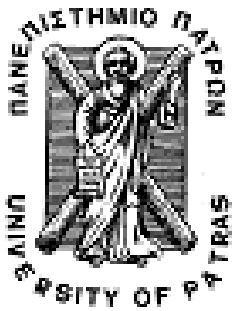
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ



ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΙΑΔΙΚΑΣΤΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2010-2011)

ΥΠΕΥΘΥΝΟΙ ΔΙΔΑΣΚΟΝΤΕΣ ΕΡΓΑΣΤΗΡΙΟΥ: Α. ΦΩΚΑ, Κ. ΣΤΑΜΟΣ

3ο 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/) .

- Συναρτήσεις (κεφάλαιο Functions)
- Πίνακες (κεφάλαιο Arrays)
- Συμβολοσειρές (κεφάλαιο Strings)
- Χρήση Βιβλιοθηκών (κεφάλαιο Libraries)

'Ασκηση 1η

Γράψτε πρόγραμμα το οποίο θα ζητάει από τον χρήστη να δώσει ένα αλφαριθμητικό (string) Ν (ορίζεται με σταθερά) το πολύ χαρακτήρων και στη συνέχεια θα ελέγχει εάν το αλφαριθμητικό string που εισήχθη μπορεί να διαβαστεί κανονικά και ανάποδα. Όταν εντοπίσει τέτοιο string θα το εκτυπώνει και στη συνέχεια θα εκτυπώνει τους ASCII ακεραίους που θα αντιστοιχούν στους χαρακτήρες του string π.χ.

Insert String (10 chars max): abccba
The string abccba can be read backwards
ASCII representation: 656667676665

```
#include <stdio.h>
#include <string.h>
main()
{
    char s[20];
      int i,len, flag=1;

      printf("Give string: ");
      scanf("%s", s);

      len=strlen(s);
      for (i=0; i<len/2;i++)
             if (s[i]!=s[len-1-i])
                  flag=0;

      if (flag==1)
             printf("The string can be read backwards \n");
         else
             printf("The string cannot be read backwards \n");

      for (i=0; i< len; i++)
       printf("%d \n", s[i]);
}
```

Άσκηση 2η

Λέξη ονομάζεται μια ακολουθία χαρακτήρων που δεν περιέχει τον κενό χαρακτήρα. Φτιάξτε ένα πρόγραμμα που δέχεται από το χρήστη μια ακολουθία χαρακτήρων και υπολογίζει και τυπώνει το πλήθος των λέξεων που περιέχει η ακολουθία. Επίσης εμφανίζει κάθε λέξη στην οθόνη (μία λέξη σε κάθε γραμμή). **Προσοχή:** ανάμεσα στις λέξεις της δοσμένης ακολουθίας μπορεί να υπάρχουν περισσότεροι από έναν κενοί χαρακτήρες.

Dwste mia akolou8ia xarakthrwn: this is some input string
this
is
some
input
string
Yparxoun 5 lekseis.

```
#include <stdio.h>
#define N 255

main()
{
   char string[N];
   char buffer[N];
   int i,j, words;
   char c;

   i = 0;
   printf("Dwste mia akolou8ia xarakthrwn: ");
   do{
        string[i++] = getchar();
   }while(string[i-1] != '\n');

   i = j = 0;

   if (string[i] == ' ')
      words = 0;
   else
      words = 1;

   while(string[i] != '\0')
   {
        if (string[i] == ' ')
        {
                if(i+1 < N && string[i+1] != ' ')
                {
                        words++;
                        printf("\n");
                }
        }
        else
                printf("%c", string[i]);
        i++;
   }

   printf("Yparxoun %d lekseis.\n\n", words);
}
```

Άσκηση 3η

Να αποθηκεύσετε σε ένα δυσδιάστατο πίνακα τις θερμοκρασίες 4 πόλεων (Αθήνα, Θεσσαλονίκη, Πάτρα, Ηράκλειο) για κάθε μέρα του Δεκεμβρίου 2007. Οι τιμές να δίνονται μέσα στο πρόγραμμα ως εξής:

{{2,0,-3,-7,6,15,10,11,4,2,12,23,13,14,5,1,-1,-3,-7,0,5,3,10,9,14,20,14,12,10,11,6},

{4,2,12,21,13,14,5,1,-1,-3,-2,0,5,3,10,9,14,20,14,12,10,11,6,2,0,-3,4,-4,15,10,11},
{6,15,10,11,4,2,12,2,0,-3,4,25,13,14,5,1,10,9,14,25,14,12,-1,-3,-9,0,5,3,12,10,4},
{2,15,10,0,-3,4,6,11,4,13,14,5,2,12,12,1,-1,-3,-3,9,14,0,5,3,10,20,10,11,6,14,12}}

Στη συνέχεια να βρείτε και να εκτυπώσετε την θερμότερη και την ψυχρότερη μέρα του μήνα για κάθε μία πόλη. Π.χ.
Αθήνα: Θερμότερη μέρα (23 C), 13 Δεκ – Ψυχρότερη μέρα (-7 C), 4 Δεκ, 20 Δεκ

Στη συνέχεια να βρείτε και να εκτυπώσετε την θερμότερη και την ψυχρότερη πόλη για κάθε μέρα του μήνα. Π.χ.

1 Δεκ: Θερμότερη πόλη (6 C), Πάτρα – Ψυχρότερη πόλη (2 C), Αθήνα, Ηράκλειο

Υπόδειξη: Εσωτερικά στο πρόγραμμά σας μπορείτε να αποθηκεύετε την πόλη ως έναν αριθμό 1 έως 4 και μόνο στην εκτύπωση να εμφανίζετε το όνομα της πόλης.

```
#include <stdio.h>

void printCityName (int i)
{
       switch(i)
       {
              case 0: printf("Athina");
                      break;
              case 1: printf("Thessaloniki");
                      break;
              case 2: printf("Patra");
                      break;
              case 3: printf("Hrakleio");
                      break;
       }

       return;
}

main()
{
   int temp[4][31] =
   {{2,0,-3,-7,6,15,10,11,4,2,12,23,13,14,5,1,-1,-3,-7,0,5,3,10,9,14,20,14,12,10,11,6},
    {4,2,12,21,13,14,5,1,-1,-3,-2,0,5,3,10,9,14,20,14,12,10,11,6,2,0,-3,4,-4,15,10,11},
    {6,15,10,11,4,2,12,2,0,-3,4,25,13,14,5,1,10,9,14,25,14,12,-1,-3,-9,0,5,3,12,10,4},
    {2,15,10,0,-3,4,6,11,4,13,14,5,2,12,12,1,-1,-3,-3,9,14,0,5,3,10,20,10,11,6,14,12}};

   int i,j;
   int max, min;

   for (i = 0; i < 4; i++)
   {
       max = temp[i][0];
       min = temp[i][0];

       for (j = 0; j < 31; j++)
       {
           if (temp[i][j] > max)
              max = temp[i][j];

           if (temp[i][j] < min)
              min = temp[i][j];
       }

       printCityName(i);

       printf(": Thermoterh mera (%d C)", max);

       for (j = 0; j < 31; j++)
           if (temp[i][j] == max)
```

```
            printf(", %d Dek", j+1);

      printf(" - Psixroteri mera (%d C)", min);

      for (j = 0; j < 31; j++)
          if (temp[i][j] == min)
             printf(", %d Dek", j+1);

      printf("\n");
  }

  printf("\n\n");

  for (i = 0; i < 31; i++)
  {
      max = temp[0][i];
      min = temp[0][i];

      for (j = 0; j < 4; j++)
      {
          if (temp[j][i] > max)
             max = temp[j][i];

          if (temp[j][i] < min)
             min = temp[j][i];
      }


      printf("%d Dek: Thermoterh polh (%d C)", i+1, max);

      for (j = 0; j < 4; j++)
          if (temp[j][i] == max)
          {
             printf(", ");
             printCityName(j);
          }

      printf(" - Psixroteri polh (%d C)", min);

      for (j = 0; j < 4; j++)
          if (temp[j][i] == min)
          {
             printf(", ");
             printCityName(j);
          }

      printf("\n");
  }

}
```


