εμφανίζει αντίστροφα,
επιστρέφει το πλήθος των ψηφίων

```
#include <stdio.h>  
#include <stdlib.h>  
  
int reverse(int num1);  
  
int main(int argc, char *argv[])  
{  
    int number, plthtos;  
    printf("Press an Integer Number, Please\n\n");  
    scanf("%d",&number);  
    plthtos=reverse(number);  
    printf("\nΤΟ ΠΛΗΘΟΣ ΤΩΝ ΨΗΦΙΩΝ ΤΟΥ ΑΡΙΘΜΟΥ  
%d ΕΙΝΑΙ %d\n\n",number,plthtos);  
    system("PAUSE");  
    return 0;  
}
```

```
int reverse(int num1)  
{  
    int y,p,cnt=0;  
    do  
    {  
        y=num1%10;  
        p=num1/10;  
        printf("%d",y);  
        cnt++;  
        num1=p;  
    } while (p!=0);  
    return cnt;  
}
```

Είδη Συναρτήσεων

Συναρτήσεις που επιστρέφουν μη αριθμητικές τιμές

```
#include <stdio.h>
#include <stdlib.h>
typedef char* string;
string MonthName(int month);
int n;
char *str;

int main(int argc, char *argv[])
{
    printf("Press an Integer Number, Please\n");
    scanf("%d", &n);
    str=MonthName(n);
    printf("\nthe month is %s\n", str);
    system("PAUSE");
    return 0;
}
```

```
string MonthName(int month)
{
    switch (month){
        case 1: return("January");
        case 2: return("February");
        case 3: return("March");
        case 4: return("April");
        case 5: return("May");
        case 6: return("June");
        case 7: return("July");
        case 8: return("August");
        case 9: return("September");
        case 10: return("October");
        case 11: return("November");
        case 12: return("Decemmmber");
        default: return("Illegal month");
    }
}
```

Είδη Συναρτήσεων

Συναρτήσεις που επιστρέφουν τιμές τύπου bool

Κατηγορηματικές συναρτήσεις

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

bool IsEven(int n);

int main(int argc, char *argv[])
{
    int i;
    for(i=1; i<=10; i++){
        if (IsEven(i)) printf("%2d\n", i);
    }
    system("PAUSE");
    return 0;
}
```

```
bool IsEven(int n)
{
    return(n%2==0);
}
```

```
2
4
6
8
10
```

Μεταβίβαση Παραμέτρων

Το Σώμα της Συνάρτησης αποτελείται

την επικεφαλίδα

- τον τύπο της τιμής επιστροφής, το όνομα & τα ορίσματα.
- π.χ. **int larger_of(int a, int b)**

και το κυρίως σώμα

- περιβάλλεται σε **άγκιστρα { }**
- περιέχει **τοπικές** μεταβλητές
- περιέχει προτάσεις, δομές ελέγχου, δομές επανάληψης
- και την πρόταση **return (;**

Μεταβίβαση Παραμέτρων

Οι τοπικές μεταβλητές ακολουθούν τους ίδιους κανόνες με τις μεταβλητές του κυρίως προγράμματος.

ονοματολογία μεταβλητών

τύποι μεταβλητών

Μία συνάρτηση μπορεί να περιέχει πολλά **return**

Πολλές προτάσεις **return** αποτελούν έναν αποτελεσματικό τρόπο επιστροφής διαφορετικών τιμών από μια συνάρτηση.

Μεταβίβαση Παραμέτρων

```
int larger_of (int a,int b)
```

```
{  
    if(a>b)  
        return a;  
    else  
        return b;  
}
```

Τοπικές μεταβλητές

Επιστρεφόμενη τιμή

