Homework #3 - Make a Random Quote Generator

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

- **A constructor (__init__) method**: The constructor will initialize a new *Random_Quote_Generator object* from the passed list of all possible quotes and their respective authors.
    - Set **quote_list** to the passed list of all possible quotes
    - Set **author_list** to the passed list of authors
    - Set **quote_history_list** to an empty list. This will hold the indices of all of the quotes that have been generated.

- ***quote* method**: Returns a random quote and its author from the **quote_list** and the **author_list** respectively**.** It randomly picks an index from 0 to the number of possible quotes minus one *(hint: use the random module)*. Add the index for the quote to the end of the **quote_history_list**. Return a string containing the quote and the author at that index (not the index itself) in the following format:

```
Quote : We cannot solve our problems with the same thinking we used when we
 created them. - Albert Einstein
```

- ***__str__* method**: If no quotes have been generated yet, it should return "No quotes so far!" otherwise it should return "Most recent: *Quote - Author*" as shown below.

```
Most recent: We cannot solve our problems with the same thinking we used wh
en we created them. - Albert Einstein
```

- ***print_history* method**: Prints the content of the quote_history_list with the index number in [] and each quote and author on a separate line. It does not return anything.

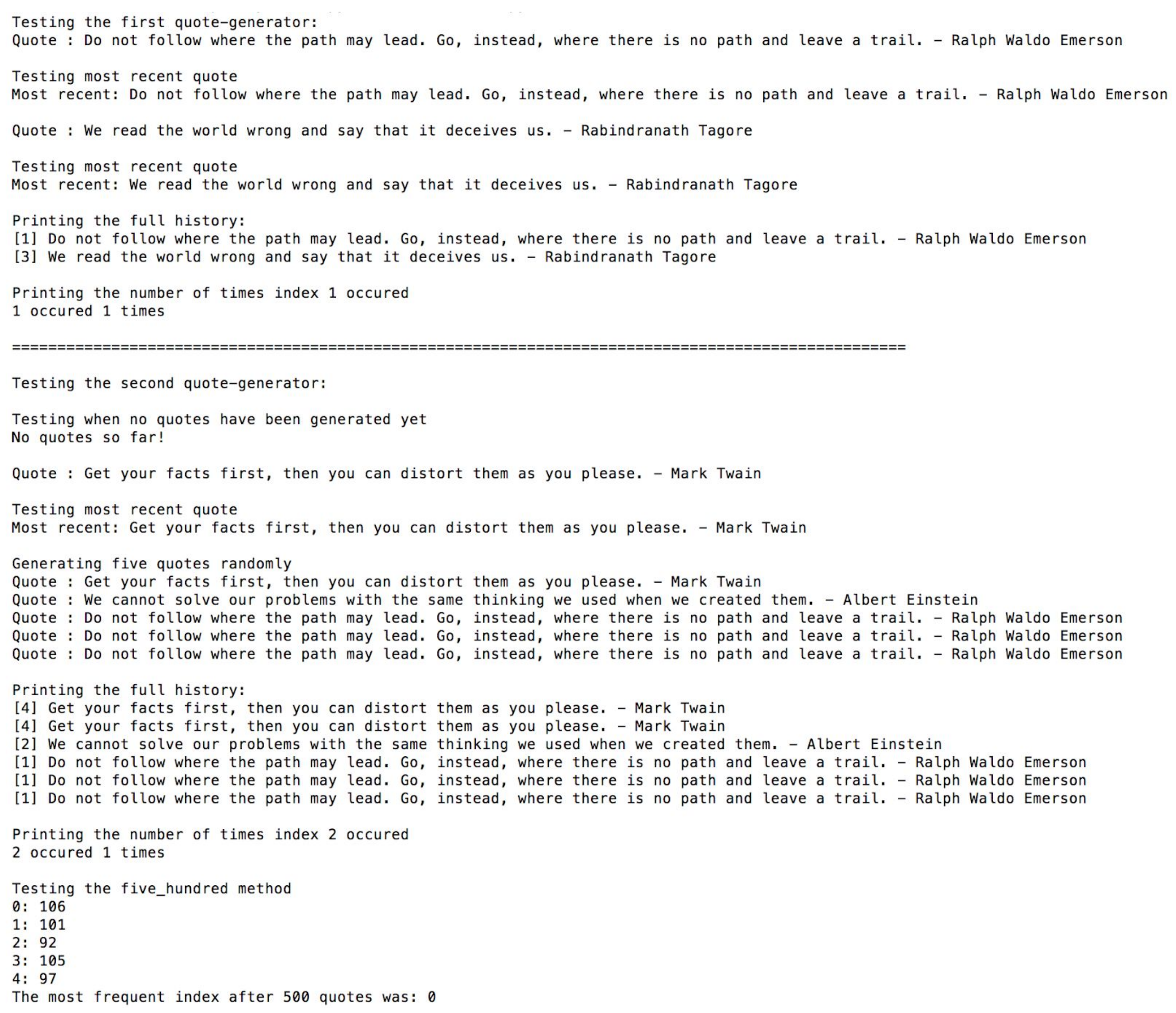
```
[2] We cannot solve our problems with the same thinking we used when we cre
ated them. - Albert Einstein
[4] Get your facts first, then you can distort them as you please. - Mark T
wain
```

