

## ΠΡΟΤΑΣΕΙΣ, ΕΚΦΡΑΣΕΙΣ ΚΑΙ ΤΕΛΕΣΤΕΣ

### 1. Η ΕΝΝΟΙΑ ΤΩΝ ΤΕΛΕΣΤΩΝ

Ένας τελεστής είναι ένα σύμβολο που οδηγεί την 'C' να εκτελέσει κάποια λειτουργία σε μεταβλητές και εκφράσεις. Οι τελεστές της 'C' χωρίζονται σε διάφορες κατηγορίες:

- Οι αριθμητικοί τελεστές
- Τελεστές Αύξησης-Μείωσης
- Οι σχεσιακοί τελεστές
- Οι λογικοί τελεστές

#### 1.1 Αριθμητικοί Τελεστές

##### 1.1.1 Περιγραφή

Σύμβολο	Σημασία
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση (πηλίκο ή πραγματική διαίρεση)
%	Υπόλοιπο

Πίνακας 1: Αριθμητικοί Τελεστές

### 1.1.2 Σειρά Προτεραιότητας

Η προτεραιότητα καθορίζει ακριβώς πως η 'C' υπολογίζει τύπους σε αριθμητικές παραστάσεις. Η προτεραιότητα των τελεστών είναι η ίδια με αυτή της σχολικής άλγεβρας. Ο Παρακάτω Πίνακας απεικονίζει την προτεραιότητα των μαθηματικών τελεστών.

Σειρά	Τελεστής
Πρώτη	*, /, %
Δεύτερη	+, -

*Πίνακας 2: Προτεραιότητα Αριθμητικών Τελεστών*

### ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>
main()
{
    int a; int b;
    a=6+2*3-4/2; b=10/5*2-2+1
    printf("a equals to %d\n",a);
    printf("b equals to %d\n",b);
}
```

Στην οθόνη του υπολογιστή εμφανίζεται το παρακάτω αποτέλεσμα:

```
a equals to 10
b equals to 3
```

### 1.1.3 Χρήση Παρενθέσεων

Για να αλλάξετε τη σειρά προτεραιότητας μπορείτε να χρησιμοποιήσετε παρενθέσεις. Οι παρενθέσεις βρίσκονται ένα επίπεδο πιο πάνω από τον πολλαπλασιασμό, τη διαίρεση και το υπόλοιπο στον πίνακα προτεραιότητας.

#### ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>
main()
{
    int a; int b;
    a=20/(3+5)%2; b=5*(5+(6-2)+1);
    printf("a equals to %d\n",a);
    printf("b equals to %d\n",b);
}
```

Στην οθόνη του υπολογιστή εμφανίζεται το παρακάτω αποτέλεσμα

```
a equals to 0
b equals to 10
```

### 1.1.4 Πολλαπλές Εκχωρήσεις

Αν σε μια παράσταση εμφανίζονται δύο ή περισσότερα σύμβολα '=' καθένα εκτελεί μια εκχώρηση. Αυτό εισάγει ένα νέο θέμα στη σειρά προτεραιότητας. Στη C το σύμβολο '=' σημαίνει πάντα εκχώρηση της τιμής στα δεξιά στην μεταβλητή που βρίσκεται στα αριστερά.

#### ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>
main ()
{
    float a; float b; float c;
    a=scanf("%f",&a);
    c=5.0+(b=9.0-c);
    printf("a equals to %.2f\n",a); printf("c equals to %.2f\n",c);
}
```

Στην οθόνη του υπολογιστή εμφανίζεται το παρακάτω αποτέλεσμα

