

Εντολές και μπλοκ

- Μια **παράσταση** όπως

`x = 0, ή i++, ή printf(...)`

- γίνεται **εντολή** όταν ακολουθείται από ελληνικό ερωτηματικό (;)

`x = 0; i++; printf(...);`

- Το (;) είναι σημείο τερματισμού εντολών.
- Τα **άγκιστρα** { και } χρησιμοποιούνται για την ομαδοποίηση δηλώσεων και εντολών σε μια σύνθετη εντολή ή **μπλοκ** ώστε να είναι ισοδύναμα με μία μόνο εντολή.
- Μετά το δεξί άγκιστρο που τερματίζει κάποιο μπλοκ δεν μπαίνει (;)
- Άγκιστρα περιβάλλουν:
 - τις εντολές μιας συνάρτησης π.χ.,
`main() {...}`
 - πολλαπλές εντολές σε δομές ελέγχου και επανάληψης

Εντολές ελέγχου

- Οι **εντολές** σ' ένα πρόγραμμα, εκτός και αν οριστεί διαφορετικά, εκτελούνται **σειριακά** (από πάνω προς τα κάτω και από τα αριστερά προς τα δεξιά)
- Αν θέλουμε να αλλάξουμε τη σειρά των πράξεων πρέπει να χρησιμοποιήσουμε τις εντολές ελέγχου



Εντολές ελέγχου

Τις εντολές ελέγχου τις διακρίνουμε σε :

- Απλής ή Διπλής επιλογής (if-else)
- Πολλαπλής επιλογής (switch)

if

Ο γενικός τύπος της εντολής ελέγχου if είναι :

if (παράσταση)

εντολή

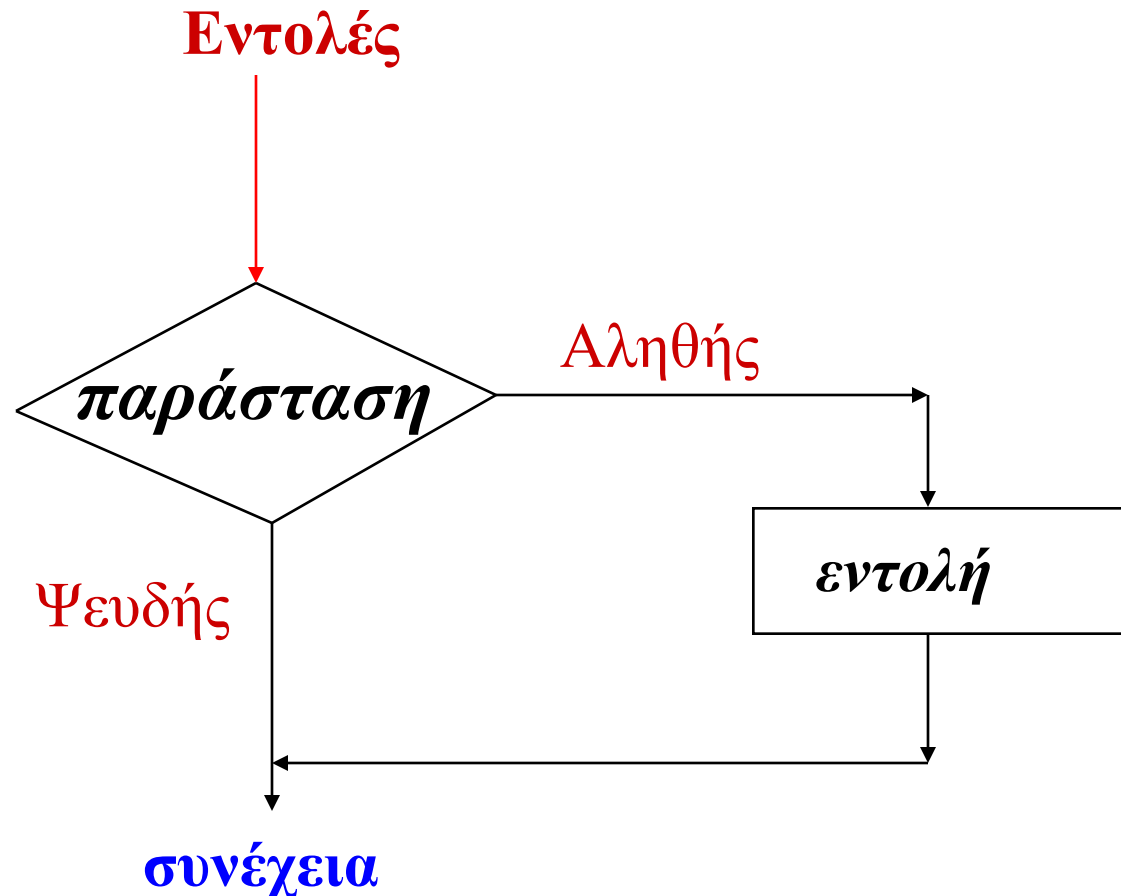
- Επειδή το if απλώς ελέγχει την αριθμητική τιμή μιας παράστασης μπορούμε να κάνουμε μερικές συντομεύσεις στον κώδικα
- Μπορούμε να γράψουμε:

