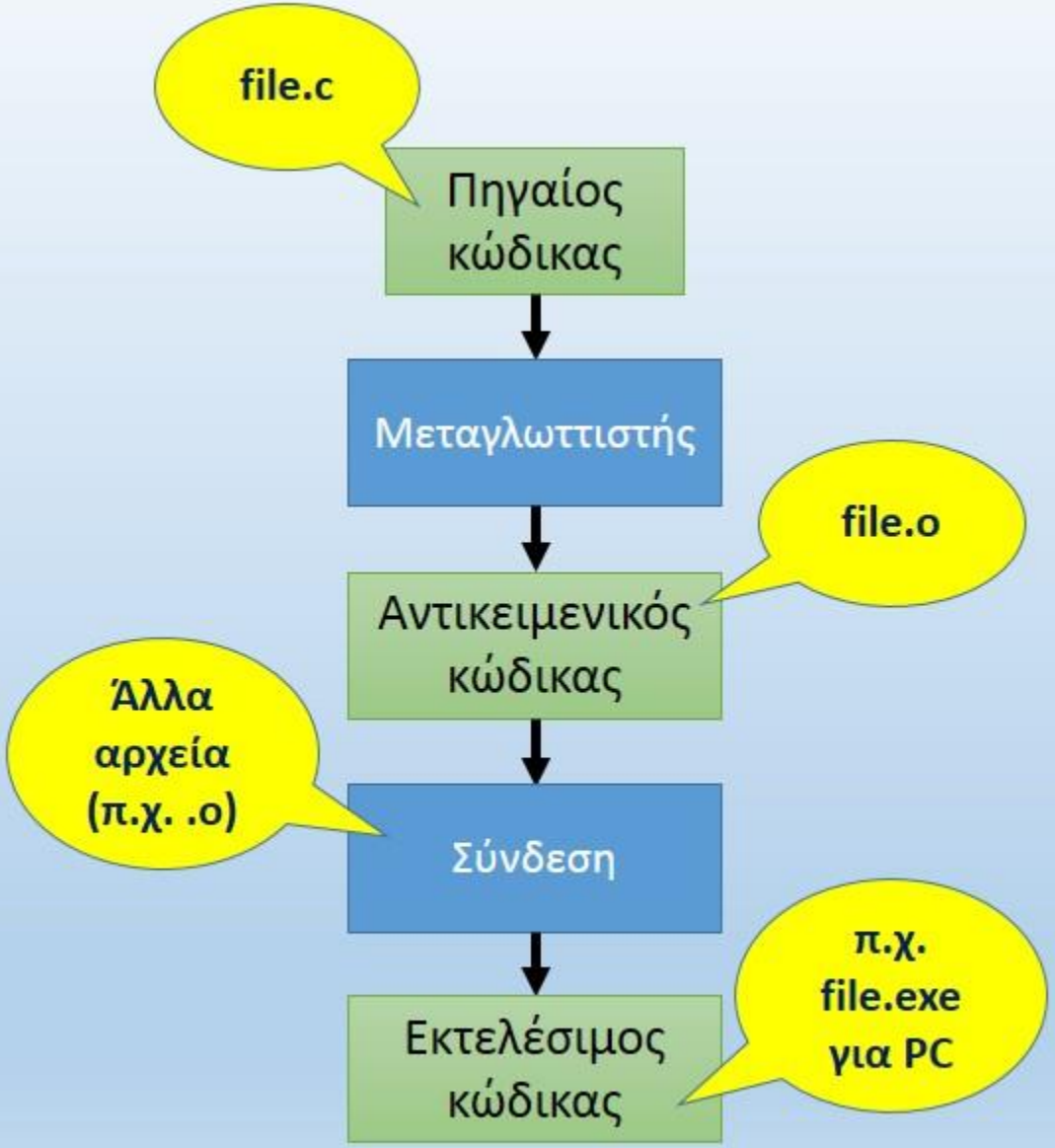
Εισαγωγή στη γλώσσα C

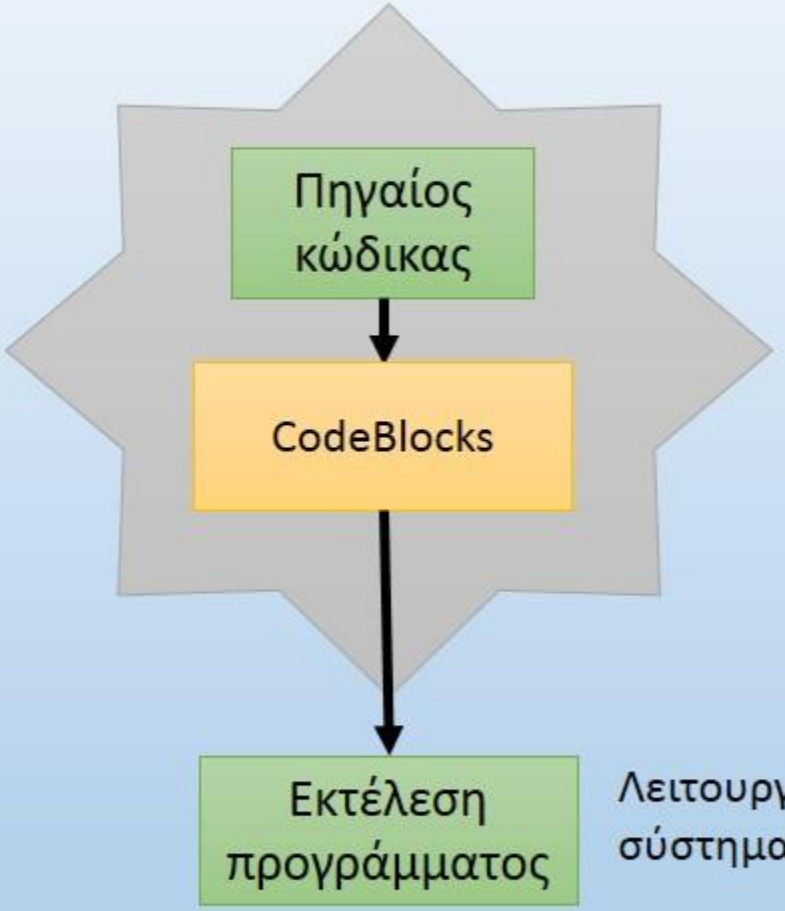
Βασικά παραδείγματα εισόδου/εξόδου
δεδομένων (πληκτρολόγιο/οθόνη)

Πρόχειρη παρουσίαση με πρακτικά παραδείγματα για χρήση στο εργαστήριο

Παναγιώτης Παπάζογλου
Αν. Καθηγητής

Από τον πηγαίο στον εκτελέσιμο κώδικα


file.c
Πηγαίος κώδικας
Μεταγλωττιστής
file.o
Αντικειμενικός κώδικας
Άλλα αρχεία (π.χ. .o)
Σύνδεση
π.χ. file.exe για PC
Εκτελέσιμος κώδικας


Πηγαίος κώδικας
CodeBlocks
Εκτέλεση προγράμματος
Λειτουργικό σύστημα

ΠΡΟΣΟΧΗ:

Δοκιμάστε όλα τα προγράμματα
των διαφανειών στο λογισμικό CodeBlocks

```
#include <stdio.h>
main()
{
   printf("Hello, this is my first program");
}
```

Το πρώτο μας πρόγραμμα!

Κάθε πρόγραμμα C περιέχει υποχρεωτικά τη συνάρτηση **main** από την οποία ξεκινά και η εκτέλεση του προγράμματος

Ενσωμάτωση βιβλιοθήκης. Περιέχει συναρτήσεις για είσοδο/ έξοδο (πληκτρολόγιο / Οθόνη)

```
#include <stdio.h>

main()

{
    printf("Hello, this is my first program");
}
```

