CS 46A Fall 2022

Homework 07

Requirements

1. You must name your classes exactly as specified. Otherwise Codecheck will not be able to process your submission and you will get no credit.
2. When you are finished with your code, submit it to Codecheck one final time then download the .signed.zip file.
3. You must upload all three signed.zip files together to Canvas and you should double check the files in Canvas to make sure all three zip files are uploaded.
4. Do not open the downloaded zip files. The files are digitally signed, and the grader program will check that they have not been opened.
5. Due time: 10 pm, Saturday, Oct 22. Submissions before the due time are not late.

   You will lose five points if your submission is marked Late in Canvas.

6. Grace time: 10 am, Sunday, Oct 23. Submissions before the grace time will not be rejected.

   You will receive no points if your submission is rejected by Canvas.

Remember to follow our Programming Style Requirements!

Problem 7A

Write an application called `Temperatures` to read a collection of double numbers, each representing the high temperature in Fahrenheit for a day. You can assume there are at least two doubles to read in, since you need to find the two highest temperatures. The input will terminate when the user enters any string that cannot be converted to a double.

Do the following

- find and print the count of negative temperatures
- find and print the count of temperatures below freezing Fahrenheit (32.0)
- find and print the two highest temperatures

You will only read the inputs one time and do all the processing as you go. You cannot use arrays or array lists.

You must use one Scanner object and call Scanner methods to process input. You cannot use try-catch statement or call any parse methods.

To find the two highest temperatures, you will need to use two variables, **high** and **secondHigh**. The tricky part is initializing them. You cannot initialize them to 0 since the temperatures can be negative. Since there are at least two temperature values to input, you could input and process the first two integers before the loop.

Must define and use the following constant inside the main() method.

```
public static final double FREEZE_POINT = 32.0;
```

You can complete the program according to the following pseudocode.

```
Read in the first temperature and assign it to high
Update the two counts if needed
Read in the second temperature and assign it to secondHigh
Update the two counts if needed
If secondHigh is larger than high
        Swap the two values
Prompt for the next temperature
While the next input token represents a double
        Read in the next temperature and assign it to a variable
        Update the two counts if needed
        If the input value > secondHigh
                Set secondHigh to the input value
                If secondHigh > high
                        Swap the two high values
        Prompt for the next temperature
Output the results
```

Notice that the two highest temperatures could be the same, and you do not want to swap two identical values.

Sample run

```
Enter the first temperature Fahrenheit: 90
Enter the second temperature Fahrenheit: 50.5
Enter another temperature Fahrenheit or Q to quit: 20.5
Enter another temperature Fahrenheit or Q to quit: -10.5
Enter another temperature Fahrenheit or Q to quit: 40
Enter another temperature Fahrenheit or Q to quit: 90
Enter another temperature Fahrenheit or Q to quit: 0
Enter another temperature Fahrenheit or Q to quit: Java
Number of negative temperatures: 1
Number of below freezing point temperatures: 3
The highest temperature: 90.0
The second highest temperature: 90.0
```

Codecheck link for 7A

Problem 7B

Write a Java application **`DecisionsWithInput`** to read and process a set of integers. Stop reading input when the next input token is not an integer.

Do the following things

- Find and print the sum of all the odd numbers (remainder is not 0 when divided by 2)
- Find and print the smallest of the inputs
- Determine if the number 5 is in the input. Print "5 is my lucky number!" if it is in the inputs, otherwise print "no fives"
- Print the number of values that are positive (greater than 0)

You will only read the inputs one time and do all the processing as you go. No arrays or array lists yet.

You must use one Scanner object and call Scanner methods to process input. You cannot use try-catch statement or call any parse methods.

Must define and use the following two constants inside the main() method.

```
final int LUCKY_NUMBER = 5;
final int EVEN_NUMBER_BASE = 2;
```

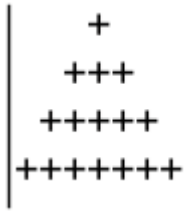
Sample run

```
Enter an integer or Q to quit: 7
Enter an integer or Q to quit: 6
Enter an integer or Q to quit: 5
Enter an integer or Q to quit: 4
Enter an integer or Q to quit: 3
Enter an integer or Q to quit: 2
Enter an integer or Q to quit: 1
Enter an integer or Q to quit: x
The sum of all odd numbers: 16.
The smallest number: 1.
5 is my lucky number!
The count of positive numbers: 7.
```

Codecheck link for 7B

Problem 7C

Write an application called **`Triangle`** to input an integer between 3 and 10 as the number of rows, then draw an isosceles triangle of the given number of rows using plus sign (+). For example, if the input integer is 4, there will be four rows and the output will look like this - with the last row starting at the left edge:



You can assume the input is an integer but must check the range.

You must use nested for loops to draw the triangle. You cannot use if statements to check the integer value then draw the triangle accordingly.

You must use one Scanner object and call Scanner methods to process input. You cannot use try-catch statement or call any parse methods.

Numbers 2, 3, and 10 are not considered magic numbers for this problem.

Sample run

```
Enter an integer between 3 and 10 (inclusive): -1
Out of range: -1.
Enter an integer between 3 and 10 (inclusive): 0
Out of range: 0.
Enter an integer between 3 and 10 (inclusive): 1
Out of range: 1.
Enter an integer between 3 and 10 (inclusive): 2
Out of range: 2.
Enter an integer between 3 and 10 (inclusive): 3
  +
 +++
+++++
```

Codecheck link for 7C