



**Δ.Π.Θ** Δομημένος Προγραμματισμός

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ**

# Δείκτες σε Συναρτήσεις

**Dr. Αθανάσιος Μπαλαφούτης**  
Εργαστηριακό Διδακτικό Προσωπικό  
Τομέας Συστημάτων Παραγωγής  
Εργαστήριο Ρομποτικής και Αυτοματισμών  
abalafou@pme.duth.gr  
Γραφείο 304, τηλ.: 25410 – 79892

# Κλήση Συνάρτησης με Τιμή

Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int a){
    a = a + 1;
}

int main(){
    int a = 10;

    Increment(a);

    printf("a = %d\n", a);
}
```

## Κλήση με Τιμή (Call by Value)

- Ονομάζεται έτσι επειδή η τιμή της μεταβλητής περνά στη συνάρτηση.
- Οι αλλαγές που γίνονται μέσα στη συνάρτηση δεν επηρεάζουν την αρχική μεταβλητή.

# Κλήση Συνάρτησης με Τιμή



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Η μεταβλητή **a** εντός της `Increment`, δεν είναι ίδια με την **a** στη `main` !

```
#include <stdio.h>

void Increment(int a){
    a = a + 1;
    printf("Διεύθυνση του a στην Increment: %p\n", &a);
}

int main(){
    int a = 10;

    Increment(a);

    printf("Διεύθυνση του a στη main: %p\n", &a);
}
```

# Κλήση Συνάρτησης με Τιμή



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

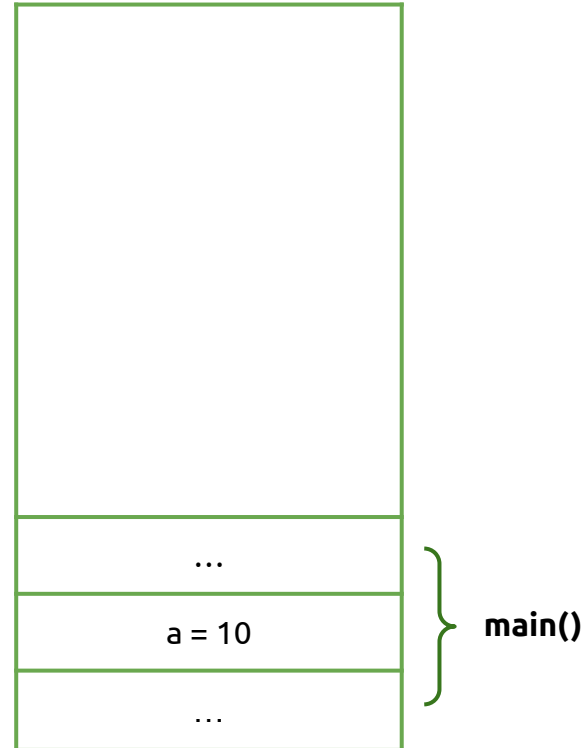
Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int a){
    a = a + 1;
}

int main(){
    int a = 10; ←
    Increment(a);
    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Τιμή



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

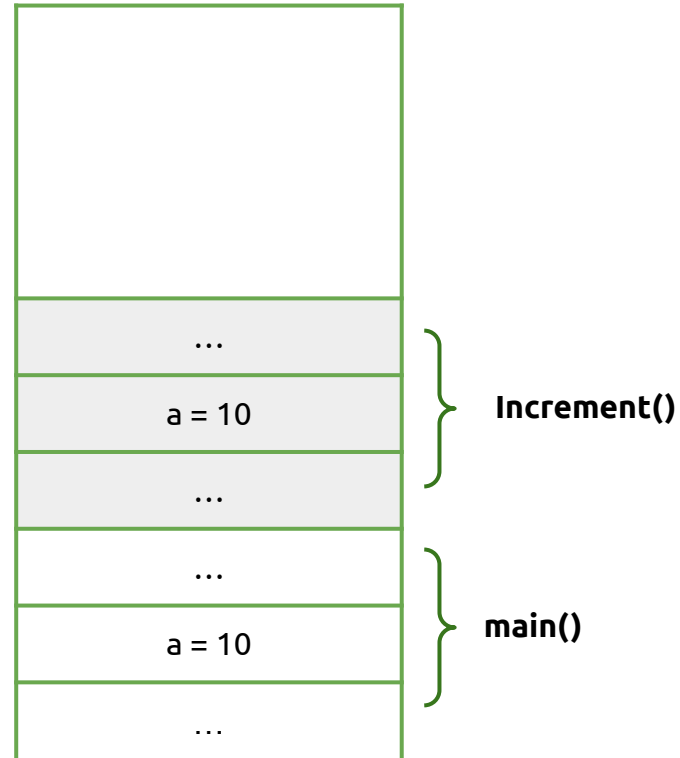
Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int a){ ←
    a = a + 1;
}

int main(){
    int a = 10;
    Increment(a);
    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Τιμή



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

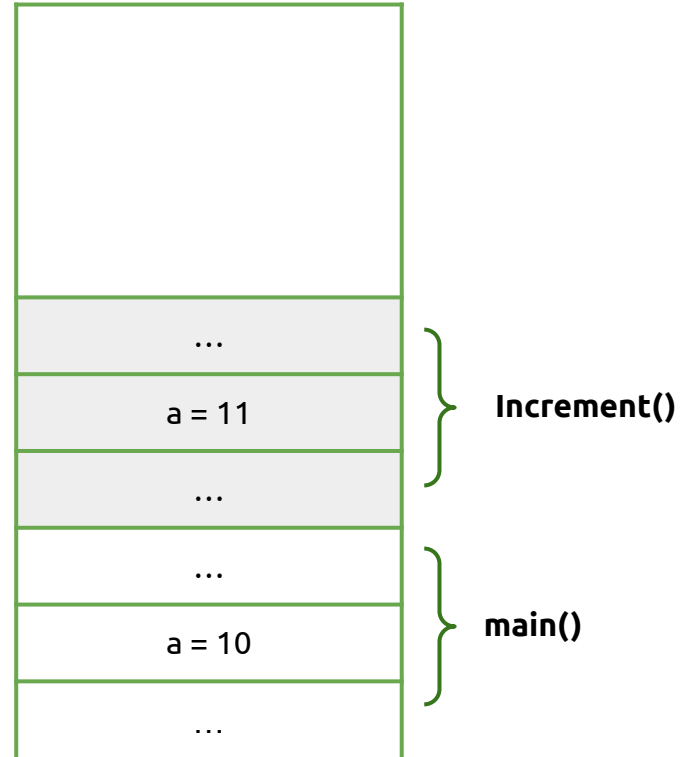
Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int a){
    a = a + 1;
}

int main(){
    int a = 10;
    Increment(a);
    printf("a = %d\n", a);
}
```

