香港浸會大學
HONG KONG BAPTIST UNIVERSITY

COMP2026/COMP2045 Practical Test

Quiz Date: 22/10/2022, 1:30pm-3:30pm

Duration: 120 minutes

Instructions

- Create a new IntelliJ project, and name it as "XXXXXXXX_ptest1" where "XXXXXXXX" is your student ID. Save all your work in this project.
- For submission, zip the src folder, rename the zip file to "XXXXXXXX_ptest1.zip" where "XXXXXXXX" is your student ID. Submit the this zip file to Moodle.
- You should ensure that all your answers are included in the zip file you have submitted. No submissions would be accepted after the end of the Practical Test.
- The duration for this Practical Test is 2 hours.
- This is a open books, open notes test. **You are not supposed to make any communication with other people or open any webpage during the test. Therefore, it is considered as violating exam regulation if you are using a phone or open Chrome/Edge/Firefox.**
- Total marks: 100 (marking out of 80, with 20 marks as bonus)
- There are TWO PARTS in this Practical Test:
    - Part A (Regular questions) – 80 marks
    - Part B (Challenging questions) – 20 marks
- NOTE: Only those who score over 55 marks in Part A are qualified for Part B.
- All programs that you submit must contain comments with proper indentation.
- **You are NOT allowed to use any data structure that has not yet been taught in the course, e.g. ArrayList, Vector, HashMap, etc.** All methods of String class can be used. API that has been taught at Lab/Lecture (e.g. `Scanner`,`ThreadLocalRandom` can be used too.
- **You are allowed to add additional method**
- Follow the output of the program as given by the sample.
- **Warning: cheating in practical test fail the course directly and your case will be recorded by the University.**

Part A (80%)

Q1 Canvas (30%)

A data file `data.txt` or `data2.txt` stores the coordinates of pixels of an image. The content of `data.txt` is as follow:

```
8 10
1 5 *
2 5 *
7 4 +
4 9 +
2 4 +
3 5 o
4 4 o
```

It represents the following image.

```
------------
|          |
|     *    |
|    +*    |
|     o    |
|    o    +|
|          |
|          |
|    +     |
------------
```

The first line indicates the height (8) and the width (10) of an image. From second line onward, each line represent a coordinate (row, col) and the symbol that be printed at that coordinate. For example, the second line of the file is `1 5 *` which means row 1 (counted from top), column 5 (counted from left) is a `*` symbol.

You are given the skeleton code Canvas.java. Please complete the required method to finish the task. After you have completed the program, you can try to read data from `data2.txt` which will produce a nice image.

Q2 Slicing (30%)

We are going to cut an array into a smaller size array or some smaller size of arrays. Please work on the skeleton code Slicing.java.

There are three parts of the program.

- Part A, you will need to slice a 1D-integer array with a given starting index `start` and ending index `end`. The method `slice1DArray` should slice the input array and return a 1D-array. The method `printArray` shall print an 1D-array.
- Part B, you will need to slice a 2D-integer array with the given starting coordinate `(r1, c1)` and ending coordinate `(r2, c2)`. The method `slice2DArray` should slice the input array and return a 2D-array. The overloaded method `printArray` shall print a 2D-array
- Part C, you will need to evenly slice a 2D-integer array with the given height `height` and width `width` of the result array.It should produce some number of array slices. The method `slice3DArray` should slice the input array and return a 3D array, or an array of 2D array.

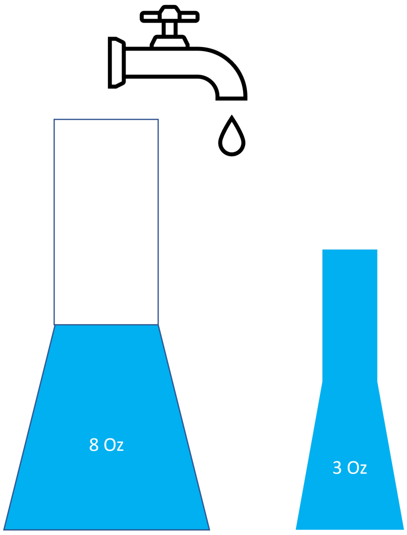
Note: you may assume that the input to these methods had already been validated and error free. For example, in part A, `start` is always smaller or equals to `end` and both variables are greater than 0 and smaller than `input.length`. In part C, variable `height` and `width` will always divide the input size, i.e. `array.length % height == 0` and `array[0].length % width == 0`.

