School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
ELEN2004/ELEN2020 - Software Development I

Lab 2 - Loops

Lab 2 consists of 2 tasks. Each week, one task must be completed and submitted (refer to the submission schedule under Assignments on Ulwazi.) Note that the following test and submission instructions apply for all tasks.

Testing your code

You should refer to the provided set of inputs and outputs listed in the lab brief to determine if your code is running correctly. Given the set of inputs your code should be able to reproduce the output provided. Make sure to test it with additional values not in the lab brief to ensure it is working correctly.

Submission instructions

Your source code (.cpp) must be submitted to the *Lab 2 Task X* link under *Assignments* on Ulwazi (where **X** is the task number). **Your submission must include your source code and flow chart in PDF or image file**. Zip only the source code and flow chart. Name your zipped file using the format ***FirstnameLastname_Studentnumber.zip***.

It is important that you follow **good programming practices** in your code. This includes

- indent code correctly (use the AStyle formatting tool in Code::Blocks),
- add necessary comments and choose meaningful names of variables in your code,
- add the date and author at the top of your source code. Add the following comment at the start of each source code file.

```
1 // Student number:
2 // Date:
```

Task 1 - Random n sided dice, number generator

A dice is an n-sided shape with the sides labelled by numbers 1, 2, ..., n. The most common dice used to play various games is the 6-sided dice. You can however have a dice that has many more sides. Write a C++ program that rolls multiple dice $d = 1, 2, \ldots 9$, outputs each dice's roll and the sum of all the rolled dice. At each turn when the dice are rolled, all the dice have the same number of sides n. Your code must do the following

- read in a number **d-n** from a file named **input.txt**, which represents the number of dice d and sides n each dice has, then
- generate a random number that would be found on each dice. Write the corresponding numbers from each dice, as well as the sum of all the numbers on each dice to a file named **output.txt**.

First draw the flow chart and then attempt the code.

Example 1

Examples of the input and output files can be seen below in Listings 1 and 2

Listing 1: Example of **input.txt**

```
1-6
3-10
2-8
8-4
1-12
5-20
```

In the first line in listing 1, one 6-sided dice is rolled, while in the second line, three 10-sided dice are rolled.

Listing 2: Example of **output.txt**

```
6-6
4-10-1-15
5-1-6
2-3-1-4-5-2-1-1-19
11-11
17-2-1-16-3-39
```

In the first line in listing 2, one dice rolled a six. In the second line, three dice rolled '4', '10' and '1' respectively. The sum of the three outcomes is '15', hence the '4-10-1-15'.

Example 2

A second example of the input and output files can be seen below in Listings 3 and 4

Listing 3: Example of **input.txt**

```
1-6
2-4
2-4
1-6
```

Listing 4: Example of **output.txt**

```
1-1
1-3-4
1-4-5
1-1
```

NOTE: your output may differ from the outputs given in the values that appear on each line. This is because your random number generator may result in a different plausible value being selected. You must ensure that the value in the output file, x, falls within $1 \leq x \leq n$ for each dice.

Hint: you will need to use the **rand()** function. It is a good idea to seed the random number generator. Refer to your textbook or the internet for more information regarding this.

Task 2 - Right angled triangle

Write a C++ program that

- reads in one positive integer value, x, from a file called **input.txt** and
- writes to a file called **output.txt** a right-angled isosceles triangle of base length x, made of **'asterisk' (*), 'single space' and 'carriage return' characters only**. The hypotenuse in this case must face the *left*, with its equal sides on the *right* and *bottom*.

Your code should be able to draw a right angled triangle for any given positive integer.

First draw a flow chart and then attempt the code.

Example 1

For example your input file may contain the value 4:

Listing 5: Example of **input.txt**

```
4
```

Your output file, **output.txt**, for the given **input.txt**, should appear as follows:

Listing 6: Example of **output.txt**

```
      *
    * *
  *   *
* * * *
```

NOTE: Consecutive stars are separated by a single space. For clarity of where the spaces appear below is the solution with all spaces replaced by a '#' symbol. **There may be additional whitespace characters beyond the right side of the triangle.**

Listing 7: Each **#** represents the position of a single space character

```
1 #######*#
2 #######*#
3 ##*####*#
4 *#*#*#*#
```

Example 2

A second example is given below:

Listing 8: Example of **input.txt**

```
1 3
```

Your output file, **output.txt**, for the given **input.txt**, should appear as follows:

Listing 9: Example of **output.txt**

```
1     *
2   * *
3 * * *
```

NOTE: Consecutive stars are separated by a single space. For clarity of where the spaces appear below is the solution with all spaces replaced by a '#' symbol. **There may be additional whitespace characters beyond the right side of the triangle.**

Listing 10: Each **#** represents the position of a single space character

```
1 #####*#
2 ##*#*#
3 *#*#*#
```