CME1212 Algorithms and Programming II
Homework 2

Upload your source code files from *DEUZEM SAKAI* until **23 April 2023, 23:55.**

Write a program in the Java programming language for the matching pairs card game: ***"Memory Game"***.

It is a game that requires finding identical pairs.
In other words, it is a game where the user needs to match pairs of tiles.

Tiles should include animal words such as dog, cat, bird, rabbit, etc.

A sample file:
animals.*txt*

Dog
Cat
Bird
Rabbit
Fish
Duck
Sheep
Horse
Cow
Lion
…

The Beginning of the Game

There is a text file (D:*animals.txt*). The number of animals is unknown. Read the file and insert them in an ***AnimalSLL***.

A sample **single linked-list (SLL)** that includes animals:

AnimalSLL

head

There are two single linked-lists (**SLL1** and **SLL2**) with *n* in size. The second SLL contains the matches of tiles in the first SLL.
At the beginning of the game, take the value of *n* from the user.

The game boards (**SLL1 and SLL2**) must be randomly filled with distinct pairs. You should randomly select *n* animals from *animalSLL* to fill game boards. Each element in a SLL should be different from the others. For example, a SLL doesn't contain two dogs.

Example:

SLL1 

SLL2 

Game Playing

Playing is very simple - the computer turns over two tiles randomly, one tile from the first SLL and one tile from the second SLL. If they are identical, the program deletes them from the game boards (***SLLs***), if not, it tries again.

The End of the Game

When all pairs are identified (when all tiles are deleted from the game boards (***SLLs***)), the game will be over.

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

The followings should be printed at each iteration.
- the number of tries (step) and
- the score.

The program ends after printing the high score table on the screen.

Scoring

The scoring principle is as follows:

- Each time the computer makes a successful match, the score should be increased by 20 points.
- If the computer fails to match, the score should be decreased by 1 point.

High Score Table

Read an unsorted file "D:*highscoretable.txt*" and then create two **SLLs (SLL3** and **SLL4)**, which are sorted by score as follows:

Example:

SLL3 Pelin → Kaan → Ali → Yeliz → Cem → Can → Ece → Sibel → Remzi → Nazan

SLL4 180 → 160 → 150 → 140 → 120 → 100 → 90 → 80 → 70 → 65

Ece 90
Kaan 160
Sibel 80
Ali 150
Nazan 65
Yeliz 140
Cem 120
Pelin 180
Can 100
Remzi 70

Notes:
1- The number of elements in the high score table is unknown, but it must contain maximum 12 items.
2- If more than one player has the same score, the older one must be added first.

If he/she gets a score within the top-12 results, he/she must be inserted into the *high-score SLLs (*SLL3 and SLL4*)*.
If the same score exists in the table, the new score should be inserted to the next of them.

At the end of the game, the new score table should be written to the same file.

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

Sample output:

```
Animal SLL: dog cat bird rabbit fish duck sheep horse cow lion monkey elephant snake turtle

Please enter n
5

SLL1: dog  horse  duck  rabbit  monkey                    score=0
SLL2: monkey  rabbit  horse  duck  dog

Randomly generated numbers: 2   5                        step=1

SLL1: dog  horse  duck  rabbit  monkey                    score=-1
SLL2: monkey  rabbit  horse  duck  dog

Randomly generated numbers: 1   5                        step=2

SLL1: horse  duck  rabbit  monkey                        score=19
SLL2: monkey  rabbit  horse  duck

Randomly generated numbers: 2   2                        step=3

SLL1: horse  duck  rabbit  monkey                        score=18
SLL2: monkey  rabbit  horse  duck

Randomly generated numbers: 2   3                        step=4

SLL1: horse  duck  rabbit  monkey                        score=17
SLL2: monkey  rabbit  horse  duck

Randomly generated numbers: 4   1                        step=5

SLL1: horse  duck  rabbit                                score=37
SLL2: rabbit  horse  duck

Randomly generated numbers: 1   2                        step=6

SLL1: duck  rabbit                                       score=57
SLL2: rabbit   duck

Randomly generated numbers: 2   2                        step=7

SLL1: duck  rabbit                                       score=56
SLL2: rabbit   duck

Randomly generated numbers: 2   2                        step=8

SLL1: duck  rabbit                                      score=55
SLL2: rabbit   duck

Randomly generated numbers: 2   1                        step=9

SLL1: duck                                              score=75
SLL2: duck

Randomly generated numbers: 1   1                        step=10

SLL1:                                                   score=95
SLL2:

The game is over.

High Score table:
Pelin  180
```

```
Kaan    160
Ali     150
Yeliz   140
Cem     120
Can     100
You      95
Ece      90
Sibel    80
Remzi    70
```

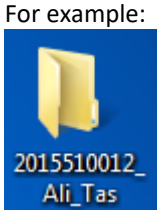
Notes

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.

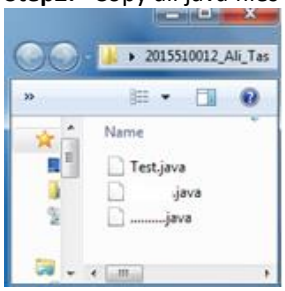
Don't use array or array list or list data structures embedded in Java.
Don't use stack or queue.
Don't use a string as the primary data structure in the main solution, instead of SLL.

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.