**National University of Computer & Emerging Sciences**

**Karachi Campus**

**Data Structures (CS201)**

**Programming Assignment # 1**

**Instructions**

**Due Date:** September 30, 2012

**Submission:** The assignment has to be submitted via slate. You must submit the source code files with proper naming convention for example (Question No. 1 of Assignment No. 1) you should give A1Q1.cpp. You should copies all questions for the assignment in a single folder (named as your id e.g. K102000) and zipped it before uploading to Slate.

Add your name and roll number at the beginning of each program, in comments.

**Plagiarism:** Any form of collaboration with other students will not be.

**Problem # 1 Magic Print (With Constraints)**

Write a program which for any given positive n, print integers from 1 to n and then print from n-1 to 1 using only a single loop without any explicit if-else statement and not using any other memory location except loop variable.

**Input / Output**

Enter a Positive Integer less than 10: 7

Magic Print for n=7: 1 2 3 4 5 6 7 6 5 4 3 2 1

**Problem #2 Block of Numbers**

You are given a two-dimensional array of 10 X 10 consists of natural numbers in between (1-9), you are required to find the largest possible block (a sub array) within it that contains natural numbers between (a-b) a sub sequence of natural numbers in between (1-9). For example:

Array 10 X 10 with (1-9)

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

1 2 4 2 5 6 7 2 2 1

Output for (1-4)

Array 10 X 10 with (1-9)

1 2 4 2

1 2 4 2

1 2 4 2

1 2 4 2

1 2 4 2

1 2 4 2

1 2 4 2

1 2 4 2

1 2 4 2

1 2 4 2

Output for (2-7)

Array 10 X 10 with (1-9)

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

2 4 2 5 6 7 2 2

You are only required to print the dimension of the largest possible two-dimensional sub array

of sub-sequence. For the above example 4X10 and 8X10 will be the output.

**Problem # 3 Bisection Methods**

Professor Robert came across an interesting problem, he wants to find a root of function   
**f(x)= ax-1** in between the value of xl = 0 and xu =1, he knows that there exists a root in this boundary. He knows the Newton’s Bisection methods for finding the root of a equation, He want that you people help him in implementing the same algorithm on C programming language.

  
allowed.

All you need to help him, in writing a program that will print on screen the successive steps of the algorithm along with the error and the value of the function at the bisection points. Refer to the table produced for the calculation manually. (if you need further help, plz see the annexure)



**Problem # 4 Shortest path through a maze**

All mazes are not perfect. In a perfect maze there is exactly one path from each point to each other point. However, in many mazes, there are multiple paths. For example, the diagrams below show three solutions for the same maze:



length = 13

length = 15

length = 13

None of these solutions, however, is optimal. The shortest path through the maze has a

path length of 11:



**Problem # 6 Counting In-between**

Given two positive integers **a** and **b**, we usually write the numbers between a and b, inclusive, in a list. Your task is to write a program that counts the number of occurrences of each digit using single loop. For example, if a = 1024 and b = 1032, the list will be 1024 1025 1026 1027 1028 1029 1030 1031 1032

There are ten 0s in the list, ten 1s, seven 2s, three 3s, and etc.

Sort your output with respect to number of occurrences using shell sort and any other stable sort.

**Input / Output**

Enter 2-Positive integers (a b): 1024 1032

There are: 0-10, 1-10, 2-7, 3-3, 4-1, 5-1, 6-1, 7-1, 8-1 and 9-1.