Στοιβά (Stack)



# Κλήση Συνάρτησης με Τιμή



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

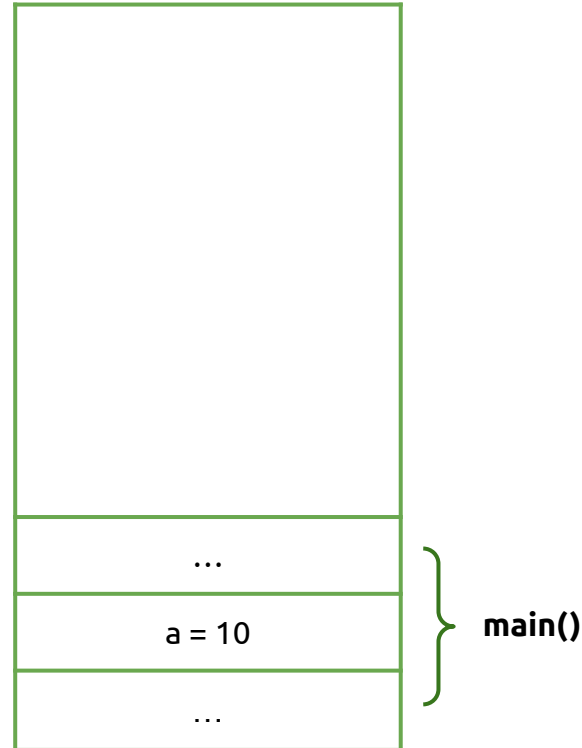
Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int a){
    a = a + 1;
}

int main(){
    int a = 10;
    Increment(a);
    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Τιμή