if (παράσταση)

αντί για

if (παράσταση != 0)

Η εντολή **if** απλής επιλογής



if - else

Ο γενικός τύπος της εντολής ελέγχου if - else είναι :

if (παράσταση)

εντολή₁

else

εντολή₂

- Το τμήμα else είναι προαιρετικό.
- Υπολογίζεται η παράσταση και αν είναι αληθής (δηλαδή έχει μη μηδενική τιμή) εκτελείται η εντολή₁
- Αν είναι ψευδής (δηλαδή έχει μηδενική τιμή) και αν υπάρχει τμήμα else εκτελείται αντίστοιχα η εντολή₂

Εντολή **if** διπλής επιλογής

- Ο γενικός τύπος της εντολής `if` διπλής επιλογής (με ομάδες εντολών σαν αντικείμενα) είναι :

`if` (παράσταση)

{

ομάδα εντολών 1

}

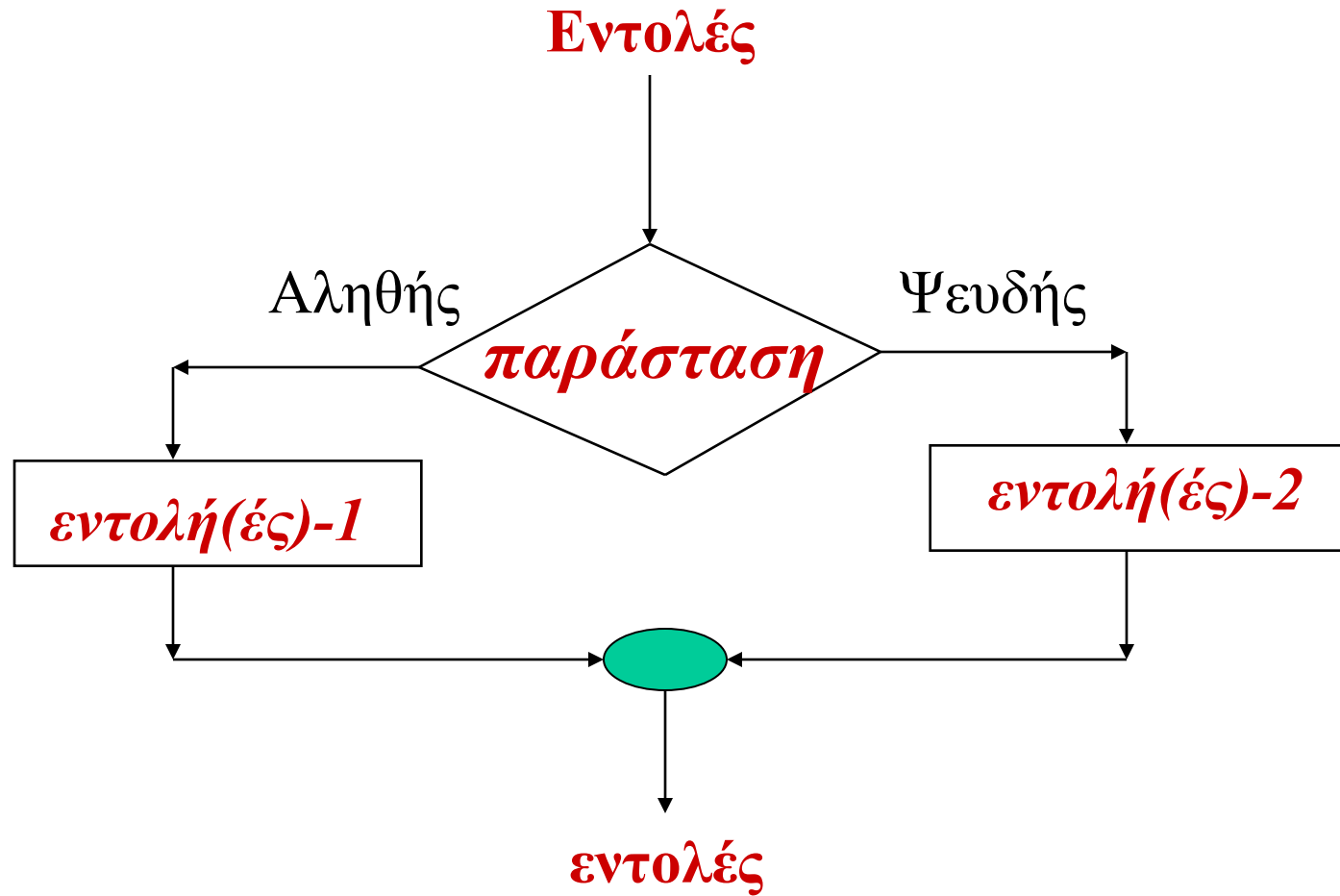
`else`

{

ομάδα εντολών 2

}

Διάγραμμα ροής



Ασάφεια

Το else συνδυάζεται με το εσωτερικό if όπως δείχνει και η εσοχή:

```
if (n > 0)
    if (a > b)
        z = a;
    else
        z = b;
```

Εναλλακτικά πρέπει να χρησιμοποιήσουμε άγκιστρα:

```
if (n > 0) {
    if (a > b)
        z = a;
}
else
    z = b;
```

Παράδειγμα

Η ασάφεια είναι καταστροφική στο παρακάτω παράδειγμα

```
if (n >= 0)
    for (i = 1; i < n; i++)
        if (s[i] > 0 ) {
            printf("...");
        }
else /* ΛΑΘΟΣ */
    printf("λάθος --- ο n είναι αρνητικός\n");
```

- Η εσοχή δείχνει αδιαμφισβήτητα τι θέλουμε αλλά ο compiler θα συνδέσει το else με το εσωτερικό if