ΑΡΧΗ και ΤΕΛΟΣ ενότητας main

Συνάρτηση εμφάνισης στην οθόνη

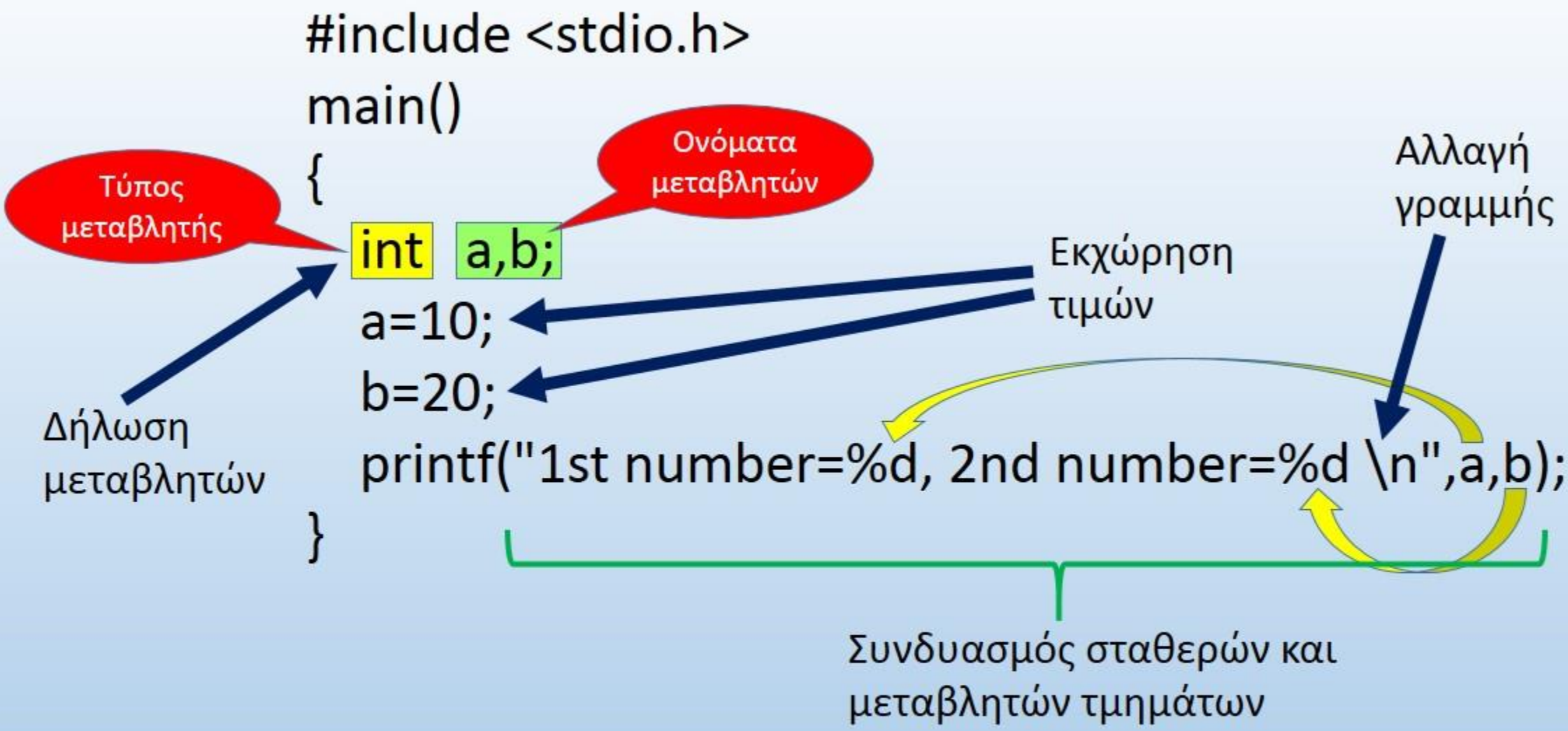
```
#include <stdio.h>
int main()
{
    printf("Hello, this is my first program");
    return 0;
}
```

Τύπος αποτελέσματος που θα επιστραφεί στο Λ.Σ. μετά το πέρας της εκτέλεσης της main

Επιστροφή τιμής στο Λ.Σ. μετά το πέρας της εκτέλεσης της main

Μορφοποιημένη έξοδος

```
#include <stdio.h>
main()
{
   int a,b;
   a=10;
   b=20;
   printf("1st number=%d, 2nd number=%d \n",a,b);
}
```



Αποτέλεσμα στην οθόνη

```
1st number=10, 2nd number=20
```

Μερικοί τύποι μεταβλητών

Τύπος	Byte (τυπικό)	Εύρος τιμών
char	1	-128..127
short int	2	-32,768..32,767
int	4	-2,147,483,648.. 2,147,483,647
long int	4	-2,147,483,648.. 2,147,483,647
float	4	1.2E-38 to 3.4E+38
double	8	2.3E-308 to 1.7E+308
unsigned char	1	0..255
unsigned short int	2	0..65535
unsigned int	4	0..4,294,967,295
unsigned long int	4	0..4,294,967,295

* Στην ANSI C ο τύπος int έχει εύρος -32768..32767

Μερικοί ειδικοί χαρακτήρες εκτύπωσης και μορφοποίησης

Χαρακτήρας μορφοποίησης	Περιγραφή
%d	Ακέραιος με πρόσημο
%c	Απλός χαρακτήρας
%f	Αριθμός κινητής υποδιαστολής
%s	Αλφαριθμητικό
%u	Μη προσημασμένος ακέραιος

Χαρακτήρας εκτύπωσης	Αποτέλεσμα εκτύπωσης
\n	Αλλαγή γραμμής
\b	Backspace
\t	Tab
\”	Διπλά εισαγωγικά

```
#include <stdio.h>
main()
{
  char c;
  short int si;
  int i;
  long int li;
  float f;
  double d;
  unsigned char uc;
  unsigned short int usi;
  unsigned int ui;
  unsigned long int uli;
  printf("%u %u %u %u %u %u %u %u %u
%u",sizeof(c), sizeof(si), sizeof(i), sizeof(li),
sizeof(f), sizeof(d), sizeof(uc), sizeof(usi),
sizeof(ui), sizeof(uli));
}
```

Χρήση του τελεστή **sizeof()** για την εύρεση δεσμευμένων byte ανά τύπο μεταβλητής

Αποτέλεσμα

1 2 4 4 4 8 1 2 4 4

Άσκηση 1 – πρόσθεση δύο ακέραιων και δύο πραγματικών αριθμών

```
#include <stdio.h>
main()
{
  int a, b, ab;
  float c, d, cd;
  a=124;
  b=91;
  c=33.8976;
  d=12.9899;
  ab=a+b;
  cd=c+d;
  printf("a=%d, b=%d, ab=%d\n",a,b,ab);
  printf("c=%2.3f, d=%2.2f, cd=%2.2f",c,d,cd);
}
```

```
#include <stdio.h>
main()
{
  int a=124, b=91, ab;
  float c=33.8976, d=12.989, cd;
  ab=a+b;
  cd=c+d;
  printf("a=%d, b=%d, ab=%d\n",a,b,ab);
  printf("c=%2.3f, d=%2.2f, cd=%2.2f",c,d,cd);
}
```

Ψηφία ακέραιου μέρους

Ψηφία δεξιά της υποδιαστολής

```
a=124, b=91, ab=215
c=33.898, d=12.99, cd=46.89
```

Άσκηση 2 – Τι θα εμφανίσουν τα ακόλουθα προγράμματα ;

2.1

```
#include <stdio.h>
main()
{
  int a1=123,a3;
  float b1=33.8976, b3, b4;

  a3=b1;
  b3=a1;
  b4=(float) a1;

  printf("a3=%d, b3=%f, b4=%3.0f\n",a3,b3,b4);
}
```

a3=33, b3=123.000000, b4=123

2.2

```
#include <stdio.h>
main()
{

  int a=12;
  float b=33.12;

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

Τυχαίο αποτέλεσμα γιατί είναι λανθασμένη η μορφοποίηση εξόδου

Άσκηση 2 – Τι θα εμφανίσουν τα ακόλουθα προγράμματα ;

2.3

```
#include <stdio.h>
main()
{
  int a=12, b=5;
  float c, d;
  c=a/b;
  d=(float)a/b;

  printf("12/5=%f\n", a/b);
  printf("12/5 float=%f\n",(float)a/b);
  printf("c=%f\n",c);
  printf("d=%f",d);
}
```

```
12/5=-1.#QNAN
12/5 float=2. 400000
c=2.000000
d=2.400000
```

Ανάγνωση από το πληκτρολόγιο

Για ανάγνωση αριθμητικών δεδομένων χρησιμοποιείται το **&**

```
#include <stdio.h>
main()
{
 int age, after10;
 float weight;
 char name[10];

 printf("Enter your age:");
 scanf("%d",&age);
 printf("Enter your name:");
 scanf("%s",name);
 printf("Enter your weight:");
 scanf("%f",&weight);
 after10=age+10;
 printf("Name:%s, Age=%d, After 10 years:%d, Weight:%2.2f",name,age,after10,weight);
}
```

```
Enter your age:47
Enter your name:Panos
Enter your weight:78.4
Name:Panos, Age=47, After 10 years:57, Weight:78.40
```

Άσκηση 3 – Να γραφεί πρόγραμμα που θα διαβάζει τρεις αριθμούς κινητής υποδιαστολής και θα εμφανίζει το μέσο όρο

```
#include <stdio.h>
main()
{

  float a,b,c;
  printf("Enter three numbers:");
  scanf("%f %f %f",&a,&b,&c);
  float d=(a+b+c)/3;
  printf("Mean value=%2.2f",d);

}
```

Το αποτέλεσμα στη μορφή XX.KK

```
Enter three numbers:2.5 8.9 33.1
Mean value=14.83
```

Άσκηση 4 – Να γραφεί πρόγραμμα που θα εμφανίζει ξεχωριστά τα ψηφία ενός τριψήφιου αριθμού, καθώς και το αντίστοιχο άθροισμά τους.

