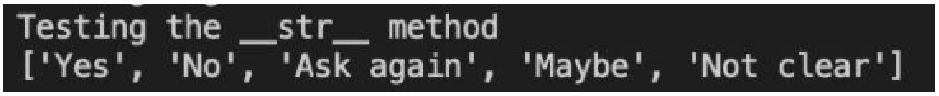
Homework #3 - Make a Magic 8 Ball

For this assignment, you will be writing a *Magic 8 Ball* class with the following:

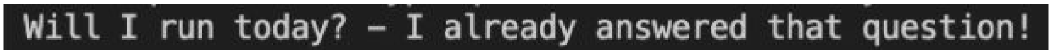
- **A constructor (__init__) method**: The constructor will initialize a new *Magic_8* object from the passed list of all possible answers.
    - Set ***answer_list*** to the passed list of possible answers.
    - Set ***question_list*** to an empty list. This will hold all the questions that have been 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, For example : "No, Yes, Better not tell you now."

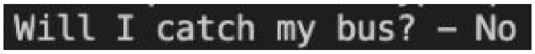


```
Testing the __str__ method
['Yes', 'No', 'Ask again', 'Maybe', 'Not clear']
```

- ***check_question*** method: Checks if the current question is already in the ***question_list*** and if so returns *"I already answered that question!"*, otherwise it adds the current question to the ***question_list*** and returns the answer from ***shake_ball***.



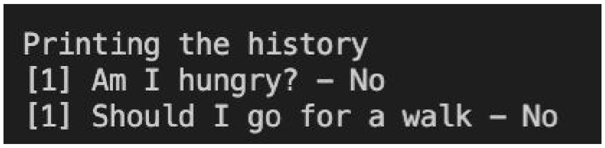
```
Will I run today? - I already answered that question!
```



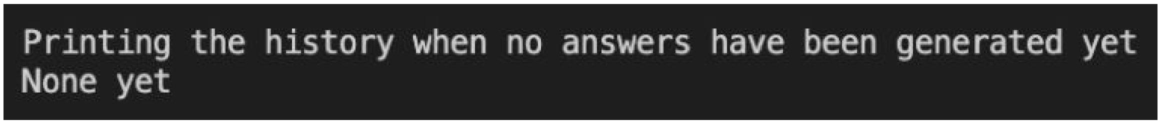
```
Will I catch my bus? - No
```

- ***shake_ball* method**: Returns a random answer from the **answer_list.** It randomly picks an index from 0 to the number of possible answers minus one *(hint: use the random module)*. It adds the index to the end of the **answer_history_list**. It returns a string containing the answer at that index (not the index).

- ***print_history* method**: Prints the content of the **answer_history_list** with the answer index in [ ] and each question and answer on a separate line. It does not return anything. If there are no items in **answer_history_list** it should print **"None yet".**

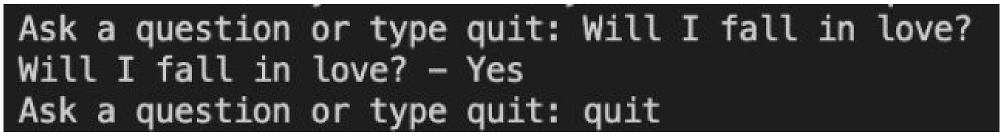


```
Printing the history
[1] Am I hungry? - No
[1] Should I go for a walk - No
```



```
Printing the history when no answers have been generated yet
None yet
```

