



ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

- Τελεστές - Κατηγορίες
- Εκφράσεις - Κατηγορίες
- Υπολογισμός εκφράσεων
- Προτάσεις - Κατηγορίες

Τελεστές (Operators)

- Τελεστής = σύμβολο που αναπαριστά μία συγκεκριμένη στοιχειώδη διεργασία η οποία ενεργεί πάνω σε κάποια δεδομένα και παράγει κάποιο αποτέλεσμα
- Τελεστέοι = τα δεδομένα (σταθερές, μεταβλητές, κλήσεις συναρτήσεων)

Κατηγορίες Τελεστών

- Αριθμητικοί (+, -, *, /, %)
- Λογικοί (&&, ||, !)
- Σύγκρισης ή Συσχετιστικοί (>, >=, ==, !=, <, <=)
- Διαχείρισης ψηφίων (>>, <<, &, |, ^, ~)
- Διαχείρισης μνήμης (&, [], .., ->)
- Ειδικοί

Τελεστές διαχείρισης ψηφίων

&	AND για bit
	OR για bit
^	αποκλειστικό OR για bit
<<	ολίσθηση αριστερά
>>	ολίσθηση δεξιά
~	συμπλήρωμα ως προς 1

Ο τελεστής AND για bit χρησιμοποιείται για εφαρμογή μάσκας σε ένα σύνολο από bits . Για παράδειγμα η εντολή $n = n \& 0177$ μηδενίζει όλα τα bit του n εκτός από τα επτά κατώτερης τάξης.

Ειδικοί Τελεστές (1)

- Μοναδιαίοι αυξησης/μείωσης (++, --)

++x, x++ ή x=x+1;

- Σύνθετοι ανάθεσης (+=, -=, *=, /=)

x+=10; ή x=x+10;

x *=y+1; ή x=x*(y+1);

- Υπό συνθήκη (? :)

<εκφρ1> ? <εκφρ2> : <εκφρ3>

x>y? x: y

συνθήκη ελέγχου

- sizeof (π.χ. sizeof(int), sizeof(x+y))

Τελεστής υπό συνθήκη

```
If (a<b)
  z=a;
else
  z=b;
```

Η παραπάνω παράσταση υπολογίζει τον μεγαλύτερο από τους a,b και τον τοποθετεί στην z. Η ίδια παράσταση υπό συνθήκη μπορεί να γραφεί με τον τριαδικό τελεστή “?:” ως εξής:

```
z=(a>b)?a:b; /* z=max(a,b) */
```

Παράδειγμα

```
int x=10;
```

```
int y=20;
```

```
++x;
```

```
y=--x;
```

```
y+=x--;
```

```
y-=x++
```

x

y

11

20

10

10

9

20

10

11

Εκφράσεις (1)

- Έκφραση = συνδυασμός ενός ή περισσότερων τελεστών και ενός ή περισσότερων τελεστών
- Σημειογραφία Εκφράσεων
 - Ενδοθεματική (infix notation)
π.χ. $x+y$
 - Προθεματική (prefix notation)
π.χ. $+x y$
 - Μεταθεματική (postfix notation)
π.χ. $x y +$

Εκφράσεις (2)

■ Κατηγορίες εκφράσεων

- Αριθμητικές (τελεστές \rightarrow αριθμητικοί, τελεστές \rightarrow αριθμ. τιμές/μεταβλητές/εκφράσεις, αποτέλεσμα \rightarrow αριθμητικού τύπου) π.χ. $(5*x+y/4)*8$
- Σύγκρισης (τελεστές \rightarrow σύγκρισης, τελεστές \rightarrow αριθ. τιμές/μεταβλητές/εκφράσεις, αποτέλεσμα \rightarrow λογικού τύπου: 0, 1)
π.χ. $x > 5$, $a \neq b$, $x == 3$, $(x+y) \geq 4$, $(x-y) > (x+z)$
- Λογικές (τελεστές \rightarrow λογικοί, τελεστές \rightarrow παραστάσεις με τιμές λογικές, αποτέλεσμα \rightarrow λογικού τύπου: 0, 1)
π.χ. $(x < 5) \&\& (x \geq 1)$, $(x == 0) \|\ (y == 0)$

