

Δομημένος Προγραμματισμός

Τμήμα Επιχειρηματικού Σχεδιασμού και
Πληροφοριακών Συστημάτων

www.bpis.teicrete.gr

Συναρτήσεις στη C

Με τον όρο «συνάρτηση» ορίζεται μία ακολουθία εντολών που έχουν **ομαδοποιηθεί και εκτελούνται με την κλήση ενός συμβολικού ονόματος**

Στη C κάθε συνάρτηση είναι ένα μικρό πρόγραμμα, με τις δικές του δηλώσεις μεταβλητών, και τις δικές του εντολές.

Μια συνάρτηση

- έχει συγκεκριμένο όνομα (κλήση συνάρτησης με το όνομα της)
- είναι ανεξάρτητη,
- εκτελεί συγκεκριμένη εργασία
- μπορεί να επιστρέψει κάποιο αποτέλεσμα

Συναρτήσεις - Functions

Η έννοια των λειτουργικών μονάδων προγράμματος
νέες συναρτήσεις

υπάρχουσες συναρτήσεις (`scanf()`, `printf()`, `pow()`...)

Φιλοσοφία του «διαίρει και βασίλευε»
(`divide and conquer`)

Χωρίζουμε το πρόγραμμα σε μέρη

Τα μέρη αντιστοιχούν σε επιμέρους και μικρότερα
προβλήματα

Αναλογία με μια επιχείρηση:

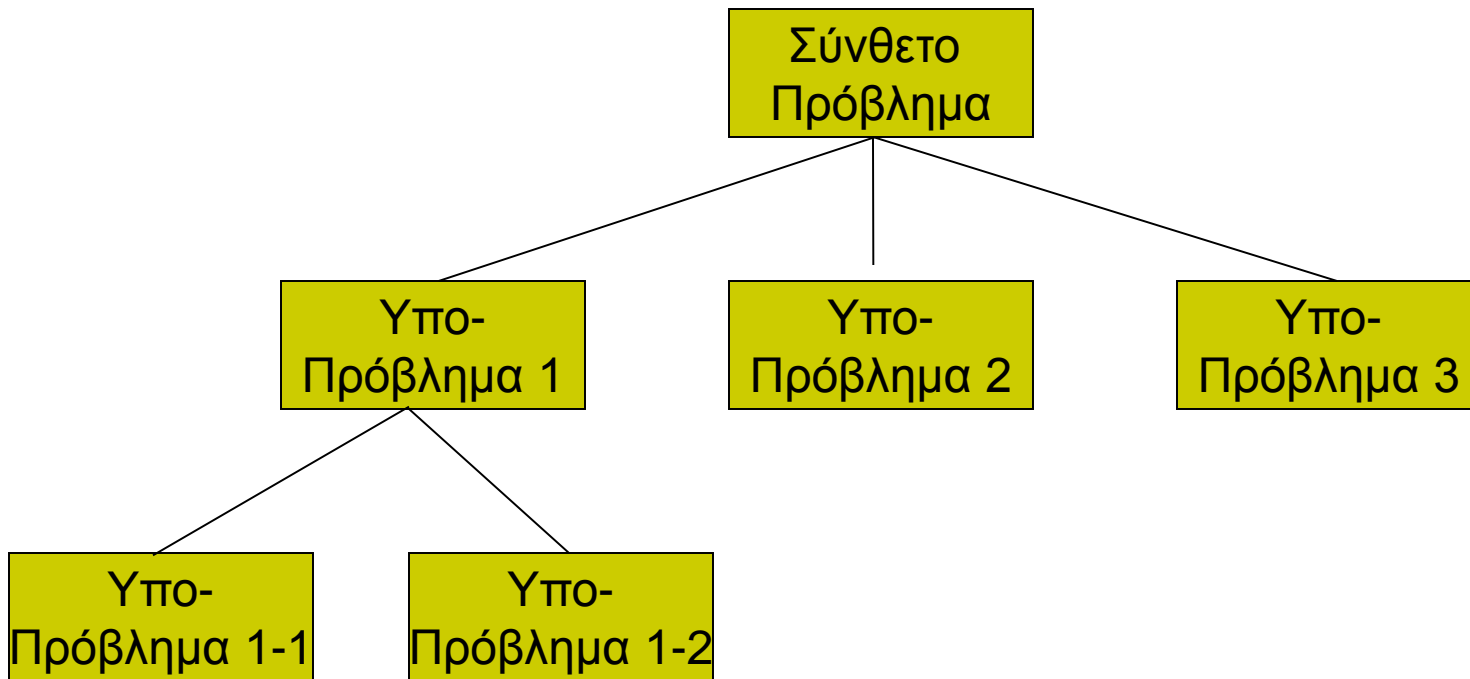
- Ο προϊστάμενος αναθέτει μια δουλειά σε κάποιον υφιστάμενο
- Παίρνει το αποτέλεσμα χωρίς να «ενδιαφέρεται» για τις λεπτομέρειες

