CME1212 Algorithms and Programming II
Homework 2

Upload your program from *DEUZEM SAKAI* until **08 May 2022, 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 Single Linked List (SLL).

The game consists of 10 turns. In each turn, a player throws three dices. The numbers must be inserted into his/her SLL. After that, the player can gain a point if one of the following combinations exists in his/her SLL:

Category	**Description**	**Score**	**Example**
Yahtzee	You must have at least 4 of the same number.	10	
Large straight	You must have at least 6 consecutive numbers	30	

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

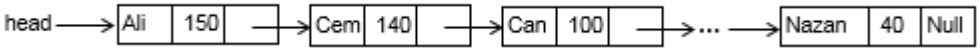
In one turn, a player can get points for multiple combinations.

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

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

High Score Table

Read the unsorted file "D:*HighScoreTable.txt*" and create a sorted Single Linked List as follows:




```
Pelin
80
Kaan
50
Ali
150
Yeliz
60
Cem
140
Can
100
Ece
90
Sibel
80
Remzi
70
Nazan
40
```

If the player earns a score within the top-10 results, he/she will be displayed in the High-Score table.
Add into SLL. If the same score exists in the table, the new score should be inserted to the next of them.
Delete the last record from the list.
The new score table should be written to the file ("D:*HighScoreTable.txt*").

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

This homework will be graded by Res.Asst.Dr. Göksu TÜYSÜZOĞLU.
You can ask your questions her from the "**FORUM → Homework 2- Questions**" part of the *DEUZEM SAKAI* software.

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: 10
Player2: 1 3 1 5 4 4 1 3 3 score: 0

Turn: 4
Player1: **4** **5** **1** 4 **2** **3** **6** 1 score: 10
Player2: 1 3 1 5 4 4 1 3 3 2 2 2 score: 0

Player1: 4 1 score: 40
Player2: 1 3 1 5 4 4 1 3 3 2 2 2 score: 0

Turn: 5
Player1: 4 1 5 5 6 score: 40
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: 40
Player2: 3 5 4 4 3 3 1 score: 20

Turn: 6
Player1: **4** **1** **5** 5 **6** **2** **3** 1 score: 40
Player2: 3 5 4 4 3 3 1 4 5 1 score: 20

Player1: 5 1 score: 70
Player2: 3 5 4 4 3 3 1 4 5 1 score: 20

Turn: 7
Player1: **5** 1 **5** **5** **5** score: 70
Player2: 3 5 4 4 3 3 1 4 5 1 2 5 2 score: 20

Player1: 1 score: 80
Player2: 3 5 4 4 3 3 1 4 5 1 2 5 2 score: 20

...

Turn: 10
Player1: 6 2 4 4 score: 120
Player2: 3 4 3 1 4 1 2 5 5 score: 50

Game is over.

The winner is player 1.

```
High Score Table
Ali 150
Cem 140
You 120
Can 100
...
Kaan 50
```

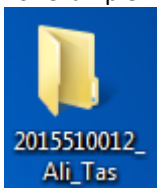
Notes

1- In your program, you can use Single Linked List (SLL) as you want, but you must use only SLL, don't use other data structures.

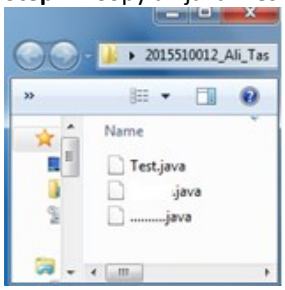
Don't use array or array list or list data structures embedded in Java.
Don't use stack or queue.
Don't use a string data type in the main solution.

2- 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 *DEUZEM SAKAI*

3- Don't use ENIGMA or any other extra library.

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

5- 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)**.

6- Your program must work correctly under all conditions. Try to control all possible errors.

7- You should use meaningful variable names, appropriate comments, and good prompting messages.