NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS
COMPUTER PROGRAMMING (CS 103) - SPRING 2014
ASSIGNMENT # 2

Due Date: February 17, 2013 (11:55 pm)

Submission Instructions:
1. Submit the source code of the assignment via slate.

2. Use proper naming convention to name the file containing source code. For example, the file containing the source code for second question of the third assignment should be named as **A3Q2.cpp**, third question of the second assignment as **A2Q3.cpp**, etc.

3. Please write your name and roll number at the beginning of the each program using comments.

Plagiarism: Plagiarism is not allowed. If found plagiarized a case would be submitted to the DC committee against you.

Note: Don't upload zip file.

Q1. You have been given a puzzle consisting of a row of squares each containing an integer, like this:

3 6 4 1 3 4 2 5 3 0

The shade on the initial square is a marker that can move to other squares along the row. At each step in the puzzle, you may move the marker the number of squares indicated by the integer in the square it currently occupies. The marker may move either left or right along the row but may not move past either end. For example, the only legal first move is to move the marker three squares to the right because there is no room to move three spaces to the left.

The goal of the puzzle is to move the marker to the 0 at the far end of the row. In this configuration, you can solve the puzzle by making the following set of moves:

Starting position	3 6 4 1 3 4 2 5 3 0
Step 1: Move right	3 6 4 1 3 4 2 5 3 0
Step 2: Move left	3 6 4 1 3 4 2 5 3 0
Step 3: Move right	3 6 4 1 3 4 2 5 3 0

Step 4: Move right

3 6 4 1 3 4 2 5 3 0

Step 5: Move left

3 6 4 1 3 4 2 5 3 0

Step 6: Move right

3 6 4 1 3 4 2 5 3 0

Even though this puzzle is solvable—and indeed has more than one solution—some puzzles of this form may be impossible, such as the following one:

3 1 2 3 0

In this puzzle, you can bounce between the two 3's, but you cannot reach any other squares.

Write a program that prompts the user to enter the size of the array, allocate memory using new operator, ask the user to enter elements of the array and displays if the puzzle is solvable. As part of this program, write a recursive function **int Solvable(int start, int squares[], int size)** that takes a starting position of the marker along with the array of squares and the number of elements. The function should return **1** if it is possible to solve the puzzle from the starting configuration and **0** if it is impossible.

You may assume all the integers in the array are positive except for the last entry, the goal square, which is always zero. The values of the elements in the array must be the same after calling your function as they are beforehand, (which is to say if you change them during processing, you need to change them back!)

Q2. Binomial coefficients are the numeric factors of the products in a power of a binomial such as $(x + y)^n$. For example, $(x + y)^2 = x^2 + 2\,x\,y + y^2$ has the coefficients `1 2 1`. Binomial coefficients can be calculated using Pascal's triangle:

```
                1                       n = 0
            1       1
        1       2       1
    1       3       3       1
1       4       6       4       1       n = 4
```

Each new level of the triangle has `1`'s on the ends; the interior numbers are the sums of the two numbers above them. Write a program that includes a recursive function to produce a list of binomial coefficients for the power `n` using the Pascal's triangle technique. For example,

Input = 2 Output = 1 2 1

Input = 4 Output = 1 4 6 4 1

Q3. Write a program that asks the user to enter 10 integers, stores these numbers in an array. The program should ask if the user would like to search an element in the array, in which case the program should then prompt the user to enter the number to look for. Use a recursive function find(int index, int target, int array[]) that does sequential search for a target

in this array. Display an appropriate message to the user after searching the element. The program should again ask if the user would like to search another element in the same array and repeat this process if necessary.

Q4. The value of π can be determined by the series equation

$$\pi = 4 * (1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 \ldots)$$

Write a program that prompts the user to enter a positive odd number *n* and uses recursion to approximate the value of π using the given formula including term up through *1/n*.

Q5. Solve the **Nine Queens** problem recursively. The objective is to place nine queens on an empty chessboard so that no queen is "attacking" any other, i.e., no two queens are in the same row, the same column, or along the same diagonal?

Show a solution on the screen and wait for the user to press a key before displaying the next solution.

Q6. Write a function that accepts an array of integers and a number indicating the number of elements as arguments. The function should recursively calculate the sum of all the numbers in the array. This function should be part of a program that reads the number of integers to be entered by the user, allocate space for that many integers, reads all the numbers from the user, and finally use this function to calculate the sum of the numbers in the array.

Q7. Write a recursive function that accepts two arguments into the parameters x and y. The function should return the value of x times y. You are required to compute the product as repeated addition, for example: 7*4=4+4+4+4+4+4+4. Write a program that reads two numbers from the user and uses this function to find the product of the given numbers. You may assume that the numbers entered by the user are both greater than zero.

Q8. Write a function that accepts an integer argument (>=1) and returns the sum of all the integers from 1 up to the number passed as an argument. For example, if 50 is passed as the argument, the function will return the sum of 1, 2, 3, 4, ... 50. Use recursion to calculate the sum. Write a program that reads an integer from the user and displays the sum using this function.

Q9. Write a recursive function that accepts a c-string as its argument and prints the string in reverse order. Demonstrate the function in a driver program that reads a string from the user and uses this function to it in reverse order.

Q10. A palindrome is any word, phrase, or sentence that reads the same forward and backward. Here are some well-known palindromes:

Able was I, ere I saw Elba

A man, a plan, a canal, Panama

Desserts, I stressed

Kayak

Write a bool function that uses recursion to determine if a c-string argument is a palindrome. The function should return true if the argument reads the same forward and backward. Demonstrate the function in a program that allows the user to enter a string and uses this function to determine if the input string is a palindrome.