Συναρτήσεις – Functions Πλεονεκτήματα

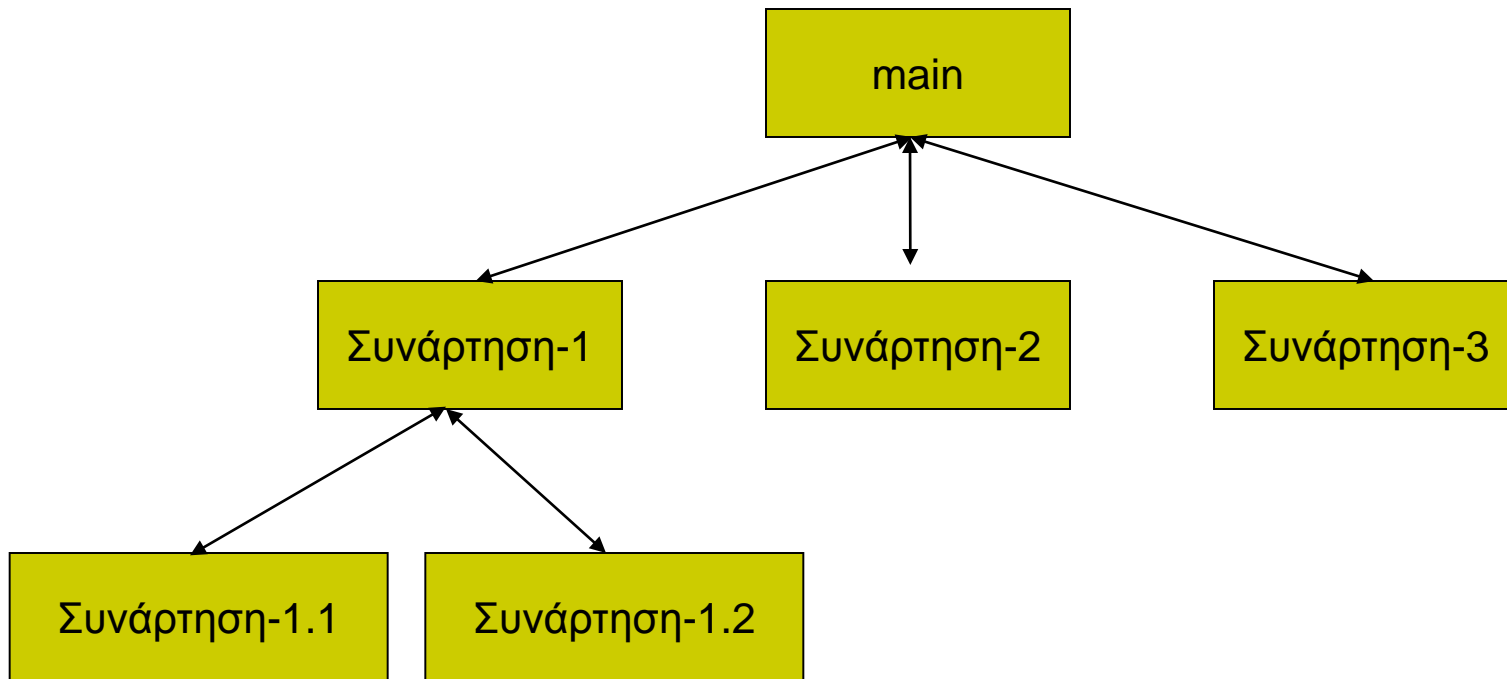
Τα μικρότερα προβλήματα είναι πιο εύκολο να τα διαχειριστούμε νοητικά

