

Εντολές επιλογής – Επαναλήψεις (if, switch, while)

Οι σημειώσεις αυτές έχουν σαν στόχο την μάθηση εντολών επιλογής (if, switch, while) που ελέγχουν τη ροή εκτέλεσης ενός προγράμματος. Πρώτα όμως, είναι αναγκαίο να περιγραφούν συνθήκες και λογικές εκφράσεις οι οποίες έχουν άμεση σχέση με τις εντολές αυτές.

Συνθήκες

Ένα πρόγραμμα επιλέγει μεταξύ εναλλακτικών δηλώσεων ελέγχοντας τη τιμή μεταβλητών. Για παράδειγμα, η πιο κάτω πρόταση:

αν ο γενικός βαθμός ενός φοιτητή σε ένα εξάμηνο είναι πέντε και άνω, τότε ο φοιτητής έχει επιτύχει, διαφορετικά έχει αποτύχει

περιλαμβάνει την ακόλουθη έκφραση:

γενικός βαθμός ενός φοιτητή σε ένα εξάμηνο >=πέντε

η οποία συγκρίνει τον γενικό βαθμό και αποτιμεί σε 1 (true) όταν ο γενικός βαθμός είναι μεγαλύτερος ή ίσος του πέντε, ή σε 0 (false) όταν ο γενικός βαθμός δεν είναι μεγαλύτερος του πέντε. Τέτοια έκφραση ονομάζεται συνθήκη η οποία είναι είτε αληθής είτε ψευδής. Αν είναι αληθής τότε θα εκτελεστεί μια ομάδα εντολών, διαφορετικά αν είναι ψευδής η ομάδα εντολών θα αγνοηθεί και δε θα εκτελεστεί.

Τελεστές

Τελεστής	Έννοια
<	Μικρότερο από
>	Μεγαλύτερο από
<=	Μικρότερο ή ίσο από
>=	Μεγαλύτερο ή ίσο από
==	Ίσο με
!=	Άνισο με

Οι τελεστές μπορούν να εφαρμοστούν μεταξύ μεταβλητών, ή μεταξύ μεταβλητών και σταθερών τιμών.

Λογικοί Τελεστές

Με τους λογικούς τελεστές:

- && (and)
- || (or)
- ! (not)

μπορούμε να δημιουργήσουμε πιο πολύπλοκες συνθήκες ή λογικές εκφράσεις.

Πίνακας αληθείας τελεστή && (and)

Τελεστέος A	Τελεστέος B	A && B
0	0	0
0	1	0
1	0	0
1	1	1

Πίνακας αληθείας τελεστή || (or)

Τελεστέος A	Τελεστέος B	A B
0	0	0
0	1	1
1	0	1
1	1	1

Πίνακας αληθείας τελεστή ! (not)

Τελεστέος A	!A
0	1
1	0

Προτεραιότητες

Τελεστής	Προτεραιότητα
Κλήση συνάρτησης	Υψηλότερη ↓ Χαμηλότερη
! + - & (unary operators)	
* / %	
+ -	
< <= >= >	
== !=	
&&	
=	

Δήλωση if

Η δήλωση if με μια εναλλακτική λύση έχει την εξής σύνταξη:

```
if (συνθήκη)
{
Εντολή 1;
Εντολή 2;
.....
}
```

Αν η συνθήκη είναι αληθής, τότε οι εντολές μέσα στα { } θα εκτελεστούν. Αν είναι ψευδής οι εντολές αυτές θα αγνοηθούν και το πρόγραμμα θα συνεχίσει μετά το }.

Η δήλωση if με δύο εναλλακτικές λύσεις έχει την εξής σύνταξη:

```
if (συνθήκη)
{
Εντολή 1;
Εντολή 2;
.....
}

else
{
Εντολή 1;
Εντολή 2;
.....
}
```

Αν η συνθήκη είναι αληθής, τότε οι εντολές μέσα στα πρώτα { } θα εκτελεστούν. Αν είναι ψευδής οι εντολές αυτές θα αγνοηθούν και το πρόγραμμα θα εκτελέσει τις εντολές που βρίσκονται μέσα στα δεύτερα { }.

Στο παράδειγμα που είδαμε πιο πάνω, η πρόταση:

αν ο γενικός βαθμός ενός φοιτητή σε ένα εξάμηνο είναι πέντε και άνω, τότε ο φοιτητής έχει επιτύχει, διαφορετικά έχει αποτύχει

μεταφράζεται στη c ως εξής (δημιουργώντας τη μεταβλητή generalGrade):

```
if ( generalGrade >= 5 )
{
    printf("Epitixia");
}
else
{
    printf("Apotixia");
}
```

Nested if – πολλαπλές εναλλακτικές αποφάσεις

Μπορούμε να χρησιμοποιήσουμε δηλώσεις if οι οποίες να βρίσκονται μέσα σε άλλες δηλώσεις if για απεικόνιση πολλαπλών εναλλακτικών αποφάσεων.

