

6

Κεφάλαιο

Ο τύπος δεδομένων float



Ο τύπος δεδομένων float

Το κεφάλαιο αυτό εισάγει και αναλύει τον τρόπο χρήσης των αριθμών κινητής υποδιαστολής (floating point) στη γλώσσα C. Εξετάζεται ο τρόπος με τον οποίο η C χειρίζεται τις σταθερές και τις μεταβλητές τύπου `float`. Επίσης γίνεται αναφορά σε συναρτήσεις που χρησιμοποιούνται για το χειρισμό δεδομένων κινητής υποδιαστολής (δεκαδικών αριθμών).

Σταθερές τύπου float

Μια σταθερά τύπου `float` δεν είναι τίποτε άλλο από ένας δεκαδικός αριθμός. Για να θεωρηθεί τύπου `float`, ένας αριθμός πρέπει απαραίτητα να περιέχει το χαρακτήρα της υποδιαστολής (`.`), για παράδειγμα

123.45

123.0

123.

0.1

είναι παραδείγματα σταθερών τύπου `float`.

Μεταβλητές τύπου float

Υπάρχουν δύο τύποι μεταβλητών κινητής υποδιαστολής (floating point), ο τύπος `float` και ο τύπος `double`. Οι μεταβλητές τύπου `float` καταλαμβάνουν 4 bytes, ενώ οι τύπου `double` 8. Οι μεταβλητές τύπου `double` έχουν μεγαλύτερη ακρίβεια στα δεκαδικά ψηφία.

Μια μεταβλητή τύπου `float` δηλώνεται όπως παρακάτω:

`float όνομα.μεταβλητής;`

`double όνομα.μεταβλητής;`

`long float όνομα.μεταβλητής;`

όπου το `long float` είναι ισοδύναμο με το `double`.

Χρήση δεδομένων τύπου κινητής υποδιαστολής

Οι διαθέσιμοι τελεστές για πράξεις μεταξύ δεδομένων τύπου κινητής υποδιαστολής (floating point) είναι πολύ περιορισμένοι σε σχέση με τούς τελεστές που εφαρμόζονται σε δεδομένα τύπου `int` και `char`. Οι τελεστές που εφαρμόζονται σε δεδομένα τύπου `float` (ή `double`) αναφέρονται στο διπλανό πίνακα.

Τελεστής	Λειτουργία
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση

 Παρατηρούμε ότι οι αριθμητικοί τελεστές `%`, `++` και `--` δεν εφαρμόζονται σε δεδομένα τύπου `float`. Μερικοί μεταγλωττιστές της C (όπως ο DEV C/C++) επιτρέπουν και σε μεταβλητές τύπου `float` τη χρήση των τελεστών `++` και `--`.

Η χρήση της `printf()` με δεδομένα κινητής υποδιαστολής

Ας θεωρήσουμε το επόμενο πρόγραμμα του οποίου το αποτέλεσμα φαίνεται στο διπλανό πλαίσιο:

```
main()
{
    float a,b;
    a=21.234;
    b=5467.1;
    printf("%f\n",a);
    printf("%f\n",b);
}
```

21.234
5467.1

Παρατηρούμε ότι αν και τα αποτελέσματα εμφανίζονται σωστά, μπορεί να προτιμούσαμε τις χιλιάδες κάτω από τις χιλιάδες, τις εκατοντάδες κάτω από τις εκατοντάδες κ.ο.κ.

Για να επιτύχουμε μορφοποιημένη (φορμαρισμένη) έξοδο αριθμών, η ακολουθία `%f` στο αλφαριθμητικό ελέγχου της `printf()` μπορεί να διαμορφωθεί ως εξής:

`%n.mf`

όπου **n** ο συνολικός αριθμός των θέσεων που θα καταλάβει ο αριθμός (μαζί με την υποδιαστολή) και **m** ο αριθμός των θέσεων μετά από την υποδιαστολή. Έτσι, **%8.3f** σημαίνει ότι ο αριθμός θα εμφανιστεί σε 8 θέσεις από τις οποίες οι 3 θα είναι για το δεκαδικό μέρος.