- Το πρόβλημα διαιρείται (divide) σε δύο ή περισσότερα υπο-προβλήματα
 - Προτιμάται τα υπο-προβλήματα να έχουν περίπου το ίδιο μέγεθος
- Το κάθε υπο-πρόβλημα λύνεται ξεχωριστά (conquer)
- Εάν χρειάζεται, οι λύσεις των υπο-προβλημάτων συνδυάζονται για να βρεθεί η λύση στο αρχικό πρόβλημα (conquer)
- Κάθε υπο-πρόβλημα αντιστοιχεί σε μία **συνάρτηση**

Μεθοδολογία επίλυσης προβλήματος



Οργάνωση Προγράμματος



Συναρτήσεις – Functions Πλεονεκτήματα

- **Πιο εύκολη διαχείριση κώδικα**
 - Debugging
 - Updating
- **Επαναχρησιμοποίηση κώδικα**
 - Χρήση υπαρχόντων συναρτήσεων σε καινούρια προγράμματα για τη λύση νέων προβλημάτων (code reuse)
- **Αποφυγή επανάληψης του ίδιου κώδικα πολλές φορές μέσα σε ένα πρόγραμμα**
- **Αφαίρεση (abstraction)**
 - «κρύβονται» λεπτομέρειες που δεν είναι απαραίτητες στον προγραμματιστή

Ορολογία σχετικά με Συναρτήσεις

Πρωτότυπο συνάρτησης (**function declarations**)

Δήλωση/Ορισμός συνάρτησης (**function definitions**)

Παράμετροι συνάρτησης (**formal parameters**)

Ορίσματα συνάρτησης (**actual parameters**)

Σώμα συνάρτησης (**function body**)

Κλήση συνάρτησης (**function call**)

Επιστροφή τιμών (**return values**)

Μαθηματικές Συναρτήσεις

- Η βιβλιοθήκη `math.h` διαθέτει μια σειρά από χρήσιμες μαθηματικές συναρτήσεις

Συνάρτηση	Περιγραφή
<code>sqrt(x)</code>	Τετραγωνική ρίζα, $x \geq 0$
<code>pow(x,y)</code>	x^y
<code>log(x)</code>	Λογάριθμος του x , $x > 0$
<code>exp(x)</code>	e^x
<code>fabs(x)</code>	Απόλυτη τιμή του x
<code>cos(x)</code>	Συνημίτονο x
<code>sin(x)</code>	Ημίτονο x
<code>tan(x)</code>	Εφαπτομένη x

Χρήση “προκατασκευασμένων” Συναρτήσεων

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

```
int main()
{
    double riza;
    double num=9.0;

    riza = sqrt(num);

    printf("%f",riza);
    return 0;
}
```

Άσκηση 1 Μαθηματικές Συναρτήσεις

Υπολογίστε την περιφέρεια και το εμβαδόν ενός κυκλικού δίσκου με ακτίνα που του δίνουμε εμείς από το πληκτρολόγιο, χρησιμοποιώντας τη συνάρτηση pow(x,y).

$$\text{Περιφέρεια δίσκου} = 2\pi r$$

$$\text{Εμβαδόν δίσκου} = \pi r^2$$

Άσκηση 1 Μαθηματικές Συναρτήσεις

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define PI 3.14159

int main()
{
    int aktina;
    double periferia, embadon;

    printf("Doste tin aktina toy kyklikoy diskoy:");
    scanf("%d", &aktina);

    periferia = 2 * PI * aktina;
    embadon = PI * pow(aktina, 2);

    printf("O diskos me aktina: %d\n", aktina);
    printf("exei periferia: %.3f\n", periferia);
    printf("kai embadon: %.3f\n", embadon);

