

Γλώσσα Προγραμματισμού C++

ΦΥΛΛΑΔΙΟ ΕΡΓΑΣΤΗΡΙΟΥ 1

Τα δεδομένα

Τα δεδομένα (πληροφορίες-data) είναι απαραίτητα στοιχεία ενός προγράμματος, καθώς οι βασικές λειτουργίες ενός προγράμματος είναι η επεξεργασία αυτών των δεδομένων και η εξαγωγή αποτελεσμάτων (δηλαδή άλλα δεδομένα). Απαιτείται λοιπόν η δέσμευση κάποιων χώρων μνήμης για να αποθηκευτούν αυτά τα δεδομένα, κατά την διάρκεια της εκτέλεσης του προγράμματος. Αυτοί οι χώροι μνήμης που τους χρησιμοποιούμε για την φύλαξη δεδομένων όταν εκτελείται ένα πρόγραμμα, ονομάζονται ανάλογα με την χρήση τους σταθερές ή μεταβλητές.

Κάθε γλώσσα προγραμματισμού μας δίνει την δυνατότητα με διάφορους τρόπους να δηλώσουμε κάποιους τύπους μεταβλητών ή σταθερών. Με τη δήλωση των μεταβλητών/σταθερών δεσμεύεται και η αντίστοιχη μνήμη την οποία μπορούμε να την χρησιμοποιήσουμε για την αποθήκευση δεδομένων.

Πρώτα δεσμεύουμε τη μνήμη και μετά την χρησιμοποιούμε για την αποθήκευση δεδομένων.

Οι σταθερές

Αρκετά προγράμματα απαιτούν ορισμένα δεδομένα τα οποία δεν αλλάζουν (ή δεν πρέπει ν' αλλαχθούν) ποτέ κατά την διάρκεια της εκτέλεσης του. Αυτά τα δεδομένα συνήθως καθορίζονται μια φορά και χρησιμοποιούνται όσο συχνά επιθυμούμε, κατά την διάρκεια λειτουργίας του προγράμματος. Για παράδειγμα, όταν γράφετε ένα πρόγραμμα, μπορείτε να καθορίσετε ότι η τιμή του π είναι 3.14159 και να χρησιμοποιείτε αυτή την τιμή όταν την χρειάζεστε, ξέροντας ότι είναι διαθέσιμη και σωστή. Οι σταθερές ορίζονται με εντολές του προπεξεργαστή με την οδηγία #define και συνήθως χρησιμοποιούμε κεφαλαία γράμματα για να τις περιγράψουμε:

```
# define PI 3.14159
```

Προσέξτε! αυτή η εντολή δεν τελειώνει με το ερωτηματικό ";". Επίσης υπάρχει κενό μεταξύ του PI και του 3.14159.

Οι μεταβλητές

Αντίθετα με τις σταθερές, οι τιμές των μεταβλητών είναι δυνατόν να αλλάξουν κατά την διάρκεια της εκτέλεσης ενός προγράμματος. Οι γλώσσες προγραμματισμού μας δίνουν την δυνατότητα να καθορίσουμε μεταβλητές και στην συνέχεια να τους δώσουμε όποια τιμή επιθυμούμε, που έχει βέβαια σχέση με το πρόγραμμα. Η δήλωση των μεταβλητών γίνεται συνήθως στην αρχή του προγράμματος.

Οι τύποι δεδομένων στην C

Ένας τύπος στην C, αλλά και σε πολλές άλλες δημοφιλείς γλώσσες προγραμματισμού, καθορίζει τον τρόπο χειρισμού κάποιας μεταβλητής. Συγκεκριμένα προσδιορίζει το είδος των τιμών που μπορεί να αποθηκεύσει, και καθορίζει ένα σύνολο λειτουργιών που είναι δυνατόν να εφαρμοσθούν πάνω στις μεταβλητές αυτού του τύπου. Η C έχει κάποιους

βασικούς τύπους δεδομένων, οι οποίοι έχουν προκαθοριστεί και μπορούν να χρησιμοποιηθούν οπουδήποτε σε ένα πρόγραμμα:

A. Ο τύπος **char** χρησιμοποιείται για να αποθηκεύσει χαρακτήρες.

Παραδείγματα δήλωσης (και συνεπώς δέσμευσης μνήμης):

```
char ch1; // Δήλωση
unsigned char ch2;
char ch3='A'; //Δήλωση & αρχική τιμή
```

B. Ο τύπος **int** για τον χειρισμό ακεραίων αριθμών με πρόσημο (... , -2, -1, 0, 1, 2, 3, ...) ή χωρίς πρόσημο (0, 1, 2, 3,...)

