Northeastern Illinois University
CS200-1, Programming I, Summer 2017
Homework 8
Due date: Thursday 7/27/2017 at 1:00 p.m.

<u>Problem 1</u>:

A class of 12 students is given a multiple choices exam of 10 questions where each question has 5 choices A-D-C-D-E, after grading the exams, the student's answers are stored in a 12 by 10 **`char`** array named **`answers`** as follows:

A	D	B	A	B	E	A	D	E	B
B	C	E	C	E	E	E	D	E	B
A	B	D	E	A	D	C	D	A	B
B	B	D	E	B	E	C	C	A	D
A	B	D	E	C	A	B	D	E	C
A	B	D	E	D	C	E	B	C	E
B	D	E	C	A	D	B	E	B	B
C	D	D	C	B	E	A	D	E	B
B	D	E	A	B	D	E	A	C	D
B	E	E	A	A	E	A	B	E	B
B	D	E	A	B	D	E	D	E	B
B	D	E	C	B	E	B	D	C	E

Each row of the above array represents the answers of one student.

The key (correct answers) for the exam is also represented by a **char** array named **key** as follows:

B	D	E	A	B	E	E	D	E	B

Create a class named Problem1, the program would do the following:

- The program should have a **`main`** method, for now just leave it empty.

- Create a method named **`numberOfStudentsPassed`** which takes a **`2D char`** array named **`answers`**, and a **`1D char`** named **`key`**.

- The method returns an **`integer`** value.

- The method finds and returns the number of students who passed the exam.

- A student passes the exam if he/she gets **AT LEAST 6 ANSWERS CORRECT, in other words, at least 6 answers which matches the key**.

- Go back to the main method and make the proper call to your **`numberOfStudentsPassed`** method.

- If you run your program with the above **`answers`** and **`key`** arrays, your method should return the number **`7`** which is the number of students who passed the exam.

- Copy the output to a text file named Problem1.txt

Problem 2:

- What is the **exact** output for the program below?
- Print and use the tracing worksheet provided on the next page. You are required to trace the program by hand in order to get credit for the question. Show your work as well as the output on the tracing worksheet.
- Scan the tracing worksheet, then save it as **.pdf** file.
- Your output should go in the output box provided on the next page.

```
public class Tracing
{
   public static void main(String [] args)
   {
      int [][] scores = {{5, 7, 3, 4, 1},
                         {12, 8, 73, 23, 6}};

      int [][] data = {{5,   7,  3,   4,  1},
                       {12,  8,  73, 23,  6},
                       {436, 94, 99, 17, 70}};
      whatsTheOutputA(scores);
      whatsTheOutputB(data);
      whatsTheOutputA(scores);
      System.out.println("Done!!!");
   }
   public static void whatsTheOutputA(int[][] a)
   {
      int i, j;

      for(i = 0; i <= a.length - a.length; i++)
      {
         for(j = i; j < a[i].length - 1; j++)
         {
            a[i][a[i].length - 1] = a[i][j] + a[i][a[i].length - 1];
            System.out.print(a[i][a[i].length - 1] + "  ");
         }
         System.out.println();
      }
   }
   public static void whatsTheOutputB(int[][] a)
   {
      for(int i = 0; i < a.length; i++)
      {
         for(int j = 0; j < a[i].length; j++)
         {
            if(i == j)
            {
               System.out.print(a[i][j] + "  ");
            }
         }
      }
      System.out.println();
   }
}
```

Tracing Worksheet

Output:

Memory box:

Problem 3:

Create a class named Problem3, the program would do the following

- The program should have a **main** method, for now just leave it empty.

- Create a method named **linearize**, which takes a **2D integer** array **arr** as a parameter and returns a **1D integer** array.

- The method should copy all the elements of the **2D** array **arr** into a new **1D** array.

- The elements are to be copied row by row, beginning with the first element in the first row and proceeding to the last element on the first row, then moving to the second row and copying the elements beginning with the first element in the second row to the last element in the second row, and so on until all elements have been copied. Once all elements have been copied, the method is to return the new **1-dimensional** array.

