

## Εργαστήριο 5

Εντολή if και παραλλαγές:  
if-else, πολλαπλές if, πολλαπλές if-else.  
Απλές και σύνθετες εντολές.  
Εντολή switch.

Προτεραιότητα τελεστών (συνέχεια):

() παρενθέσεις  
+, - (πρόσημα), ++, --  
\*, /, %  
+, -  
<, >, <=, >=  
==, !=  
&&  
||  
=, \*=, /=, %=, +=, -=

Στη C οι λογικές συνθήκες έχουν αριθμητικές τιμές:  
0 (μηδέν) αν είναι ψευδής και 1 αν είναι αληθής.

Η C όμως δέχεται και **οποιοσδήποτε** αριθμητικές παραστάσεις ως συνθήκες!  
Στη C μια τέτοια παράσταση **ισχύει** αν έχει **μη μηδενική τιμή**.  
Η παράσταση δεν ισχύει μόνο αν η τιμή της είναι μηδέν  
(βλέπε παρακάτω το πρόγραμμα που δοκιμάζει τις διακλαδώσεις **if**).

### Διακλαδώσεις (conditional statements)

Υπάρχουν τριών ειδών διακλαδώσεις στη C :  
**if, if else** και **switch**.

#### Διακλάδωση if.

Στο πρόγραμμα γράφουμε :

```
if (συνθήκη) {ομάδα εντολών}
```

πχ.

```
if (i>0) { printf("positive"); }
```

**Παρατήρηση.** Όταν μέσα στην ομάδα εντολών υπάρχει μόνο μία εντολή τότε τα άγκιστρα δεν χρειάζονται.

```
if (i>0) printf("positive");
```

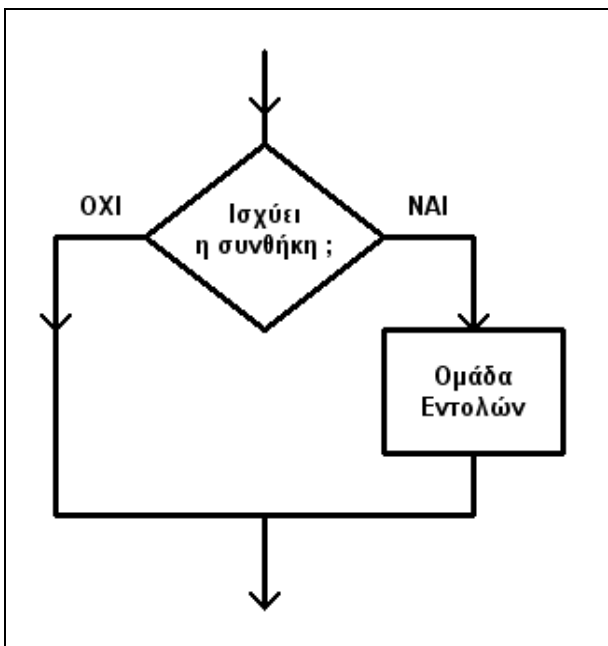
Πχ.

```
if ( year%4 == 0 && year%100 != 0 ) printf("dissekto");
```

Πχ.

```
if (i==0) { printf("zero, setting i=1"); i=1;}
```

Το λογικό διάγραμμα της **if**.



Όταν το πρόγραμμα φτάνει στην εντολή **if**, υπολογίζει πρώτα την λογική τιμή της συνθήκης.

Αν η συνθήκη ισχύει (έχει μη μηδενική τιμή), τότε εκτελείται ομάδα εντολών.

Αν η συνθήκη δεν ισχύει, δεν γίνεται τίποτα.

Στο τέλος (και στις δύο περιπτώσεις) το πρόγραμμα προχωράει προς την επόμενη εντολή του προγράμματος.

**Παράδειγμα.** Λογικές συνθήκες με **if**.

```
#include <stdio.h>
#include <stdlib.h>
main(){ int a;

    a = 0;

    printf("%d\n", (a==1)); // 0
    printf("%d\n", (a==0)); // 1

    if (100) printf(" 100 ok\n"); // NAI
    if (-100) printf("-100 ok\n"); // NAI
    if (0.25) printf(" 0.25 ok\n"); // NAI
    if (-0.25) printf("-0.25 ok\n"); // NAI

    if (0) printf("0 ok\n"); // OXI

    if (a==0) printf("a==0\n"); // NAI
    if (a==1) printf("a==1\n"); // OXI

    if (a=0) printf("a=0\n"); // OXI ! Γιατί;
    if (a=1) printf("a=1\n"); // NAI ! Γιατί;

    system("pause");
}
```

**Διακλάδωση if else.**

Παρατηρήστε ότι το διάγραμμα της **if** δεν είναι συμμετρικό.

Πολλές φορές, αν η συνθήκη δεν ισχύει, θέλουμε να εκτελέσουμε κάποιες άλλες εντολές.

Αυτή τη δυνατότητα μας προσφέρει η διακλάδωση **if else**.

