Assignment – 2

Who wants to be a millionaire?

In this assignment, you need to implement a popular TV show "Who wants to be a millionaire?". We believe most of you are familiar with this show: A contestant is asked increasingly difficult questions to become a millionaire. Questions are multiple choice and the contestant must choose the correct answer to see the next question. After viewing a question, the contestant can leave the game with the money already won rather than attempting an answer. If the contestant answers a question incorrectly, then all of their winnings are lost, except that the 1,000TL and 15,000TL prizes are guaranteed: if a player gets a question wrong above these levels, then the prize drops to the previous guaranteed prize. List of money prize for the number of questions correctly answered is given below:

1 question answered correctly 500 TL
2 questions answered correctly 1.000 TL
3 questions answered correctly 2.000 TL
4 questions answered correctly 3.000 TL
5 questions answered correctly 5.000 TL
6 questions answered correctly 7.500 TL
7 questions answered correctly 15.000 TL
8 questions answered correctly 30.000 TL
9 questions answered correctly 60.000 TL
10 questions answered correctly 125.000 TL
11 questions answered correctly 250.000 TL
12 questions answered correctly 1.000.000 TL

In our version of the show, contestants are given two lifelines that can be utilized after seeing a question. These are "50/50" and "Double Dip". In "50/50", the computer removes two of the incorrect answers. The contestant can leave the game if she/he is not sure about the answer. In "Double Dip", the contestant can make two guesses; however, she/he cannot leave the game. The question must be answered. These two lifelines can be used only once.

Implementation

You are given a file that contains questions. Please open this file with a text editor such as notepad++. This is a comma separated file. A line from the file and the definition of the file content is given below:

"1142","1","Bir kisiyi agirlamaktan mutluluk duyuldugunu ifade etmek icin nerede yeri oldugu soylenir?","2013-03-27 14:25:43","Avucumuzun icinde","Dizimizin dibinde","Basimizin ustunde","Gozumuzun onunde","C","44"

Definition

- question_id: ID for a question in our dataset. Used as a foreign key in other tables.
- no: Level of the question in gameshow- 1 to 12-. Often is an indicator of the question difficulty.

- question: Text of the question in Turkish.
- time: Server timestamp of the question. **(disregard this information)**
- choiceA: Text of first choice in Turkish.
- choiceB: Text of second choice in Turkish.
- choiceC: Text of third choice in Turkish.
- choiceD: Text of fourth choice in Turkish.
- correct_choice: Letter for the correct choice.
- program_id: The foreign key from the Program table that identifies the program the question has appeared in **(disregard this information)**

The dataset is available at: https://goo.gl/EfchSB. Please download the file named question.csv.

- Your implementation should start by asking a randomly selected level 1 question. As the contestant answers questions correctly, you should increase the level of the question by 1. At each level, questions must be chosen randomly.
- Before starting the game, a screen should welcome you and ask whether you want to start a new show or see the show report. The show report option should read the file, which you stored earlier contestants and their prizes. Therefore, your implementation should first ask the name of the contestant. After the contestant leaves or completes answering questions, you should save how much prize she/he wins to a text file.
- While implementing our version of the show, it would be a good idea to create a class that has the following fields: question_id, level_no, question, choiceA, choiceB, choiceC, choiceD, correctChoice. Read the file line by line and parse the line into your class. Since there are many question to load in your classes, use arrays to keep them in the memory.

Submission

This is a group project. Please form groups of 2 (at least) or 3 (at most) students.

Your assignment is due on 01/05/2018 (23:59). Upload your files in a compressed format (zip, rar, etc.) to https://goo.gl/qsbvG4. Students can submit the assignment until 04/05/2018 (23:59). Those will receive a penalty of 10 points per day.

References

Data set, who wants to be a millionaire. https://github.com/bahadiri/Millionaire