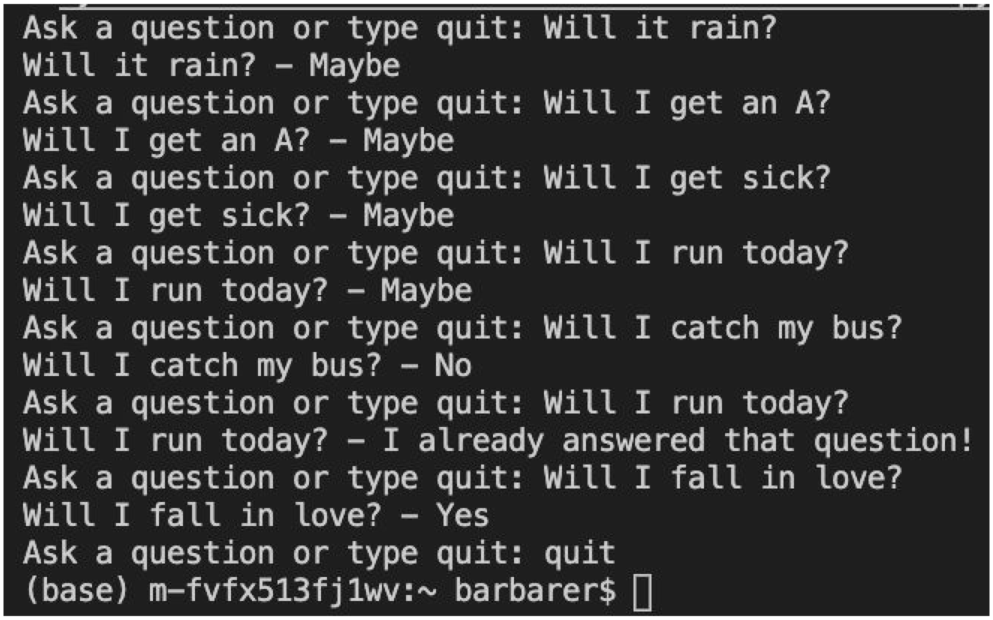
- ***main()* function**:   Loops until the user types “quit” getting a question from the user, calls the ***check_question*** method, and prints the question and response from **check_question** as "*question* - *answer*" as shown below.



```
Ask a question or type quit: Will I fall in love?
Will I fall in love? - Yes
Ask a question or type quit: quit
```