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

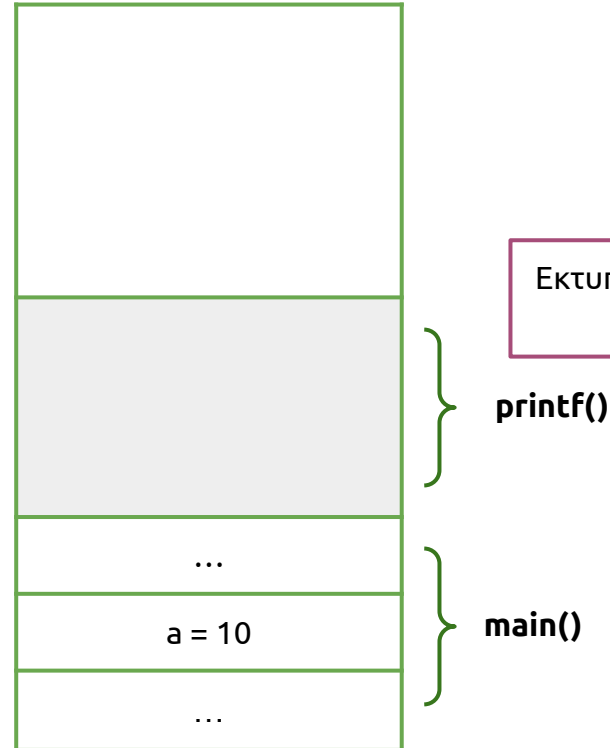
Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int a){
    a = a + 1;
}

int main(){
    int a = 10;
    Increment(a);
    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Τιμή



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int a){
    a = a + 1;
}

int main(){
    int a = 10;
    Increment(a);
    printf("a = %d\n", a);
} ←
```

Στοίβα (Stack)



Εκτυπώνει στην οθόνη:  
a = 10

Απελευθερώνεται η  
μνήμη με το τέλος του  
προγράμματος

# Κλήση Συνάρτησης με Αναφορά



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int *p){
    *p = *p + 1;
}

int main(){
    int a = 10;

    Increment(&a);

    printf("a = %d\n", a);
}
```

## Κλήση συνάρτησης με αναφορά (call by reference)

- Ονομάζεται έτσι επειδή η διεύθυνση μνήμης της μεταβλητής περνά στη συνάρτηση.
- Οι αλλαγές που γίνονται μέσα στη συνάρτηση επηρεάζουν την αρχική μεταβλητή.
- Χρησιμοποιείται με δείκτες (pointers)

