Α' Εξάμηνο
ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΟΜΗΜΕΝΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Εργαστήριο – 1η εβδομάδα.
Κοζάνη, 5 Οκτωβρίου 2007.

Βασικά στοιχεία γραφής προγραμμάτων με τη C:
α) Case Sensitive!!!
β) Δυνατότητα εισαγωγής σχολίων με /* και */
γ) Ελεύθερο format αρκεί να μη σπάζουν τα tokens. Προτιμούμε όμως κάποιο συνεπές και ευανάγνωστο στυλ.
δ) Παρομοίως τα σχόλια μπαίνουν παντού αρκεί να μη σπάνε τα tokens, να μην είναι ένθετα και να ανοίγουν και κλείνουν σωστά με /* και */ αντιστοίχως.
ε) Οι εντολές τελειώνουν με ;
στ) Οι διάφορες συναρτήσεις αλλά και τα τμήματα κώδικα (βασικές δομές) στο εσωτερικό τους αρχίζουν με { και τελειώνουν με }
ζ) Πάντα υπάρχει μία συνάρτηση που παίζει το ρόλο του κύριου προγράμματος και η οποία υποχρεωτικά λέγεται main .

Διόρθωση σφαλμάτων (debugging):
Σε αντιστοιχία με τα στάδια συγγραφής, προεπεξεργασίας, καθαυτό μεταγλώττισης και εκτέλεσης ενός προγράμματος, εντοπίζονται και τα εξής είδη σφαλμάτων:
α) συντακτικά, επειδή δεν τηρείται κάποιος κανόνας της C. Αυτά εντοπίζονται από το μεταγλωττιστή στο στάδιο της μεταγλώττισης.
β) Σφάλματα εκτέλεσης ή σημασιολογικά ή έξυπνα σφάλματα. Αυτά "ξεφεύγουν" από το προηγούμενο στάδιο επειδή δεν έχουν σχέση με παραβίαση γραμματικών κανόνων, αλλά εντοπίζονται κατά την εκτέλεση και ο υπολογιστής μας τυπώνει ένα σχετικό μήνυμα που μας κατατοπίζει και το πρόγραμμα συνήθως τερματίζεται. Π.χ. προσπάθεια πρόσβασης σε περιοχή της μνήμης που δεν έχει δεσμευτεί.
γ) Λογικά σφάλματα. Αυτά δε μπορεί να τα εντοπίσει ο μεταγλωττιστής ή ο υπολογιστής γιατί έχουν να κάνουν με τη σχεδίαση του αλγορίθμου. Υπάρχει περίπτωση ένα λάθος να εντοπιστεί ακόμη και μετά από μήνες παραγωγικής δουλειάς! Γι' αυτό ένα πρόγραμμα πρέπει να δοκιμάζεται εκτενώς με εφαρμογή σε διάφορες περιπτώσεις προβλημάτων των οποίων ξέρουμε τη λύση.
Στην εξεύρεση και διόρθωση των σφαλμάτων βοηθούν και σχετικά προγράμματα σχεδιασμένα για το σκοπό αυτό (debuggers).

Για να αποφύγουμε μερικά συχνά λάθη προσέχουμε τα εξής:
- οι εντολές να τελειώνουν με ;
- να κλείνουν τα άγκιστρα (για κάθε { να υπάρχει και ένα })
- Παρομοίως να κλείνουν τα σχόλια (/* και */) και να μην υπάρχουν ένθετα σχόλια

Πρόγραμμα p1-1
Κάτι απλό για να ξεκινήσουμε.

```
#include <stdio.h>

void main(void)
(
   printf("\nHello!\n");
}
```

Να γίνουν δοκιμές με δύο ή περισσότερους χαρακτήρες \n, καθώς και χωρίς καθόλου αλλαγή γραμμής. Επίσης, με \t, \v, \b, \r και \a. Πειραματιστείτε με τις ακολουθίες διαφυγής σε διάφορες θέσεις.
Προσοχή! Οι χαρακτήρες διαφυγής να είναι πάντα μέσα στα εισαγωγικά “ “.