```
#include <stdio.h>
main()
{

   int num, eka, dek, mon, sum;
   printf("N=");
   scanf("%d", &num);
   printf("N=%d\n",num);
   eka=(int)(num/100);
   num=num-eka*100;
   dek=(int)(num/10);
   mon=num-dek*10;
   sum=eka+dek+mon;
   printf("eka=%d, dek=%d, mon=%d, sum=%d",eka, dek,
mon, sum);
}
```

```
N=761
N=761
eka=7, dek=6, mon=1, sum=14
```

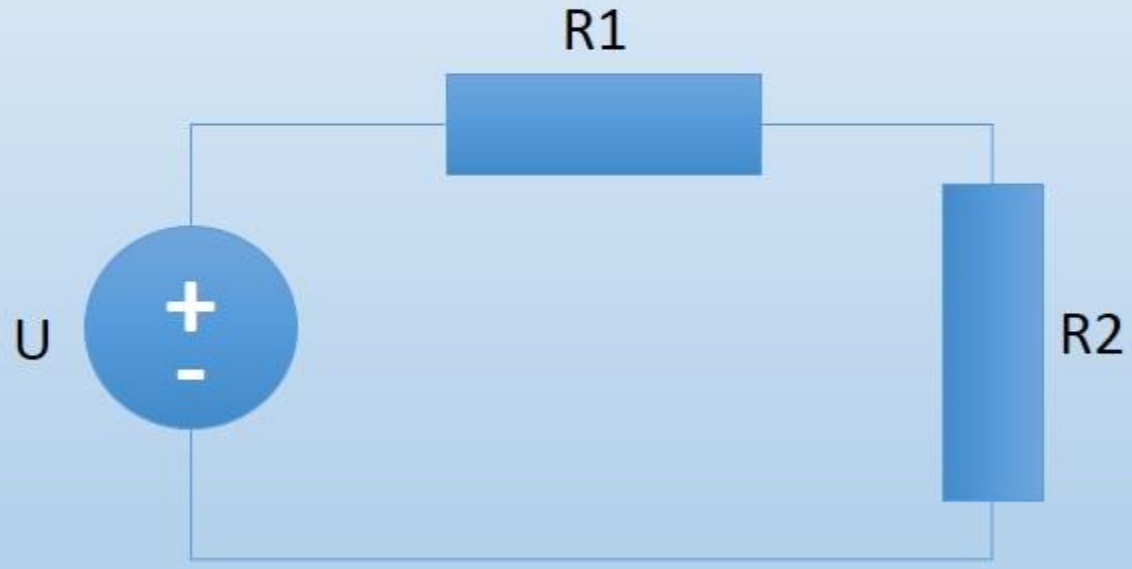
Άσκηση 5 – Να γραφεί πρόγραμμα που θα διαβάζει και θα εμφανίζει έναν ακέραιο αριθμό, έναν αριθμό κινητής υποδιαστολής, καθώς και μια λέξη έως 5 χαρακτήρες.

Άσκηση 6 – Να γραφεί πρόγραμμα που θα διαβάζει την ακτίνα ενός κύκλου και θα εμφανίζει το εμβαδό.

Άσκηση 7 – Να γραφεί πρόγραμμα που θα αντιμεταθέτει τις τιμές δύο μεταβλητών

Άσκηση 8

Να αναπτυχθεί πρόγραμμα που θα υπολογίζει τη συνολική αντίσταση αλλά και το ρεύμα του κυκλώματος. Οι τιμές των στοιχείων του κυκλώματος εισάγονται από το πληκτρολόγιο.



Εισαγωγή στη γλώσσα C

Έλεγχος & Επανάληψη

Πρόχειρη παρουσίαση με πρακτικά παραδείγματα για χρήση στο εργαστήριο

Παναγιώτης Παπάζογλου
Αν. Καθηγητής

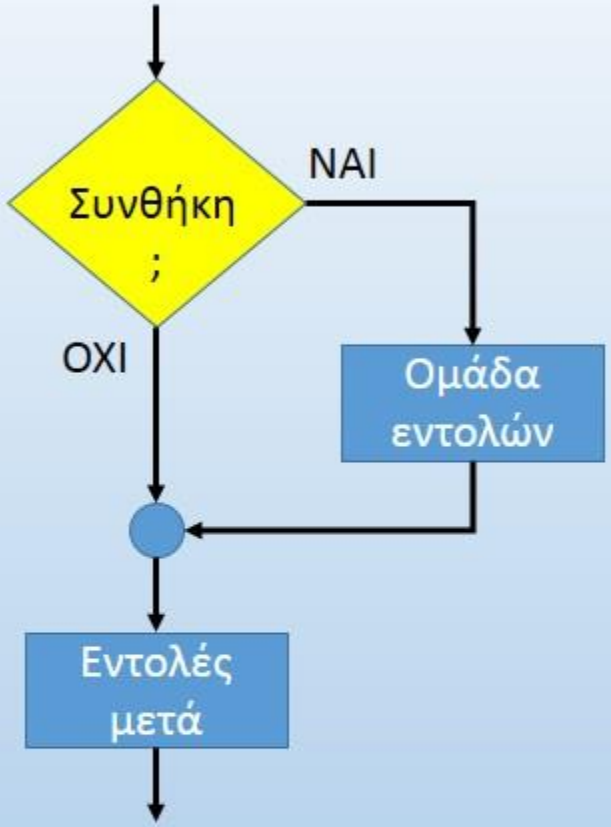
Υλοποίηση ελέγχου

Εντολή if

```
if (συνθήκη)
{

     /* Ομάδα εντολών */

}

/* Εντολές μετά */
```

ΑΡΧΗ σχόλιου

ΤΕΛΟΣ σχόλιου



Τι θα εμφανίσουν τα ακόλουθα προγράμματα ;

```
int x=0;
if (x==1)
  printf("Kalimera\n");
  printf("Kali ebdomada\n");

printf("Kalo mina\n");
printf("Kali douleia");
```

```
int x=0;
if (x==1)
{
  printf("Kalimera\n");
  printf("Kali ebdomada\n");
}

printf("Kalo mina\n");
printf("Kali douleia");
```

Εντολή if

Τι θα εμφανίσουν τα ακόλουθα προγράμματα ;

```
int x=0;
if (x==1)
  printf("Kalimera\n");
  printf("Kali ebdomada\n");

printf("Kalo mina\n");
printf("Kali douleia");
```

Kali ebdomada
Kalo mina
Kali douleia

```
int x=0;
if (x==1)
{
  printf("Kalimera\n");
  printf("Kali ebdomada\n");
}

printf("Kalo mina\n");
printf("Kali douleia");
```

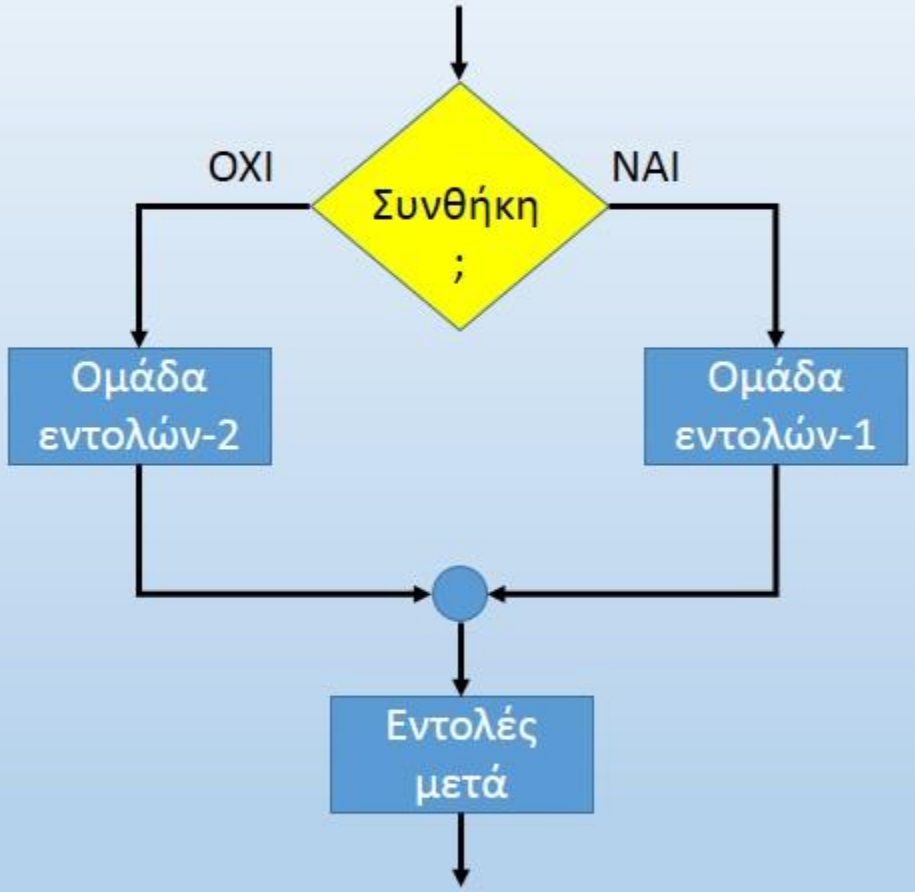
Kalo mina
Kali douleia

Τελεστές ελέγχου

Τελεστής	Περιγραφή	Παράδειγμα
==	Έλεγχος ισότητας ή όχι	(a==b)
!=	Έλεγχος ισότητας ή όχι	(a!=b)
>	Μεγαλύτερο	(a>b)
<	Μικρότερο	(a<b)
>=	Μεγαλύτερο ή ίσο	(a>=b)
<=	Μικρότερο ή ίσο	(a<=b)

Εντολή if-else