The expected output is given below

```
Part A
Original Array looks like:
10 20 30 40 50 60
Slice from 1 to 4
20 30 40 50
Slice from 3 to 3
40


Part B
Original Array looks like:
11 12 13 14
21 22 23 24
31 32 33 34

Slice 2D array from (1, 0) to (2, 2)
21 22 23
31 32 33

Slice 2D array from (1, 1) to (1, 3)
22 23 24


Part C
Original Array looks like:
11 12 13 14 15 16
```

```
21 22 23 24 25 26
31 32 33 34 35 36
41 42 43 44 45 46
Slice the array into 2-by-3 arrays.
-----Slice 0-----
11 12 13
21 22 23
-----Slice 1-----
14 15 16
24 25 26
-----Slice 2-----
31 32 33
41 42 43
-----Slice 3-----
34 35 36
44 45 46
Slice the array into 2-by-6 arrays.
-----Slice 0-----
11 12 13 14 15 16
21 22 23 24 25 26
-----Slice 1-----
31 32 33 34 35 36
41 42 43 44 45 46
```

Q3 Flasks (20%)



You are required to pour water between two flasks. These flasks has different capacity and the water inside (volume) may not exceed these capacity. You can fully fill a flask, empty a flask and pouring water between two flasks. Study the program Flask.java and finish the program that gives the following output.

```
Step1
Flask 1:(0/8)
Flask 2:(0/3)

Step2
Flask 1:(8/8)

Step3
Flask 1:(5/8)
Flask 2:(3/3)

Step4
Flask 1:(3/8)
Flask 2:(0/3)

Step5
Flask 1:(2/8)
Flask 2:(3/3)

Do you know how to make a 1?
Sorry, this is not part of the question
```

Part B (20%) - eligible only if you obtain 55 or above in Part A.

Q4 Connect4(20%)

We modify the Connect4 game from your assignment so that it allows the players to remove his/her token from the game. To do that, the player will first type `@` which is a special command to remove a token. Then the player needs to enter the coordinates of his token in the format of `"row col"`. This command will be considered invalid if the coordinates do not contain the player's token and the player will need to enter a new command or place the token again. When the token is removed, all tokens above it will drop by one cell due to gravity.

This special move will allow a tie (i.e., both player connect4 at the same time) and suicide (let the opponent wins the game). This also implies changes of check and checkmate. Yet, to obtain full marks for this question, you only need to correctly implement

- the token removal;
- validating the input;
- correct the game over messages (in case of suicide);
- update check method (not checkmate).

You are allowed to change any part of your code. You are adviced to use your own `Connect4.java` program code to attempt this quiz.

Some sample out is as below.

```
 01234567
 --------
|        |
|        |
|        |
| 112    |
| 221    |
| 12212  |
 --------
Check!
Player 2, please enter a command. Press 'h' for help
@
Remove Token. Please enter the coordinate (row, col)
5 1
Wrong input!, please do again
Player 2, please enter a command. Press 'h' for help
@
Remove Token. Please enter the coordinate (row, col)
4 1
 01234567
 --------
|        |
|        |
|        |
|  12    |
| 121    |
| 12212  |
 --------
```

After a token at `(4, 1)` is removed, other tokens above it fall. Noted that a player can only remove his own token.

```
Player 1, please enter a command. Press 'h' for help
3
 01234567
 --------
|        |
|        |
|  11    |
|  21    |
|21221   |
|121122  |
 --------
Check!
Player 2, please enter a command. Press 'h' for help
@
Remove Token. Please enter the coordinate (row, col)
3 2
 01234567
 --------
|        |
|        |
|   1    |
|  11    |
|21221   |
|121122  |
 --------
Player 1 wins
```

In this example, it was a check to player 2 because player 1 can win by removing `(5,2)`. Player 2 decide to end the game by suicide, which connects player 1 in a diagonal.

```
 01234567
 --------
|        |
|        |
|    1   |
| 1  2   |
| 2221   |
| 1112   |
 --------
Check!
Player 2, please enter a command. Press 'h' for help
@ 5 4
Remove Token. Please enter the coordinate (row, col)
 01234567
 --------
|        |
|        |
|        |
| 1  1   |
| 2222   |
| 1111   |
 --------
Tie
```

This example shows an example of a tie game.