Άσκηση 4η

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα το οποίο θα καταγράφει τις κινήσεις ενός ρομπότ κατά την κίνηση του σε ένα χώρο. Οι δυνατές κινήσεις αντιστοιχούν στις τιμές εμπρός=1, πίσω=2, δεξιά=3 και αριστερά=4. Ακολούθως πρέπει να κάνει τα εξής:

1. Να αποθηκεύει τις τιμές αυτές σε ένα μονοδιάστατο πίνακα (θα αποθηκεύει τις 1000 πρώτες κινήσεις που πραγματοποίησε). Χρησιμοποιήστε την παραγωγή τυχαίων αριθμών στο πεδίο τιμών [1,4] για το γέμισμα του πίνακα με τιμές.

2. Γράψτε συνάρτηση η οποία δέχεται σαν όρισμα τον πίνακα με τις κινήσεις του ρομπότ και να βρίσκει πόσες φορές το ρομπότ μας αφού κινήθηκε δεξιά έκανε πίσω μετά από δύο θέσεις. Να επιστρέφεται ο αριθμός αυτός στην main() και να εμφανίζεται στην οθόνη.

3. Γράψτε συνάρτηση η οποία δέχεται σαν όρισμα τον πίνακα με τις κινήσεις του ρομπότ και να βρίσκει πόσες φορές βρέθηκε ο συνδυασμός κινήσεων αριστερά – δεξιά – εμπρός. Να επιστρέφεται ο αριθμός αυτός στην main() και να εμφανίζεται στην οθόνη.

4. Γράψτε συνάρτηση η οποία δέχεται σαν όρισμα τον πίνακα με τις κινήσεις του ρομπότ και να βρίσκει πόσες φορές έκανε μια πλήρη περιστροφή κατά τη φορά των δεικτών του ρολογιού. Μια πλήρη περιστροφή πραγματοποιείται όταν το ρομπότ κινηθεί 3 συνεχόμενες φορές με στροφή στα δεξιά. Να επιστρέφεται ο αριθμός αυτός στην main() και να εμφανίζεται στην οθόνη.

5. Γράψτε συνάρτηση η οποία δέχεται σαν όρισμα τον πίνακα με τις κινήσεις του ρομπότ και να βρίσκει τη συχνότερη κίνηση που πραγματοποίησε συνολικά το ρομπότ. Να επιστρέφεται ο αριθμός αυτός στην main() και να εμφανίζεται στην οθόνη.

6. Γράψτε συνάρτηση η οποία δέχεται σαν όρισμα τον πίνακα με τις κινήσεις του ρομπότ και να βρίσκει τη σπανιότερη κίνηση που πραγματοποίησε συνολικά το ρομπότ. Να επιστρέφεται ο αριθμός αυτός στην main() και να εμφανίζεται στην οθόνη.

7. Γράψτε συνάρτηση η οποία δέχεται σαν όρισμα τον πίνακα με τις κινήσεις του ρομπότ και για κάθε φορά που το ρομπότ έκανε δύο φορές πίσω, να αντικαθιστά την δεύτερη οπισθοχώρηση με μια κίνηση αριστερά. Η συνάρτηση δεν επιστρέφει τίποτα στη main().