```
if (συνθήκη)
{

  /* Ομάδα εντολών-1 */

}

else
{
 /* Ομάδα εντολών-2 */

}

/* Εντολές μετά */
```



Παράδειγμα

```
if (x==1)
{
   printf("X=1");
}
else
{
   printf("X<>1");
}

printf(“Εντολές μετά”);
```

Εντολή if και λογικοί τελεστές

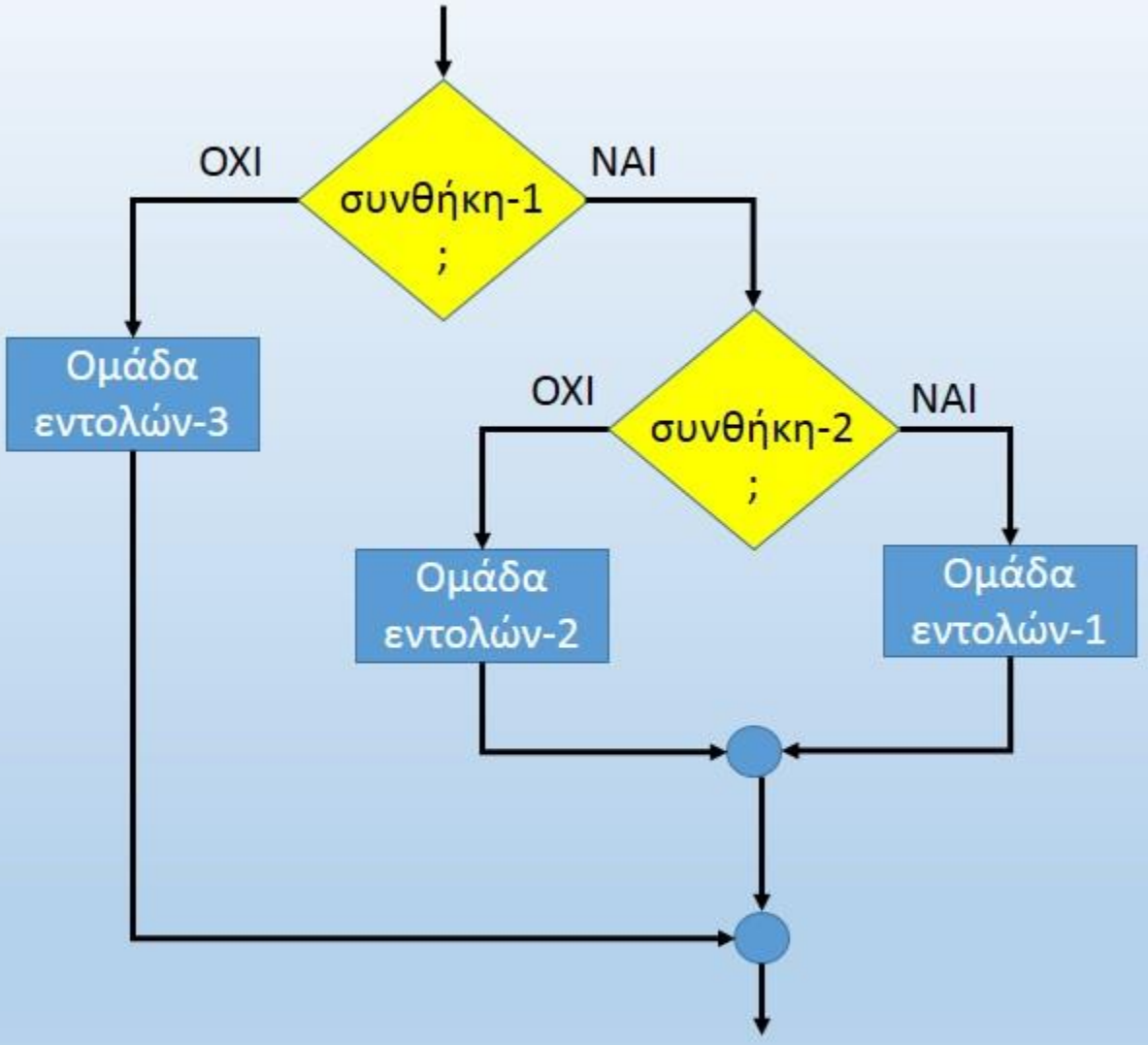
```
if (συνθήκη-1 || συνθήκη-2)
{

  /* Ομάδα εντολών */

}
```

Η ομάδα εντολών εκτελείται αν είναι αληθής η συνθήκη-1 ή η συνθήκη-2

|| → Τελεστής OR
&& → Τελεστής AND

Ένθετα if

```
if (συνθήκη-1)
        if (συνθήκη-2)
                ομάδα-εντολών-1
        else
                ομάδα-εντολών-2
else
        ομάδα-εντολών-3

Επόμενη εντολή
```



Άσκηση 1: Πρόγραμμα που θα διαβάζει έναν ακέραιο αριθμό από το πληκτρολόγιο και θα εμφανίζει κατά περίπτωση αν είναι αρνητικός, μηδέν ή θετικός

```
#include <stdio.h>
main()
{

int x;
printf("Enter x:");
scanf("%d",&x);
if (x==0)
   printf("X=0");
else
   if (x<0)
      printf("X<0");
   else
      printf("X>0");

}
```

Άσκηση 2: Πρόγραμμα που θα διαβάζει έναν ακέραιο αριθμό από το πληκτρολόγιο και θα εμφανίζει κατά περίπτωση αν ανήκει ή όχι στο διάστημα [10,20]

