

6^η Εργαστηριακή Άσκηση

Σκοπός:

- Εξοικείωση στη χρήση συναρτήσεων (ορισμός και κλήση συναρτήσεων)
- Κατανόηση της αναδρομής και του τρόπου δημιουργίας αναδρομικών συναρτήσεων

ΣΥΝΑΡΤΗΣΕΙΣ

- Συνάρτηση (function): σύνολο εντολών που έχει ομαδοποιηθεί και τους έχει αποδοθεί ένα όνομα
- Κλήση (calling) : η εκτέλεση των εντολών της συνάρτησης που έχει κληθεί
- Ορίσματα (arguments): οι παραστάσεις που επιτρέπουν στο καλούν πρόγραμμα να μεταβιβάζει πληροφορία στη συνάρτηση π.χ. printf("Hello, World);
- Επιστροφή (returning): η επάνοδος στο καλούν πρόγραμμα μετά το τέλος εκτέλεσης των εντολών της συνάρτησης
- Επιστροφή τιμής (returning value): Η επιστροφή κάποιας τιμής στο καλούν πρόγραμμα μετά το τέλος εκτέλεσης εντολών μιας συνάρτησης π.χ. n1 = GetInteger();

Δήλωση συναρτήσεων (πριν τη main)

- Πρωτότυπο Συνάρτησης (function prototype)
 - επιστρεφόμενος_τύπος όνομα(όρισμα₁, όρισμα₂,...)
 - επιστρεφόμενος_τύπος: ο τύπος δεδομένων της τιμής που επιστρέφει η συνάρτηση
 - όνομα: το όνομα της συνάρτησης
 - όρισμα: τύπος δεδομένων και όνομα μεταβλητής που μεταβιβάζεται ως όρισμα από το καλούν πρόγραμμα στη συνάρτηση
- void
 - ορίζει ότι μια συνάρτηση δεν επιστρέφει κάποια τιμή ή δεν δέχεται κάποιο όρισμα
 - π.χ. void function1(int a);
 - π.χ. int function2(void);

Ορισμός Συνάρτησης (μετά τη main)

Για κάθε μη ενσωματωμένη (δική μας) συνάρτηση πρέπει να ορίσουμε το σώμα της στο πρόγραμμα (μετά την main)

```
<τύπος> (<όνομα - συν> (<παραμέτροι>);  
{  
    <δηλώσεις τοπικών μεταβλητών>  
    <προτάσεις>  
}
```

- return(παράσταση);
 - παράσταση: η τιμή που επιστρέφει η συνάρτηση στο καλούν πρόγραμμα
- return;
 - αν η συνάρτηση δεν επιστρέφει κάποια τιμή
 - ο επιστρεφόμενος τύπος είναι void

Κλήση Συνάρτησης (μέσα στη main)

Καλείται η συνάρτηση για εκτέλεση με συγκεκριμένα ορίσματα

```
<όνομα - συν> (<παραμ1>, <παραμ2>, ..., <παραμN>);
```

Τρόπος Χρήσης Συναρτήσεων

```
main () {
int a, b;
f(a,b); /* καθορίζω τα δεδομένα πάνω στα οποία θα γίνει η επεξεργασία
- πραγματικές παράμετροι*/
}
void f(int x, int y) { /* τυπικές παράμετροι */
}
```

