Προγραμματισμός Υπολογιστών

Δομές ελέγχου (1-3)

Νικόλαος Ζ. Ζάχαρης
Καθηγητής

Πανεπιστήμιο Δυτικής Αττικής
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Η δήλωση if

Ο πιο απλός τρόπος για να ελέγξουμε την ροή ενός προγράμματος είναι με την βοήθεια της δήλωσης if. Η σύνταξη της δήλωσης είναι :

```
if( Ακεραία_Τιμή)
{
   Δήλωση_1
   Δήλωση_2
   ..............
}
```

Μέσα στο if μπορεί να είναι απλές δηλώσεις (statements), όπως για παράδειγμα a = 2; ή σύνθετες εντολές όπως για παράδειγμα, να είναι άλλες δηλώσεις if

Αν η **Ακεραία Τιμή** είναι διάφορη του 0 τότε και μόνο τότε θα εκτελεστούν όλες οι δηλώσεις που εμπεριέχονται μέσα στο if

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

int main(int argc, char *argv[]) {

  if(1) {
     printf("A statement inside if\n");
  }
   printf("A statement outside if");
  return 0;
}
```

```
A statement inside if
A statement outside if
```

Στο παράδειγμα η δήλωση if είναι πάντα αληθής οπότε πάντα εμφανίζεται το μήνυμα. Αν αλλάξουμε το if(1) σε if(0) τότε δεν θα εμφανίζεται το μήνυμα.

Τα άγκιστρα δεν είναι απαραίτητα αν μέσα στο if υπάρχει μια δήλωση. Όμως είναι αρκετά βοηθητικά στη κατανόηση του κώδικα οπότε προτείνεται να τα χρησιμοποιείται πάντα στο κώδικα σας.

Η δήλωση if..else

Η δήλωση if μπορεί να συνοδεύεται από την δήλωση else ή οποία εκτελείται στην περίπτωση που η **Ακεραία Τιμή** είναι 0. Η σύνταξη των δηλώσεων είναι :

```
if(Ακεραία_Τιμή)
{
   δηλώσεις
}
else {
   δηλώσεις
}
```

Το σύνολο των δηλώσεων που θα εκτελεστεί μόνο αν η ***Ακεραία_Τιμή*** είναι διάφορή του 0

Το σύνολο των δηλώσεων που θα εκτελεστεί μόνο αν η ***Ακεραία_Τιμή*** είναι 0

Στο διπλανό παράδειγμα επειδή το if περιέχει τη τιμή 0 τότε θα εκτελεστούν μόνο οι δηλώσεις του else. Αν αλλάξουμε το if(0) σε if(1) τότε θα εκτελεστούν οι δηλώσεις του τμήματος if και θα αγνοηθούν οι εντολές μέσα στο else.

```
int main(int argc, char *argv[]) {

   if(0) {
            printf("The statement inside if\n");
   }
   else  {
             printf("The statement inside else\n");
   }
   printf("A statement outside if..else");
   return 0;
}
```

```
The statement inside else
A statement outside if..else
```

Η πολλαπλή δομή if

Η δήλωση if μπορεί να πάρει τη παρακάτω μορφή, η οποία είναι συνδυασμός της if και της δήλωσης if..else και μας επιτρέπει να δημιουργήσουμε μια πολλαπλή μορφή ελέγχου της ροής του κώδικα.

```
if(Ακεραία_Τιμή_1)
{
   δηλώσεις
}
else if(Ακεραία_Τιμή_2)
{
   δηλώσεις
}
else if(Ακεραία_Τιμή_3)
{
   δηλώσεις
}
else {
   δηλώσεις
}
```

Οι **Ακέραιες Τιμές** ελέγχονται μια προς μία αρχίζοντας από την **Ακεραία_Τιμή_1**, μετά τη **Ακεραία_Τιμή_2** και ούτω καθ' εξής. Εκτελούνται οι δηλώσεις της πρώτης τιμής που είναι διάφορη του 0 και συνεχίζει στο τέλος της δομής.

Μπορούμε να χρησιμοποιήσουμε όσα **else if** επιθυμούμε στο πρόγραμμα μας.

Το τελευταίο else είναι προαιρετικό και το χρησιμοποιούμε όταν θέλουμε να εκτελέσουμε εντολές στην περίπτωση που δεν είναι αληθής καμία από τις ανωτέρω δηλώσεις if.

Παράδειγμα εκτέλεσης σε πολλαπλή δομή