```
#include <stdio.h>
main()
{

int x;
printf("Enter x:");
scanf("%d",&x);
if ((x>=10) && (x<=20))
   printf("X E [10,20]");
else
   printf("X /E [10,20]");
}
```

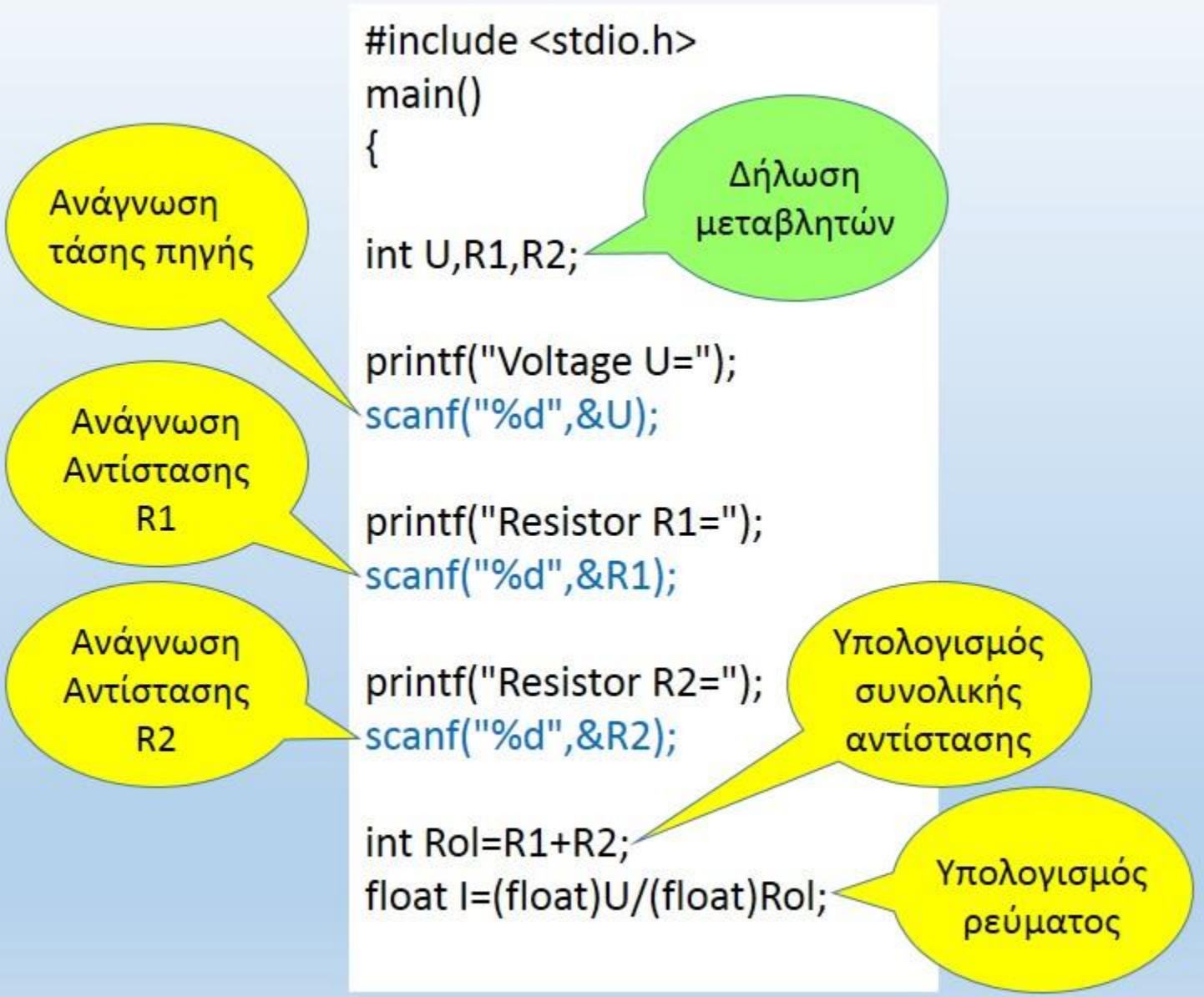
Άσκηση 3: Η άσκηση 2 αλλά με λογικό OR

```
#include <stdio.h>
main()
{

int x;
printf("Enter x:");
scanf("%d",&x);
if ((x<10) || (x>20))
   printf("X /E [10,20]");
else
   printf("X E [10,20]");
}
```

Άσκηση

Να αναπτυχθεί πρόγραμμα που θα υπολογίζει:
(α) τη συνολική αντίσταση
(β) το ρεύμα του κυκλώματος
(γ) την ισχύ των αντιστάσεων

Οι τιμές των στοιχείων του κυκλώματος εισάγονται από το πληκτρολόγιο, ενώ το πρόγραμμα θα εμφανίζει την αντίσταση με τη μικρότερη ισχύ.


#include <stdio.h>
main()
{
int U,R1,R2;
printf("Voltage U=");
scanf("%d",&U);
printf("Resistor R1=");
scanf("%d",&R1);
printf("Resistor R2=");
scanf("%d",&R2);
int Rol=R1+R2;
float I=(float)U/(float)Rol;
Ανάγνωση τάσης πηγής
Δήλωση μεταβλητών
Ανάγνωση Αντίστασης R1
Ανάγνωση Αντίστασης R2
Υπολογισμός συνολικής αντίστασης
Υπολογισμός ρεύματος

Υπολογισμός ισχύος στις αντιστάσεις

```
float PR1=(I*I)*R1;
float PR2=(I*I)*R2;

printf("Rol=%d, I=%.4fA, I=%.2fmA, PR1=%.2fW,
PR2=%.2fW\n",Rol,I,I*1000,PR1,PR2);

if (PR1<PR2)
   printf("MIN Power ==> PR1");
else
   if (PR2<PR1)
      printf("MIN Power ==> PR2");
   else
      printf("PR1=PR2");

}
```

Εμφάνιση αποτελεσμάτων

Εύρεση ελάχιστης ισχύος

Υλοποίηση βρόχων επανάληψης

Βρόχος επανάληψης

- Δομή που περιλαμβάνει μια ομάδα εντολών που εκτελείται επαναληπτικά
- Περιλαμβάνει μετρητή που αρχικοποιείται πριν το βρόχο
- Ο έλεγχος της επανάληψης γίνεται με την κατάλληλη συνθήκη
- Σε κάθε επανάληψη, ενημερώνεται ο μετρητής

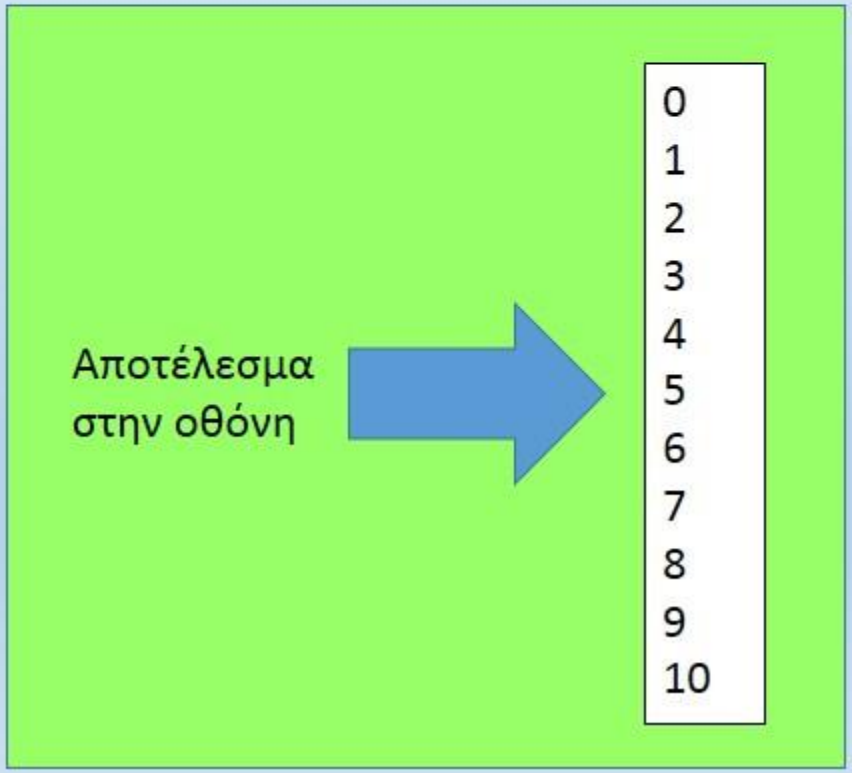
Δομή for

for (αρχική έκφραση; συνθήκη; τελική έκφραση)

```
/*
        Εντολές που επαναλαμβάνονται όσο
        η συνθήκη είναι αληθής
*/
```

Δομή for - ΠΑΡΑΔΕΙΓΜΑ

```
int i;
for (i=0; i<=10; i++)
        printf("%d\n",i);
```



Τελεστής

++

i++ → i=i+1

i+=1 → i=i+1

Τελεστής

--

i-- → i=i-1

i-=1 → i=i-1

Θέση Τελεστή

```
#include <stdio.h>
main()
{
int a,b,e=1,f=1;
a=e++;
b=++f;
printf("a=%d, b=%d, e=%d, f=%d",a,b,e,f);
}
```

Αποτέλεσμα στην οθόνη

a=1, b=2, e=2, f=2

a=e++;

Πρώτα γίνεται η εκχώρηση του e στο a και μετά αυξάνεται το περιεχόμενο του e

b=++f;

Πρώτα γίνεται η αύξηση του f και μετά εκχωρείται το περιεχόμενό του στο b

Βρόχος while-do

```
while (συνθήκη)
{

/*
Εντολές που
επαναλαμβάνονται όσο η
συνθήκη είναι αληθής
*/

}
```

Βρόχος while-do - ΠΑΡΑΔΕΙΓΜΑ

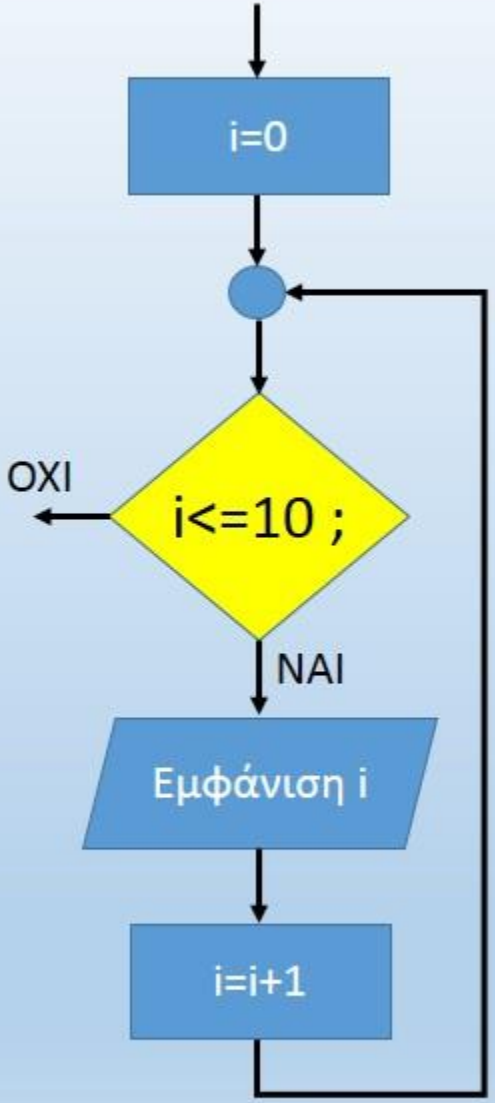
Παράδειγμα

Αρχικοποίηση μετρητή

συνθήκη

