

CSE 1141 - COMPUTER PROGRAMMING I
Programming Assignment # 5
DUE DATE: 14/12/2018 - 23:00 (No extension)

In this assignment, you will write the following programs using arrays.

1. Write a program which takes a number (**N**) from user, and rolls 10 fair six-sided dice, **N** times. To roll a single dice, you can simply generate a random number between 1 and 6. Use an array to tabulate the number of times each possible total (between 10 and 60) occurs. For example, if the number is 1 for all ten dices, the total of them will be 10. Then print out a text histogram of the results, as illustrated below. A histogram is a graphical representation of a statistical distribution that shows how many elements fall into a set of values.

Example Run:

Welcome to dice simulator program.

Enter a value: 1000

```
10:
11:
12:
13:
14:
15:
16:
17:
18: *
19: ****
20:
21: ***
22: *****
23: *********
24: *****************
25: *****************
26: *****
27: *****
28: *****
29: *****
30: *****
31: *****
32: *****
33: *****
    *****
34: *****
35: *****
36: *****
    *****
37: *****
38: *****
39: *****
40: *****
41: *****
42: *****
43: *****
44: *****
```

```

45: *****
46: *****
47: *****
48: ***
49: **
50:
51:
52: *
53:
54:
55:
56:
57:
58:
59:
60:

```

2. In the second question, you will write a program which have multiple stages. In the first part, you need to take a string from the user which shows the content of a 2D array. This string should contain a set of numbers separated by dashes (-), and each number is used for a binary representation of a row. Then, generated binary digits are located to the corresponding row of your 2D array as in the example below.

Example:

If the user enters a string such as “4-1-7”, then you should print the following grid:

```

-----
| 1 | 0 | 0 |
-----
| 0 | 0 | 1 |
-----
| 1 | 1 | 1 |
-----

```

- The grid above shows the initial content of your table. You should create another 2D array which shows the updated version of your original table, one step ahead.
- The cells in the second array should contain the values of ‘0’ or ‘1’ by looking its eight neighbors (*left, right, up, down, and diagonal* neighbors).
- The rules to give a value to a cell listed below:
 - If a cell contains the value of ‘1’ in the first array, there are three rules to update its value in the second array:
 - If a cell has fewer than two ‘1’ neighbor cells, it is converted to ‘0’.
 - If a cell has more than three ‘1’ neighbor cells, it is converted to ‘0’.
 - If a cell has exactly two or three ‘1’ neighbor cells, it remains as ‘1’.
 - If a cell contains the value of ‘0’ in the first array, the following rule applies to update its value in the second array:
 - If a cell has exactly three ‘1’ neighbor cells, it is converted to ‘1’.
 - Otherwise, it remains as ‘0’.

- Each of these rules should be applied to the second array in the next step.

Example:

```
-----
| 1 | 0 | 0 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |
-----
```

It is converted to the following table after 1st step:

```
-----
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
-----
```

Therefore, your program should take the number of steps in the beginning, and should show the content of your second array after each step. In the end, your program should print the decimal value of each row by using dashes (-) at the end of final step.

Example Run 1:

Welcome to our program.

Enter a string: 5-2-7

Enter a number of steps: 3

```
-----
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |
-----
```

It is converted to the following table after 1st step:

```
-----
| 0 | 1 | 0 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |
-----
```

It is converted to the following table after 2nd step:

```
-----
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
-----
```

It is converted to the following table after 3rd step:

0	0	0	0
0	1	0	0
0	1	0	0

The decimal value for the second table after 3 steps: 0-2-2.

Example Run 2:

Welcome to our program.

Enter a string: 5-2-16

The number 16 cannot be represented with 3x3 array!

Enter a string: 10-5-15-0

Enter a number of steps: 2

1	0	1	0
0	1	0	1
1	1	1	1
0	0	0	0

It is converted to the following table after 1st step:

0	1	1	0
0	0	0	1
1	1	0	1
0	1	1	0

It is converted to the following table after 2nd step:

0	0	1	0
1	0	0	1
1	1	0	1
1	1	1	0

The decimal value for the second table after 2 steps: 2-9-13-14.

Rules for the second question:

- a. The size of your grid should be determined based on the given string. You should not ask the number of rows or columns to the user.
- b. You are required to create exactly **two** 2D arrays (*Please do not create more 2D arrays!*). You should copy the content of your second array to the first array after each step. Then, you should fill the second array by examining the first array, again.
- c. You are required to implement the problem by using minimum of three methods (More methods are also acceptable.)
- d. One of your methods should be the **convertDectoB** method which takes a decimal value and the size of binary representation, then it should return a single dimensional array filled with the binary representation.

```
public static int[] convertDectoB(int number, int size)
```

For example, if the number is 1, and the size is 3, then you should return an array with the following content: {0,0,1}.

- e. You should handle necessary error checking cases. For example, the user cannot enter a string such as “**4-1-16**”, since the value of 16 cannot be represented with 3 fields in binary notation (1000₂). In such a case, print an appropriate error message.
- f. You are not allowed to use ready methods (library methods) to convert a decimal value to binary, and vice versa.
- g. Your program should run with any size of string, and any size of 2D arrays.

Submission Instructions:

Please zip and submit your files using filename YourNumberHW5.zip (ex: 150713852HW5.zip) to Canvas system (under Assignments tab). Your zip file should contain the following 4 files:

1. Java source code for Q1 (Pro5_1_150713852.java)
2. Java class file for Q1 (Pro5_1_150713852.class)
3. Java source code for Q2 (Pro5_2_150713852.java)
4. Java class file for Q2 (Pro5_2_150713852.class)

Submission Notes:

1. Write a comment at the beginning of your program to explain the purpose of the program.
2. Write your name and student ID as a comment.
3. Include necessary comments to explain your actions.
4. Select meaningful names for your variables and class name.
5. You are allowed to use the materials that you have learned in lectures & labs.
6. Do not use things that you did not learn in the course.

7. **Program submissions** should be done through the Canvas class page, under the assignments tab. Do not send program submissions through e-mail. E-mail attachments will not be accepted as valid submissions.
8. You are responsible for making sure you are turning in the right file, and that it is not corrupted in anyway. We will not allow resubmissions if you turn in the wrong file, even if you can prove that you have not modified the file after the deadline.
9. In case of any form of **copying and cheating** on solutions, all parts will get **ZERO** grade. You should submit your own work. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties.
All types of plagiarism will result in zero grade from the homework.
10. No late submission will be accepted.

Grading:

Question 1 → 30 points

- Taking an input (5 points)
- Simulating rolling of 10 dices N times (15 points)
- Printing histogram (10 points)

Question 2 → 60 points

- Taking an input string and constructing 1st array (10 points)
- Generating the second array based on the given rules (15 points)
- Showing the content of second array for N steps (10 points)
- Use of only two 2D arrays (10 points)
- Necessary error checking (5 points)
- Use of minimum three methods (10 points)

Submission Format → 10 points

- 2 java files + 2 class files
- Comments are necessary
- Make sure that your class files can be executed on another computer.