Στο πρόγραμμα γράφουμε :

```
if (Συνθήκη) Ομάδα Εντολών A else Ομάδα Εντολών B
```

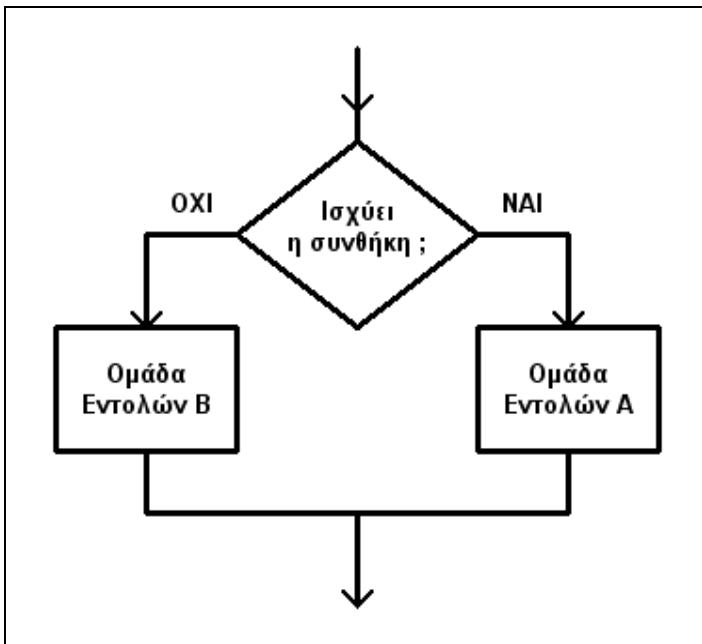
Πχ.

```
if ( i==0 )
    printf("i is zero\n");
else
    printf("i is non zero\n");
```

Πχ.

```
if ( 1<=i && i<=5 )
    printf("i is 1,2,3,4 or 5\n");
else
    printf(" i < 1 or i >5 \n");
```

Το λογικό διάγραμμα της **if else**.



Όταν το πρόγραμμα φτάνει στην εντολή **if else**, υπολογίζει πρώτα την λογική τιμή της συνθήκης.

Αν η συνθήκη ισχύει (έχει μη μηδενική τιμή), τότε εκτελείται η ομάδα εντολών Α.

Αν η συνθήκη ΔΕΝ ισχύει (έχει τιμή μηδέν), τότε εκτελείται η ομάδα εντολών Β.

Στο τέλος (και στις δύο περιπτώσεις) το πρόγραμμα προχωράει προς την επόμενη εντολή του προγράμματος.

**Παράδειγμα.** Υπολογισμός του min και max

```
#include <stdio.h>
#include <stdlib.h>
main() { int a,b,min,max;

    printf("give a b -> "); scanf("%d%d", &a, &b);
    if (a<b) min = a; else min = b;
    if (a<b) max = b; else max = a;
    printf("min = %d max = %d\n", min, max);

    system("pause");
}
```

**Ένθετες** (nested) εντολές **if else**.

Οι ομάδες εντολών της **if else** είναι δυνατών να περιέχουν μέσα επίσης διακλαδώσεις.

πχ.

```
if (i>0)
    { if (i>3)  i++; else i--; }
else
    { if (i<-4) i++; else i--; }
```

**Παράδειγμα.** Οι ρίζες της δευτεροβάθμιας εξίσωσης  $Ax^2 + Bx + C = 0$

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

main(){ float A,B,C,D,x1,x2;
    printf("give A B C -> ");
    scanf("%f%f%f", &A, &B, &C);

    D = B*B - 4*A*C;

    if (D<0) printf("No real roots \n");
    else {

        if (D>0){
            x1 = (-B + sqrt(D))/(2*A);
            x2 = (-B - sqrt(D))/(2*A);

            printf("roots : x1 = %f, x2 = %f \n", x1, x2);

        } else {
            x2 = -B/(2*A);
            printf("double root : x = %f \n", x2);
        }
    }

