

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ



ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΙΑΔΙΚΑΣΤΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2009-2010)  
ΥΠΕΥΘΥΝΟΙ ΔΙΔΑΣΚΟΝΤΕΣ ΕΡΓΑΣΤΗΡΙΟΥ: Α. ΦΩΚΑ, Κ. ΣΤΑΜΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

1<sup>ο</sup> ΣΕΤ ΑΣΚΗΣΕΩΝ

Οι ασκήσεις αυτού του φυλλαδίου καλύπτουν τα παρακάτω θέματα και δίνονται ενδεικτικά οι αντίστοιχες ενότητες στο βιβλίο The GNU C Programming Tutorial που μπορείτε συμβουλευτείτε (<http://crasseux.com/books/ctutorial/>) .

- Τύποι δεδομένων, δήλωση μεταβλητών, αρχικοποίηση μεταβλητών, Μετατροπή τύπων (κεφάλαιο Variables and Declarations)
- Τελεστές, προτεραιότητα τελεστών, λογικοί τελεστές, Εκφράσεις (κεφάλαιο Expressions and Operators)
- printf () και scanf() ([http://www.cs.utah.edu/~phister/K\\_n\\_R/appb.html](http://www.cs.utah.edu/~phister/K_n_R/appb.html))

### Άσκηση 1η

Σημειώστε ποιες από τις παρακάτω εκφράσεις είναι σωστές και ποιες λάθος;

- Η διαίρεση  $12/5$  έχει σαν αποτέλεσμα 2.
- Η πράξη `int x=9.9;` βάζει στον ακέραιο x τον αριθμό 10.
- Η ανάθεση `x=60+(5!=7)` έχει αποτέλεσμα 60.
- Αν αρχικά `x=13 % 4` και `y=100` τότε μετά από την ανάθεση `x=y=(x+2)` θα ισχύει ότι `x=3` και `y=3`.
- Η έκφραση `(2>5) && (x=2)` τοποθετεί το 2 στην μεταβλητή x.
- Η έκφραση `(3<1) || (x=2)` τοποθετεί το 2 στην μεταβλητή x.
- Έστω ότι `x= (16/5==16%5)`. Τότε η `printf("%d ",x)` θα εκτυπώσει 0.
- Έστω ότι `x=5+031`. Τότε η `printf("%d ",x)` θα εκτυπώσει 30.
- Έστω ότι `x='\\103'`. Τότε η `printf("%c",x)` θα εκτυπώσει το γράμμα C.
- Η ανάθεση `a=a++`; είναι συντακτικά ορθή.

### Άσκηση 2η

Να γραφεί πρόγραμμα που να εμφανίζει στην οθόνη τις παρακάτω εκφράσεις:

- `printf("That's all folks!!!\\n");`
- `'c'`
- `/?\\`

**Υπόδειξη:** Ελέγξτε στη παρακάτω ιστοσελίδα πώς εμφανίζονται οι ειδικοί χαρακτήρες. <http://crasseux.com/books/ctutorial/Special-characters.html>

**Άσκηση 3η**

Έχουμε το παρακάτω τμήμα κώδικα. Βρείτε τι θα εκτυπωθεί πριν τρέξετε τον κώδικα.

```
int z;
int x=1;
int y=3;
x=--y;
printf("%d %d\n", x,y);
z=(x++)-(--y);
printf("%d %d %d\n", x,y,z);
y=(z--)+2;
printf("%d %d\n", y,z);
x= 12 % 5 / 1.5;
y= 101 + (int)(1.5);
z= 'd' - 3;
printf("%d %d %c\n", x,y,z);
```

**Άσκηση 4η**

Λογικές Εκφράσεις: Στην λογική πρόταση A && B, αν το A είναι ψευδές τότε όλη η πρόταση είναι ψευδής ανεξάρτητα από την B. Αντίστοιχα, στην πρόταση A || B, αν το A είναι αληθές τότε η πρόταση είναι αληθής ανεξάρτητα από το B. Γράψτε το παρακάτω πρόγραμμα:

```
main(void)
{
    int i, j;
    int res;
    scanf("%d", &i);
    scanf("%d", &j);
    res = (++i == 3) && (++j == 4);
    printf("%d\n", i);
    printf("%d\n", j);
    printf("%d\n", res);
}
```