```
int i=0;
while(i<=10)
{
   printf("%d\n",i);
   i++;
}
```

Ενημέρωση μετρητή



Βρόχος do-while

```
do
{

/*
Εντολές που
επαναλαμβάνονται όσο η
συνθήκη είναι αληθής
*/

}
while (συνθήκη);
```

Βρόχος do-while - ΠΑΡΑΔΕΙΓΜΑ

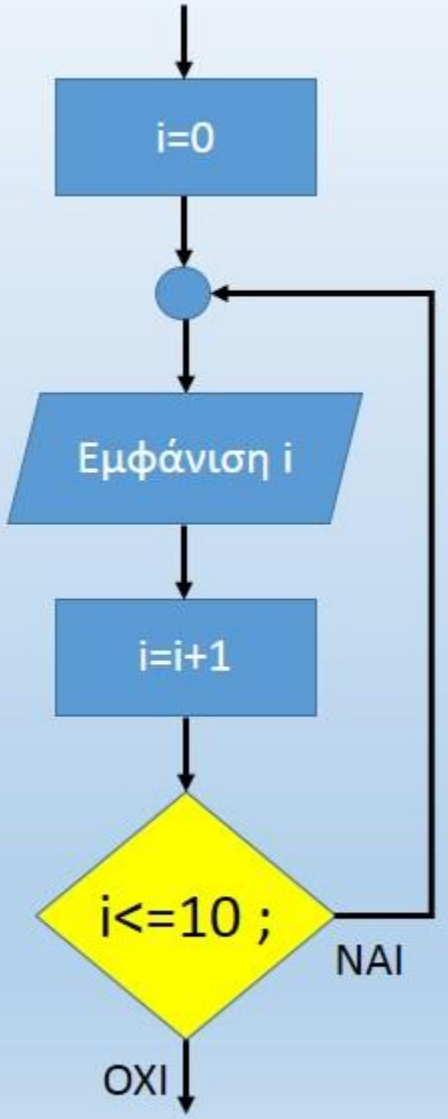
Παράδειγμα

```
int i=0;
do
{
   printf("%d\n",i);
   i++;
}
while(i<=10);
```

Αρχικοποίηση μετρητή

Ενημέρωση μετρητή

συνθήκη

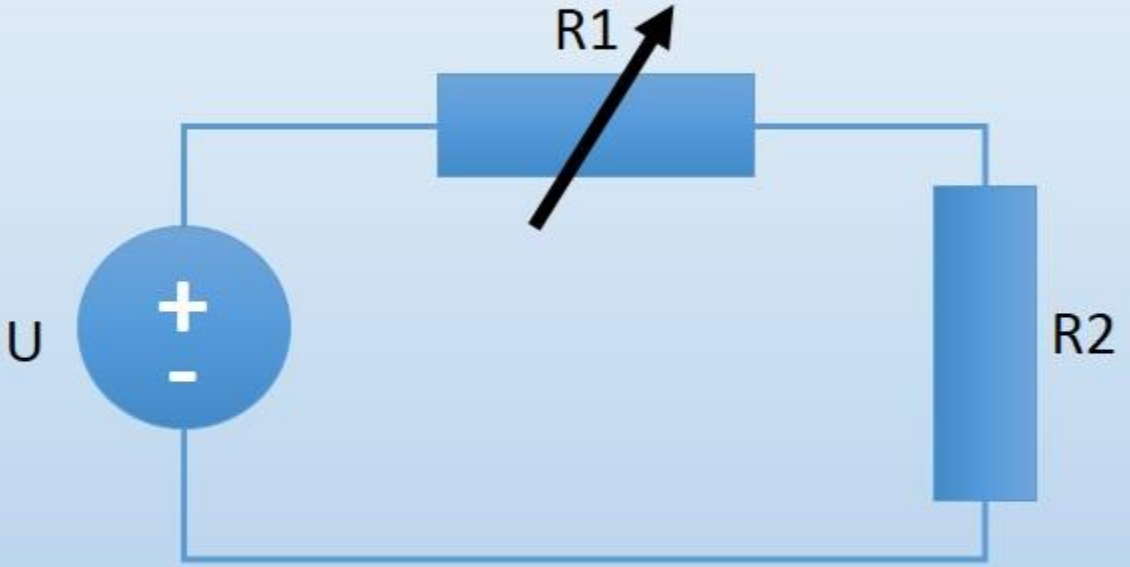


Ποιες είναι οι διαφορές των δομών
while-do και do-while ;

Άσκηση

Να αναπτυχθεί πρόγραμμα που θα υπολογίζει:
(α) τη συνολική αντίσταση
(β) το ρεύμα του κυκλώματος

Οι τιμές των στοιχείων του κυκλώματος εισάγονται από το πληκτρολόγιο. Όλοι οι υπολογισμοί θα γίνονται για μεταβλητή αντίσταση R1 στο εύρος [A,B] με βήμα 10Ω.



Αποτέλεσμα στην οθόνη. Εισάγουμε υποθετικά U=10, A=100, B=150 και R2=130

Voltage U=10
Resistor R1 [A,B], A=100
Resistor R1 [A,B], B=150
Resistor R2=130

R1	R2	Rol	I(A)	I(mA)
---	---	----	-----	-----
100	130	230	0.043	43.48
110	130	240	0.042	41.67
120	130	250	0.040	40.00
130	130	260	0.038	38.46
140	130	270	0.037	37.04
150	130	280	0.036	35.71

Πρόγραμμα...

```
#include <stdio.h>
main()
{

int U,R1A,R1B,R2,Rol;
float I;

printf("Voltage U=");
scanf("%d",&U);

printf("Resistor R1 [A,B], A=");
scanf("%d",&R1A);

printf("Resistor R1 [A,B], B=");
scanf("%d",&R1B);
```

```
printf("Resistor R2=");
scanf("%d",&R2);