Πρόγραμμα p1-2
Εδώ εισάγουμε δύο από τους τύπους μεταβλητών για να δείξουμε πώς να τους τυπώνουμε με την εντολή printf. Επίσης, εισάγεται η έννοια της απόδοσης τιμής σε μία μεταβλητή.

```
#include <stdio.h>

void main(void)
{
   int i = 1, j = 2;
   float a = 0.5, b = 2;
   printf("Οι αριθμοί i και j είναι %d και %d αντιστοίχως\n", i, j);

   printf("Το άθροισμα των i και j είναι %d\n", i+j);
   printf("Οι αριθμοί a και b είναι %f και %f αντιστοίχως\n", a, b);
   printf("Το γινόμενο των a και b είναι %f\n", a*b);
}
```

Βάλτε λάθος format για τη μορφοποίηση των μηνυμάτων εξόδου: %d αντί για %f και το αντίστροφο. Παρατηρείστε τι συμβαίνει με τα μηνύματα εξόδου.

Πρόγραμμα p1-3
Γενίκευση του προηγούμενου, για να εισάγουμε την εντολή ανάγνωσης scanf. Προσοχή! Δεν ξεχνάμε το σύμβολο & που είναι απαραίτητο σε αυτή τη νέα εντολή.

```
#include <stdio.h>

void main(void)
{
   int i, j;
   float a, b;

   printf("Δώσε μου έναν ακέραιο αριθμό: ");
   scanf("%d", &i);
   printf("Δώσε μου άλλον έναν ακέραιο: ");
   scanf("%d", &j);
   printf("Οι αριθμοί i και j είναι %d και %d αντιστοίχως\n", i, j);
   printf("Το άθροισμα τους είναι %d\n\n", i+j);

   printf("Δώσε μου έναν πραγματικό αριθμό: ");
   scanf("%f", &a);
   printf("Δώσε μου άλλον έναν πραγματικό: ");
   scanf("%f", &b);
   printf("Οι αριθμοί a και b είναι %f και %f αντιστοίχως\n", a, b);
   printf("Το γινόμενο των a και b είναι %f\n", a*b);
}
```

Πρόγραμμα p1-4
Παραλλαγή του προηγούμενου για να δώσει την έννοια της συνάρτησης ως δομικού λίθου ενός

σύνθετου προγράμματος. Επίσης, εισάγεται η έννοια της εμβέλειας μιας μεταβλητής.

```
#include <stdio.h>

void intSum(void)
{
   int i, j;

   printf("Δώσε μου έναν ακέραιο αριθμό: ");
   scanf("%d", &i);
   printf("Δώσε μου άλλον έναν ακέραιο: ");
   scanf("%d", &j);
   printf("Οι αριθμοί i και j είναι %d και %d αντιστοίχως\n", i, j);
   printf("Το άθροισμα τους είναι %d\n", i+j); /* εδώ έβαλα μόνο μια φορά την ακολουθία
                                                  διαφυγής \n */
}

void floatSum(void)
{
   float a, b;

   printf("Δώσε μου έναν πραγματικό αριθμό: ");
   scanf("%f", &a);
   printf("Δώσε μου άλλον έναν πραγματικό: ");
   scanf("%f", &b);
   printf("Οι αριθμοί a και b είναι %f και %f αντιστοίχως\n", a, b);
   printf("Το γινόμενο των a και b είναι %f\n", a*b);
}

void main(void)
{
   intSum();
   printf("\n");
   floatSum();
}
```

Πρόγραμμα p1-5

Παραλλαγή του προηγούμενου για να δώσει την έννοια της επανάληψης και του σχετικού μπλοκ εντολών, να τονίσει την έννοια της εμβέλειας μεταβλητών και να δώσει παράδειγμα συνάρτησης με συγκεκριμένο τύπο επιστρεφόμενης τιμής.

```
#include <stdio.h>

int intSum(void)
{
   int x, number, count, sum;

   count = 0;
   sum = 0;
   printf("Δώσε μου ένα πλήθος ακέραιων αριθμών: ");
   scanf("%d", &number);
   while(count < number)
```

```
    {
        printf("Δώσε μου τον υπ' αριθμόν %d όρο: ", count+1);
        scanf("%d", &x);
        count = count+1;
        sum = sum + x;
    }
    return sum;
}

void main(void)
{
    int sum;

    sum = intSum();
    printf("Το άθροισμα όλων είναι %d\n", sum);
}
```