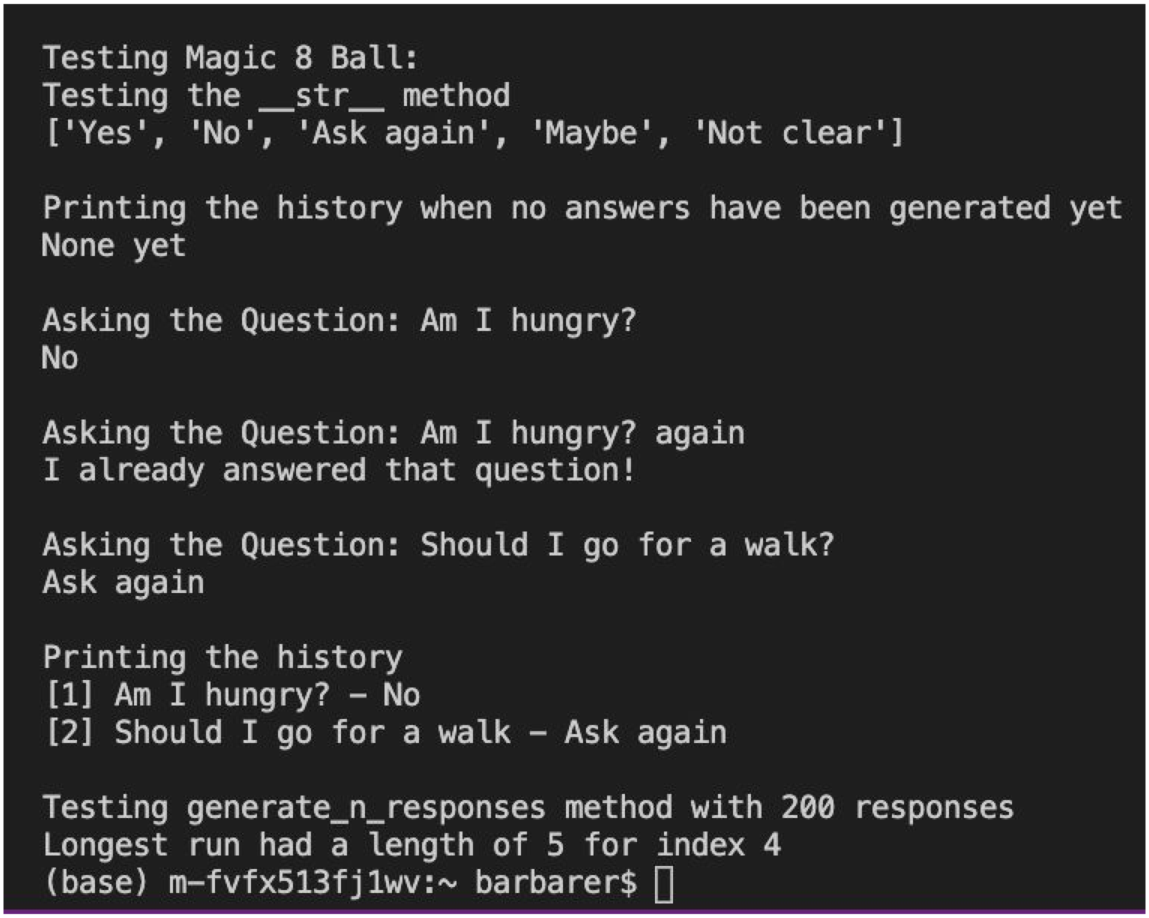
- Example Output From HW3.py

Sample output from the main method:



```
Ask a question or type quit: Will it rain?
Will it rain? - Maybe
Ask a question or type quit: Will I get an A?
Will I get an A? - Maybe
Ask a question or type quit: Will I get sick?
Will I get sick? - Maybe
Ask a question or type quit: Will I run today?
Will I run today? - Maybe
Ask a question or type quit: Will I catch my bus?
Will I catch my bus? - No
Ask a question or type quit: Will I run today?
Will I run today? - I already answered that question!
Ask a question or type quit: Will I fall in love?
Will I fall in love? - Yes
Ask a question or type quit: quit
(base) m-fvfx513fj1wv:~ barbarer$ ▯
```

Sample output from the test method:



```
Testing Magic 8 Ball:
Testing the __str__ method
['Yes', 'No', 'Ask again', 'Maybe', 'Not clear']

Printing the history when no answers have been generated yet
None yet

Asking the Question: Am I hungry?
No

Asking the Question: Am I hungry? again
I already answered that question!

Asking the Question: Should I go for a walk?
Ask again

Printing the history
[1] Am I hungry? - No
[2] Should I go for a walk - Ask again

Testing generate_n_responses method with 200 responses
Longest run had a length of 5 for index 4
(base) m-fvfx513fj1wv:~ barbarer$ ▯
```

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

NOTE 2: You are welcome to replace the answers we have provided in the *main function* with your favorite responses

Grading Rubric - Total of 60 points

- 5 points - the ***__init__*** method sets the object's ***answer_list*** correctly to the passed ***answer_list*** and sets both the object's ***answer_history_list*** and ***question_list*** to an empty list
- 5 points - the ***__str__*** method returns a string with all answers in ***answer_list*** separated by commas : "Yes, No, It depends"
- 5 points - the ***check_answer*** method returns *"I already answered that question!"* if the question has already been asked
- 10 points - the ***check_answer*** method calls the ***shake_ball*** method and returns the answer when the user asks a new question and adds the passed question to the ***question_list***.

- 10 points - the ***shake_ball*** method returns a random answer and saves the index of the answer at the end of the ***answer_history_list***
- 5 points - the **print_history** function prints **"None Yet"** when there are no items in ***answer_history_list***.
- 10 points - ***print_history*** prints "[index] Question - Answer" for each of the questions in the **question_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 - response*"**.

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

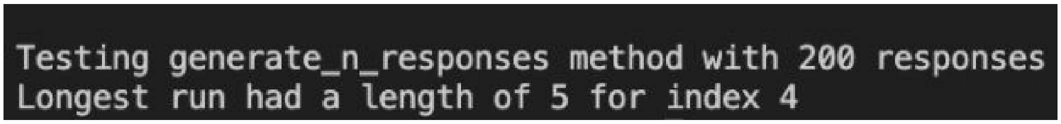
Extra Credit - 6 points

Implement the following method: Create the ***generate_n_responses*** method. It takes a number as an input: n, Ex: 200. It generates random answers n times by calling ***shake_ball*** and returns the index and length of the longest consecutive run for an answer index. You should first reset ***answer_history_list*** to an empty list. A run is a repetition of the same number consecutively in a list.

Ex: If 10 random answers were [1,5,6,3,2,4,1,**4,4,4**] then three 4's is the longest run

Hence the function should return “**longest run was length of 3 for index 4”**

Extra Credit Example Output:



```
Testing generate_n_responses method with 200 responses
Longest run had a length of 5 for index 4
```