```
1  if(0)
   {
      δηλώσεις
   }
2  else if(1)
   {
      δηλώσεις      3
   }
   else if(1)
   {
      δηλώσεις
   }
   else {
      δηλώσεις
   }
   4
```

Στο διπλανό παράδειγμα, αρχικά θα γίνει έλεγχος του πρώτου if το οποίο έχει τιμή 0 (**1**) οπότε συνεχίζει στο δεύτερο if το οποίο έχει τιμή 1 (**2**) οπότε εκτελούνται οι δηλώσεις του, (**3**) και η ροή του προγράμματος συνεχίζει στο τέλος της δομής (**4**).

Δεν έχει καμία σημασία αν στη συνέχεια υπάρχουν και άλλες δηλώσεις if με τιμές διάφορες του 0. Εκτελούνται μόνο οι δηλώσεις του πρώτου if που περιέχει τιμή διάφορη του 0 και η ροή του προγράμματος συνεχίζει στο τέλος της πολλαπλής δομής.

Τελεστές σύγκρισης

Η τοποθέτηση μια ακεραίας τιμής 0 ή 1 μέσα σε ένα if έχει μικρή αξία σε πραγματικά προβλήματα αφού δεν πρόκειται να αλλάξει η τιμή κατά τη διάρκεια της εκτέλεσης του προγράμματος. Συνήθως μέσα σε ένα if τοποθετούμε μια συνθήκη και για την δημιουργία της θα πρέπει να χρησιμοποιήσουμε έναν από τους παρακάτω τελεστές σύγκρισης.

==	σημαίνει **ισούται**
!=	σημαίνει **διάφορο**
<	σημαίνει **μικρότερο**
>	σημαίνει **μεγαλύτερο**
<=	σημαίνει **μικρότερο ή ίσο**
>=	σημαίνει **μεγαλύτερο ή ίσο**

Το αποτέλεσμα μιας συνθήκης είναι **αληθές (**αντιστοιχεί σε οποιαδήποτε τιμή διάφορη του 0**)** ή **ψευδές (**αντιστοιχεί στη τιμή 0**)**. Για παράδειγμα όταν εκτελούμε την συνθήκη **x == 3** είναι σαν να κάνουμε την ερώτηση, αν είναι η τιμή του x είναι ίση με 3. Η απάντηση σε αυτή την ερώτηση είναι αλήθεια – που σημαίνει ότι ναι η μεταβλητή x έχει την τιμή 3 – ή είναι ψέματα – που σημαίνει ότι η μεταβλητή x έχει μια οποιαδήποτε άλλη τιμή εκτός από 3.

Η C δεν έχει ένα τύπο δεδομένων για λογικές τιμές αλλά θεωρεί τη ακεραία τιμή 0 σαν ΨΕΥΔΗ και οποιαδήποτε άλλη σαν ΑΛΗΘΗ, συνήθως χρησιμοποιείται το 1.

Παράδειγμα :

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


int main(int argc, char *argv[])
{
    int x = (10 == 10);
    printf("%d",x);   // Εκτυπώνεται η τιμή 1 που αντιστοιχεί στη τιμή ΑΛΗΘΗΣ
    return 0;
}
```

Στον ανωτέρω κώδικα δημιουργούμε μια μεταβλητή x στην οποία αναθέτουμε το αποτέλεσμα της ερώτησης αν η τιμή 10 ισούται με την τιμή 10. Αυτό είναι αλήθεια οπότε στην μεταβλητή x αποθηκεύεται η τιμή 1 που αντιστοιχεί στη τιμή ΑΛΗΘΗΣ.

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


int main(int argc, char *argv[])
{
    int y = (5 >= 10);
    printf("%d",y);   // Εκτυπώνεται η τιμή 0 που αντιστοιχεί στη τιμή ΨΕΥΔΗΣ
    return 0;
}
```

Στον ανωτέρω κώδικα δημιουργούμε μια μεταβλητή y στην οποία αποδίδουμε το αποτέλεσμα της ερώτησης αν η τιμή 5 είναι μεγαλύτερη από την τιμή 10. Αυτό είναι ψέματα οπότε στην μεταβλητή y αποθηκεύεται η τιμή 0 που αντιστοιχεί στη τιμή ΨΕΥΔΗΣ.

Παράδειγμα

Να διαβαστεί ένας αριθμός και να εκτυπωθεί η απόλυτη τιμή του.

Αρχή Αλγόριθμου

Διάβασε Α

Αν Α < 0 **Τότε** Α ← Α*(-1)

Εκτύπωσε Α

Τέλος Αλγόριθμου

Σημείωση

Η απόλυτη τιμή ενός αριθμού είναι πάντα θετικός αριθμός, οπότε αν διαβάσουμε από το πληκτρολόγιο μια αρνητική τιμή θα πρέπει να τη μετατρέψουμε σε θετική τιμή. Για τη μετατροπή ενός αρνητικού σε θετικό αριθμό θα πρέπει να τον πολλαπλασιάσουμε με το -1.

Η υλοποίηση του προγράμματος για την εύρεση της απόλυτης τιμής

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

int main(int argc, char *argv[])
{
    int a;

    printf("Type a number  : ");
    scanf("%d", &a);

    if(a < 0)
    {
        a = a * (-1);
    }

    printf("The absolute value is : %d", a);

   return 0;
}
```