Σημείωση: Η εντολή

```
count = count+1;
```

μπορεί να γραφεί και ως

```
count++;
```

ενώ η εντολή

```
sum = sum + x;
```

γράφεται και

```
sum +=  x;
```

Πρόγραμμα p1-6

Άλλο ένα παράδειγμα επανάληψης, αυτή τη φορά συνδυασμένο και με επιλογή/λήψη απόφασης. Πρόκειται για παραλλαγή του αλγόριθμου εύρεσης μέγιστου και ελάχιστου που αναφέρθηκε στο μάθημα.

```
#include <stdio.h>

void main(void)
{
    int x, number, count, minimum=0, maximum=0;

    printf("Δώσε μου ένα πλήθος θετικών ακέραιων αριθμών: ");
    scanf("%d", &number);

    printf("Δώσε μου τον υπ' αριθμόν %d όρο: ", 1);
    scanf("%d", &x);
    minimum = x;
    if(x > maximum)
        maximum = x;
    count = 1;

    while(count < number)
    {
        printf("Δώσε μου τον υπ' αριθμόν %d όρο: ", count+1);
        scanf("%d", &x);
        if(x < minimum)
```

```
        minimum = x;
      if(x > maximum)
        maximum = x;
      count++;
   }

   printf("Η μέγιστη τιμή είναι %d και η ελάχιστη είναι %d \n", maximum, minimum);
}
```

Πρόγραμμα p1-7
Παραλλαγή του προηγούμενου με χρήση συνάρτησης για την υλοποίηση του αλγόριθμου και εισαγωγή της έννοιας των παραμέτρων, καθώς και του περάσματος ορισμάτων μέσω αναφοράς στη διεύθυνση.

```
#include <stdio.h>

void miniMax(int *maximum, int *minimum)
{
   int x, number, count;

   *minimum = 0;
   *maximum = 0;
   printf("Δώσε μου ένα πλήθος θετικών ακέραιων αριθμών: ");
   scanf("%d", &number);

   printf("Δώσε μου τον υπ' αριθμόν %d όρο: ", 1);
   scanf("%d", &x);
   *minimum = x;
   if(x > *maximum)
      *maximum = x;
   count = 1;

   while(count < number)
   {
      printf("Δώσε μου τον υπ' αριθμόν %d όρο: ", count+1);
      scanf("%d", &x);
      if(x < *minimum)
         *minimum = x;
      if(x > *maximum)
         *maximum = x;
      count++;
   }
}

void main(void)
{
   int m, n;

   miniMax(&m,&n);
   printf("Η μέγιστη τιμή είναι %d και η ελάχιστη είναι %d \n", m, n);
}
```

Πρόγραμμα p1-8
Πρόγραμμα που εξηγεί παραστατικά την έννοια και τις πρακτικές συνέπειες του περάσματος ορισμάτων με τιμή σε αντίθεση με το πέρασμα μέσω αναφοράς.

```
#include <stdio.h>

void testFunc(int x, int y)
{
   printf("\nΟι τιμές των x και y είναι %d και %d, αντιστοίχως", x, y);
   x = x+2;
   y = y-x;
   printf("\nΟι τιμές των x και y τώρα είναι %d και %d, αντιστοίχως", x, y);
}

void main(void)
{
   int x, y;
   printf("\nΔώσε έναν ακέραιο x :");
   scanf("%d", &x);
   printf("\nΔώσε άλλον έναν ακέραιο y :");
   scanf("%d", &y);
   testFunc(x, y);
   printf("\nΟι τιμές των x και y μετά την κλήση είναι %d και %d, αντιστοίχως\n", x, y);
}
```