CS101- Algorithms and Programming I

Lab 04

Lab Objectives: `while` loops

- For all labs in CS 101, your solutions must conform to these CS101 style guidelines (rules!)
- Create a Lab04 workspace (i.e. the folder H:\private\cs101\lab04). This assignment has parts a, b, and c, each of which should be placed in a separate project within the same Lab04 workspace. Note: only one project is active at a time. To work (Build/Run) a different project, right click on the project's name and select "Set as active project".
- You can only use `while` statements for this lab assignment. You cannot use any other repetition statements. You are not allowed to use arrays or any other data structures to store user input.

---

a. Create a new project Lab04a. Write a program that takes an integer, `n`, from the user and calculates the largest power of `2` less than `n`. Display the result on the screen. For example, if the user entered n = 64 the computer should give the result as 32, because
the largest power of 2 less than 64 is $2^5 = 32$.
You should validate the user input as shown in the sample runs.
Note: You are not allowed to use Math class functions to solve this problem.

Sample runs:

```
> run Lab04a
Enter a positive integer n >= 2:  1
The input n should be positive > 1. Exiting...
> run Lab04a
Enter a positive integer n >= 2:  0.5
The input n should be an integer. Exiting...
> run Lab04a
Enter a positive integer n >= 2:  64
The largest power of 2 less then 64 is 32 (= 2 ^ 5)
> run Lab04a
Enter a positive integer n >= 2:  65
The largest power of 2 less then 65 is 64 (= 2 ^ 6)
> run Lab04a
Enter a positive integer n >= 2:  250
The largest power of 2 less then 250 is 128 (= 2 ^ 7)
> run Lab04a
Enter a positive integer n >= 2:  257
The largest power of 2 less then 257 is 256 (= 2 ^ 8)
>
```

b. Create a new project Lab04b. Write a guessing game between the user (player) and the computer. The program should generate a random integer between 1 and 100. The user should be prompted to guess the number, or enter 0 to quit the game. Display a message to the player according to the following:

- If the player's guess is within 20 (but more than 10) of the correct number, display "You are getting warmer."
- If the player's guess is within 10 or less of the correct number, display "You are hot!"
- If the player's guess is more than 20 away from the correct number, display "You are cold…"

The game should keep inputting guesses until either the player quits or guesses correctly. You should validate the player's guess, and prompt for new value when invalid input is entered. When the game finishes display an appropriate message to the player. See sample runs below.

Note: You can use Math.random() method which generates a random double value between 0 and 1. You will have to do some calculations to convert the number to an int between 1 and 100.

Sample runs:

```
> run Lab04b
Enter an integer between 1 and 100 (0 to quit): 0
Thanks for playing....
> run Lab04b
Enter an integer between 1 and 100 (0 to quit): -2
Invalid input...Enter a number between 1 and 100 (0 to quit): -8
Invalid input...Enter a number between 1 and 100 (0 to quit): 4
You are getting warmer.
Enter a number between 1 and 100 (0 to quit): 0
Thanks for playing....
> run Lab04b
Enter an integer between 1 and 100 (0 to quit): 50
You are getting colder...
Enter a number between 1 and 100 (0 to quit): 20
You are getting warmer.
Enter a number between 1 and 100 (0 to quit): 10
You are getting hot!
Enter a number between 1 and 100 (0 to quit): 8
You are getting hot!
Enter a number between 1 and 100 (0 to quit): 4
You are getting hot!
Enter a number between 1 and 100 (0 to quit): 2
Great! The number was 2
You guessed it in 6 tries!
>
```

c. Create a new project Lab04c.

1. Write a program that inputs a sequence of integers from the user until the user enters an input which is not an integer and does the followings:
    - Display all adjacent duplicates. For example, if the input is 3 3 5 5 5 4 7 7 4 6 2 2 3 3 3 3 8 0.5, the program should display “Duplicates: 3 5 7 2 3”.
    - Display the maximum and minimum of the inputs.
    - Display the sum and average of the inputs.
    - Display the count of the inputs and the number of even inputs.

Note: *Do not store all values read from the user; simply process each one as it is read!*

Note: You are not allowed to use Math class methods to solve this problem.

Sample runs:

```
> run Lab04c
Enter a sequence of integers (not-integer to process input): z
No values entered.
> run Lab04c
Enter a sequence of integers (not-integer to process input): QQ
No values entered.
> run Lab04c
Enter a sequence of integers (not-integer to process input): 335554 77 46223333 80.5
Duplicates: 3 5 7 2 3
***************
Max: 8 Min: 2
Sum: 73 Average: 4.294117647058823
Count: 17 Even count: 6
> run Lab04c
Enter a sequence of integers (not-integer to process input): 4842X
No adjacent duplicates...
***************
Max: 8 Min: 2
Sum: 18 Average: 4.5
Count: 4 Even count: 4
>
```

2. In the same folder as your .class file, create a text file, "testdata.txt", containing some test data, e.g. the values 3 3 5 5 5 4 7 7 4 6 2 2 3 3 3 3 8 X, space separated or one value per line.

   Open a command prompt and navigate to the folder containing your class file. From the command prompt, run your program by typing the command "java Lab04c < testdata.txt", which should cause your program to read its input from the specified file, rather than the keyboard. It will still output its results to the command prompt. *Note: you can have your program read data from one file and send the output to another file --as you did in the previous lab assignments-- by*

typing "java Lab04c < testdata.txt > results.txt" Try it, then open "results.txt" to see what happened!

Download and save realdata.txt text file, then run your program reading input from the given file `realdata.txt`.

Sample runs:

```
adayanik@ADAY-NB2:$ java Lab04c
Enter a sequence of integers (not-integer to process input): 0 0 3 2 2 2 1 1 4 5 5 Q
Duplicates: 0 2 1 5
***************
Max: 5 Min: 0
Sum: 25 Average: 2.272727272727273
Count: 11 Even count: 6
adayanik@ADAY-NB2:$
adayanik@ADAY-NB2:$
adayanik@ADAY-NB2:$ java Lab04c < realdata.txt
Enter a sequence of integers (not-integer to process input): Duplicates: 33 78 65
***************
Max: 99 Min: 0
Sum: 6943 Average: 48.21527777777778
Count: 144 Even count: 65
adayanik@ADAY-NB2:$
```

Note: The last sample runs were taken on Ubuntu. Command window on Windows or Mac will look slightly different.