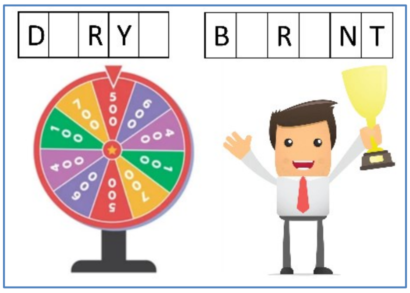
CME1212 Algorithms and Programming II
Homework 1



Upload your source code files from *DEUZEM SAKAI* until **10 April 2022, 23:55.**

Write a Java program for a simple version of the ***"Wheel of Fortune"*** **(Çarkıfelek)** game. Please keep in mind that the official "*Wheel of Fortune*" rules and instructions are different depending on the game version you will implement.

The main aim of the game is to solve the word puzzle and gain money as you can.

Description About the Game

There is only one player.

The game is played with a special wheel that determines the dollar value of the called letters. The wheel is split into 8 segments as follows: 10, 50, 100, 250, 500, 1000, Double Money, and Bankrupt.

The computer randomly determines a word and then a player tries to guess it by suggesting letters. The word should be a country in English such as Turkey, Germany, France, Italy, etc. There are 195 countries in the world.

The letters of the hidden word represented by tiles are face down. A player spins the wheel by naming the letters one-by-one to solve the puzzle. Both consonants and vowels can be guessed by the player after spinning the wheel. If the player guesses a letter that is part of the word, the computer writes it in all its correct positions. If the called letter cannot be found in the word, the player cannot earn any money from this turn.

Before guessing a letter, a player must spin the wheel to determine how much each letter, if revealed. Every spin is risky because the wheel also includes bankrupt, resulting in the loss of all earnings to that point. In the case of "Double Money" segment, the current amount of money is doubled if the player guesses correctly.

The game is over when the player completes the whole word correctly. The player gets the money he/she earns.

Initially

Read the <u>unsorted</u> file "D:\countries.txt" and then create a **Stack (S1)**, which is <u>sorted</u> alphabetically, as follows:

Turkey
France
Spain
Germany
Solomon Islands
Poland
South Korea
Japon
India
Kenya
Malta
Netherlands

S1	Afghanistan	Albania	Algeria	...	Germany	...	Yemen	Zambia	Zimbabwe

Add all letters in a **Stack (S2)**.

S2	A	B	C	...	V	W	X	Y	Z

Read the <u>unsorted</u> file "D:\HighScoreTable.txt" and create two **Stacks (S3 and S4)** as follows:

Pelin	1530
Kaan	550
Ali	3850
Yeliz	1650
Nazan	2740
Can	2100
Ece	1500
Sibel	1680
Remzi	1620
Cem	400

S3	Ali	Nazan	Can	Sibel	Yeliz	...	Cem
S4	3850	2740	2100	1680	1650	...	400

The stacks S3 and S4 must be sorted by scores in decreasing order.

The Beginning of the Game

Generate a random number (*x*), print this number on the screen, and get x^{th} word of S1.
For example; if the randomly generated number is 3, the country "ALGERIA" is selected from the list.

Store this word in a **Queue (Q1)**.
In addition to Q1, create an empty Queue, named **Q2**, for the game board.

Example:

Q1	A	L	G	E	R	I	A
Q2	-	-	-	-	-	-	-

Game Playing

The game board (Q2) must be seen on the screen, permanently.

The program should randomly select a letter from S2. After that, the letter should be deleted from S2.

If the selected letter does not exist in Q1, the player does not earn anything in this step.
If the user suggests a letter that exists in Q1, the computer writes it in all its correct positions in Q2.

For example;
If the selected letter is "*E*", then Q2 should be as follows:

Before

Q2	-	-	-	-	-	-	-

After

Q2	-	-	-	E	-	-	-

The program must display all steps until the game is over. In other words, the program must repeatedly print Q2, the current step, the score, and the remaining letters (S2) on the screen.

Scoring

If the guess is correct, then a prize according to the wheel segment is paid. The amount in the wheel segment is multiplied by the number of times that the letter appears in the word.

End of the Game

The game is over when the player completes the whole word correctly.

High Score Table

If the player earns a score within the top-10 results, he/she will be displayed in the High-Score table.
Add into S3 and S4. If the same score exists in the table, the new score should be inserted to the next of them. Delete the last record.
The new score table should be written to the same 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 1 - Questions**" part of the *DEUZEM SAKAI* software.

Sample output:

```
Randomly generated number: 59

Word:  - - - -               Step:   1      Score: 0          ABCDEFGHIJKLMNOPQRSTUVWXYZ
Wheel: 100
Guess: I
Word:  - I - I               Step:   2      Score: 200        ABCDEFGHJKLMNOPQRSTUVWXYZ
Wheel: 10
Guess: R
Word:  - I - I               Step:   3      Score: 200        ABCDEFGHJKLMNOPQSTUVWXYZ
Wheel: 1000
Guess: F
Word:  F I - I               Step:   4      Score: 1200       ABCDEGHJKLMNOPQSTUVWXYZ
Wheel: 250
Guess: S
Word:  F I - I               Step:   5      Score: 1200       ABCDEGHJKLMNOPQTUVWXYZ
Wheel: Double Money
Guess: J
Word:  F I J I               Step:   6      Score: 2400       ABCDEGHKLMNOPQTUVWXYZ

You win ₺2400 !!!

High Score Table
Ali   3850
Nazan 2740
You   2400
Can   2100
...
Kaan  550
```

Notes

1- In your program, you can use the stack and queue data structures as you want, but you must use only stack and queue.
Don't use other data structures such as an array or arraylist or list.
Don't use STRING data type in the main solution, instead of a stack or queue.

2- The stack class has only the following methods: push, pop, peek, isFull, isEmpty, and size.
Don't add a new method into the stack class.

For example, don't write a *display* method in the Stack class.
For example, don't write a *search* method in the Stack class.

All other methods must be written in the ***main*** program.

3- The queue class has only the following methods: enqueue, dequeue, peek, isFull, isEmpty, and size.
Don't add a new method into the Queue class.

For example, don't write a *display* method in the Queue class.
For example, don't write a *search* method in the Queue class.

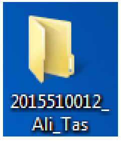
All other methods must be written in the ***main*** program.

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

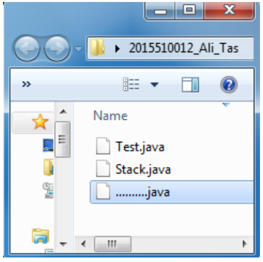
5- Don't use stack and queue classes embedded in Java. Write your own Stack and Queue classes.

6- 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*

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

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

9- 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)**.

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

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