Σύνθετες συνθήκες

Όλες οι γλώσσες προγραμματισμού μας δίνουν την δυνατότητα να δημιουργήσουμε σύνθετες συνθήκες οι οποίες αποτελούνται από περισσότερες από μια, απλές συνθήκες. Για παράδειγμα δύο απλές συνθήκες είναι :

x >= 10	Η τιμή του x είναι μεγαλύτερη ή ίση με το 10
x <= 50	Η τιμή του x είναι μικρότερη ή ίση με το 50

Αυτές οι συνθήκες μπορούν να συνδυαστούν ώστε να δημιουργήσουν μια σύνθετη συνθήκη όπου η ερώτηση θα είναι :

Είναι η τιμή του x μεγαλύτερη ή ίση με το 10 **ΚΑΙ** η τιμή του x είναι μικρότερη ή ίση με το 50

Σε αυτή την περίπτωση η σύνθετη συνθήκη θα εκτιμηθεί συνολικά και θα έχει σαν αποτέλεσμα την τιμή αλήθεια (δηλαδή 1) αν η x έχει οποιαδήποτε τιμή από 10 μέχρι και 50 και θα έχει τιμή ψέματα (δηλαδή 0) αν η x έχει οποιαδήποτε άλλη τιμή.

Για την δημιουργία μιας σύνθετης συνθήκης θα πρέπει ανάμεσα στις απλές συνθήκες να τοποθετήσουμε ένα λογικό τελεστή.

Λογικοί τελεστές

Τελεστής	*Σημασία*
&&	Λογικό ΚΑΙ
||	Λογικό Η
!	Λογικό ΟΧΙ

Οι λογικοί τελεστές χρησιμοποιούνται για την συνένωση των αποτελεσμάτων από τις απλές συνθήκες. Για παράδειγμα :

`int k = (x > 2) && (x < 10);`	k = 1 αν ισχύουν και οι δυο συνθήκες. Για παράδειγμα, το x έχει τιμή 8
`int k = (x > 2) || (x < 10);`	k = 1 αν ισχύει μια από τις δυο συνθήκες Για παράδειγμα, το x έχει τιμή 4
`int k = !(x > 2);`	k = 1 αν δεν ισχύει η συνθήκη x > 2. Για παράδειγμα, το x έχει τιμή 0

Το λογικό ΚΑΙ

Ο πίνακας αλήθειας του λογικού τελεστή ΚΑΙ μας δείχνει πως θα εκτιμηθεί η σύνθετη συνθήκη σε σχέση με την αλήθεια των απλών συνθηκών.

Συνθήκη Α **x >= 2**	**Συνθήκη Β** **x <= 50**	**Α && Β** **(x >= 2) && (x <= 50)**
0	0	0
0	1	0
1	0	0
1	1	1

Από τον ανωτέρω πίνακα βλέπουμε ότι η σύνθετη συνθήκη είναι 1 (αληθής) μόνο στην περίπτωση που και οι δύο απλές συνθήκες είναι αληθείς. Σε οποιαδήποτε άλλη περίπτωση, η σύνθετη συνθήκη είναι 0 (ψευδής). Το λογικό ΚΑΙ χρησιμοποιείται όταν θέλουμε να αληθεύουν όλες οι επιμέρους συνθήκες. Στο παράδειγμα η σύνθετη συνθήκης είναι αληθής μόνο όταν η τιμή του x είναι ανάμεσα στο 2 και στο 50 (π.χ. Όταν το x έχει τιμή 7). Σε οποιαδήποτε άλλη περίπτωση είναι ψευδής (π.χ. Όταν το x έχει τιμή 67).

Το λογικό Ή

Ο πίνακας αλήθειας του λογικού τελεστή Η μας δείχνει πως θα εκτιμηθεί η σύνθετη συνθήκη σε σχέση με την αλήθεια των απλών συνθηκών.

Συνθήκη Α **x < 0**	**Συνθήκη Β** **x > 50**	**Α || Β** **(x < 0) || (x > 50)**
0	0	0
0	1	1
1	0	1
1	1	1

Από τον ανωτέρω πίνακα βλέπουμε ότι η σύνθετη συνθήκη είναι 1 (αληθής) στις περιπτώσεις που και οι δύο απλές συνθήκες είναι αληθείς ή μόνο η μια εκ των δύο απλών συνθηκών. Το λογικό Η χρησιμοποιείται όταν θέλουμε να αληθεύει μία εκ των επιμέρους συνθηκών. Στο παράδειγμα η σύνθετη συνθήκης είναι αληθής μόνο όταν η τιμή του x είτε είναι μικρότερη του 0 ή είναι μεγαλύτερη από το 50. (π.χ. Όταν το x έχει τιμή -8). Σε οποιαδήποτε άλλη περίπτωση είναι ψευδής (π.χ. Όταν το x έχει τιμή 34).