Παραγωγή τυχαίων αριθμών:

Για την παραγωγή τυχαίων αριθμών χρησιμοποιήστε την γεννήτρια τυχαίων αριθμών rand() της γλώσσας C όπως στο παρακάτω παράδειγμα.

```
#include <stdlib.h>
#include <time.h>

/* συνάρτηση RandomInteger για την παραγωγή τυχαίου αριθμού μέσα σε
καθορισμένο πεδίο τιμών [low, high]. */
int RandomInteger(int low, int high)
{
      int k;
      double d;
```

```
      d = (double) rand() / ((double)RAND_MAX + 1);
      k = (int) (d * (high – low + 1));

      return (low + k);
}

main()
{
      int i, number;

      /* αρχικοποίηση γεννήτριας παραγωγής τυχαίων αριθμών με βάση την
        συνάρτηση srand() */
      srand((int)time(NULL));

      /*κλήση της RandomInteger μέσα σε for για την παραγωγή 10
        τυχαίων αριθμών στο πεδίο τιμών [1,9] */
      for (i = 0; i < 10; i++)
      {
            number = RandomInteger(1, 9);
            Printf("%d\n", number);
      }
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define N 1000
#define FORWARD 1
#define BACKWARD 2
#define RIGHT 3
#define LEFT 4

int RandomInteger(int low, int high)
{
  int k;
  double d;

  d = (double) rand() / ((double)RAND_MAX + 1);
  k = (int) (d * (high - low + 1));

  return (low + k);
}

/**
 * Returns the number of patterns of the form R?B (right movement, any movement, backwards movement)
 *@param Array of integers
```

```
 *@returns number of exc2 patterns
 */
int FindR_B(int Array[])
{
	int i=0,counter=0;

	for (i=0; i < N-2; i++)
		if ((Array[i]==RIGHT) && (Array[i+2]==BACKWARD))
			counter++;

	return counter;
}

int findLRF(int Array[])
{
	int i=0;
	int count=0;

	for(i=0; i < (N-2); i++)
		if( (Array[i] == LEFT) && (Array[i+1] == RIGHT) && (Array[i+2] == FORWARD) )
			count++;

	return count;
}

/**
 *Returns the number of the circular patterns of the form RRR ( right , right , right)
 *@param Array of integers
 *@returns number of clockwise circles done by the robot
 */
int FindCircle(int Array[])
{
	int i=0,counter=0;

	for (i=0; i < N-2; i++)
		if ((Array[i]==RIGHT) && (Array[i+1]==RIGHT) && (Array[i+2]==RIGHT))
			counter++;

	return counter;
}

/**
 *Returns the movement (1-4) which has the least frequency
 *@param Array of integers
 *@returns the least probable movement (the movement with the minimum frequency)
 */
int FindRareMovement(int Array[])
```

```
{
	int i=0,counter=0,FreqMatrix[4]={0,0,0,0},min=0;

	for (i=0; i < N; i++)
		FreqMatrix[Array[i]-1]++;

	for (i=1; i < 4; i++)
		if (FreqMatrix[min] > FreqMatrix[i])
			min=i;

	return min;
}

int findMaxFreq(int Array[])
{
	int i=0;
	int Freqs[4]={0,0,0,0};
	int max = 0;

	for(i = 0; i < N; i++)
		Freqs[Array[i]-1]++;

	for(i = 1; i < 4; i++)
		if(Freqs[max] < Freqs[i])
			max = i;

	return max;
}

void replaceBBWithBL(int Array[])
{
	int i=0;

	for(i = 0 ; i < (N-1); i++)
		if((Array[i] == BACKWARD) && (Array[i+1] == BACKWARD) )
			Array[i+1] = LEFT;
}

main()
{
    int moves[N];
    int i;

    srand((int)time(NULL));

    for (i = 0; i < N; i++)
         moves[i] = RandomInteger(1, 4);
```

```
       printf("Number of R?B moves %d\n",FindR_B(moves));
       printf("LRF count: %d\n",findLRF(moves));
       printf("Number of circular moves %d\n",FindCircle(moves));
       printf("The movement with the least frequency %d\n",FindRareMovement(moves));
       printf("max freq: %d\n",findMaxFreq(moves));
      replaceBBWithBL(moves);
   printf("Replaced second backward movement with left movement\n");
}
```