printf("R1\tR2\tRol\tI(A)\tI(mA)\n");
printf("---\t---\t---\t-----\t-----\n");
int i=R1A;
while(i<=R1B)
{
   Rol=i+R2;
   I=(float)U/(float)Rol;
   printf("%d\t%d\t%d\t%.3f\t%.2f\n",i,R2,Rol,I,I*1000);
   i+=10;
}
}
```

Εισαγωγή στη γλώσσα C

Συναρτήσεις

Πρόχειρη παρουσίαση με πρακτικά παραδείγματα για χρήση στο εργαστήριο

Παναγιώτης Παπάζογλου
Αν. Καθηγητής

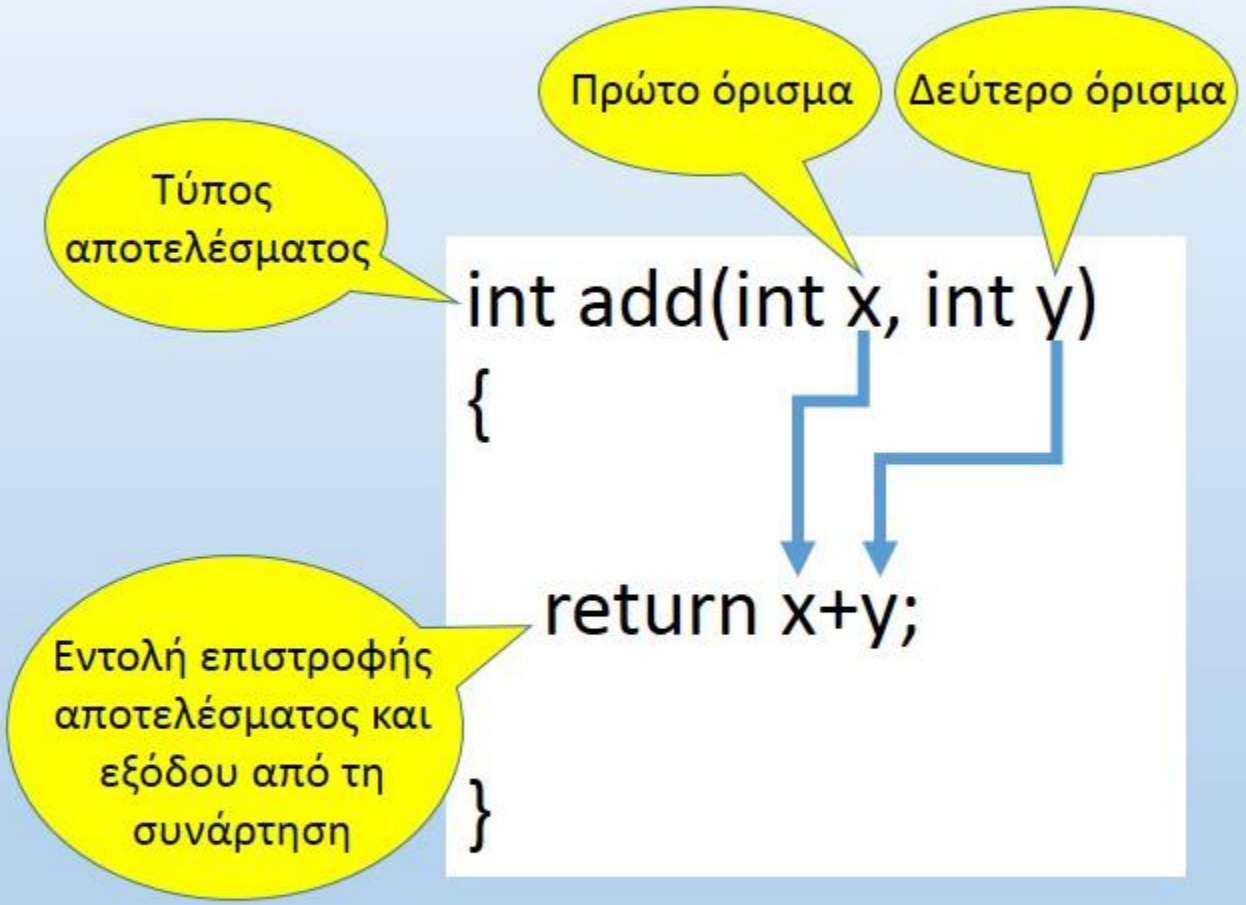
Συνάρτηση

- 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


Πρώτο όρισμα
Δεύτερο όρισμα
Τύπος αποτελέσματος
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);

   c=add(a,b);

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

}
```

Δήλωση συνάρτησης

```
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;
}
```

Εισαγωγή στη γλώσσα C

Πίνακες

Πρόχειρη παρουσίαση με πρακτικά παραδείγματα για χρήση στο εργαστήριο

Παναγιώτης Παπάζογλου
Αν. Καθηγητής

Πίνακας

- Συλλογή ομοειδών δεδομένων (π.χ πίνακας ακέραιων αριθμών)
- Τα δεδομένα αυτά καταλαμβάνουν διαδοχικές θέσεις μνήμης
- Έχει ένα μοναδικό όνομα που συνοδεύεται από ένα δείκτη για την προσπέλαση των δεδομένων που περιέχει
- Οι πίνακες μιας ή δύο διαστάσεων αποτελούν τις συνηθέστερες μορφές

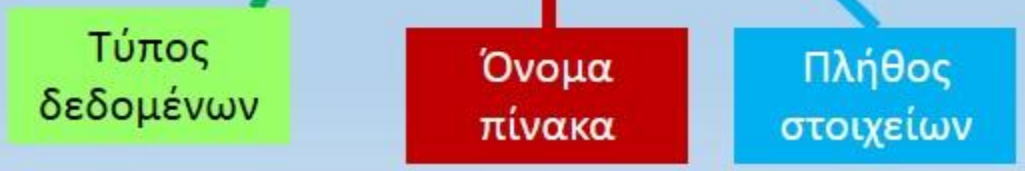
Πίνακας μιας διάστασης ή μονοδιάστατος πίνακας

Δήλωση πίνακα

```
Τύπος_δεδομένων ΟΝΟΜΑ_ΠΙΝΑΚΑ [πλήθος_στοιχείων]
```

Παραδείγματα

```
int    array_a[20];
char   array_b[10];
float  array_c[5];
```



Αρχικοποίηση πινάκων

Ένας πίνακας μπορεί να αρχικοποιηθεί με τη δήλωσή του
(καθορίζοντας ή όχι το πλήθος των στοιχείων)

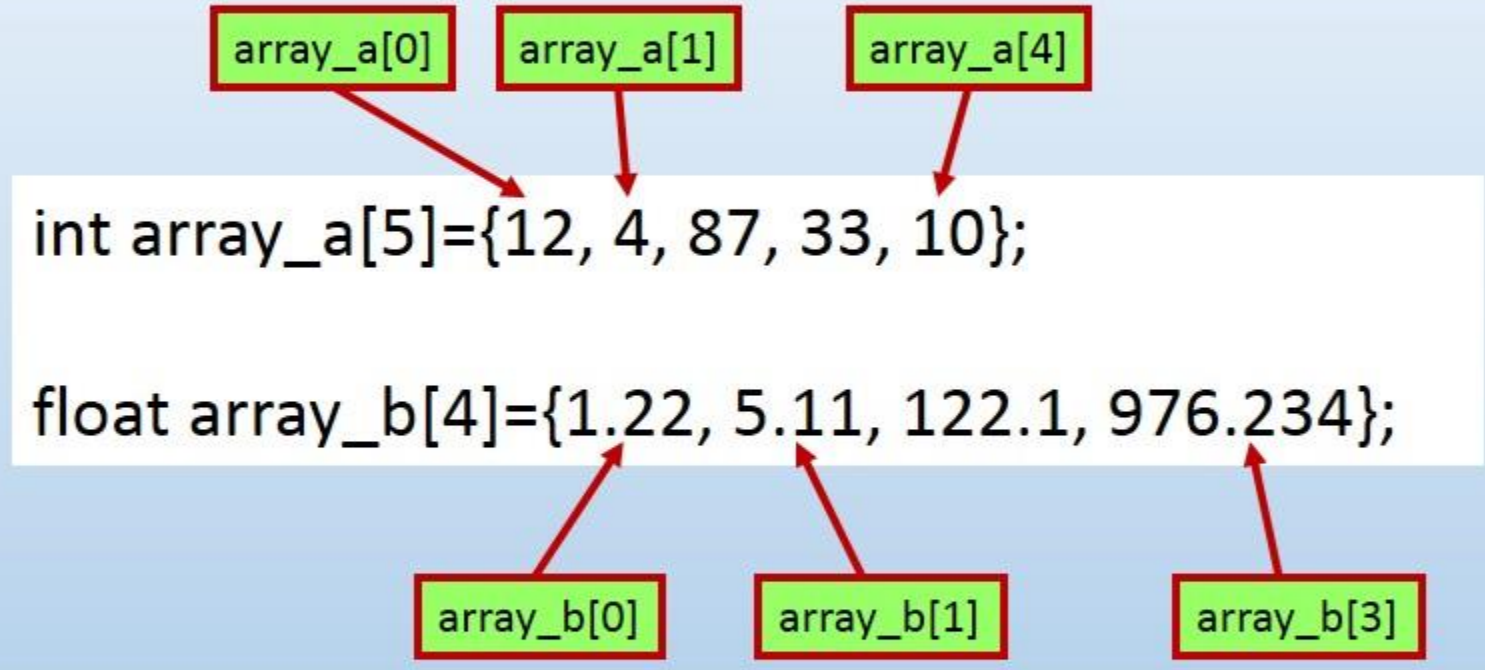
Αρχικοποίηση περιεχομένων με καθορισμό πλήθους στοιχείων

```
int array_a[5]={12, 4, 87, 33, 10};

float array_b[4]={1.22, 5.11, 122.1, 976.234};
```

Αρχικοποίηση περιεχομένων με αυτόματο καθορισμό πλήθους στοιχείων

```
int array_a[]={12, 4, 87, 33, 10};

float array_b[]={1.22, 5.11, 122.1, 976.234};
```

Προσπέλαση στοιχείων πίνακα



Πίνακας δύο διαστάσεων ή δισδιάστατος πίνακας

Δήλωση πίνακα

Τύπος_δεδομένων ΟΝΟΜΑ_ΠΙΝΑΚΑ [**πλήθος_γραμμών**][**πλήθος_στηλών**]