Παράδειγμα: Συνάρτηση Υπολογισμού Δύναμης

```
#include <stdio.h>
int power(int m, int n);

main() {
int i, res;
for (i=0; i<10; i++) {
res = power (2, i);
printf("%d %d \n", i, res);
}
return 0;
}

int power(int base, int n) {
int i, p;
p =1;
for (i =1; i <=n; i++)
p = p*base;
return p;
}
/* να καταχωρούνται τιμές σε τρεις ακέραιες μεταβλητές
και μετά να καλείται μια συνάρτηση που θα υπολογίζει τη μεγαλύτερη
τιμή */

#include <stdio.h>
main()
{
int a, b, c;
int max;
int maximum();
/* εδώ δηλώνουμε τις συναρτήσεις που θα χρησιμοποιήσει το πρόγραμ-
μα - μια συνάρτηση επιστρέφει μία και μόνο μία τιμή στο όνομά της και
όταν την δηλώνουμε, γράφουμε πριν από το όνομά της τον τύπο δεδομένων
της, τον τύπο δεδομένων δηλ. της τιμής που επιστρέφει */
printf("\nΔώσε τον πρώτο αριθμό : ");
scanf("%d", &a);
printf("\nΔώσε τον δεύτερο αριθμό : ");
scanf("%d", &b);
printf("\nΔώσε τον τρίτο αριθμό : ");
scanf("%d", &c);
max = maximum(a, b, c); /* καλείται η συνάρτηση */
printf("\nΗ μεγαλύτερη τιμή είναι : ", max);
scanf("%d", &a);
} /* end of main */

/* εδώ γράφουμε τις εντολές της συνάρτησης */
```

```

int maximum(a1, b1, c1)
int a1, b1, c1;
/* Τα a1, b1 και c1 είναι τα ορίσματα της συνάρτησης, αλλά είναι και
τοπικές μεταβλητές και εδώ δηλώνεται ο τύπος δεδομένων τους */
{
int max1;
max1=a1;
if (b1>max1)
    max=b1;
if (c1>max1)
    max=c1;
return max1;
} /* end of sum() */

/* Να καταχωρούνται τιμές σε μια ακέραια μεταβλητή και σε μια
μεταβλητή χαρακτήρα και μετά να καλείται μια συνάρτηση τύπου void που
θα εκτυπώνει τον χαρακτήρα τόσες φορές όσο είναι η τιμή της ακέραιας
μεταβλητής - εκμάθηση των συναρτήσεων τύπου void */

#include <stdio.h>
main()
{
int a;
char ch;
void ektyp();
/* μια συνάρτηση τύπου void δεν επιστρέφει καμία τιμή ή επηρεάζει
δύο ή περισσότερες τιμές στο κυρίως πρόγραμμα και όταν τη δηλώνουμε,
γράφουμε πριν από το όνομά της τη λέξη void */
clrscr();
printf("\nΔώσε τον αριθμό : ");
scanf("%d", &a);
printf("\nΔώσε τον χαρακτήρα : ");
scanf("%c", &ch);
ektyp(a, ch); /* μια συνάρτηση τύπου void καλείται με απλή αναγραφή
του ονόματός της και των ορισμάτων της */
scanf("%d", &a);
} /* end of main */

void ektyp(a1, ch1)
int a1;
char ch1;
{
int i;
for (i=1; i<=a1; i++)
    printf(ch);
/* μια συνάρτηση τύπου void δεν έχει εντολή return */
} /* end of ektyp() */

```

Αναδρομικές Συνάρτησεις

- Ορισμός συνάρτησης μέσω κλήσης του εαυτού της
- Μία τεχνική επίλυσης προβλημάτων
- Π.χ. εύρεση αθροίσματος $1+2+\dots+n$
- Κλασική λύση

```

int sum (int n) {
s=0;
for (i=1; i<=n; i++)
    s=s+i;

```

```
return s;}
```

- Αναδρομική λύση
 $\text{sum}(n) = \text{sum}(n-1) + n$

```
int sum (int n) {  
if (n<=1)  
return n;  
Else  
return (sum(n-1)+n);}
```

Ασκήσεις

- Συνάρτηση υπολογισμού του παραγοντικού ενός ακεραίου
- Συνάρτηση υπολογισμού αθροίσματος, διαφοράς και γινομένου δύο αριθμών.
- Υπολογισμός ύψωσης δύναμης ακεραίου σε μη αρνητικό ακέραιο
- Υπολογισμός n -οστού όρου ακολουθίας Fibonacci

1. Να γίνει ένα πρόγραμμα το οποίο να ζητάει έναν ακέραιο από το πληκτρολόγιο, έστω το n , και να επιστρέφει στην οθόνη το n -παραγοντικό ($n!$). Ο υπολογισμός του παραγοντικού να γίνει σε ξεχωριστή συνάρτηση πρώτα μη αναδρομικά και κατόπιν αναδρομικά.

2. Να γίνει μια αναδρομική συνάρτηση για την ύψωση ενός ακεραίου σε έναν άλλο και να συγκριθεί με τις προηγούμενες συναρτήσεις που έκαναν τους ίδιους υπολογισμούς σε μη αναδρομική μορφή. Επίσης, να γίνει πρόγραμμα που να την χρησιμοποιεί.

(δοκιμάστε και με αρνητικό εκθέτη)

3. Να γίνει μία συνάρτηση, με αναδρομική μορφή, που να υπολογίζει και να επιστρέφει την τιμή του n -οστού όρου της ακολουθίας του Fibonacci. Κατόπιν, να γραφεί πρόγραμμα που να την χρησιμοποιεί.