- Με την ευκαιρία προσέξτε ότι υπάρχει (;) μετά το
printf("λάθος --- ο n είναι αρνητικός\n");
εφόσον το else ακολουθείται από εντολή, που πάντα τερματίζει με (;)

else-if

Η κατασκευή:

if (παράσταση)

εντολή

else if (παράσταση)

εντολή

else if (παράσταση)

εντολή

else if (παράσταση)

εντολή

else

εντολή

- Περιγράφει μια διακλαδωμένη απόφαση
- οι παραστάσεις υπολογίζονται με τη σειρά
- αν κάποια παράσταση είναι αληθής εκτελείται η εντολή που συνδέεται με αυτήν και όλη η αλυσίδα τερματίζεται
- το τελευταίο else είναι η προεπιλεγμένη περίπτωση όπου τίποτα από τα παραπάνω δεν είναι αληθές
- Αν δεν ισχύει κάτι συγκεκριμένο μπορεί και να παραλειφθεί

Το πρόγραμμα του Μαγικού Αριθμού

```
#include <stdio.h>

main( )
{
    system("chcp 1253");
    int magic = 123;
    int guess;
    printf("Πληκτρολογήστε ένα ακέραιο αριθμό : " );
    scanf(" %d", &guess);
    /* Διαβάζει ένα ακέραιο αριθμό */
    if (guess == magic)
        printf("*** Σωστό ***");
}
```

Το Πρόγραμμα του Μαγικού Αριθμού που εμφανίζει και τα λανθασμένα μηνύματα

```
#include <stdio.h>

main( )
{
    int magic = 123;
    int guess;
    printf("Πληκτρολογήστε ένα ακέραιο αριθμό : " );
    scanf(" %d", &guess);
    if ( guess == magic )
        printf("*** Σωστό ***");
    else
        printf("..Λάθος..");
}
```