# Κλήση Συνάρτησης με Αναφορά



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

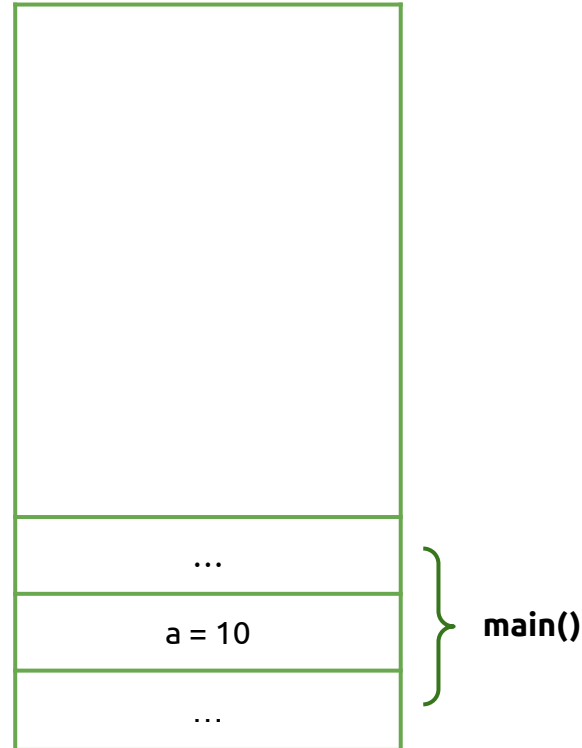
```
#include <stdio.h>

void Increment(int *p){
    *p = *p + 1;
}

int main(){
    int a = 10; ←
    Increment(&a);

    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Αναφορά

Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

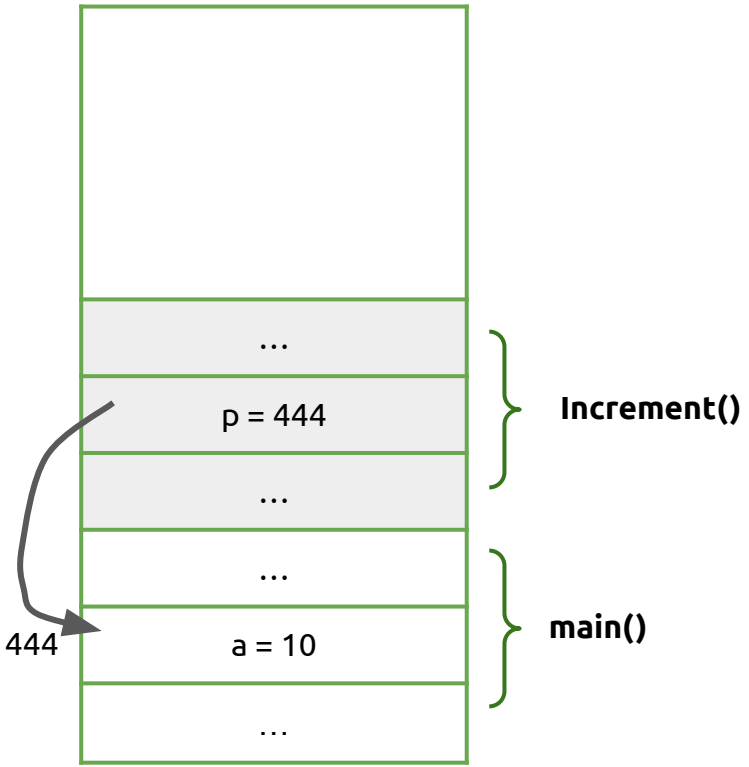
void Increment(int *p){ ←
    *p = *p + 1;
}

int main(){
    int a = 10;

    Increment(&a);

    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Αναφορά

Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

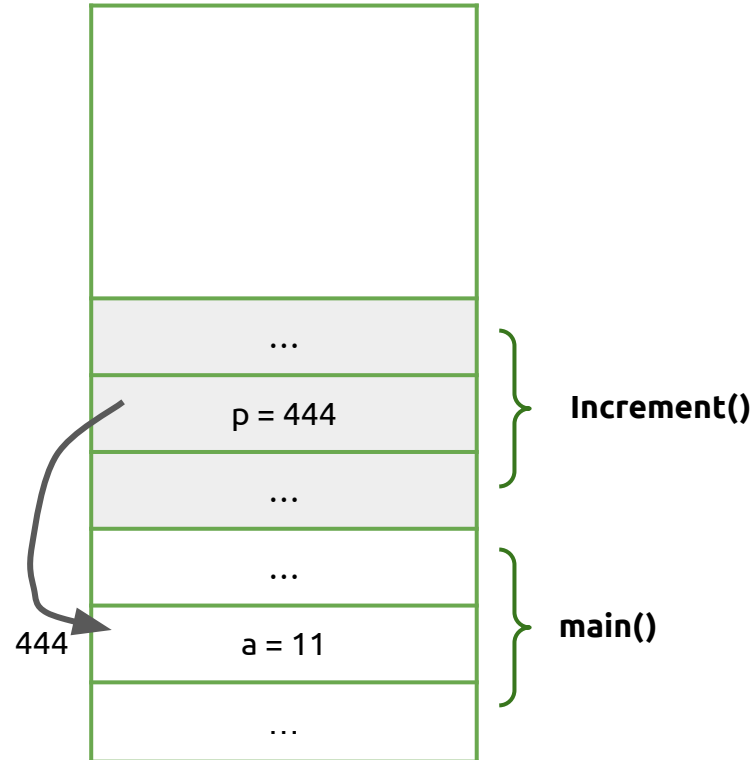