Παραδείγματα δήλωσης:

```
int alpha;
short metritis=0, beta;
unsigned int arithmos_foititon;
```

Παράδειγμα χρήσης:

```
alpha=5;
printf("the integer is:%d", alpha); //εμφάνιση στη οθόνη
```

Γ. Ο τύπος **float** που χρησιμοποιείται για τον χειρισμό πραγματικών αριθμών (περιλαμβάνουν υποδιαστολή: 34.5, 23.0, 0.987).

Παραδείγματα δήλωσης:

```
float mesos_oros;
float tasi = 5.3;
```

Παράδειγμα χρήσης:

```
alpha=12.3;
printf("the float is:%f", alpha); //εμφάνιση στη οθόνη
```

Περισσότερα για τους τύπους δεδομένων στη θεωρία του μαθήματος

Ονομασίες μεταβλητών

Οι ονομασίες των μεταβλητών πρέπει να αρχίζουν με γράμμα ή με κάτω παύλα. Στην περιγραφή της ονομασίας μπορείτε να χρησιμοποιήσετε και αριθμούς, αλλά δεν επιτρέπονται άλλοι ειδικοί χαρακτήρες. Τα πεζά και κεφαλαία γράμματα παίζουν ρόλο (είναι διαφορετικά!). Ένας συγκεκριμένος αριθμός χαρακτήρων είναι σημαντικοί για τον compiler και αυτός ο αριθμός εξαρτάται από τον συγκεκριμένο compiler που χρησιμοποιείτε (πχ για την turbo C τα πρώτα 32 ψηφία είναι σημαντικά).

Οι τελεστές στη C

Η C όπως και κάθε γλώσσα προγραμματισμού χρησιμοποιεί διάφορους τελεστές για να εκτελέσει πράξεις, να αναπαραστήσει αριθμητικές και λογικές εκφράσεις και να δημιουργήσει δομές ελέγχου. Έτσι λοιπόν στην C ανά κατηγορία έχουμε τους τελεστές που ακολουθούν.

Ο τελεστής καταχώρησης:

Για το τελεστή καταχώρησης (ή καλύτερα ανάθεσης σε μεταβλητή -assignment) χρησιμοποιείται το σύμβολο του ίσον (=) που έχει βέβαια διαφορετική έννοια από ότι στα μαθηματικά. Η εντολή:

```
var2 = 52;
```

σημαίνει ότι δίνω την τιμή 52 στην μεταβλητή var2. Αν δώσω παρακάτω την εντολή:

```
var2 = var2 +1;
```

που ενώ μαθηματικά δεν "στέκει", "αυξάνω" κατά μία μονάδα την τιμή που ήδη υπάρχει καταχωρημένη στην μεταβλητή var2, δηλαδή η νέα τιμή της var2 μετά την εκτέλεση της συγκεκριμένης εντολής θα είναι 53.

Φυσικά δεν μπορείτε να καταχωρήσετε τιμή σε μία σταθερά

Η C υποστηρίζει ακόμα πολλαπλή καταχώρηση. Δηλαδή ισχύει το:

```
x=y=z=54;
```

Εδώ οι καταχωρήσεις γίνονται από δεξιά προς τα αριστερά, δηλαδή πρώτα παίρνει την τιμή 54 η μεταβλητή z, μετά η μεταβλητή y και τέλος η x.

Οι αριθμητικοί τελεστές:

Πρόκειται για τους:

+ Πρόσθεση - Αφαίρεση * Πολλαπλασιασμός / Διαίρεση

% Ακέραιο υπόλοιπο (modulo) (πχ., 25 % 4 μας δίνει 1. Εφαρμόζεται μόνο σε ακεραίους)

Για κάθε τελεστή απαιτούνται 2 όροι (εκτός από τον τελεστή "-", που είναι δυνατόν να χρησιμοποιηθεί και σαν πρόσημο), οι οποίοι μπορεί να είναι σταθερές ή μεταβλητές.

```
Ilikia = etos1 - etos2;
```

```
printf("%d", 95-55);
```

```
Celsius = (-25);
```

Τα κενά διαστήματα δεν παίζουν ρόλο και μπορείτε να χρησιμοποιήσετε παρενθέσεις για μαθηματικές παραστάσεις:

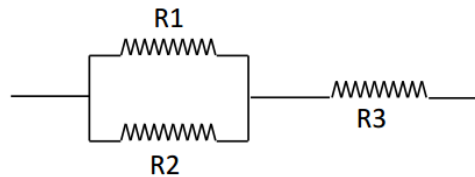
```
result = -(b+2)*a + (c +3*(a-b));
```

Η C ακολουθεί την αλγεβρική προτεραιότητα στους αριθμητικούς τελεστές. Δηλαδή μεγαλύτερη προτεραιότητα έχουν οι παρενθέσεις, μετά το πρόσημο (δηλαδή το μείον), ακολουθούν ο πολλαπλασιασμός, η διαίρεση και το modulo και τέλος οι τελεστές πρόσθεσης και αφαίρεσης. Η εντολή καταχώρησης έχει μικρότερη προτεραιότητα από όλες

τις πράξεις. Η φορά των πράξεων είναι από αριστερά προς δεξιά για όλους τους αριθμητικούς τελεστές, εκτός αν χρησιμοποιήσετε το μείον σαν πρόσημο, οπότε η φορά είναι από δεξιά προς αριστερά!

Παράδειγμα με υπολογισμό στην C

Να γραφτεί ένα πρόγραμμα με το οποίο να δίνετε 3 τιμές σε μεταβλητές που αφορούν στις 3 αντιστάσεις R1, R2 και R3 του κάτωθι κυκλώματος (σε ohm) και στην συνέχεια αυτό να υπολογίζει και να εκτυπώνει την ολική αντίσταση Rtotal.



Ενδεικτική λύση:

```
#include <stdio.h>

main() {

float R1, R2, R3, Rtotal; // δηλώνουμε τις μεταβλητές -δέσμευση μνήμης
R1=1; // δίνουμε τιμή στη 1η αντίσταση
R2=1; // δίνουμε τιμή στη 2η αντίσταση
R3=1; // δίνουμε τιμή στη 3η αντίσταση
Rtotal=(R1*R2)/(R1+R2) + R3; // υπολογίζουμε την ολική αντίσταση
printf("Oliki antistasi: %f \n", Rtotal); //εμφανίζουμε το αποτέλεσμα
}
```

Γράψτε το παραπάνω πρόγραμμα και δοκιμάστε το με διάφορες τιμές των R1, R2, R3.

Ελέγξτε τα αποτελέσματα.

Ασκήσεις για το εργαστήριο

Γράψτε και τρέξτε τα παρακάτω προγράμματα.

- **TRIGONO.CPP** Γράψτε ένα πρόγραμμα σε C το οποίο θα δίνει τιμές σε 2 μεταβλητές σε μια πλευρά ενός τριγώνου (a) και στο ύψος (h) που αντιστοιχεί σε αυτή, θα υπολογίζει το εμβαδόν (μεταβλητή E) του τριγώνου και θα το εμφανίζει στην οθόνη.
- **CELSIUS.CPP** Να γραφτεί ένα πρόγραμμα το οποίο θα δίνει σε μια μεταβλητή μια θερμοκρασία σε κλίμακα Φαρενάιτ (F) και θα βρίσκει την αντίστοιχη θερμοκρασία σε κλίμακα Κελσίου (C). Η σχέση μεταξύ κλίμακας Κελσίου (C) και Φαρενάιτ (F) είναι

$$C = 5.0/9.0 * (F-32)$$

Προσοχή στη διαίρεση! Αν χρησιμοποιήσουμε ακέραιες σταθερές (δηλαδή γράψουμε 5/9 αντί του σωστού 5.0/9.0) τότε το αποτέλεσμα θα είναι επίσης ακέραιο (δηλαδή μηδέν).

- **VATHMOS.CPP** Να γραφτεί ένα πρόγραμμα που θα υπολογίζει το τελικό βαθμό (tb) ενός φοιτητή, ο οποίος θα προκύπτει από τους βαθμούς τριών (3) εργαστηριακών ασκήσεων (b1, b2, b3) όταν η 1η άσκηση μετρά 15%, η δεύτερη 35% και η τρίτη 50%.
- **EMVADON.CPP** Να γραφτεί ένα πρόγραμμα, στο οποίο θα δίνουμε μια τιμή στη πλευρά του τετραγώνου σε μέτρα στο σχέδιο και αυτό θα υπολογίζει και θα τυπώνει το εμβαδόν της σκιασμένης επιφάνειας.