Το πρόγραμμα του Μαγικού Αριθμού με μια αλληλοδιαδοχή if – else if

```
#include <stdio.h>

main( )
{   int magic = 123;
    int guess;
    printf("Πληκτρολογήστε ένα ακέραιο αριθμό : ");
    scanf(" %d", &guess);
    if (guess == magic) {
        printf("** Σωστό **");
        printf("Ο μαγικός αριθμός είναι = %d", magic);
    } else if (guess > magic)
        printf("..Λάθος.. Μεγαλύτερος ");
    else
        printf("..Λάθος.. Μικρότερος");
}
```

Το εναλλακτικό ?

```
#include <stdio.h>

main( )
{
    int magic = 123;
    int guess;
    printf("Πληκτρολογήστε ένα ακέραιο αριθμό : " );
    scanf(" %d", &guess);
    if (guess == magic) {
        printf("** Σωστό **");
        printf(" Ο μαγικός αριθμός είναι : %d",magic);
    }
    else
        guess > magic ? printf("Μεγαλύτερος ") : printf("Μικρότερος");
}
```

Η εντολή **switch**

- Πρόκειται για μια διακλαδωμένη απόφαση
- ελέγχει αν μια παράσταση ταυτίζεται με μια τιμή από ένα σύνολο σταθερών ακέραιων τιμών και διακλαδώνεται ανάλογα.
- Ο γενικός τύπος της εντολής switch είναι :

```
switch (μεταβλητή) {  
    case σταθερά1:  εντολές; break;  
    case σταθερά2 :  εντολές; break;  
    case σταθερά3 :  εντολές; break;  
    .  
    default :  εντολές;  
}
```


Παράδειγμα της εντολής **switch**

```
menu ( )  
{  
    char ch;  
    printf("1. Έλεγχος Ορθογραφίας \n");  
    printf("2. Διόρθωση των λαθών \n");  
    printf("3. Εμφάνιση των λαθών \n");  
    printf(" Πάτα ένα άλλο πλήκτρο για έξοδο \n");  
    printf(" Δώσε την επιλογή σου : ");  
    ch = getchar( );
```

(συνεχίζεται...)

Παράδειγμα της εντολής switch

(συνέχεια)

```
switch (ch) {  
    case '1' :  
        check_spelling( );  
        break;  
    case '2' :  
        correct_errors( );  
        break;  
    case '3' :  
        display_errors( );  
        break;  
    default :  
        printf ("Δεν έγινε επιλογή") ;  
}
```

Μέτρηση εμφανίσεων ψηφίου, λευκού διαστήματος, χαρακτήρων

```
#include <stdio.h>

main() /* count digits, white space, others */
{
    int c, i, nwhite, nother, ndigit[10];

    nwhite = nother = 0;
    for (i = 0; i < 10; i++)
        ndigit[i] = 0;
    while ((c = getchar()) != EOF) {
        switch (c) {
            case '0': case '1': case '2': case '3': case '4':
            case '5': case '6': case '7': case '8': case '9':
                ndigit[c-'0']++;
                break;
            case ' ':
            case '\n':
            case '\t':
                nwhite++;
                break;
            default:
                nother++;
                break;
        }
    }
    printf("digits =");
    for (i = 0; i < 10; i++)
        printf(" %d", ndigit[i]);
    printf(", white space = %d, other = %d\n",
        nwhite, nother);
    return 0;
}
```

Άσκηση

- Να γράψετε μία συνάρτηση η οποία θα επιστρέφει την τιμή της μεγαλύτερης από τις δύο παραμέτρους της.

```
int max(int x, int y)
{
    if ( x > y )
        return x;
    else
        return y;
}
```

Άσκηση

Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει έναν ακέραιο αριθμό από το πληκτρολόγιο και θα εμφανίζει το μήνυμα Hello Student και την τιμή του ακέραιου αριθμού που έχει πληκτρολογηθεί .
Το πρόγραμμα να σταματά χωρίς μήνυμα, όταν ο εισαγόμενος αριθμός είναι το μηδέν