CSE 1141 - COMPUTER PROGRAMMING I
Programming Assignment # 5
DUE DATE: 17/12/2017 - 23:59 (No extension)

Q1. In this question, you will write a program that grades multiple-choice tests. Assume that there are 40 questions, and the answer key will be given by the user. Your program will take the answers of each student as a single string, and it will grade the exam based on the given key. Your program should print the total number of correct and incorrect answers separately for each student.

There are several constraints regarding the program:

i. There might be 4 different cases for answers (“A”, “B”, “C”, “D”).

ii. A student can leave a question blank, in which case his/her answer will be given with a dash ("-").

iii. Your program should also provide detailed information about the student’s score. Suppose that there are four different types of subjects (or chapters). These are represented with S1, S2, S3, and S4. The question numbers for each subject are the followings:
   - The question numbers for subject 1 (S1) are: 1, 5, 9, 13, …. 37.
   - The question numbers for subject 2 (S2) are: 2, 6, 10, 14, …. 38.
   - The question numbers for subject 3 (S3) are: 3, 7, 11, 15, …. 39.
   - The question numbers for subject 4 (S4) are: 4, 8, 12, 16, …. 40.

iv. Your program should display the percentage of success of each student in each subject.
   - Example: Suppose that student Leyla has the following sub-totals:
     - She has 8 correct and 2 incorrect answers for S1.
     - She has 7 correct and 3 incorrect answers for S2.
     - She has 5 correct and 5 incorrect answers for S3.
     - She has 2 correct and 8 incorrect answers for S4.
   - Then, your program should display the following output:

```
Leyla has 22 correct and 18 incorrect answers in total.
Her percentage of success is:
S1        S2        S3        S4
---       ---       ---       ---
80%       70%       50%       20%
```

- Your program should take the answer key first, then it should ask the student’s name and his/her answers. Then it should print the output for that student. This process should continue repeatedly. If the user enters 0, that means the program ends. It should be noted that the answer key will be given once, for each execution.

- Error Checking:
  - If the answers key or the students’ answers have less than 40 entries, then give an appropriate error message.

```
Illegal Input!
```

Here are some execution examples:

Execution 1:

```
Hello Teacher!
Please enter the answer key:
AABBAACCCCBABADDAAAADDDDAACCAAAABBBCCDCC

Please enter the name of student: Leyla
Please enter the answers of Leyla:
ABBBACCDCDA-BAACAAC-DDBAAABDCAABBBBA-DCC


Leyla has 22 correct and 18 incorrect answers in total.
The percentage of success is:
S1        S2        S3        S4
---       ---       ---       ---
80%       70%       50%       20%

Please enter the name of student: Ali
Please enter the answers of Ali:
DBAB-ACCBABADCDDBBAADDDD-A-CCDAAABCCCDBC


Ali has 23 correct and 17 incorrect answers in total.
The percentage of success is:
S1        S2        S3        S4
---       ---       ---       ---
20%       50%       60%       100%


Please enter the name of student: 0
Program ends. Bye :)
```

Execution 2:

```
Hello Teacher!
Please enter the answer key:
CCBBAACCBADABBCCBDDAABCDDCBACBADBCCADDBD

Please enter the name of student: Ahmet
Please enter the answers of Ahmet:
AB---ACC-CDA-BAA-
Illegal Input!

Please enter the name of student: Ahmet
Please enter the answers of Ahmet:
AB---ACCCDA-BAA-AAC-DDB-AAB-CAA-BBB--BBD
```

```
Ahmet has 10 correct and 30 incorrect answers in total.
The percentage of success is:

S1          S2          S3          S4
---         ---         ---         ---
30%         10%         40%         20%

Please enter the name of student: 0
Program ends. Bye :)
```

Q2. In this question, you will write a program that tests whether a two-dimensional list has seven-length string "CSE1141", either horizontally, vertically, or diagonally (forward→ or backward← and up↑ or down↓). Your program should prompt the user to enter the number of rows and columns of the list and then you will generate a two-dimensional list with random letters between 'A' to 'Z' and '0' to '9'. Your program then displays **True** if the list contains "CSE1141", either horizontally, vertically, or diagonally (forward→ or backward← and up↑ or down↓). Otherwise, it will display False. Here are some examples of the true cases:

F	4	M	G	I	G	P
5	5	1	C	V	0	1
Z	G	S	K	U	T	9
6	5	6	1	1	9	1
K	3	R	1	V	H	7
C	S	E	1	1	4	1
5	7	0	3	0	3	8

C	4	M	G	I	G	P
S	5	1	C	V	0	1
E	X	D	0	R	7	8
1	5	6	1	1	9	1
1	3	R	1	V	H	7
4	S	E	1	1	4	1
1	1	O	P	T	R	E

C	4	M	G	I	G	P
S	S	1	C	V	0	1
E	X	E	0	R	7	8
1	5	6	1	1	9	1
1	3	R	1	1	H	7
4	S	E	1	1	4	1
1	1	O	P	T	R	1

It should be noted that you do not need to highlight the true case with yellow. We just add them to read it easily. Please do not forget to leave a space (' ') after printing each item in the list.