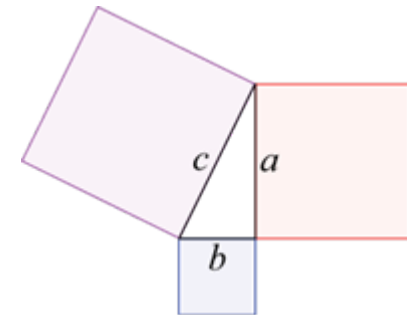
    return 0;
}
```

Άσκηση 2 Μαθηματικές Συναρτήσεις

Υπολογίστε την υποτείνουσα ενός τριγώνου χρησιμοποιώντας το Πυθαγόρειο θεώρημα καθώς και το εμβαδόν του τριγώνου χρησιμοποιώντας τη συνάρτηση `sqrt`.

Για τον υπολογισμό της υποτείνουσας χρησιμοποιήστε το Πυθαγόρειο θεώρημα:

$$c = \sqrt{a^2 + b^2}.$$



- Για την **τετραγωνική ρίζα** χρησιμοποιήστε τη συνάρτηση `sqrt()`
- Για το **a²** και το **b²** χρησιμοποιήστε τη συνάρτηση `pow(x,y)`

Άσκηση 2 Μαθηματικές Συναρτήσεις

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

int main()
{
    float sidea, sideb, sidec, area;

    printf("Doste tin pleyra A toy orthogoniou trigonou:");
    scanf("%f", &sidea);
    printf("Doste tin pleyra B toy orthogoniou trigonou:");
    scanf("%f", &sideb);

    sidec = sqrt(pow(sidea,2)+pow(sideb,2));    /* ypologismos pleyras c */
    area = 0.5 * sidea * sideb;                /* ypologismos embadou */

    printf("To trigono me pleyres %.3f kai %.3f\n", sidea, sideb);
    printf("exei ypotinousa: %.3f\n", sidec);
    printf("kai embadon: %.3f\n", area);

    return 0;
}
```

Άσκηση 3 Μαθηματικές Συναρτήσεις

Γράψτε πρόγραμμα σε C που να διαβάζει σε έναν πίνακα τις θερμοκρασίες μιας εβδομάδας από το πληκτρολόγιο σαν δεκαδικούς αριθμούς και στη συνέχεια υπολογίζει τη μέγιστη θερμοκρασία κατά την απόλυτη τιμή χρησιμοποιώντας τη συνάρτηση `fabs()`.

Άσκηση 3 Μαθηματικές Συναρτήσεις

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define WEEK 7

main()
{
    float temp[WEEK];
    int i, max = 0;

    printf("Doste tis thermokrasies tis ebdomadas:\n");
    for (i = 0; i < WEEK; i++)
    {
        printf("%d> ", i+1);
        scanf("%f", &temp[i]);
    }

    for (i = 0; i < WEEK; i++)
    {
        if ( fabs(temp[i]) > fabs(temp[max]))
            max = i;
    }
    printf("H megisti kata apolyto thermokrasia tis ebdomadas einai h %.3f thn %d hmera\n", temp[max], max+1);
    return 0;
}
```

Ορισμός Συναρτήσεων

Επιστρεφόμενος-τύπος όνομα-συνάρτησης (λίστα-παραμέτρων)

```
{  
  Δηλώσεις μεταβλητών;  
  Εντολές;  
}
```

Όνομα-συνάρτησης

οποιοδήποτε δεκτό όνομα

Επιστρεφόμενος-τύπος

τύπος των δεδομένων που επιστρέφει η συνάρτηση

void : δεν επιστρέφει τίποτα

Λίστα-παραμέτρων

δηλώσεις παραμέτρων, χωρίζονται με κόμμα (τύπος παράμετρος, τύπος παράμετρος, ...)

Ορισμός Συναρτήσεων