- ***print_count_for_num* method**: Takes in a parameter *num* which specifies which index to look for. Prints the number of times that index occurs in the quote_history_list.

```
1 occured 0 times
```

Example Output From HW3.py

You can also refer to the example output provided as an image file separately. The file is called ***ExampleOutput_HW3.png***



```
Testing the first quote-generator:
Quote : Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson

Testing most recent quote
Most recent: Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson

Quote : We read the world wrong and say that it deceives us. - Rabindranath Tagore

Testing most recent quote
Most recent: We read the world wrong and say that it deceives us. - Rabindranath Tagore

Printing the full history:
[1] Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson
[3] We read the world wrong and say that it deceives us. - Rabindranath Tagore

Printing the number of times index 1 occured
1 occured 1 times

===================================================================================================

Testing the second quote-generator:

Testing when no quotes have been generated yet
No quotes so far!

Quote : Get your facts first, then you can distort them as you please. - Mark Twain

Testing most recent quote
Most recent: Get your facts first, then you can distort them as you please. - Mark Twain

Generating five quotes randomly
Quote : Get your facts first, then you can distort them as you please. - Mark Twain
Quote : We cannot solve our problems with the same thinking we used when we created them. - Albert Einstein
Quote : Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson
Quote : Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson
Quote : Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson

Printing the full history:
[4] Get your facts first, then you can distort them as you please. - Mark Twain
[4] Get your facts first, then you can distort them as you please. - Mark Twain
[2] We cannot solve our problems with the same thinking we used when we created them. - Albert Einstein
[1] Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson
[1] Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson
[1] Do not follow where the path may lead. Go, instead, where there is no path and leave a trail. - Ralph Waldo Emerson

Printing the number of times index 2 occured
2 occured 1 times

Testing the five_hundred method
0: 106
1: 101
2: 92
3: 105
4: 97
The most frequent index after 500 quotes was: 0
```

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 quotes and names we have provided in the *main function* with your favorite quotes and authors

Grading Rubric - Total of 60 points

5 points - the __*init*__ method sets the object's quote_list and author_list correctly (the instance variables)
5 points - the __*init*__ method sets the object's quote_history_list to an empty list
10 points - the *quote* method correctly picks a random index between 0 and the number of quotes in the quote_list minus one
5 points - the *quote* method saves the picked index at the end of the quote_history_list
5 points - the *quote* method returns the quote and the corresponding author
5 points - the __*str*__ method returns a string “Most recent: *Quote - Author*” with the text of the last quote and author
5 points - the __*str*__ method returns a string telling the user “No quotes so far!” if there haven't been any calls to *quote* method yet
10 points - *print_history* prints "[index] Quote - Author" for each of the quotes in the quote_history_list in order and on a separate line.
10 points - *print_count_for_num* correctly prints the number of times an index occurs in the quote_history_list

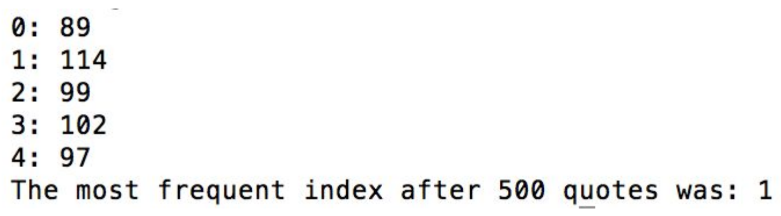
This grading rubric shows how you will gain points, but not all the ways you could lose points.

Extra Credit - 6 points

Implement the following method:

five_hundred* method**: Finds the most frequently chosen index after telling 500 quotes. In this method, reset the ***quote_history_list instance variable to the empty list, execute the ***quote*** method 500 times, print how many times each index occurred, and print the most frequently occurring index. Choose any one of the topmost common indices if there is a tie.

Extra Credit Example Output:



```
0: 89
1: 114
2: 99
3: 102
4: 97
The most frequent index after 500 quotes was: 1
```