Παραδείγματα

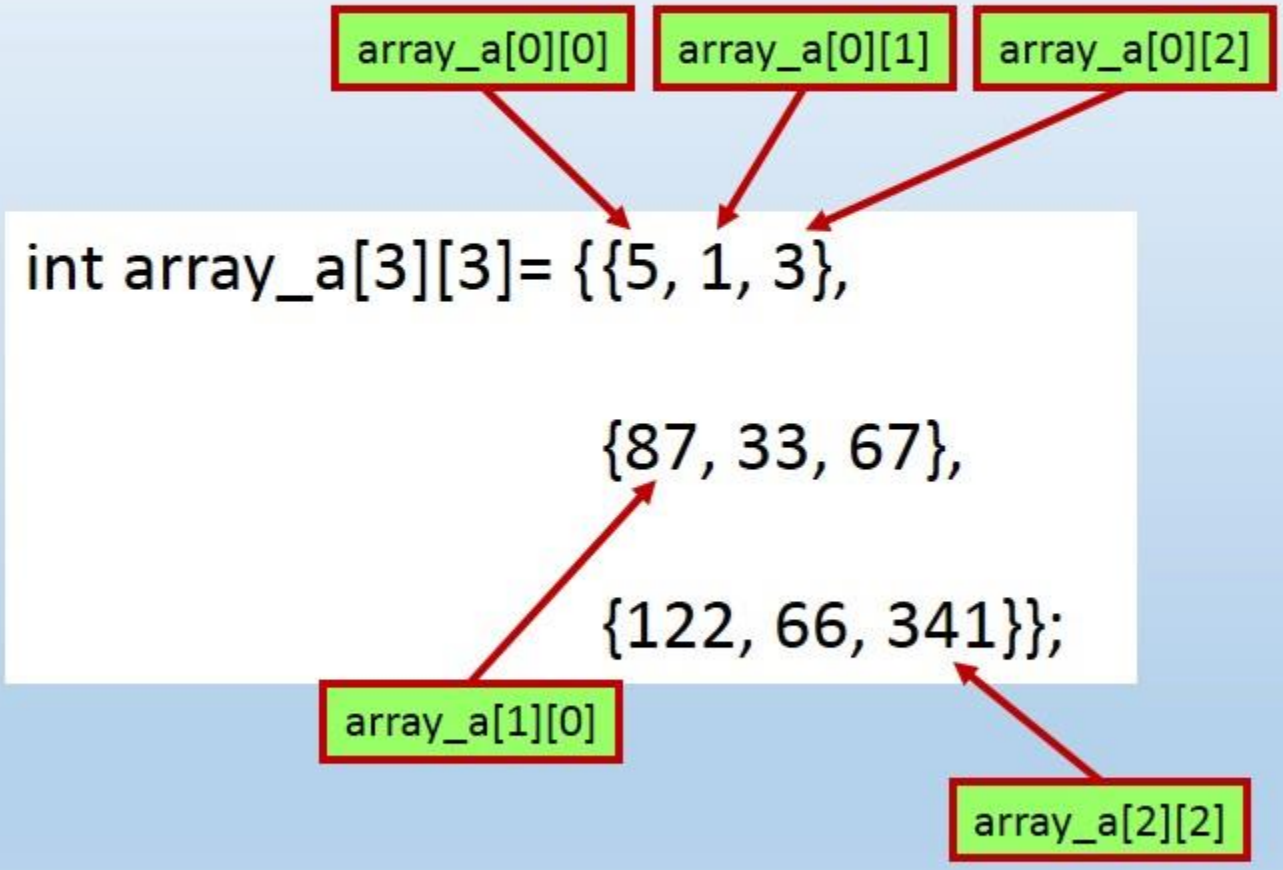
```
int    array_a[20][5];
char   array_b[10][4];
float  array_c[5][5];
```

Αρχικοποίηση πινάκων

Αρχικοποίηση περιεχομένων με καθορισμό πλήθους γραμμών και στηλών

```
int array_a[3][3]= {{5, 1, 3},
                    {87, 33, 67},
                    {122, 66, 341}};
```

Προσπέλαση στοιχείων πίνακα

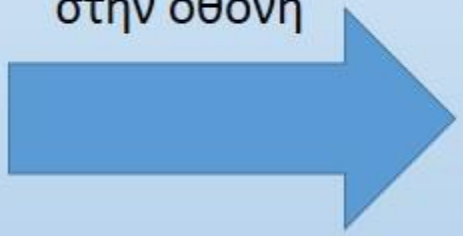


Άσκηση 1

Να γραφεί πρόγραμμα που να αρχικοποιεί ένα μονοδιάστατο πίνακα 10 θέσεων και στη συνέχεια να εμφανίζει τα περιεχόμενά του στην οθόνη.

```
#include <stdio.h>

int array_a[10]={100, 66, 78, 124, 21, 344, 322, 11, 19, 18};

main()
{
int i;
for(i=0;i<10;i++)
   printf("array_a[%d]=%d\n",i,array_a[i]);
}
```

Αποτέλεσμα στην οθόνη



```
array_a[0]=100
array_a[1]=66
array_a[2]=78
array_a[3]=124
array_a[4]=21
array_a[5]=344
array_a[6]=322
array_a[7]=11
array_a[8]=19
array_a[9]=18
```

Άσκηση 2

Να γραφεί πρόγραμμα που να γεμίζει ένα μονοδιάστατο πίνακα 10 θέσεων από το πληκτρολόγιο και στη συνέχεια να εμφανίζει τα περιεχόμενά του στην οθόνη σε οριζόντια διάταξη.

```
#include <stdio.h>

int array_a[10];

main()
{
int a,i;
for(i=0;i<10;i++)
{
   printf("array_a[%d]:",i);
   scanf("%d",&a);
   array_a[i]=a;
}

for(i=0;i<10;i++)
   printf(" %d",array_a[i]);

}
```

Αποτέλεσμα στην οθόνη

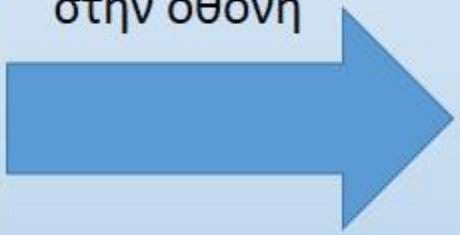
```
array_a[0]:3
array_a[1]:21
array_a[2]:77
array_a[3]:89
array_a[4]:345
array_a[5]:2
array_a[6]:6
array_a[7]:88
array_a[8]:12
array_a[9]:34
 3 21 77 89 345 2 6 88 12 34
```

Άσκηση 3

Να γραφεί πρόγραμμα που να γεμίζει ένα μονοδιάστατο πίνακα 10 θέσεων από το πληκτρολόγιο και στη συνέχεια να υπολογίζει και να εμφανίζει το άθροισμα των στοιχείων του.

```
#include <stdio.h>

int array_a[10];

main()
{
int a,i;
for(i=0;i<10;i++)
{
   printf("array_a[%d]:",i);
   scanf("%d",&a);
   array_a[i]=a;
}
int sum=0;
for(i=0;i<10;i++)
   sum+=array_a[i];

printf("Sum=%d",sum);

}
```

Αποτέλεσμα στην οθόνη



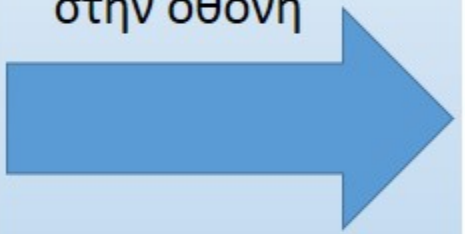
```
array_a[0]:2
array_a[1]:3
array_a[2]:4
array_a[3]:1
array_a[4]:2
array_a[5]:8
array_a[6]:0
array_a[7]:4
array_a[8]:2
array_a[9]:1
Sum=27
```

Άσκηση 4

Να γραφεί πρόγραμμα που να γεμίζει ένα μονοδιάστατο πίνακα 10 θέσεων από το πληκτρολόγιο και στη συνέχεια να εμφανίζει το πλήθος των αρνητικών, μηδενικών και θετικών αριθμών.

```
#include <stdio.h>
int array_a[10];
int mik=0,zer=0,meg=0;
main()
{
int a,i;
for(i=0;i<10;i++)
  {
   printf("array_a[%d]:",i);
   scanf("%d",&a);
   array_a[i]=a;
  }
int sum=0;
for(i=0;i<10;i++)
   if (array_a[i]<0)
      mik++;
   else
      if (array_a[i]==0)
         zer++;
      else
         meg++;

printf("[<0]:%d, [=0]:%d, [>0]:%d",mik,zer,meg);
}
```

Αποτέλεσμα στην οθόνη



```
array_a[0]:0
array_a[1]:9
array_a[2]:-2
array_a[3]:-3
array_a[4]:-4
array_a[5]:7
array_a[6]:0
array_a[7]:1
array_a[8]:2
array_a[9]:3
[<0]:3, [=0]:2, [>0]:5
```