|  |  |  |
| --- | --- | --- |
|  |  | **ISM 6225**  **Distributed Information systems** |
| Prof Agrawal TA: Khaninder Kommagoni | | |

Assignment 1 – Programming Introduction

Primary objective: Develop familiarity with essential programming constructs

Secondary objective: develop comfort with using the IDE and GitHub

*Estimated time: 25 hours*

## Introduction

Full-stack application development is an essential skill needed to succeed and even survive in business analytics and/ or information systems roles, especially as AI takes over many rudimentary tasks formerly performed by analysts. This assignment introduces the essential programming constructs such as variables, selection, loops, methods and arrays used to build such applications. Specifically, this assignment avoids the use of API methods and object-oriented programming. Those tasks are left for later assignments. This assignment also does not check for efficiency in program implementation. That is something you will develop over a lifetime in the profession. Rather, the focus is on simple programming exercises for students to learn basic industry best practices. One design goal for this assignment was to focus tightly on introductory programming structures, with low probability that students would find ready-to-use solutions online.

This is an individual assignment, to give every student the opportunity to develop the necessary skills to become a productive contributor to project teams in this class and beyond.

## Activity

Use a programming language of your choice to define methods to do the operations specified in the method signatures and hints below. The methods are listed in the recommended sequence of development. A starter Program.cs file is included in the appendix.

## Submission

Push the code to GitHub and submit the URL to Canvas. Also, get the output from a sample run that shows the use of all required methods and upload/push a screenshot to GitHub. This serves as a quick check. Submit your self-reflection as a comment to the assignment.

## Grading scheme

Each method carries 1 point. You will be graded on the following aspects for each question:

Logic (including appropriate organization of logic into methods) : 0.5

Handling all reasonable corner cases : 0.2

Descriptive comments explaining the logic to reviewer : 0.2

Self-reflection (time taken, learning, and recommendations) : 0.1

## Method specifications

**QUESTION 1:**

/\*n – number of rows for the pattern, integer (int)

\* summary : This method prints a triangle pattern using \*.

\* For example n = 5 will display the output as:

\*

\* \*

\* \*\*\*

\* \*\*\*\*\*

\* \*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*

\*

\* returns : N/A

\* return type : void

\*/

private static void PrintTriangle(int n)

**QUESTION 2:**

/\* n2 – number of terms of the series, integer (int)

\* This method prints the following series(Odd numbers) till n terms: 1, 3, 5, 7, 9,

\* 11… and their sum.

\* For example, if n2 = 5, output will be

\*

\* The odd numbers are : 1, 3, 5, 7, 9

\* The sum is : 25

\*

\* Returns : N/A

\* Return type: void

\*

\*/

private static void PrintSeriesSum(int n2)

**QUESTION 3:**

/\* An array is *monotonic* if it is either monotone increasing or monotone decreasing.

\* An array A is monotone increasing if for all i <= j, A[i] <= A[j].  An array A is

\* monotone decreasing if for all i <= j, A[i] >= A[j].

\* Return true if and only if the given array A is monotonic

\* For example:

\* Input: A = [1,2,2,6] will return the output: true

\* Input: A = [3,3,2,1] will return the output : true

\* Input: A = [4,5,2,3] will return the output: false

\* Input: A = [1,1,1,1] will return the output : true

\* returns : Boolean Value

\* return type : bool

\*/

public static bool MonotonicCheck(int [] A)

**QUESTION 4:**

/\* Given an array of integers and an integer n, you need to find the number of unique

\* n-diff pairs in the array. Here a n-diff pair is defined as an integer pair (i, j),

\* where i and j are both numbers in the array and their absolute difference is n.

\* Example 1:

\* Input: [3, 1, 4, 1, 5], k = 2

\* Output: 2

\* Explanation: There are two 2-diff pairs in the array, (1, 3) and (3, 5).

\* Although we have two 1s in the input, we should only return the number of unique

\* pairs.

\* Example 2:

\* Input:[1, 2, 3, 4, 5], k = 1

\* Output: 4

\* Explanation: There are four 1-diff pairs in the array, (1, 2), (2, 3), (3, 4) and

\* (4, 5).

\* Example 3:

\* Input: [1, 3, 1, 5, 4], k = 0

\* Output: 1

\* Explanation: There is one 0-diff pair in the array, (1, 1).

