**CME1212 Algorithms and Programming II**

**Homework 3**

Upload your program from *Google Classroom* until **31 March 2019, 23:55.**

Write a Java program for a simple version of ***Yahtzee*** game.

The objective of *Yahtzee* is to get as many points as possible by rolling three dices.

Two people play this game. Each player has a queue.

The game consists of 15 turns. In each turn, a player throws three dices. The numbers must be inserted into his/her queue. After that, the player can gain a point if one of the following combinations exists in his/her queue:

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Description** | **Score** | **Example** |
| Yahtzee | You must have at least 4 of the same number. | 20 |  |
| Large straight | You must have at least 6 consecutive numbers | 10 |  |

If a combination occurs, the player gets the related point and these numbers must be deleted from the queue.

In one turn, a player can get points for multiple combinations.

The program must display all steps until the game is over.

At the end of the game, the winner should be displayed, if exists.

In your program, you can use queue as you want, but you must use only queues, don’t use other data structures.

You can use *linear queue* or *circular queue*.

Don’t use normal array.

Don’t use stack.

Don’t use a string variable.

Don’t use queue class embedded in Java. Write your own queue class.

Don’t take any input from the user !!!

**Sample output:**

**Turn: 1**

Player1: 2 4 2 score: 0

Player2: 1 3 1 score: 0

**Turn: 2**

Player1: 2 4 2 2 5 1 score: 0

Player2: 1 3 1 5 4 4 score: 0

**Turn: 3**

Player1: 2 4 2 2 5 1 4 2 2 score: 0

Player2: 1 3 1 5 4 4 1 3 3 score: 0

Player1: 4 5 1 4 2 score: 20

Player2: 1 3 1 5 4 4 1 3 3 score: 0

**Turn: 4**

Player1: 4 5 1 4 2 3 6 1 score: 20

Player2: 1 3 1 5 4 4 1 3 3 2 2 2 score: 0

Player1: 4 1 score: 30

Player2: 1 3 1 5 4 4 1 3 3 2 2 2 score: 0

**Turn: 5**

Player1: 4 1 5 5 6 score: 30

Player2: 1 3 1 5 4 4 1 3 3 2 2 2 1 2 1 score: 0

Player1: 4 1 5 5 6 score: 30

Player2: 3 5 4 4 3 3 1 score: 40

**Turn: 6**

Player1: 4 1 5 5 6 2 3 1 score: 30

Player2: 3 5 4 4 3 3 1 4 5 1 score: 40

Player1: 5 1 score: 40

Player2: 3 5 4 4 3 3 1 4 5 1 score: 40

**Turn: 7**

Player1: 5 1 5 5 5 score: 40

Player2: 3 5 4 4 3 3 1 4 5 1 2 5 2 score: 40

Player1: 1 score: 60

Player2: 3 5 4 4 3 3 1 4 5 1 2 5 2 score: 40

**Turn: 8**

...

**Turn: 15**

Player1: 6 2 4 4 score: 120

Player2: 3 4 3 1 4 1 2 5 5 score: 100

Game is over.

The winner is player 1.

**Notes**

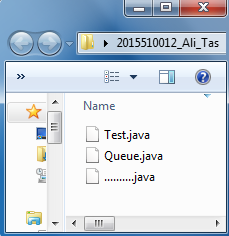
**1-** Upload format

**Step1:** Create a new folder, named by your student number and name (without any space)

For example: 2015510012\_Ali\_Tas



**Step2:** Copy all java files into this folder



**Step3:** Compress the folder **2015510012\_Ali\_Tas.zip**

**Step4:** Upload the file **2015510012\_Ali\_Tas.zip** from *google.classroom.com*

**2-** Don’t use ENIGMA or any other extra library.

**3.**If you are late, your grade will be decreased 10 points for each day. After five days, your assignment will not be accepted.

**4-** Assignment must be your individual work.

**Cheating** is strictly prohibited.

All source codes will be automatically compared with each other by using a program.

If any cheating occurs, your assignment will be graded with **zero (0)**.