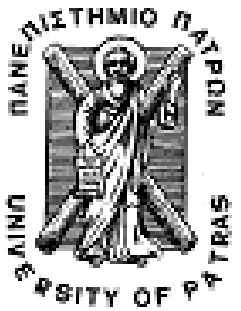
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ



ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΙΑΔΙΚΑΣΤΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2009-2010)

ΥΠΕΥΘΥΝΟΙ ΔΙΔΑΣΚΟΝΤΕΣ ΕΡΓΑΣΤΗΡΙΟΥ: Α. ΦΩΚΑ, Κ. ΣΤΑΜΟΣ

1° ΣΕΤ ΑΣΚΗΣΕΩΝ

(Ενδεικτικές Λύσεις)

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)

Άσκηση 1η

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

a. Η διαίρεση 12/5 έχει σαν αποτέλεσμα 2.
b. Η πράξη int x=9.9; βάζει στον ακέραιο x τον αριθμό 10.
c. Η ανάθεση x=60+(5!=7) έχει αποτέλεσμα 60.
d. 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.
e. Η έκφραση (2>5) && (x=2) τοποθετεί το 2 στην μεταβλητή x.
f. Η έκφραση (3<1) || (x=2) τοποθετεί το 2 στην μεταβλητή x.
g. Έστω ότι x= (16/5==16%5). Τότε η printf("%d ",x) θα εκτυπώσει 0.
h. Έστω ότι x=5+031. Τότε η printf("%d ",x) θα εκτυπώσει 30.
i. Έστω ότι x='\103'. Τότε η printf("% c",x) θα εκτυπώσει το γράμμα C.
j. Η ανάθεση a=a++; είναι συντακτικά ορθή.

```
a. Σ
b. Λ
c. Λ
d. Σ
e. Λ
f. Σ
g. Σ
h. Σ
i. Σ
j. Σ
```

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

```
#include <stdio.h>
main()
{
 printf("printf(\"That\'s all folks!!!\\n\");\n");
 printf("\'c\'\n");
 printf("\/\?\\\n");
}
```

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);
```

```
2 2
3 1 1
3 0
1 102 a
```

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.

```
Για i=4 και j=4 η έξοδος είναι 5 5 0. Η res έχει τιμή 0 (δηλ.
FALSE) γιατί πρώτα αυξάνεται η τιμή της i και j και μετά
γίνεται ο έλεγχος ισότητας == 4.

Για i=3 και j=10 η έξοδος είναι 4 11 0.
```

- 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.

```
Τώρα πρώτα γίνεται ο έλεγχος ισότητας και μετά αυξάνεται η τιμή
των i και j. Συνεπώς, για να έχει η μεταβλητή res τιμη 1 πρέπει
να δώσουμε για το i τιμή 3 ή για το j τιμή 4.

Για τις προηγούμενες τιμές έχουμε έξοδο 5 5 1 και 4 10 1. Η res
έχει τιμή 1 γιατί πρώτα γίνεται ο έλεγχος ισότητας i++ == 3 και
μετά αυξάνεται η τιμή της i. Σε αυτή την περίπτωση η τιμή της j
παραμένει 10 και δεν έχει αυξηθεί κατά 1 όπως θα περιμέναμε.
Αυτό γίνεται διότι στην εντολή res = (i++ == 3) || (j++ == 4);
ο έλεγχος της πρώτης συνθήκης μας δίνει TRUE και συνεπώς η τιμή
όλης της παράστασης θα είναι TRUE ανεξαρτήτως της τιμής της
δεύτερης συνθήκης δεδομένου ότι συνδέονται με OR. Γι' αυτό το
λόγο η C δεν προχωράει στον υπολογισμό της τιμής της δεύτερης
συνθήκης, και έτσι δεν εκτελείται το j++.
```

- 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.

```
Για i=3 και j=4 η έξοδος είναι 4 5 1. Η res έχει τιμή 1 (δηλ.
TRUE) γιατί πρώτα γίνονται οι έλεγχοι ισότητας και μετά
αυξάνονται οι τιμές.

Για i=10 και j=4 η έξοδος είναι 11 4 0. Η τιμή της j δεν έχει
αυξηθεί διότι στον υπολογισμό της τιμής της res η πρώτη συνθήκη
ήταν FALSE και δεδομένου ότι οι δύο συνθήκες συνδέονται με AND
δεν γίνεται ο υπολογισμός της τιμής της δεύτερης συνθήκης, αφού
η res θα είναι FALSE ανεξαρτήτως της τιμής της, και έτσι δεν
εκτελείται το j++.
```

- 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.

Τώρα πρώτα αυξάνεται η τιμή των i και j και μετά γίνεται ο έλεγχος ισότητας. Συνεπώς, για να έχει η μεταβλητή res τιμη 1 πρέπει να δώσουμε για το i τιμή 2 και για το j τιμή 3.

Για τις προηγούμενες τιμές έχουμε έξοδο 4 4 0 και 11 4 0. Εδώ βλέπουμε ότι η j στην πρώτη περίπτωση δεν έχει γίνει 5 διότι αντίθετα με την προηγούμενη περίπτωση δεν χρειάστηκε να υπολογιστεί η τιμή και των δύο συνθηκών και έτσι δεν εκτελέστηκε η j++.

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.


```
#include <stdio.h>

int main()
{
  long mebibytes=0;
  long kibibytes=0;
  long gibibytes=0;
  long bytes=0;
  long initial_bytes=0;

  printf("Dose bytes: ");
  scanf("%ld", &initial_bytes);

  /*Calculate Gibibytes*/
  kibibytes = initial_bytes / 1024;
  mebibytes = kibibytes / 1024;
  gibibytes = mebibytes / 1024;

  /*Now calcuate the rest*/
  mebibytes = mebibytes - 1024*gibibytes;
  kibibytes = kibibytes - 1024*(mebibytes + 1024*gibibytes);
  bytes       =       initial_bytes       -       1024*(kibibytes       +
1024*(mebibytes+1024*gibibytes));

  printf("Ta %ld bytes analyontai ws eksis:\n",initial_bytes);
```

```
  printf("%d Gibibytes kai %d Mebibytes kai %d Kibibytes kai %d
bytes\n",gibibytes,mebibytes,kibibytes,bytes);
}
```

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.

```
#include <stdio.h>

int main()
{
      int h1, m1, s1, h2, m2, s2, dh, dm, ds;
      printf("Enter time T1 (hh:mm:ss) :: ");
      scanf("%d:%d:%d", &h1, &m1, &s1);
      printf("Enter time T2 (hh:mm:ss) :: ");
      scanf("%d:%d:%d", &h2, &m2, &s2);

      ds = s2 - s1;
      dm = m2 - m1;
      dh = h2 - h1;

      if(ds < 0){
            ds += 60; dm -= 1;
      }

      if(dm < 0){
            dm += 60; dh -= 1;
      }

      if(dh < 0){
            printf("\t T2 is before T1 !! \n");
      }
      else
      {
            printf("Interval between [%d:%d:%d] and [%d:%d:%d]:
                  %d:%d:%d\n",h2, m2, s2, h1, m1, s1, dh, dm, ds);
      }
}
```

Οι