```
?10.5  
a equals to 10.50  
b equals to 3.50
```

### 1.1.5 Σύνθετες Εκχωρήσεις

Συχνά στον προγραμματισμό, θέλετε να αλλάξετε την τιμή μιας μεταβλητής. Δηλαδή, θέλετε να προσθέσετε ή να πολλαπλασιάσετε την τιμή με μια παράσταση και να εκχωρήσετε το αποτέλεσμα ξανά στην αρχική μεταβλητή. Αυτό μπορεί να γίνει χρησιμοποιώντας τον ακόλουθο πίνακα.

Τελεστής	Παράδειγμα	Ισοδύναμο
+=	b+=200;	b=b+200;
-=	salary-=30;	salary=salary-30;
*=	factor*=2.5;	factor=factor*2.5;
/=	Num_1/=3.15;	Num_1=Num_1/3.15;
%=	Daynum%=7;	Daynum=Daynum%7;

Πίνακας 3: Σύνθετοι Τελεστές

### 1.2 Τελεστές Αύξησης και Μείωσης

Η C έχει δύο μοναδικούς τελεστές που προσθέτουν ή αφαιρούν 1 σε μεταβλητές. Ο ακόλουθος πίνακας παρουσιάζει τη σχέση των τελεστών αύξησης-μείωσης με άλλους τύπους παραστάσεων που ήδη γνωρίζετε.

Τελεστής	Τύπος	Παράδειγμα	Ισοδύναμο
++	Επιθεματικός	++b;	b=b+1;
--	Επιθεματικός	--b;	b=b-1;
++	Προθεματικός	b++;	b=b+1;
--	Προθεματικός	b--;	b=b-1;

Πίνακας 4: Τελεστές Αύξησης-Μείωσης

Αν μια μεταβλητή χρησιμοποιείται με επιθεματικό τελεστή τότε εκτελείται πρώτα ο τελεστής αύξησης ή μείωσης πριν χρησιμοποιηθεί η τιμή της μεταβλητής. Αντίθετα, αν μια μεταβλητή χρησιμοποιείται με προθεματικό τελεστή τότε ο τελεστής αύξησης ή μείωσης τελείται αφού χρησιμοποιηθεί η τιμή της μεταβλητής.

## ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>
main ()
{
    int a; int b;
    a=6; b=++a-1;
    printf("a equals to %d\n",a); printf("b equals to %d\n",b);
}
```

Στην οθόνη του υπολογιστή εμφανίζεται το παρακάτω αποτέλεσμα:

```
a equals to 7
b equals to 6
```

## ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>
main()
{
    int a; int b;
    a=6; b=a++-1;
    printf("a equals to %d\n",a); printf("b equals to %d\n",b);
}
```

Στην οθόνη του υπολογιστή εμφανίζεται το παρακάτω αποτέλεσμα:

```
a equals to 7
b equals to 5
```

## 1.3 Σχεσιακοί Τελεστές

### 1.3.1 Περιγραφή

Οι σχεσιακοί τελεστές χρησιμοποιούνται για να συγκρίνουν εκφράσεις θέτοντας ερωτήσεις όπως 'Είναι η  $y$  μεγαλύτερη από το 50;' ή 'Είναι η  $x$  ίση με 0'. Μια έκφραση που περιέχει ένα σχεσιακό τελεστή υπολογίζεται είτε ως αληθής (Διαδικό 1) ή ως ψευδής (Διαδικό 0). Οι έξι σχεσιακοί τελεστές της 'C' παρουσιάζονται στον παρακάτω Πίνακα.

Τελεστής	Σύμβολο
Ίσον	==
Μεγαλύτερο από	>
Μικρότερο από	<
Μεγαλύτερο ή ίσο	>=
Μικρότερο ή ίσο	<=
Άνισο	!=

Πίνακας 5: Σχεσιακές Τελεστές

### 1.3.2 Προτεραιότητα Σχεσιακών Τελεστών

Όπως και οι μαθηματικοί τελεστές που παρουσιάστηκαν προηγουμένως, οι σχεσιακοί τελεστές έχουν κάποια προτεραιότητα καθένας, που καθορίζει με ποια σειρά θα εκτελεστεί μια έκφραση πολλών τελεστών. Για τους σχεσιακούς τελεστές, ισχύουν τα ακόλουθα:

- Όλοι οι σχεσιακοί τελεστές έχουν χαμηλότερη προτεραιότητα από τους μαθηματικούς τελεστές
- Υπάρχουν επίσης προτεραιότητα δύο επιπέδων μέσα στους σχεσιακούς τελεστές όπως δίνεται και στον παρακάτω πίνακα

Τελεστής	Σχετική Προτεραιότητα
< <= > >=	1
!= ==	2

Πίνακας 6: Προτεραιότητα Σχεσιακών Τελεστών

## 1.4 Λογικοί Τελεστές

Οι λογικοί τελεστές της 'C' επιτρέπουν το συνδυασμό δύο ή περισσότερων σχετικών εκφράσεων σε μία έκφραση που υπολογίζεται είτε σε αληθής είτε ψευδής. Ο Πίνακας 7 παραθέτει τους τρεις λογικούς τελεστές της 'C'.

Τελεστής	Σύμβολο	Παράδειγμα	Υπολογίζεται σε
AND	&&	x1 && x2	Αληθής (1) μόνο αν η x1 και η x2 είναι αληθής. Ψευδής (0) διαφορετικά
OR		x1    x2	Αληθής (1) αν είτε η x1 ή η x2 είναι αληθής. Ψευδής (0) μόνο αν και οι δύο είναι ψευδής
NOT	!	!x1	Ψευδής (0) αν η x1 είναι αληθής. Αληθής (1) αν η x1 είναι ψευδής


Πίνακας 7: Λογικοί Τελεστές

Είναι σημαντικό να γίνει κατανοητό ότι οποιαδήποτε αριθμητική τιμή ερμηνεύεται είτε ως αληθής είτε ως ψευδής, όταν χρησιμοποιηθεί σε πρόταση της C, που αναμένει μια λογική τιμή. Οι κανόνες είναι οι ακόλουθοι:

- Μια τιμή μηδέν αναπαριστά το ψευδές
- Μια μη μηδενική τιμή αναπαριστά το αληθές

## 1.4.2 Προτεραιότητα των Λογικών Τελεστών

Οι λογικές τελετές της 'C' έχουν επίσης μια σειρά προτεραιότητα και μεταξύ τους και σε σχέση με άλλους τελεστές. Ο τελεστής ! έχει προτεραιότητα ίση με τους μοναδιαίους μαθηματικούς τελεστές ++ και --. Έτσι, ο τελεστής ! έχει την υψηλότερη προτεραιότητα από τους άλλους σχεσιακούς τελεστές. Αντίθετα οι τελεστές && και || έχουν πιο χαμηλή προτεραιότητα, χαμηλότερη από όλους τους μαθηματικούς και σχεσιακούς τελεστές, αν και ο && έχει υψηλότερη προτεραιότητα από τον ||.



()
++ --! -(ως πρόσημο)
* / %
+ -
< <= > >=
== !=
&&
=

**Η σειρά προτεραιότητας φαίνεται από το βέλος .**



## 1.5 Η πρόταση if

Η σχεσιακή λογική ενσωματώνεται συνήθως σε προγράμματα 'C' με την πρόταση if. Η if είναι μια πρόταση απόφασης. Η πρόταση if ελέγχει μια σχέση με βάση:

```
if (συνθήκη)
    {μία ή περισσότερες προτάσεις της C};
```

Αν την πρόταση if ακολουθεί μια πρόταση 'C', τα άγκιστρα δεν είναι απαραίτητα (Συνίσταται να τα χρησιμοποιείται) (Προσοχή: Στην πρόταση if δεν τοποθετούμε ερωτηματικό στο τέλος της πρότασης). Η συνθήκη είναι μια σχεσιακή σύγκριση και πρέπει να περικλείεται σε παρενθέσεις και μπορεί να περιλαμβάνει μια οποιαδήποτε μεταβλητή που έχετε ορίσει ή μια αριθμητική παράσταση. Η παράσταση μπορεί να είναι μια από τις παρακάτω κατηγορίες

- Μια σταθερά
- Μια παράσταση από μεταβλητές
- Ένας συνδυασμός των παραπάνω κατηγοριών

## **ΠΑΡΑΔΕΙΓΜΑ**

```
/* Programma pou pistopoiiei tin ilikia enos foititi */
#include <stdio.h>
main()
{
    int student_age;
    printf("\nWhat is the student's age?");
    scanf("%d",&student_age);
    if (student_age<18)
    {
        printf("%c",'x07'); /*Hixitiko Sima */
        printf("**The age can not be less than 18\n");
        printf("Try again...\n");
        printf("\nWhat is the student's age?");
        scanf("%d",&student_age);
        printf("%d\n",student_age);
    }
    printf("Thank you. You entered a valid age");
}
```

Στην οθόνη του υπολογιστή εμφανίζεται το παρακάτω αποτέλεσμα:

```
What is the student's age? 6
***The age can not be less that 18***
Try again...