    system("pause");
}
```

## Διακλάδωση **switch**.

Όταν χρειάζονται πολλές διακλαδώσεις στο πρόγραμμα, η χρήση των **if else** γίνεται δύσκολη στην πράξη.

Πχ. μια επιλογή του χρήστη από μενού 10 δυνατοτήτων χρειάζεται 10 εντολές **if else**.

Η εντολή **switch** οργανώνει καλύτερα τέτοιες περιπτώσεις.

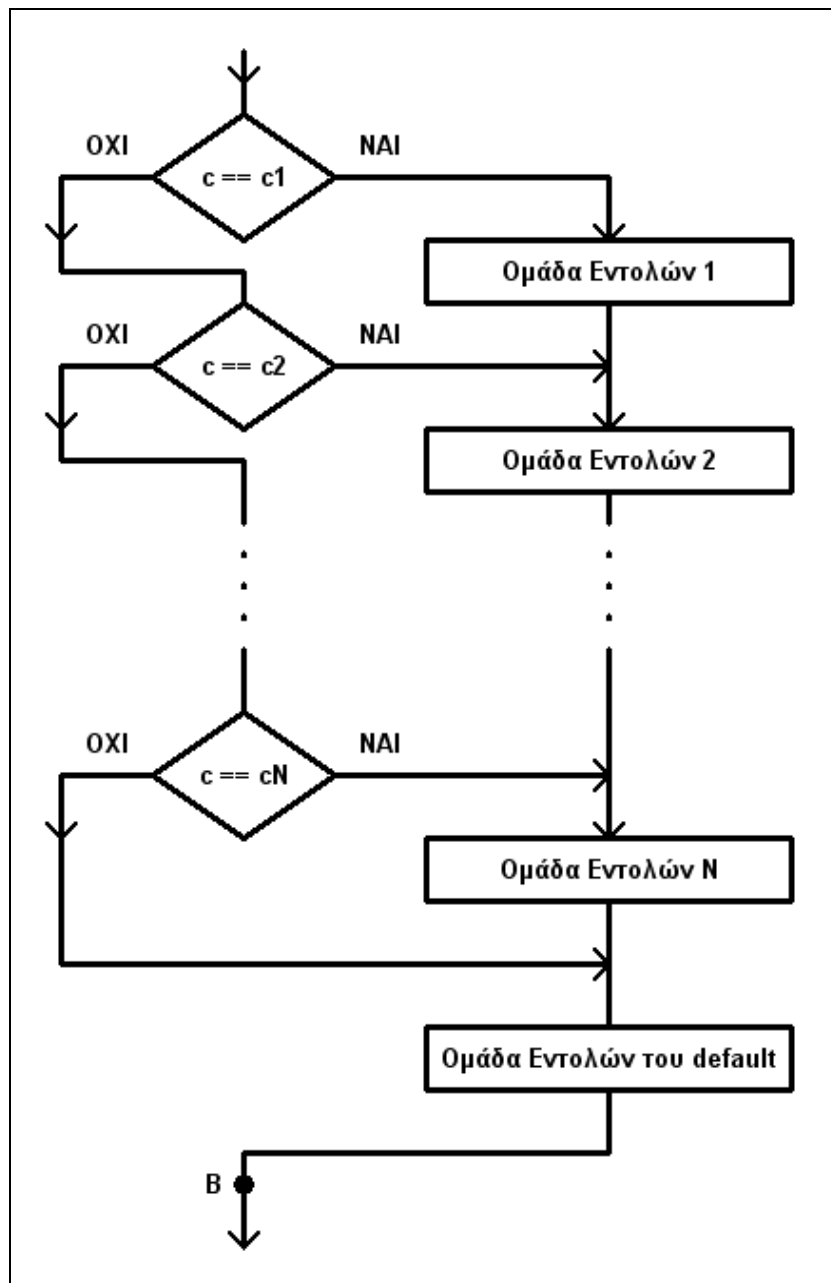
Έστω **c** μια μεταβλητή τύπου π.χ. **int** και **c1, c2,..., cN** διαφορές **σταθερές** τιμές της **c** που μας ενδιαφέρουν.

Θέλουμε να διακλαδώσουμε το πρόγραμμά μας σε περιπτώσεις τύπου **c = ci** και να εκτελέσουμε κατάλληλες εντολές.

Η εντολή **switch** χρησιμοποιείται ως εξής :

```
switch (c) {  
    case c1 : Ομάδα Εντολών 1  
    case c2 : Ομάδα Εντολών 2  
    ...  
    case cN : Ομάδα Εντολών N  
    default : Ομάδα Εντολών του default  
}
```

Για να καταλάβουμε τι ακριβώς κάνει **switch**, ας δούμε πρώτα το λογικό διάγραμμά της.



Όπως βλέπουμε, **switch** συγκρίνει διαδοχικά την τιμή **c** με σταθερές **ci** και αν **c=ci**, τότε εκτελούνται **όλες** οι παρακάτω ομάδες εντολών, μαζί με την ομάδα του **default**.

Η ομάδα του **default** εκτελείται πάντα, και εκτελείται ως τελευταία. Αυτή η συμπεριφορά της **switch** σπάνια είναι χρήσιμη. Συνήθως θέλουμε να εκτελεστεί μόνο μία ομάδα εντολών σε κάθε περίπτωση! Ευτυχώς, στις ομάδες εντολών της **switch** μπορούμε να χρησιμοποιήσουμε μια ειδική εντολή **break**. Η **break** προκαλεί άμεσο τερματισμό της **switch**, (μετακίνηση στο σημείο **B** στο διάγραμμα).

Παράδειγμα. Οργάνωση μενού με **switch** και `getche`.

```
#include <stdio.h>
#include <conio.h> // getche
main(){ char ch;

    printf("Menu :\n");
    printf("1 - Play \n");
    printf("2 - Work \n");
    printf("3 - EXIT \n");

    printf("\n Your choice -> "); ch = getche();

    printf("\n\n");

    switch (ch) {
    case '1' : { printf("Now playing ... \n"); break; }
    case '2' : { printf("Now working ... \n"); break; }
    case '3' : { printf("Bye ... \n"); break; }
    default  : { printf("Bad choice\n"); break; }
    }
}
```