Υπολογισμός Εκφράσεων (1)

- Προτεραιότητα (χωρισμός τελεστών σε ομάδες διαφορετικού επιπέδου προτεραιότητας)
- Προσεταιριστικότητα (καθορισμός κατεύθυνσης εφαρμογής τελεστών ίδιας προτεραιότητας)

Υπολογισμός Εκφράσεων (2)

ΤΕΛΕΣΤΕΣ	ΠΡΟΣΕΤΑΙΡΙΣΤΙΚΟΤΗΤΑ
() [] ->	Από αριστερά προς τα δεξιά
! ~ ++ -- - * & (τύπος) sizeof	Από δεξιά προς τα αριστερά
• / %	Από αριστερά προς τα δεξιά
+ -	Από αριστερά προς τα δεξιά
<< >>	Από αριστερά προς τα δεξιά
< <= > >=	Από αριστερά προς τα δεξιά
== !=	Από αριστερά προς τα δεξιά
&	Από αριστερά προς τα δεξιά
^	Από αριστερά προς τα δεξιά
	Από αριστερά προς τα δεξιά
&&	Από αριστερά προς τα δεξιά
	Από αριστερά προς τα δεξιά
?:	Από δεξιά προς τα αριστερά
= += -= *= %= &= ^= = <<= >>=	Από δεξιά προς τα αριστερά
, (κόμμα)	Από αριστερά προς δεξιά

Υπολογισμός Εκφράσεων (3)

- Εφαρμοστική σειρά (applicative order)

Υπολογισμός των τελεστών και εφαρμογή του τελεστή,
π.χ. $x+y$

- Υπολογισμός περιορισμένης έκτασης (short circuit evaluation)

Υπολογισμός όσων τελεστών απαιτούνται,

π.χ. $x \ \&\&y$, $x \ ||y$

Προτάσεις

Πρόταση = εντολή + προσδιορισμοί +;

Κατηγορίες

- Δήλωσης (π.χ. `int num;`)
- Ανάθεσης/καταχώρησης (π.χ. `num = 20;`)
- Ροής ελέγχου (π.χ. `if (a>b) a; else b;`)
- Κλήσης συνάρτησης (π.χ. `printf("hello");`)

Σύνθετη πρόταση = { <προτ1>; <προτ2>; ... <προτν>; }

π.χ. { `int num;`
`num=10;`
`printf("num=%d\n", num);`}

Σημαντικό: **όχι** ερωτηματικό μετά το άγκιστρο.

Προτάσεις Προεπεξεργαστή

- Ειδικές προτάσεις (δεν τελειώνουν σε;)
- Εκτελούν κάποια επεξεργασία του πηγαίου κώδικα πριν τη μεταγλώττιση
- Πρόταση συμπερίληψης
 - `#include <όνομα αρχείου>` ή `#include "όνομα αρχείου"`
 - π.χ. `#include <stdio.h>`, `#include <string.h>`
- Πρόταση μακρο-αντικατάστασης
 - `#define <όνομα> <κείμενο>`
 - π.χ. `#define PI 3.1415`, `#define sq(x) x*x`
(πρβλ. `sq(x+1)`)

Δήλωση Σταθερών

#define <όνομα> <τιμή>

#define TRUE 1

#define FALSE 0

#define JANUARY 1

const int TRUE=1;

const double e=2.17;

```
enum boolean {NO, YES};
```

```
enum months {JAN=1, FEB, MAR, APR, MAY,  
JUN, JUL, AUG, SEP, OCT, NOV, DEC};
```

Οι απαριθμήσεις παρέχουν ένα βολικό τρόπο για τη σύνδεση σταθερών τιμών με ονόματα. Οι μεταβλητές απαρίθμησης παρέχουν τη δυνατότητα ελέγχου και συχνά είναι καλύτερες από την #define.