- Αν δώσουμε στα i και j αρχικά τις τιμές 4 και 4 παρατηρήστε την έξοδο του προγράμματος. Κάντε το ίδιο για τιμές 3 και 10. Εξηγήστε αν η συμπεριφορά του προγράμματος σας φαίνεται λογική ή παρατηρείτε κάτι παράξενο και που οφείλετε αυτό.
- Αλλάξτε την εντολή `res = (++i == 3) || (++j == 4);` σε `res = (i++ == 3) || (j++ == 4);` Τι τιμές πρέπει να δώσετε για να έχει η μεταβλητή res τιμή 1; Βάλτε τις τιμές που δοκιμάσατε προηγουμένως και δείτε αν και πως αλλάζει η συμπεριφορά του προγράμματός σας.
- Αλλάξτε την εντολή `res = (++i == 3) || (++j == 4);` σε `res = (i++ == 3) && (j++ == 4);` Δώστε τιμές 3 και 4. Στη συνέχεια δώστε τις τιμές 10 και 4. Εξηγήστε τη συμπεριφορά του προγράμματος.
- Αλλάξτε την εντολή `res = (++i == 3) || (++j == 4);` σε `res = (++i == 3) && (++j == 4);` Τι τιμές πρέπει να δώσετε για να έχει η μεταβλητή res τιμή 1; Βάλτε τις τιμές που δοκιμάσατε προηγουμένως και δείτε αν και πως αλλάζει η συμπεριφορά του προγράμματός σας.

**Άσκηση 5η**

Τα προθέματα Kilo, Mega, Giga κλπ. σημαίνουν  $1000$ ,  $1000^2$ ,  $1000^3$  κλπ. Στην πληροφορική χρησιμοποιούνται τα πολλαπλάσια  $1024$ ,  $1024^2$ ,  $1024^3$ , για τα οποία υπάρχουν τα προθέματα Kibi, Mebi, Gibi κλπ. Επομένως ένα Kibibyte είναι  $1024$  bytes, ένα Mebibyte είναι  $1024$  Kibibytes και ένα Gibibyte είναι  $1024$  Mebibytes. (Αντίστοιχα, ένα Kilobyte είναι  $1000$  bytes, ένα Megabyte είναι  $1000$  Kilobytes και ένα Gigabyte είναι  $1000$  Megabytes.)

Να γραφεί πρόγραμμα σε C το οποίο να διαβάζει ένα μεγάλο ακέραιο αριθμό από bytes και να τον αναλύει σε Gibibytes, Mebibytes και Kibibytes. Πχ.

Dose bytes: 1234567890

Ta 1234567890 bytes analyontai ws eksis:

1 Gibibytes kai 153 Mibibytes kai 384 Kibibytes kai 722 bytes

Τρέξτε το πρόγραμμα σας για τις τιμές  $40\ 000$ ,  $2\ 000\ 000\ 000$ ,  $40\ 000\ 000\ 000$  και καταγράψτε τα αποτελέσματα.

**Άσκηση 6η**

Γράψτε ένα πρόγραμμα το οποίο να διαβάζει από την είσοδό του 2 χρονικούς προσδιορισμούς (έστω  $T1$  και  $T2$ ), με την μορφή HH:MM:SS όπου HH είναι ένας αριθμός ωρών, MM ένας αριθμός λεπτών και SS ένας αριθμός δευτερολέπτων. Στη συνέχεια το πρόγραμμα πρέπει να ελέγχει εάν η χρονική στιγμή  $T2$  έπεται της  $T1$ , και να εκτυπώνει το χρονικό διάστημα μεταξύ των στιγμών  $T1$  και  $T2$ , στην μορφή HH:MM:SS.

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