Για μορφοποιημένη έξοδο, το προηγούμενο πρόγραμμα θα μπορούσε να διατυπωθεί ως εξής:


```
main()
{
    float a,b;
    a=1.234;
    b=5467.1;
    printf("%8.3f\n",a);
    printf("%8.3f\n",b);
}
```

21.234
5467.100

Χρήση της συνάρτησης scanf() με δεδομένα τύπου float

Η **scanf()** χρησιμοποιείται για την είσοδο δεκαδικών αριθμών με το πληκτρολόγιο. Για παράδειγμα, το επόμενο πρόγραμμα ζητάει και εμφανίζει δύο δεκαδικούς.

```
main()
{
    float a;
    double b;
    printf("Δώσε δεκαδικό:");
    scanf("%f",&a);
    printf("Δώσε δεκαδικό:");
    scanf("%lf",&b);
    printf("Έδωσες %f και %f\n",a,b);
}
```

 Παρατηρούμε ότι όταν πρόκειται να διαβαστεί αριθμός διπλής ακρίβειας (τύπου **double**), η ακολουθία **%lf** αντικαθιστά την **%f** στο αλφαριθμητικό ελέγχου της **scanf()**.

Η συνάρτηση pow()

Η C δεν διαθέτει τελεστή ύψωσης σε δύναμη όπως άλλες γλώσσες προγραμματισμού. Η ύψωση σε δύναμη γίνεται μέσω της συνάρτησης βιβλιοθήκης `pow()`.

Η συνάρτηση `pow()` έχει δύο παραμέτρους (τη βάση και τον εκθέτη) και επιστρέφει τιμή τύπου `double`.

```
#include <math.h>
main()
{
    int a,b;
    double c;
    a=4;
    b=5;
    //κλήση της pow
    c=pow(a,b);
    printf("%d εις την %d = %f\n",a,b,c);
}
```

Το παραπάνω πρόγραμμα υπολογίζει και εμφανίζει στην οθόνη την τιμή της παράστασης 4^5 .

Παρατηρούμε ότι, μολονότι το `a` και `b` είναι ακέραιοι, το αποτέλεσμα της `pow()` είναι αριθμός τύπου `double` (από τον ορισμό της συνάρτησης) και θα πρέπει να καταχωριστεί σε ανάλογη μεταβλητή (`c`).

Η συνάρτηση `pow()` ορίζεται στο αρχείο κεφαλίδας `math.h`, το οποίο θα πρέπει να συμπεριληφθεί στο πρόγραμμα με την οδηγία `#include`.

Η λογική των *αρχείων κεφαλίδας* (header files), καθώς και η φιλοσοφία χρήσης των συναρτήσεων βιβλιοθήκης, αναλύονται στο κεφάλαιο 9.

Παραδείγματα

Π.1 Το επόμενο πρόγραμμα ζητάει δύο δεκαδικούς διπλής ακρίβειας. Υπολογίζει και εμφανίζει το άθροισμά τους, και μετά εξάγει τους αριθμούς στην οθόνη, σε 10 θέσεις με 4 δεκαδικά ψηφία.

```
main()
{
    double a,b,c;
    scanf("%lf %lf",&a,&b);
    c=a+b;
    printf("%10.4f\n",a);
    printf("%10.4f\n",b);
    printf("=====");
    printf("%10.4f\n",c);
}
```

```
15.2143
345.1234
=====
360.3377
```

Π.2 Το επόμενο πρόγραμμα ζητάει ένα δεκαδικό αριθμό. Υπολογίζει και εμφανίζει χωριστά το ακέραιο και το δεκαδικό του τμήμα.

```
main()
{
    double a,d;
    int i;
    scanf("%lf",&a);
    i=a;
    d=a-i;
    printf("%f\n",d);
    printf("%d\n",i);
}
```

Δεδομένου ότι η μεταβλητή *i* είναι τύπου *int*, με αυτή την πρόταση καταχωρίζεται στην *i* το ακέραιο μέρος της *a*.