Αριθμητικές Μετατροπές (1)

Έμμεσες

- Αν ένας από τους τελεστές είναι long double μετατρέπεται και ο άλλος σε long double
- Αλλιώς, αν ένας από τους τελεστές είναι double μετατρέπεται και ο άλλος σε double
- Αλλιώς αν ένας από τους τελεστές είναι float μετατρέπεται και ο άλλος σε float
- Αλλιώς ο char ή ο short μετατρέπεται σε int.

Τέλος αν ένας από τους 2 τελεστές είναι long μετατρέπεται και ο άλλος σε long

Αριθμητικές Μετατροπές (2)

Σε οποιαδήποτε παράσταση μπορούν να επιβληθούν ρητές μετατροπές τύπου με έναν μοναδικό τελεστή που λέγεται προσαρμογή (cast).

Στην κατασκευή

(όνομα τύπου) παράσταση

η παράσταση μετατρέπεται στον κατανομαζόμενο τύπο με βάση τους κανόνες μετατροπής που προαναφέραμε.

Π.χ `sqrt((double) n);`

```
#include <stdio.h>
main()
{
float test;
test=6/5;
printf("%f", test);
}
```

Άσκηση 1

Να γραφεί πρόγραμμα σε C το οποίο διαβάζει δύο ακεραίους αριθμούς από το πληκτρολόγιο και στην συνέχεια τυπώνει το άθροισμα, την διαφορά, το γινόμενο, το ακέραιο ηλίκο και το υπόλοιπο της διαίρεσης των δύο αριθμών αυτών. Παράδειγμα εκτέλεσης:

Για $x = 15$ και $y = 4$ η έξοδος είναι:

$$15 + 4 = 19$$

$$15 - 4 = 11$$

$$15 * 4 = 60$$

$$15 / 4 = 3$$

$$15 \text{ mod } 4 = 3$$

Άσκηση 2

Να γραφεί πρόγραμμα C το οποίο:

- (i) διαβάζει από το πληκτρολόγιο έναν double **d**
- (ii) αποθηκεύει τον d σε έναν float **f** με χρήση typecasting (float)
- (iii) τυπώνει τους d και f με ακρίβεια 12 δεκαδικών ψηφίων.

Άσκησης (1)

- i. Γράψτε πρόγραμμα σε C το οποίο ζητάει από τον χρήστη να δώσει τις συντεταγμένες δύο σημείων στο επίπεδο. Στην συνέχεια, υπολογίζει την ευκλείδεια απόσταση των σημείων αυτών και την εκτυπώνει. (Υπόδ: πρέπει να χρησιμοποιηθεί η συνάρτηση υπολογισμού της τετραγωνικής ρίζας `sqrt()` της `math.h`)
- ii. Να γραφεί πρόγραμμα σε C το οποίο διαβάζει από το πληκτρολόγιο έναν πραγματικό αριθμό. Στην συνέχεια, θεωρώντας ότι ο αριθμός αυτός αναπαριστά θερμοκρασία σε βαθμούς Φαρενάιτ, να γίνει μετατροπή σε κλίμακα Κελσίου και το αποτέλεσμα να τυπωθεί. Υπόδ: ο τύπος μετατροπής από Φαρενάιτ σε Κελσίου είναι $C = (5/9) * (F - 32)$
- iii. Να γραφεί πρόγραμμα σε C το οποίο διαβάζει από το πληκτρολόγιο δύο αριθμούς και τους εναλλάσσει.
- iv. Να γραφεί πρόγραμμα το οποίο διαβάζει ένα αριθμό από το πληκτρολόγιο, υπολογίζει την τρίτη δύναμή του και τον εκτυπώνει.
- v. Να γραφεί πρόγραμμα το οποίο διαβάζει ένα ακέραιο αριθμός x , και απομονώνει τα n bits του που αρχίζουν από τη θέση p .