**Lab Assignment – 7  
Due Date: 10/4 (Thursday), 11.59 pm  
Total Points: 100**

**Objectives:**

* Learn to use different type of selection statements in programming
* Learn to build conditional expressions using relational, equality, and logical operators

You will be working on three programs in this assignment.

1. Write a program that reads three edges for a triangle and computes the perimeter if the input is valid. Otherwise, displays that the input is invalid. The input is valid if the sum of any two edges is greater than the third edge.   
     
   For example, if your input for three edges is 1, 2, 1 the output should be:  
   ***The inputs are invalid as sum of two edges is less than or equal to the third edge.***

For example, if your input for three edges is 2, 2, and 1, the output should be:  
***The inputs are valid as sum of any two edges is greater than the third edge and the perimeter of this triangle is 5.***

**Develop test cases for the above program before writing the code.**

As an example, couple of test cases are provided below. You would need to create additional test cases with inputs like negative, zero values and any other values you think the program can be subjected to for edges. Please note that after you write the program, you will test your program with the test cases you developed and document the Actual output of the program in the table below and also check if the test passed or failed. If the test fails, you would have to analyze the problem and fix the code and run the program and retest the test case until if passes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **Inputs**  **(a, b and c are edges of triangle** | **Expected Output** | **Actual Output** | **Test Passed? (Yes/No)** |
| 1 | a = 2, b = 1, c = 2 | ***The inputs are valid as sum of any two edges is greater than the third edge and the perimeter of this triangle is 5.*** | ***The inputs are valid as sum of any two edges is greater than the third edge and the perimeter of this triangle is 5.*** | Y |
| 2 | a = 1, b = 2, c = 1 | ***The inputs are invalid as sum of two edges is less than or equal to the third edge.*** | ***The inputs are invalid as sum of two edges is less than or equal to the third edge.*** | Y |
| 3 | a = 3, b = 1, c = 2 | ***The inputs are invalid as sum of two edges is less than or equal to the third edge.*** | ***The inputs are invalid as sum of two edges is less than or equal to the third edge.*** | Y |
| 4 | a = 5, b = 6, c = 7 | ***The inputs are valid as sum of any two edges is greater than the third edge, and the perimeter of this triangle is 18*** | The inputs are valid as sum of any two edges is greater than the third edge, and the perimeter of this triangle is 18 | Y |
| 5 | a = 4, b = 10, c = 7 | ***The inputs are valid as sum of any two edges is greater than the third edge, and the perimeter of this triangle is 21*** | The inputs are valid as sum of any two edges is greater than the third edge, and the perimeter of this triangle is 21 | Y |

1. Write a program that reads a temperature as a whole number from the keyboard and outputs a “probable” season (winter, spring, summer, or fall) depending on the temperature. ***(40 points)***

|  |  |
| --- | --- |
| Temperature >= 90 | it is probably summer |
| temperature >= 70 and < 90 | It is probably spring |
| temperature >= 50 and <70 | it is probably fall |
| Temperature < 50 | It is probably winter |
| Temperature > 110 or < - 5 | Outside the valid range. |

Develop test cases for the program before you write the code. Use the table above to pick test values for temperature. Remember to pick a value on the left-hand side of the edge, a value right on the edge and a value on the right side of the edge. As an example, for condition – temperature > 110, the test values for temperatures would be 111, 110 and 109. Similarly, identify the test cases for all the other conditions and document the expected output. Finally, after you write the program, test your program using the test cases your developed and document the actual output of the program. Next, check if the program passed all the test cases your identified. If not, then analyze the code and fix the issues and retest the program.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **Inputs**  **Temperature** | **Expected Output** | **Actual Output** | **Test Passed? (Yes/No)** |
| **1** | **111** | **Outside the valid range** | **Outside the valid range..** | **Y** |
| **2** | **110** | **Outside the valid range** | **It is probably summer.** | **N** |
| **3** | **109** | **It is probably summer** | **It is probably summer.** | **Y** |

***Run the programs and use the following inputs: 100, 70, 120, 45 and -10 as the temperatures.***

1. Design and implement an application that reads an integer value representing a year from the user. The purpose of the program is to determine if the year is a leap year (and therefore has 29 days in Feb) in the Gregorian calendar.

A year is a leap year if it meets one of the following criteria

* If the year is divisible by 4 and not divisible by 100.

(or)

* If the year is divisible by 400

If a year does not meet any of the above two criteria then it is not a leap year. Display an error message for any input value less than 1582 (the year the Gregorian calendar was adopted). ***(40 points)***

**Develop the test cases for the above program and test your program against the test cases and document the actual output.**

**Run of the program for following years: 2014 and 2016**

**Things to Turn in:**

1. In a new Word file and save the file naming convention that we have used for earlier labs. *Use landscape page layout. From here on you will be using landscape page layout for your assignments and change the font of the document to Courier new.*
2. **Enter your name at the top of the document.**
3. Include the filled-out test-cases table for program – I, make sure the last two columns – Actual Output and Test Passed? are filled out.
4. Copy and paste the source code and output of Program – I, copy and paste the screen shot of the output window of Program – I for two runs of the program for **edge values: *1, 2, 1 and 2, 1, 2.***
5. Include the filled-out test-cases table for program – II, make sure the last two columns – Actual Output and Test Passed? are filled out.
6. Copy and paste the source code and output of Program – II and copy and paste the screen shots of the output window of Program – II for each run of the program for the following ***temperature values: -5, 67, and 90.***
7. Include the filled-out test-cases table for program – III, make sure the last two columns – Actual Output and Test Passed? are filled out.
8. Copy and paste the source code and output of Program – III and copy and paste the screen shots of the output window of Program – III for each run of the program for the ***years – 1580, 2014 and 2016***
9. Zip your NetBeans Project folder
10. Submit your word document along with the zipped NetBeans project folder using the ***Lab Assignment – 7 link*** on blackboard by the due date.