```
int Squarelt(int a)
{
    int result;
    result = a*a;
    return result;
}
```

Ορισμός Συναρτήσεων

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

```
int SquareIt(int a)
{
    int result;
    result = a*a;
    return result;
}
```

```
void main()
{
    int num, square;
    printf("Enter a number:");
    scanf(" %d", &num );
    square = SquareIt( num );
    printf("The square of %d is: %d", num, square);
}
```

Ορισμός Συνάρτησης

Σώμα Συνάρτησης

Επιστροφή τιμής
Συνάρτησης

Κλήση Συναρτήσεων - Function Call

Μία συνάρτηση ενεργοποιείται όταν κληθεί. Εάν η συνάρτηση δεν επιστρέφει τιμή, η κλήση της γίνεται από ένα σημείο του προγράμματος ως εξής:

```
όνομα_συναρτησης ( όρισμα1, όρισμα2, ..., όρισμαN );
```

αν επιστρέφει τιμή η κλήση της γίνεται με τον ίδιο τρόπο αλλά η συνάρτηση πλέον θα πρέπει να συμμετέχει σε κάποια έκφραση

Κλήση Συναρτήσεων - Function Call

όνομα_συνάρτησης (όρισμα1, ..., όρισμαN);

τιμή ← η τιμή που επιστρέφει ο κώδικας της
συνάρτησης όταν εκτελεστεί με τα δοσμένα
ορίσματα

```
x = SquareIt(5);
```

Στη μεταβλητή x θα ανατεθεί η τιμή που επιστρέφει η
συνάρτηση SquareIt(5)

→ 25

Ορίσματα συναρτήσεων

όνομα συνάρτησης (όρισμα1, όρισμα2, ..., τελευταίο όρισμα);

Οι τιμές **όρισμα1, όρισμα2, ..., τελευταίο όρισμα** καλούνται ορίσματα, χωρίζονται με κόμμα και περικλείονται σε παρενθέσεις, αντιστοιχούν δε ένα προς ένα στη λίστα των παραμέτρων της συνάρτησης που ορίστηκαν κατά τη δήλωσή της.

Ακόμη κι όταν δεν υπάρχουν ορίσματα οι παρενθέσεις είναι υποχρεωτικές, καθώς δηλώνουν στο μεταγλωττιστή ότι το όνομα αντιπροσωπεύει συνάρτηση και όχι μεταβλητή.

Τα ορίσματα μπορούν να είναι σταθερές, τιμές μεταβλητών ή ακόμη και τιμές εκφράσεων αλλά σε κάθε περίπτωση πρέπει να είναι ίδιου τύπου με τις παραμέτρους της συνάρτησης.

Ορίσματα συναρτήσεων

```
x=10;
```

```
print_max_two_integers( x+10, 32 );
```

```
<επόμενη πρόταση>;
```

Κλήση Συναρτήσεων

Μία συνάρτηση μπορεί να κληθεί με δύο τρόπους:

1. χρησιμοποιώντας το όνομά της και τη λίστα των ορισμάτων της σε μια αυτόνομη πρόταση π.χ.

wait(12);

2. χρησιμοποιώντας το όνομά της και τη λίστα των ορισμάτων της σε μια έκφραση π.χ.

k = SquareIt(10);

Ορισμός Συναρτήσεων

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

```
int SquareIt(int a)
{
    int result;
    result = a*a;
    return result;
}
```

Παράμετρος Συνάρτησης

Ορισμός Συνάρτησης

Σώμα Συνάρτησης

Τιμή Συνάρτησης

```
void main()
```

```
{
    int num, square;
    printf("Enter a number:");
    scanf(" %d", &num );
    square = SquareIt( num );
    printf("The square of %d is: %d", num,square);
}
```

Επιστρεφόμενη τιμή Συνάρτησης

Όρισμα Συνάρτησης

Κλήση Συνάρτησης

Συναρτήσεις με ή χωρίς επιστρεφόμενη τιμή

Εάν η συνάρτηση έχει έξοδο, αυτή θα πρέπει να επιστρέφεται με χρήση της εντολής **return** στο τέλος του σώματος της συνάρτησης.

π.χ. **return var; //var** η μεταβλητή με επιστρεφόμενη τιμή

Εάν δεν έχει έξοδο, ο <επιστρεφόμενος τύπος> αντικαθίσταται από τη λέξη **void**, π.χ.