void Increment(int *p){
    *p = *p + 1;
}

int main(){
    int a = 10;

    Increment(&a);

    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Αναφορά



Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

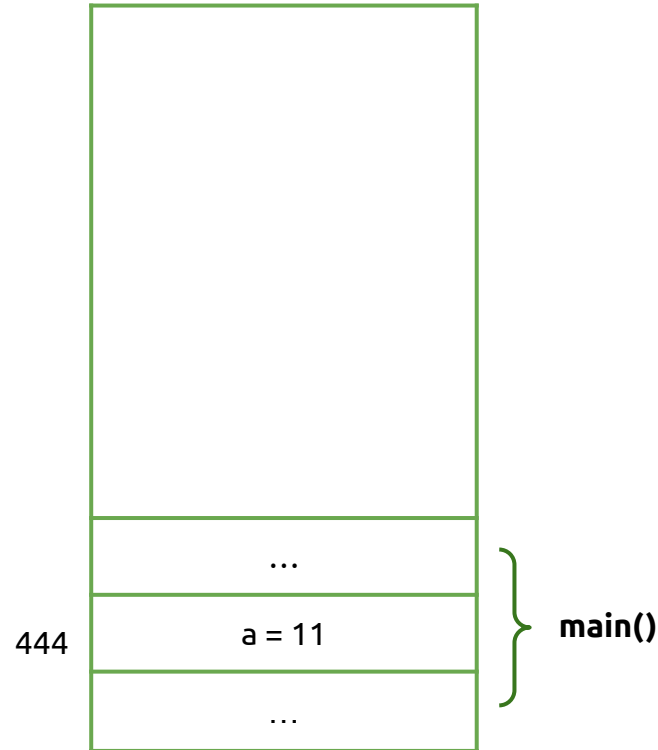
void Increment(int *p){
    *p = *p + 1;
}

int main(){
    int a = 10;

    Increment(&a); ←

    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Αναφορά



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

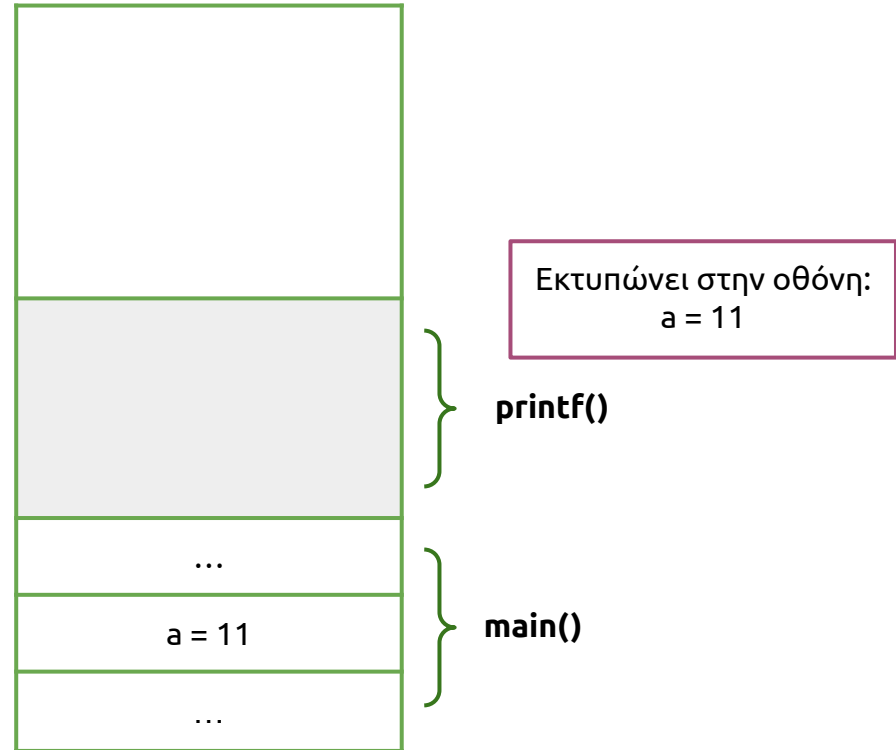
void Increment(int *p){
    *p = *p + 1;
}

int main(){
    int a = 10;

    Increment(&a);

    printf("a = %d\n", a); ←
}
```