\* Note : The pairs (i,j) and (j,i) count as same.

public static int DiffPairs(int[] nums, int k)

**QUESTION 5:**

/\* Imagine a special bulls keyboard with all keys in a single row.

\* Given a string keyboard of length 26 indicating the layout of the keyboard(indexed

\* from 0 to 25), initially your finger is at index 0. To type a character, you have

\* to move your finger to the index of the desired character.

\* The time taken to move your finger from index i to index j is |i – j|

\* You want to type a single word. Complete the function to calculate

\* how much time it takes to type it with one finger.

\* Example 1:

\* Input: keyboard = “abcdefghijklmnopqrstuvwxyz” word = “dis”

\* Output: 18

\* Explanation: Initial index 0 at a . The moves from 0 to 3 , then 3 to 8 and finally

\* from 8 to 18. Therefore total time = 3 + 5 + 10 = 18

\*

\* Example 2:

\* Input: keyboard = “hijklmnopqrstuvwxyzabcdefg” word = “gobulls”

\* Output: 79

\* returns : Integer

\* return type : int

\*/

public static int BullsKeyboard(string keyboard, string word)

QUESTION 6:

/\* Given two strings str1 and str2 and below operations that can performed on str1.

\* Find minimum number of edits (operations) required to convert ‘str1’ into ‘str2’

\* 1.Insert

\* 2.Remove

\* 3.Replace

\*

\* All of the above operations are of equal cost.

\*

\* Example 1:

\* Input: str1 = “goulls” str2 = “gobulls”

\* Output: 1

\* Explanation: We can convert str1 to str2 by inserting ‘b’

\*

\* Example 2:

\* Input: str1 = “robky” str2 = “rocky”

\* Output: 1

\* Explanation: We can convert str1 to str2 by replacing ‘b’ with ‘u’

\*

\* Example 3:

\* Input: str1 = “sunday” str2= “saturday”

\* Output: 3

\* Explanation: We can convert by replacing ‘n’ with ‘r’ and inserting ‘t’ and ‘a’

\* returns : Integer

\* return type : int

\*/

public static int StringEdit(string str1, string str2)

## Appendix: Program.cs

using System;

namespace Assignment1\_Fall20

{

    class Program

    {

        static void Main(string[] args)

        {

            int n = 5;

            PrintTriangle(n);

            int n2 = 5;

            PrintSeriesSum(n2);

            int[] A = new int[] { 1, 2, 2, 6 }; ;

            bool check = MonotonicCheck(A);

            Console.WriteLine(check);

            int[] nums = new int[] { 3, 1, 4, 1, 5 };

            int k = 2;

            int pairs = DiffPairs(nums, k);

            Console.WriteLine(pairs);

            string keyboard = "abcdefghijklmnopqrstuvwxyz";

            string word = "dis";

            int time = BullsKeyboard(keyboard, word);

            Console.WriteLine(time);

            string str1 = "goulls";

            string str2 = "gobulls";

            int minEdits = StringEdit(str1, str2);

            Console.WriteLine(minEdits);

        }

        public static void PrintTriangle(int x)

        {

            try

            {

                // Write your code here

            }

            catch

            {

                Console.WriteLine("Exception occured while computing PrintTriangle()");

            }

        }

        public static void PrintSeriesSum(int n)

        {

            try

            {

                // Write your code here

            }

            catch

            {

                Console.WriteLine("Exception occured while computing PrintSeriesSum()");

            }

        }

        public static bool MonotonicCheck(int[] n)

        {

            try

            {

                // Write your code here

            }

            catch

            {

                Console.WriteLine("Exception occured while computing MonotonicCheck()");

            }

            return false;

        }

        public static int DiffPairs(int[] J, int k)

        {

            try

            {

                // Write your code here

            }

            catch

            {

                Console.WriteLine("Exception occured while computing DiffPairs()");

            }

            return 0;

        }

        public static int BullsKeyboard(string keyboard, string word)

        {

            try

            {

                // Write your code here

            }

            catch

            {

                Console.WriteLine("Exception occured while computing BullsKeyboard()");

            }

            return 0;

        }

        public static int StringEdit(string str1, string str2)

        {

            try

            {

                // Write your code here

            }

            catch

            {

                Console.WriteLine("Exception occured while computing StringEdit()");

            }

            return 0;

        }

    }

}