Here are some execution examples:

Execution 1:

```
Please enter the number of rows: 10
Please enter the number of columns: 15

S J A 5 P 0 C 2 G Y W 0 P H S
7 7 E X 3 4 5 W 1 M O 6 3 4 2
6 1 7 6 6 C S E 1 1 4 1 2 U Y
7 G 1 W 9 E V 7 K 4 P 0 A Y O
X 5 Z 9 H 3 V 0 L 3 S R 1 6 J
E Z 2 2 6 4 O E P L X P Z 7 0
5 5 5 G 0 N 6 9 2 P F 9 Y H Y
H H S E T H 1 J 8 7 8 P 9 2 8
2 L K C 1 M 9 G T X U 2 W 0 8
Z T B U 8 5 7 S 7 N 9 S X P O

TRUE
```

Execution 2:

```
Please enter the number of rows: 7
Please enter the number of columns: 10

 1 6 3 3 2 2 X 9 I 3
 9 0 Z W Y 3 0 B Z K
 3 6 6 3 4 B 0 8 L D
 3 T V 2 5 6 1 L 2 0
 0 O Z P W X Q W Y L
 P K V 7 O 2 3 6 2 8
 F 2 L L 7 O 8 M S X
```

FALSE

Execution 3:

```
Please enter the number of rows: 9
Please enter the number of columns: 8

 F 6 5 V 9 Q 1 4
 P G 0 E 1 3 O 3
 N 0 7 I 4 2 7 7
 6 3 7 Q 1 Q 9 8
 0 6 5 3 1 5 2 8
 Z 0 P 8 E H I 8
 Z 4 6 P S J C E
 O 1 U 3 C 8 B L
 3 Y C L L H 1 F
```

TRUE

Execution 4:

```
Please enter the number of rows: 14
Please enter the number of columns: 14

 8 O A D 0 7 4 7 9 U L 2 1 Z
 0 A K 2 J G 6 9 4 1 7 I 1 2
 2 F 3 8 G 7 M R 8 T 3 9 H 0
 Y 6 3 N I 7 I 9 F J 8 S I R
 J W 8 W F 6 H E H O U Q 9 8
 1 E T 1 0 6 U 2 0 7 U 4 4 2
 9 2 C 5 N 6 X 7 Z 0 3 L U V
 Z 5 O 3 D 4 1 B 3 8 S 6 B C
 M 3 U F 3 4 7 S 4 F S R M L
 2 1 9 7 1 6 M M 3 0 1 Y C 2
 8 7 8 1 7 E 9 3 L F 7 2 Z J
 1 6 E Y S D 8 R 5 X 9 0 F N
 6 S 6 X 6 8 2 B 9 9 0 O 3 7
 C 8 K 7 0 0 C R 9 5 V 2 K 9
```

TRUE

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)

Notes:

1. Write a comment at the beginning of each 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 names.
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. 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.
8. No late submission will be accepted.

Grading:

Question 1 → 45 points

- Taking the answer key (5 points)
- Taking the student's name and the answers (5 points)
- The number of correct/incorrect answers (8 points)
- Printing the percentage of success (16 points)
- Loop termination condition - Entering 0 control (8 points)
- Error checking (3 points)

Question 2 → 45 points

- Taking inputs (5 points)
- Creating a table of row-by-column values (15 points)
- Correct execution for test inputs (25 points)
  - Finding "CSE1141" keyword either
    - horizontally, forward→ or backward← (7 points),
    - vertically, up↑ or down ↓ (7 points),

        - diagonally forward→ or backward← (7 points),
    - False case (4 points).

Submission Format → 10 points

- 2 java files + 2 class files
- Make sure that your class files can be executed on another computer.
- Make sure that the input/output of your program must be the same with the examples above (all informative strings & spaces). Otherwise, auto-grader cannot grade your code!!!
- Comments are necessary!