Πιο κάτω δίνεται ένα παράδειγμα:

```
if (x > 0)
    num_pos = num_pos + 1;
else
    if (x < 0)
        num_neg = num_neg + 1;
    else /* x ίσο με μηδέν */
        num_zero = num_zer + 1;
```

Επειδή τα nested if μπορούν να γίνουν πολύπλοκα, στις περιπτώσεις όπου κάθε ψευδής αποτίμηση (εκτός από την τελευταία) αποτελείται από μια δήλωση της μορφής if-then-else, τότε το nested if μπορεί να υλοποιηθεί με τον ακόλουθο τρόπο σύνταξης:

```
if (συνθήκη)
```

```
{
```

```
Εντολή 1;
```

```
Εντολή 2;
```

```
.....
```

```
}
```

```
else if (συνθήκη)
```

```
{
```

```
Εντολή 1;
```

```
Εντολή 2;
```

```
.....
```

```
}
```

```
....
```

```
else if (συνθήκη)
```

```
{
```

```
Εντολή 1;
```

```
Εντολή 2;
```

```
.....
```

```
}
```

Συνεπώς, το προηγούμενο παράδειγμα υλοποιείται ως εξής:

```
if (x > 0)  
    num_pos = num_pos + 1;  
  
else if (x < 0)  
    num_neg = num_neg + 1;  
  
else    /* x ίσο με μηδέν */  
    num_zero = num_zer + 1;
```

Τα δύο πιο πάνω παραδείγματα επιτυγχάνουν το ίδιο αποτέλεσμα. Το δεύτερο παράδειγμα όμως είναι το επιθυμητό, καθώς μειώνει σημαντικά την πολυπλοκότητα του κώδικα και τον κάνει πιο ευανάγνωστο.

Nested if με περισσότερες από μια μεταβλητές

Δίνεται το ακόλουθο παράδειγμα:

```
if (marital_status == 'S')
    if (gender == 'M')
        if (age >= 18 && age <=26)
            printf("All criteria are met.");
```

Η εντολή printf θα εκτελεστεί μόνο αν όλες οι συνθήκες είναι αληθείς. Το πιο πάνω παράδειγμα μπορεί να γραφτεί και με τον εξής πιο απλό τρόπο κάνοντας χρήση μόνο ενός if:

```
if ( (marital_status == 'S') && (gender == 'M') && (age >= 18) &&
    (age <=26) )
    printf("All criteria are met.");
```

Ο τρόπος αυτός έχει άμεση σχέση με τους λογικούς τελεστές που περιγράφηκαν προηγουμένως.

Switch

Το switch είναι ένας άλλος τρόπος επιλογής μιας μεταξύ πολλών εναλλακτικών αποφάσεων. Είναι ιδιαίτερα χρήσιμο στις περιπτώσεις που η επιλογή είναι βασισμένη στη τιμή μιας συγκεκριμένης μεταβλητής ή μιας έκφρασης (controlling expression). Η τιμή της έκφρασης μπορεί να είναι του τύπου **int** ή **char**, αλλά όχι **double**.

Η σύνταξη του switch έχει ως εξής:

switch (controlling expression)

```
{  
    case 'τιμή':  
        Εντολή 1;  
        Εντολή 2;  
        .....  
        break;  
  
    case 'τιμή':  
        Εντολή 1;  
        Εντολή 2;  
        .....  
        break;  
  
    .....  
  
    case 'τιμή':  
        Εντολή 1;  
        Εντολή 2;  
        .....  
        break;  
  
    Default:  
        Εντολή 1;  
}
```

Πιο κάτω δίνεται ένα παράδειγμα που υλοποιεί τον ακόλουθο πίνακα αποφάσεων:

Ταυτότητα κατηγορίας	Κατηγορία Πλοίου
B ή b	Battleship
C ή c	Cruiser
D ή d	Destroyer
F ή f	Frigate

```
switch (class)
{
    case 'B':
    case 'b':
        printf("Battleship");
        break;

    case 'C':
    case 'c':
        printf("Cruiser");
        break;

    case 'D':
    case 'd':
        printf("Destroyer");
        break;

    case 'F':
    case 'f':
        printf("Frigate");
        break;

    default:
        printf("Unknown ship class %c", class);
}
```

Πιο πάνω, class είναι μια μεταβλητή τύπου char.

Μελέτη Περίπτωσης

Να γραφτεί ένα πρόγραμμα σε C στο οποίο ο χρήστης θα καταχωρεί τρεις αριθμούς που αντιπροσωπεύουν βαθμούς σε μαθήματα. Οι τρεις αριθμοί έχουν πεδίο τιμών από 1 μέχρι και 100. Ακολούθως, το πρόγραμμα θα υπολογίζει το μέσο όρο τους και θα τυπώνει το τελικό βαθμό με βάση τα πιο κάτω κριτήρια:

Αν ο μέσος όρος βρίσκεται μεταξύ:

86-100	Τύπωσε το χαρακτήρα A.
65-85	Τύπωσε το χαρακτήρα B.
50-64	Τύπωσε το χαρακτήρα C.
25-49	Τύπωσε το χαρακτήρα D.
1-24	Τύπωσε το χαρακτήρα E.

Το πρόγραμμα να κάνει επίσης έλεγχο για την ορθότητα των βαθμών που δίνονται από το χρήστη. Σε περίπτωση που ο χρήστης δώσει βαθμό έξω από το πεδίο τιμών 1 – 100, θα τυπώνεται μήνυμα λάθους και θα επιτρέπει στο χρήστη να δώσει νέο βαθμό.