

IT-114 MID-SEM EXAM QUESTION PAPER

Q1. Create a Tic-Tac-Toe game, in which the human has to play against the computer, with the following features:

- 1.) It take inputs from the Standard input stream (keyboard),**
- 2) Game is always either won by the computer or the game goes to a draw, i.e. the computer never loses the game.**

	1	2	3
1	X	O	X
2	X	O	
3		O	

In the Tic-Tac-Toe board above, computer wins.

Input format:

You will take input from the keyboard. Each input will consist of two space-separated integers in the range of [1,3]. E.g. input "3 3" refers to the bottom-right cell. After human enters his choice in the console, you need to print the move taken by the CPU and then again human will enter his choice and so on human and CPU will play turn by turn.

Output format:

Your code should be made in a way that the only possible outputs are : "CPU Wins !!" or "Game Draw !!".

Example game play:

Player 1 Enter Your Choice:

1 1

CPU chose: 1 2

Player 1 Enter Your Choice:

1 3

CPU chose: 2 2

Player 1 Enter Your Choice:

2 1

CPU chose: 3 3

CPU Wins !!

Q2. A classroom has 50 students and 25 desks each having 2 students. Roll numbers 1-25 are strong students and roll numbers 26-50 are weak ones. There are 5 rows in the class room. Each student has a roll number assigned. It is the first day of the class. The teacher enters the classroom and finds that the students are not sitting according to any order. The teacher wants a fair distribution of the class according to merit. He decides on the following order:

- 1.) The highest roll numbered students should be sitting in the back rows (ascending order).**
- 2.) In one desk there should be a weak student and a strong student.**
- 3.) The student roll number difference at every desk is in non-descending order across desks (following a same directional scan order across the room).**
- 4.) The monitor should be in the middle of the classroom so that he can watch over the class. You can pick any one of the two possible students as a monitor (mention it in the documentation).**

TODO:

Part 1: Print the roll numbers and names of the students that are neighbors of the monitor (there can be 6 neighbors where diagonal neighbors are allowed only for front and back desks).

Part 2: The Instructor has the authority of checking/knowing, the neighbors of a given student. So when the instructor inputs the roll-number of a student, print the student's name and the neighbors of the student.

Instructor will be giving the input in the console itself.

Both parts are to be implemented.

Note: The student profile will be given in a text file in the format - <student name> <roll number> in each line.

Q3. Given a bracket sequence, consisting of these different types of brackets :

`"{", "}" , "(" , ")"" , "[" , "]"`

Write a program to check whether these brackets appear in correct sequence. By correct sequence we mean that for every opening bracket of a type there is a closing bracket of that same type.

For example, the program should print true for exp = "[()]{ }{[()>()] }]" and false for exp = "[()]"

You will not use any data-structure class from the java.util package.

If you need one, implement it. For instance, if you decide to use a linkedlist data structure, your class should have a public interface (i.e. method signatures) similar to the LinkedList class

<http://docs.oracle.com/javase/6/docs/api/java/util/LinkedList.html>. So if you need a data structure, implement that data structure and it should contain methods similar to those given in javadoc of that data structure class.