```
void print_max_two_int( int first_integer, int second_integer )
{
    if (first_integer > second_integer)
        printf( "max=%d\n",first_integer );
    else
        printf( "max=%d\n",first_integer );
}
```

Συναρτήσεις void

Αν η συνάρτηση είναι τύπου void, δεν επιστρέφει τιμή

```
void f()  
{  
    printf("Test\n");  
}
```

Λάθος:

```
x = f() + 10;
```

Ροή του Προγράμματος

- Η ροή του προγράμματος αλλάζει όταν συναντήσει την κλήση μιας συνάρτησης
- $f(a, b, c)$
 - Πρώτα υπολογίζονται οι εκφράσεις που αντιστοιχούν στα ορίσματα
 - Οι τιμές των εκφράσεων a, b, c αντιγράφονται στα ορίσματα της f
 - Ο έλεγχος μεταπηδά στην πρώτη εντολή της f
 - Οι εντολές της f εκτελούνται μέχρι να συναντηθεί
 - `return` έκφραση;
 - Τέλος συνάρτησης (δηλαδή `}`)
 - Η κλήση $f(a, b, c)$ αντικαθίσταται με την επιστρεφόμενη τιμή

Θέματα σχετικά με Συναρτήσεις

Δηλώσεις μεταβλητών μέσα σε blocks: **ΤΟΠΙΚΕΣ** μεταβλητές

Συναρτήσεις **ΔΕΝ** μπορούν να οριστούν μέσα σε άλλες συναρτήσεις

ΟΠΟΙΑΔΗΠΟΤΕ συνάρτηση **μπορεί να καλέσει** άλλες συναρτήσεις και τον εαυτό της (αναδρομή)

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

function1

Άσκηση

Γράψτε ένα πρόγραμμα που θα τυπώνει δύο ακέραιους στη μορφή **[x , y]**

Ορίστε μία συνάρτηση **print_values** όπου θα τυπώνονται οι αριθμοί

Καλέστε τη συνάρτηση στο πρόγραμμα σας ώστε να τυπωθούν όλα τα δυνατά ζεύγη τιμών για τιμές x και y από 0 έως 5

Τροποποιήστε τη συνάρτηση σας ώστε όταν τυπωθούν όλες οι τιμές του y να τυπώνεται σε νέα γραμμή η συμβολοσειρά
"-----"

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

function2

Άσκηση

Γράψτε ένα πρόγραμμα που θα υπολογίζει την τρίτη δύναμη ενός ακέραιου αριθμού

Ορίστε μία συνάρτηση **power** όπου θα υπολογίζεται η δύναμη

καλέστε τη συνάρτηση στο πρόγραμμα σας, ώστε να υπολογιστούν και να εμφανιστούν στην οθόνη οι κύβοι των αριθμών από το 1 ως το 10

Άσκηση

Τροποποιήστε τη συνάρτηση ώστε να δέχεται δύο ορίσματα. Το πρώτο είναι ο αριθμός και το δεύτερο θα παίρνει τιμές 2 και 3. Αν είναι 2 θα υπολογίζεται το τετράγωνο του αριθμού, ενώ αν είναι 3 ο κύβος του αριθμού.

Καλέστε τη συνάρτηση στο πρόγραμμά σας, ώστε να υπολογιστούν και να εμφανιστούν στην οθόνη τα τετράγωνα και οι κύβοι των αριθμών από το 1 ως το 10