```
int reverse(int num1)
```

```
{  
    int y,p,cnt=0;  
    do  
    {  
        y=num1%10;  
        p=num1/10;  
        printf("%d",y);  
        cnt++;  
        num1=p;  
    } while (p!=0);  
    return cnt;  
}
```

Όρισμα

Κλήση συναρτήσεων

Δύο τρόποι κλήσης μιας συνάρτησης

Καλείται απλά με το όνομά της & την λίστα των ορισμάτων της

π.χ. **labConnect(server_addr, server_port);**

Εκχωρείται η επιστρεφόμενη τιμή της συνάρτησης σε μια μεταβλητή.

π.χ. **k=robotExitFound();**

Θέση συναρτήσεων

Δύο τρόποι τοποθέτησης μιας συνάρτησης

στο κυρίως πρόγραμμα

Δήλωση, Ορισμός, Κλήση

Ορίζεται συνήθως στο τέλος του κυρίως προγράμματος

Η **δήλωση** και ο **ορισμός** της συνάρτησης μεταφέρεται σε ξεχωριστό αρχείο πηγαίου κώδικα.

Τα λεγόμενα αρχεία κεφαλίδας (header files)

π.χ. **Robot.h** **int RobotStop(void)**
int labLoadLabyrinth(char *name)

Αναδρομή

Αναδρομή:

είναι η αναγωγή του αρχικού προβλήματος σε μικρότερα προβλήματα του ίδιου τύπου.

Αν μια συνάρτηση περιέχει μια κλήση του εαυτού της στο εσωτερικό του τμήματος των εντολών της, τότε λέγεται ότι είναι:

μία αναδρομική συνάρτηση
(recursive function)

Αναδρομή

Ο όρος **Αναδρομή** (recursion) αναφέρεται σε μία κατάσταση στην οποία μία συνάρτηση καλεί τον εαυτό της **άμεσα** ή **έμμεσα**.

Έμμεση Αναδρομή λαμβάνει χώρα όταν μία συνάρτηση καλεί μία άλλη συνάρτηση η οποία στην συνέχεια καλεί την πρώτη συνάρτηση.

Αναδρομή

Δομή αναδρομικής συνάρτησης

* Κατάσταση διακοπής * Αναδρομικό βήμα

Πρόβλημα

Να γραφεί μια αναδρομική συνάρτηση που να υπολογίζει το παραγοντικό ενός μη αρνητικού ακεραίου αριθμού

Αναδρομή

Ανάλυση του προβλήματος

$$n! = n(n-1)!, \quad 0! = 1$$

* Αν $n == 0$
τότε να επιστραφεί η τιμή 1

(κατάσταση διακοπής)

* Αν $n \geq 1$,
τότε να επιστραφεί η : $n * \text{factorial}(n-1)$

(αναδρομικό βήμα)

Αναδρομή

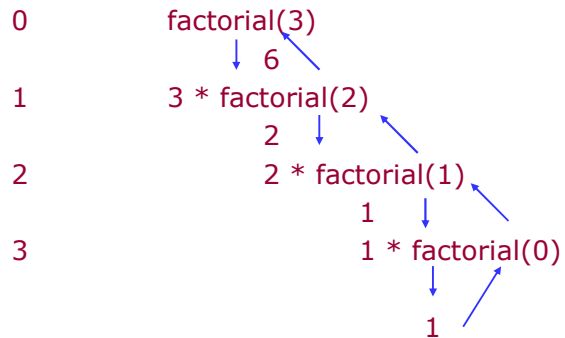
Υλοποίηση

```
int factorial (int n)
{
    if (n == 0) /* Κατάσταση διακοπής */
        return 1;
    else
        return (n * factorial (n-1)); /* Αναδρομικό βήμα */
}
```

Αναδρομή

Εκτέλεση της factorial(3);

Αναδρομικό
επίπεδο



Αναδρομή

Πρόβλημα

Να γραφεί μία συνάρτηση που να υπολογίζει το άθροισμα όλων των ακεραίων αριθμών από τον m μέχρι τον n .

Ανάλυση

Δεδομένων δύο ακεραίων αριθμών m και n ($m \leq n$) θέλουμε να υπολογίσουμε το άθροισμα

$$\text{athroisma}(m,n) = m + (m + 1) + \dots + n$$

Αναδρομή

Υλοποίηση

```
int athroisma (int m, n)
{
  if (n == m)           (κατάσταση διακοπής)
    return n;
  else
    return (n + athroisma (m, n-1));
}
```

(αναδρομικό βήμα)

Αναδρομή

Παράδειγμα

Να γραφεί συνάρτηση, η οποία να επιστρέφει την τιμή του $\sum_{i=1}^n i^k$ για δεδομένα **n** και **k**

Καταρχήν θα γράψουμε μία συνάρτηση, η οποία επιστρέφει τις τιμές της δύναμης **1^k, 2^k, 3^k, ..., n^k** για δεδομένα **k** και **n**.

Αναδρομή

Συνάρτηση δύναμης

```
int power (int k, int n)
{
  int i, product=1;
  for (i=1; i <= n; ++i)
    product*= k;
  return (product);
}
```

product = product *k;

Αναδρομή

Συνάρτηση αθροίσματος & δύναμης

```
int sum_power (int k, int n)
{
  int i, sum = 0;
  for (i=1; i <= n; ++i)
    sum+=power (k, i);
  return (sum);
}
```

sum= sum +power (k, i);