- The method should accommodate any **2D** array (including ragged arrays).

- Go back to the main method and make the proper calls to **linearize** method, and test it with the integer arrays shown in the sample input/return below.

- Copy the output to a text file named Problem3.txt

Argument array arr	**Return array**

8 9 31
7 23 73 24
13 19
	8 9 31 7 23 73 24 13 19

32 13 45
12 100 56 94
19
32 120
	32 13 45 12 100 56 94 19 32 120

Problem 4:

Create a class named Problem4, the program would do the following

- The program should have a **`main`** method, for now just leave it empty.
- Create a method named **`mirror`**, which takes a **`2D integer`** array **x** as a parameter and returns a **`2D integer`** array.
- The method should return a **`2D integer`** array with the every column of the original array **x** reversed.
- Assume that the array **x** is **NOT** a ragged array.
- Go back to the main method and make the proper calls to **`mirror`** method, and test it with the integer arrays shown in the sample input/return below.
- Copy the output to a text file named Problem4.txt

<table>
<tr><th>Argument array x</th><th>Return array</th></tr>
<tr>
<td>
<table>
<tr><td>8</td><td>9</td><td>31</td><td>4</td><td>29</td><td>12</td></tr>
<tr><td>7</td><td>23</td><td>73</td><td>12</td><td>6</td><td>30</td></tr>
<tr><td>13</td><td>42</td><td>42</td><td>18</td><td>5</td><td>15</td></tr>
<tr><td>4</td><td>7</td><td>32</td><td>43</td><td>15</td><td>17</td></tr>
</table>
</td>
<td>
<table>
<tr><td>4</td><td>7</td><td>32</td><td>43</td><td>15</td><td>17</td></tr>
<tr><td>13</td><td>42</td><td>42</td><td>18</td><td>5</td><td>15</td></tr>
<tr><td>7</td><td>23</td><td>73</td><td>12</td><td>6</td><td>30</td></tr>
<tr><td>8</td><td>9</td><td>31</td><td>4</td><td>29</td><td>12</td></tr>
</table>
</td>
</tr>
<tr>
<td>
<table>
<tr><td>32</td><td>45</td><td>67</td><td>21</td><td>3</td><td>8</td></tr>
<tr><td>12</td><td>65</td><td>23</td><td>27</td><td>12</td><td>18</td></tr>
<tr><td>19</td><td>32</td><td>14</td><td>18</td><td>19</td><td>27</td></tr>
<tr><td>39</td><td>48</td><td>29</td><td>26</td><td>32</td><td>44</td></tr>
<tr><td>17</td><td>28</td><td>37</td><td>38</td><td>22</td><td>19</td></tr>
</table>
</td>
<td>
<table>
<tr><td>17</td><td>28</td><td>37</td><td>38</td><td>22</td><td>19</td></tr>
<tr><td>39</td><td>48</td><td>29</td><td>26</td><td>32</td><td>44</td></tr>
<tr><td>19</td><td>32</td><td>14</td><td>18</td><td>19</td><td>27</td></tr>
<tr><td>12</td><td>65</td><td>23</td><td>27</td><td>12</td><td>18</td></tr>
<tr><td>32</td><td>45</td><td>67</td><td>21</td><td>3</td><td>8</td></tr>
</table>
</td>
</tr>
</table>

Extra Credit (+20 points):

Problem 4 can be solved in two different ways; one way uses nested loops to reverse (flip) the columns.

The other way uses a single loop, if you can solve **Problem 4** **both ways**, you'll get 20 points extra.

General Instructions:

- No hard copies will be collected.

- Do not send your files through the email!

- You should submit your work by the due date, **No** extensions will be given. (See syllabus for late homework policy).

- DO **NOT** turn in multiple files, only one .zip file.

What to turn in:

There should be three .java file, three .txt file and one .pdf file, put all those files into a zip file and name it <YourFirstName_YourLastName>.zip, submit the zip file into the Dropbox on D2L.

How to zip multiple files?

On Windows: Select all the files > right click > Send to > Comprised File

On Mac: Select all the files > Click/Tap with two fingers > Compress Items