Στοίβα (Stack)



# Κλήση Συνάρτησης με Αναφορά



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Τι θα εκτυπώσει το παρακάτω πρόγραμμα;

```
#include <stdio.h>

void Increment(int *p){
    *p = *p + 1;
}

int main(){
    int a = 10;

    Increment(&a);

    printf("a = %d\n", a);
}
```

Στοίβα (Stack)



Εκτυπώνει στην οθόνη:  
a = 11

Απελευθερώνεται η  
μνήμη με το τέλος του  
προγράμματος

# Άσκηση 2.5



Να γράψετε συνάρτηση που θα δέχεται δύο ακέραιες μεταβλητές και θα αντιμεταθέτει τις τιμές τους.

Στη συνέχεια να γραφεί κύριο πρόγραμμα που θα διαβάζει 2 ακέραιους αριθμούς, θα καλεί τη συνάρτηση και θα εμφανίζει τις τιμές τους μετά την αντιμετάθεση,

Π.χ. αν έχουμε τις τιμές  $a=5$  και  $b=3$ , μετά την εκτέλεση της συνάρτησης οι μεταβλητές αυτές θα περιέχουν τις τιμές:  $a=3$  και  $b=5$ .

# Άσκηση 2.5 - swap



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int swap(int *x, int *y){

    int z;

    z = *x;
    *x = *y;
    *y = z;
    return 0;
}
```

# Άσκηση 2.5 - main



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int swap(int *x, int *y){

    int z;

    z = *x;
    *x = *y;
    *y = z;
    return 0;
}
```

```
int main(){

    int a, b;
    printf("Δώσε τις τιμές στα a και b: ");
    scanf("%d %d", &a, &b);

    swap(&a, &b);
    printf("Νέα τιμή της a: %d\n", a);
    printf("Νέα τιμή της b: %d\n", b);
    return 0;
}
```

# Άσκηση 2.6



Σε ένα απλό αισθητήρα μιας συσκευής, ο χρόνος μετριέται σε δευτερόλεπτα από τα μεσάνυχτα. Έτσι για παράδειγμα, όταν το ρολόι της συσκευής δείχνει 4812, η ώρα είναι 1:20:12. Γράψτε μια συνάρτηση `GetTime` με το ακόλουθο πρωτότυπο:

```
void GetTime(long sysSecs, int *hours, int *minutes, int *seconds);
```

που να δέχεται έναν αριθμό `long` ο οποίος δηλώνει τα δευτερόλεπτα που αναφέρονται στο ρολόι της συσκευής και επιστρέφει την ώρα, τα λεπτά και τα δευτερόλεπτα που αντιστοιχούν στον κλασικό τρόπο μέτρησης του χρόνου της ημέρας.

Δοκιμάστε τη συνάρτηση, γράφοντας ένα απλό κυρίως πρόγραμμα που να διαβάσει το χρόνο σε δευτερόλεπτα, καλεί τη συνάρτηση και εμφανίζει την ώρα, όπως στο παράδειγμα.

# Άσκηση 2.6 - getTime



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

void getTime(long sysSecs, int *hours, int *minutes, int *seconds) {
    *hours = (sysSecs / 3600) % 24;    // Υπολογισμός ωρών (για 24ωρο σύστημα)
    *minutes = (sysSecs % 3600) / 60; // Υπολογισμός λεπτών
    *seconds = sysSecs % 60;          // Υπολογισμός δευτερολέπτων
}
```

# Άσκηση 2.6 - getTime



```
int main() {
    long secondsInput;
    int hours, minutes, seconds;

    printf("Εισάγετε το χρόνο σε δευτερόλεπτα από τα μεσάνυχτα: ");
    scanf("%ld", &secondsInput);

    // Κλήση της συνάρτησης
    getTime(secondsInput, &hours, &minutes, &seconds);

    // Εκτύπωση αποτελέσματος σε μορφή HH:MM:SS
    printf("Η ώρα είναι: %02d:%02d:%02d\n", hours, minutes, seconds);

    return 0;
}
```

## Άσκηση 2.7



Δ.Π.Θ

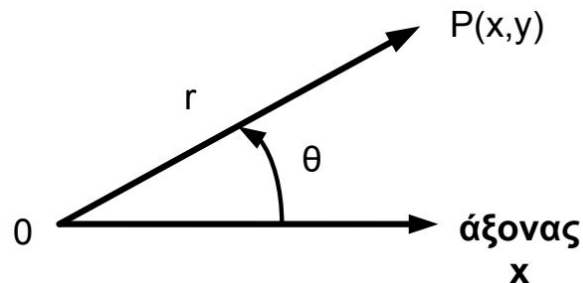
Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Να γραφεί μια συνάρτηση με όνομα `convert_to_polar` που να δέχεται ως είσοδο τις καρτεσιανές συντεταγμένες ενός σημείου  $x, y$  και να επιστρέφει τις πολικές συντεταγμένες του  $r, \theta$ . Οι πολικές συντεταγμένες υπολογίζονται σύμφωνα με τους τύπους :

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \tan^{-1} \left( \frac{y}{x} \right)$$



Η συνάρτηση θα καλείται από τη `main()`.

# Άσκηση 2.7 - convert\_to\_polar



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

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

// Συνάρτηση μετατροπής από καρτεσιανές σε πολικές συντεταγμένες
void convert_to_polar(double x, double y, double *r, double *theta) {
    *r = sqrt(x * x + y * y);           // Υπολογισμός της ακτίνας r
    *theta = atan2(y, x);               // Υπολογισμός της γωνίας θ σε ακίνια
}
```

$$r = \sqrt{x^2 + y^2} \quad \theta = \tan^{-1} \left( \frac{y}{x} \right)$$

# Άσκηση 2.7 - main



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
int main() {
    double x, y, r, theta;

    printf("Εισάγετε τις καρτεσιανές συντεταγμένες (x, y): ");
    scanf("%lf %lf", &x, &y);

    convert_to_polar(x, y, &r, &theta);