What is the student's age? 21
Thank you. You entered a valid age.
```

## **1.6 Η Πρόταση else**

Η πρόταση else δεν χρησιμοποιείται ποτέ χωρίς μια πρόταση if. Η ενότητα αυτή παρουσιάζει την πρόταση else μέσα από τη δημοφιλή πρόταση if-else και έχει την ακόλουθη σύνταξη

```
if (συνθήκη)
    {ενότητα μίας ή περισσότερες προτάσεις της C};
else
    {ενότητα μίας ή περισσότερες προτάσεις της C};
```

Το πρώτο μέρος της if-else είναι πανομοιότυπο με την απλή if. Αν η συνθήκη είναι αληθής, εκτελείται η ενότητα προτάσεων C μετά την if. Αν η συνθήκη είναι ψευδής όμως, εκτελείται η ενότητα προτάσεων C μετά την else. Ενώ η if καθορίζει τι συμβαίνει μόνο αν η συνθήκη είναι αληθής, η if-else καθορίζει τι σημαίνει και αν η συνθήκη είναι ψευδής.

**Προσοχή:** Στην τέλος της πρότασης else δεν τοποθετείται ερωτηματικό.

### **ΠΑΡΑΔΕΙΓΜΑ**

```
/* Programma pou pistoioiei an enas akeraios einai megalyteros apo 0 */
#include <stdio.h>
main()
{
    int akeraios;
    printf("\nWhat is the number?");
    scanf("%d",&akeraios);
    if(akeraios>0)
    {
        printf("\nMegalyteros apo miden\n");
    }
    else
    {
        printf("\nMikroteros i isos toy midenos\n");
    }
    printf("End of the program");
}
```

Ακολουθεί το αποτέλεσμα του προγράμματος. Το πρόγραμμα 'γνωρίζει' εάν ο ακέραιος αριθμός είναι μεγαλύτερος του μηδενός. Στην οθόνη του υπολογιστή εμφανίζεται το παρακάτω αποτέλεσμα:

```
What is the number? 6  
Megalyteros apo miden  
End of the program
```

### **1.7 Η σύνθετη πρόταση else if**

Αυτή είναι μια ένθετη μορφή του if όπως φαίνεται και παρακάτω:

```
if (συνθήκη 1)  
    {ενότητα 1 μίας ή περισσότερες προτάσεις της C};  
else if (συνθήκη 2)  
    {ενότητα 2 μίας ή περισσότερες προτάσεις της C};  
else  
    {ενότητα 3 μίας ή περισσότερες προτάσεις της C};  
επόμενη πρόταση;
```

Εάν η πρώτη συνθήκη (συνθήκη 1), είναι αληθής, εκτελούνται οι προτάσεις της ενότητας 1, πριν το πρόγραμμα συνεχιστεί με την επόμενη πρόταση. Εάν η πρώτη έκφραση δεν είναι αληθής, ελέγχεται η δεύτερη έκφραση (συνθήκη 2). Εάν είναι αληθής, εκτελούνται οι προτάσεις της ενότητας 2. Εάν και οι δύο προτάσεις είναι ψευδείς, εκτελούνται οι προτάσεις της ενότητας 3.

#### **ΠΑΡΑΔΕΙΓΜΑ**

```
/* Programma pou pistoioiei an enas pragmatikos einai megalyteros */  
/* mikroteros i isos me to miden */  
#include <stdio.h>  
main ()  
{  
    int akeraios;  
    printf("\nWhat is the number?");  
    scanf("%d",&akeraios);  
    if(akeraios>0)  
    {  
        printf("\nMegalyteros apo miden\n");  
    }  
    else if (akeraios==0)  
    {  
        printf("\Isos me to miden\n");  
    }  
    else  
    {  
        printf("\Mikroteros apo to miden\n");  
    }  
    printf("End of the program");  
}
```

Ακολουθεί το αποτέλεσμα του προγράμματος:

```
What is the number? -6  
Mikroteros apo to miden  
End of the program
```