Αν από την *a* αφαιρέσουμε το ακέραιο τμήμα της (την *i*), το αποτέλεσμα θα είναι το δεκαδικό τμήμα του αριθμού.

Ανασκόπηση Κεφαλαίου 6

- Ο τύπος δεδομένων `float` αφορά πραγματικούς αριθμούς (με δεκαδικά). Οι μεταβλητές και οι σταθερές τύπου `float` δεσμεύουν 4 byte.
- Ο τύπος δεδομένων `double` χρησιμοποιείται για περισσότερη ακρίβεια στα δεκαδικά ψηφία και δεσμεύει 8 byte.
- Σε μεταβλητές τύπου `float` και `double` δεν μπορούν να χρησιμοποιηθούν οι τελεστές `++`, `--`, και `%` που χρησιμοποιούνται σε δεδομένα τύπου `int` και `char`.
- Η `printf()` χρησιμοποιείται και για τη μορφοποιημένη εμφάνιση ενός δεκαδικού αριθμού. Για παράδειγμα, η `printf("%8.3f\n", a)` εμφανίζει ένα δεκαδικό (τα περιεχόμενα της μεταβλητής `a`), σε χώρο οκτώ θέσεων με τρία δεκαδικά.
- Για την ύψωση σε δύναμη χρησιμοποιείται η συνάρτηση `pow()`.

Ασκήσεις Κεφαλαίου 6

- 6.1 Να γραφεί πρόγραμμα το οποίο να ζητάει δύο δεκαδικούς αριθμούς, να υπολογίζει το μέσο όρο τους, και να τον εμφανίζει με δύο δεκαδικά ψηφία καταλαμβάνοντας συνολικά επτά θέσεις στην οθόνη. ★
- 6.2 Να γραφεί πρόγραμμα το οποίο να ζητάει την ακτίνα ενός κύκλου. Να υπολογίζει και να εμφανίζει το εμβαδόν του με διπλή ακρίβεια. Να χρησιμοποιηθεί η συνάρτηση `pow()` για την ύψωση σε δύναμη. Η τιμή του π 3,141592 να δηλωθεί ως σταθερά με την οδηγία `#define`. ★★
- 6.3 Τι αποτέλεσμα θα έχει το επόμενο πρόγραμμα και γιατί; ★

```
main()
{
    float d;
    int a,b;
    a=5;
```



```
b=6;
d=(a+b)/2;
printf("%f\n",d);
}
```

6.4 Ποια από τα επόμενα αληθεύουν: ★

- ☐ Οι τελεστές ++ και -- δεν μπορούν να εφαρμοστούν σε μεταβλητές τύπου `float`.
- ☐ Οι μεταβλητές τύπου `double` αποθηκεύουν απεριόριστο αριθμό δεκαδικών ψηφίων.
- ☐ Με τη συνάρτηση `printf()` δεν μπορούμε να καθορίσουμε τον ακριβή αριθμό των δεκαδικών ψηφίων που θα εμφανίζονται στην οθόνη.
- ☐ Η παράσταση `1+1.0` έχει αποτέλεσμα τύπου `float`.
- ☐ Η C έχει τελεστή για ύψωση σε δύναμη.

6.5 Να γραφεί πρόγραμμα το οποίο να υπολογίζει το υπόλοιπο της ακέραιας διαίρεσης ενός δεκαδικού αριθμού με έναν ακέραιο. Το πρόγραμμα θα ζητάει δύο αριθμούς, ένα δεκαδικό και έναν ακέραιο, και θα εμφανίζει το υπόλοιπο. Αν υποθέσουμε π.χ. ότι δίνουμε τους αριθμούς 5.14 και 2 το αποτέλεσμα είναι το 1.14. ★★ ★