    // Εκτύπωση αποτελεσμάτων (μετατροπή γωνίας σε μοίρες)
    printf("Πολικές συντεταγμένες: r = %.2f, θ = %.2f μοίρες\n", r, theta * (180.0 / M_PI));

    return 0;
}
```

# Άσκηση 2.8



Να γράψετε δύο συναρτήσεις με ονόματα:

- `convert_to_cylindrical`
- `convert_to_spherical`

που να υπολογίζουν και να επιστρέφουν τις κυλινδρικές και σφαιρικές συντεταγμένες ενός σημείου  $(x, y, z)$ .

Κυλινδρικές

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} & x &= \rho \cos \varphi \\ \varphi &= \arctan(y, x) & y &= \rho \sin \varphi \\ z &= z & z &= z \end{aligned}$$

Σφαιρικές

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} & x &= r \sin \theta \cos \varphi \\ \theta &= \arctan(\sqrt{x^2 + y^2}, z) & y &= r \sin \theta \sin \varphi \\ \varphi &= \arctan(y, x) & z &= r \cos \theta \end{aligned}$$

Στη συνέχεια, να γραφεί πρόγραμμα που:

1. Θα δημιουργεί  $N$  σημεία  $(x, y, z)$  στο διάστημα  $[0, 100]$  χρησιμοποιώντας τη συνάρτηση `rand()`. Η του  $N$  είναι γνωστή και θα πρέπει να οριστεί ως σταθερά.
2. Θα καλεί τις δύο συναρτήσεις για καθένα από τα  $N$  σημεία και θα εμφανίζει τις τιμές που επιστρέφονται από τις συναρτήσεις.

Υπόδειξη: Όλες οι τιμές να είναι τύπου `double`.

# Άσκηση 2.8 - Κυλινδρικές



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#define N 10

void convert_to_cylindrical(double x, double y, double *r, double *phi){
    *r = sqrt(x*x + y*y);
    *phi = atan(y/x);
    *phi = 180.0 * (*phi) / M_PI;
}
```

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} & x &= \rho \cos\varphi \\ \varphi &= \arctan(y, x) & y &= \rho \sin\varphi \\ z &= z & z &= z \end{aligned}$$

# Άσκηση 2.8 - Σφαιρικές



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
void convert_to_spherical(double x, double y, double z, double* r, double* theta,
                          double* phi )
{
    *r = sqrt(x*x + y*y + z*z);
    *theta = atan(sqrt(x*x + y*y)/z);
    *theta = 180.0 * (*theta) / M_PI;
    *phi = atan(y/x);
    *phi = 180.0 * (*phi) / M_PI;
}
```

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} & x &= r \sin \theta \cos \varphi \\ \theta &= \arctan(\sqrt{x^2 + y^2}, z) & y &= r \sin \theta \sin \varphi \\ \varphi &= \arctan(y, x) & z &= r \cos \theta \end{aligned}$$

# Άσκηση 2.8 - main



```
int main(){
    double x, y, z, r, phi, theta;
    int lower, upper;
    srand(time(0));
    lower = 0;
    upper = 100;
    for(int i = 0; i < N; i++){
        x = rand() % (upper - lower + 1) + lower;
        y = rand() % (upper - lower + 1) + lower;
        z = rand() % (upper - lower + 1) + lower;
        convert_to_cylindrical(x, y, z, &r, &phi);
        printf("Κυλινδρικές Συντεταγμένες για (%.2f, %.2f, %.2f): ", x, y, z);
        printf("ρ = %.2f, φ = %.2f, z = %.2f \n", r, phi, z);
        convert_to_spherical(x, y, z, &r, &theta, &phi);
        printf("Σφαιρικές Συντεταγμένες για (%.2f, %.2f, %.2f): ", x, y, z);
        printf("r = %.2f, θ = %.2f, φ = %.2f \n", r, theta, phi);
    }
}
```