Η λογική Άρνηση

Ο πίνακας αλήθειας της λογικής άρνησης μας δείχνει πως θα εκτιμηθεί η σύνθετη συνθήκη σε σχέση με την αλήθεια της απλής συνθήκης.

Συνθήκη Α **x >= 2**	**!Α** **!(x >= 2)**
0	1
1	0

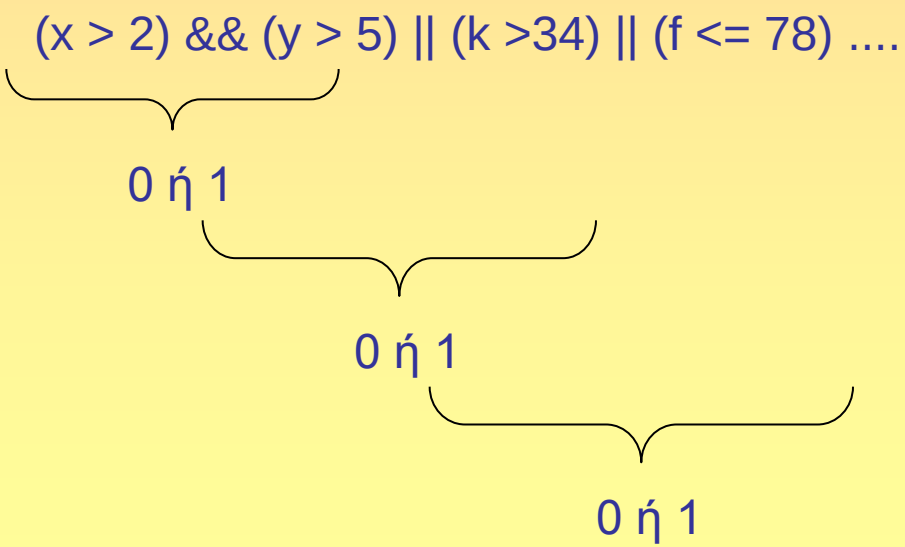
Από τον ανωτέρω πίνακα βλέπουμε ότι η σύνθετη συνθήκη είναι αληθής στην περίπτωση που η απλή συνθήκη είναι ψευδής διαφορετικά είναι αληθής. Η λογική άρνηση χρησιμοποιείται όταν θέλουμε να αντιστρέψουμε το αποτέλεσμα της απλής συνθήκης όπως στο συγκεκριμένο παράδειγμα που η συνθήκη είναι αληθής για όλες τις τιμές που μικρότερες από 2. (π.χ. Όταν το x έχει τιμή -8).

Εκτίμηση πάνω από δύο συνθήκες

Η δημιουργία μιας σύνθετης συνθήκης δεν περιορίζεται σε δύο μόνο απλές συνθήκες ούτε και στην ίδια μόνο μεταβλητή αλλά σε όσες μεταβλητές απαιτεί το πρόγραμμα μας. Για παράδειγμα, η συνθήκη

(x > 2) && (y > 5) && (k >34) && (f <= 78) ....

θα εκτιμηθεί ανά δύο απλές συνθήκες όπως παρακάτω :



Εκτίμηση πάνω από δύο συνθήκες (συνέχεια)

Στην περίπτωση που όλες οι απλές συνθήκες συνδέονται με το λογικό τελεστή ΚΑΙ και ένα εκ των αποτελεσμάτων είναι 0 (ψευδής) τότε σταματά ή εκτίμηση των επόμενων συνθηκών και ολόκληρη η σύνθετη συνθήκη παίρνει τη τιμή 0 αφού δεν έχει πλέον νόημα να συνεχιστεί η εκτίμηση των επιμέρους συνθηκών.

Στην περίπτωση που όλες οι απλές συνθήκες συνδέονται με το λογικό τελεστή Ή και ένα εκ των αποτελεσμάτων είναι 1 (αληθής – true) τότε σταματά ή εκτίμηση των επόμενων συνθηκών και ολόκληρη η σύνθετη συνθήκη παίρνει τη τιμή 1 αφού δεν έχει πλέον νόημα να συνεχιστεί η εκτίμηση των επιμέρους συνθηκών.

Σε μια σύνθετη συνθήκη μπορούμε να χρησιμοποιήσουμε παρενθέσεις για να δηλώσουμε με ακρίβεια την σειρά με την οποία θέλουμε να εκτιμηθεί η συνθήκη. Για παράδειγμα η συνθήκη :

```
(x > 2) || ( (y > 5) && (k >34) )
```

θα εκτιμηθεί διαφορετικά από τη συνθήκη :

```
( (x > 2) || (y > 5) ) && (k >34)
```