Homework #3 - Make a Magic 8 Ball Program

For this assignment, you will be writing a *Magic8Ball* class with the following:

● **A constructor (*\_\_init\_\_*) method**: The constructor will initialize a new ***Magic8Ball*** object from the passed list of all possible answers (answer\_set).

○ Set ***answer\_list*** to answer\_set (the argument of the method).

○ Set ***question\_history\_list*** to an empty list. This will hold all the questions that have been asked.

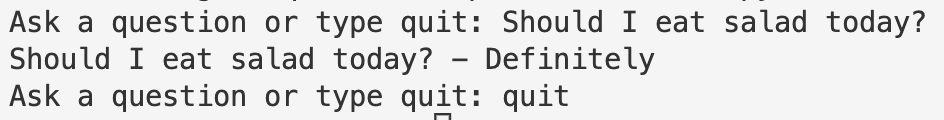
○ Set ***answer\_history\_list*** to an empty list. This will hold the indices of all of the answers that have been generated.

● ***\_\_str\_\_* method**: It should return a string with all of the answers in ***answer\_list*** separated by commas.

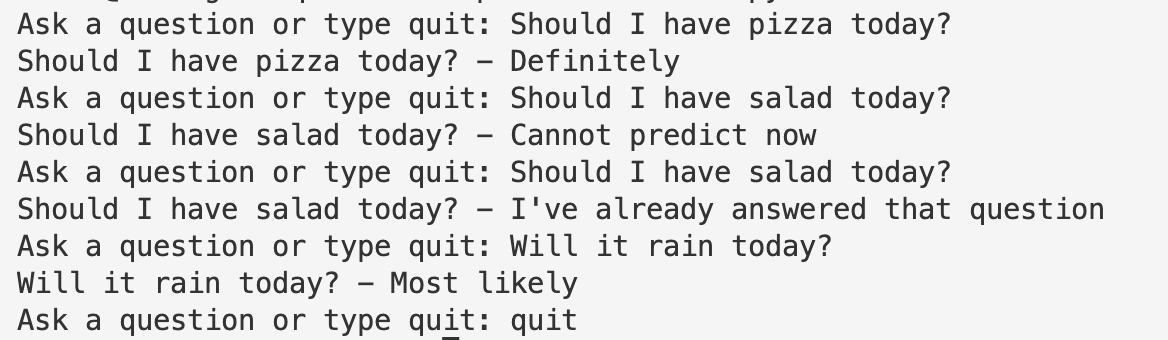
● ***get\_random\_answer* method**: This method randomly picks an answer from the ***answer\_list***. It first randomly chooses an index and appends that index to the ***answer\_history\_list***. Then it returns the answer at the randomly picked index from ***answer\_list***.

● ***shake* method**: The method takes a question and first checks if the question is already in the ***question\_history\_list***. If so, it returns a string, "I've already answered that question” Otherwise, it adds the question to the ***question\_history\_list*** and returns the answer from ***get\_random\_answer***.

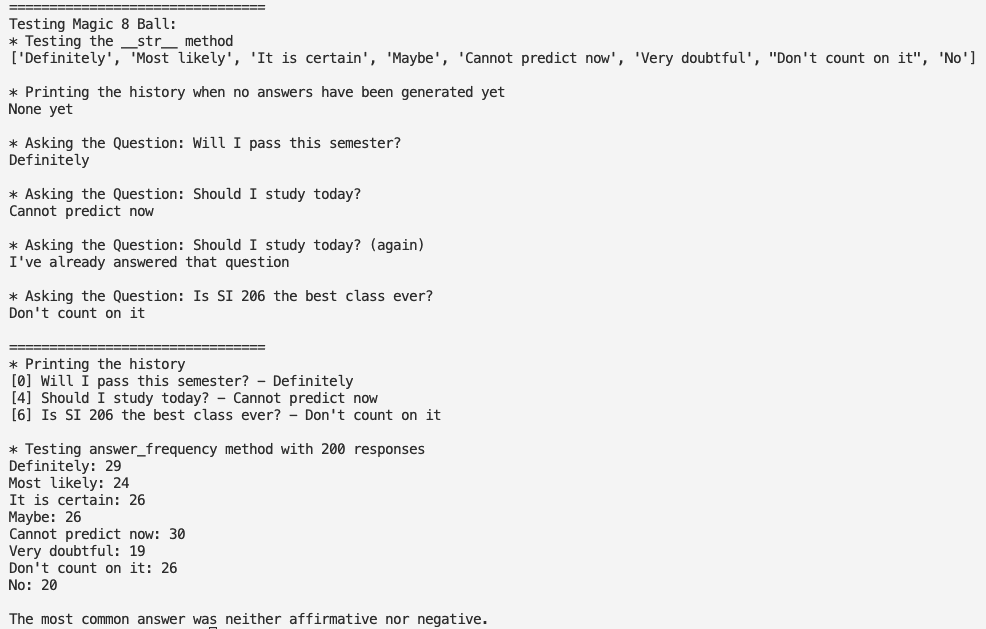
● ***print\_question\_history* method**: If there are no items in the ***answer\_history\_list***, it prints "None yet". Otherwise, the method prints "[answer index] question - answer" for each of the indices in the ***answer\_history\_list***, each on a separate line.

● ***main()* function**: Loops until the user types “quit” getting a question from the user, calls the ***shake*** method, and prints the question and response from ***shake*** as "*question* - *answer*" as shown below. 

● Example Output From HW3.py

**Sample output from the main method: **

**Sample output from the test method:**



**NOTE: Your output will not look *exactly* like this because we are using *random* and can’t predict what it will return*.***

Grading Rubric - Total of 60 points

● 5 points - the ***\_\_init\_\_*** method sets the object's ***answer\_list*** correctly to the passed argument and sets both the object's ***question\_history\_list*** and ***answer\_history\_list*** to an empty list

● 5 points - the ***\_\_str\_\_*** method returns a string with all answers in ***answer\_list*** separated by commas

* Correct answers for a list ['Yes', 'No', 'Maybe']:
  + "['Yes', 'No', 'Maybe']"
  + Yes, No, Maybe
  + Yes,No,Maybe

● 5 points - the ***shake*** method returns *“I’ve already answered that question”* if the question has already been asked

● 10 points - the ***shake*** method calls the ***get\_random\_answer*** method and returns the answer when the user asks a new question and adds the passed question to the ***question\_history\_list***.

● 10 points - the ***get\_random\_answer*** method returns a random answer and saves the index in the ***answer\_history\_list***

***●*** 5 points - the **print\_question\_history** method prints "None Yet"when there are no items in ***answer\_history\_list***

● 10 points - ***print\_question\_history*** prints "[index] Question - Answer" for each of the questions in the ***question\_history\_list*** and ***answer\_history\_list*** in order and on a separate line.

● 10 points - the ***main()*** function loops until the user enters "quit" and each time asks the users for a question and prints the "question - answer".

This grading rubric shows how you will gain points, but not all the ways you could lose points.

Extra Credit - 6 points

Create the ***answer\_frequency*** method. It takes as an argument: n, an integer. The method implements the following:

(1) It calls get\_random\_answer an 'n' number of times and records the random answer in a list. (2) It then prints the frequency of each answer in each line.

For example, it will print

Definitely: 27

Most likely: 32

It is certain: 25

... and so on.

(3) It prints whether the most common answer was "affirmative", "negative", or "neither affirmative nor negative".

Please feel free to use these predefined lists:

affirmative = ["Definitely", "Most likely", "It is certain"]

negative = ["Very doubtful", "Don't count on it", "No"]