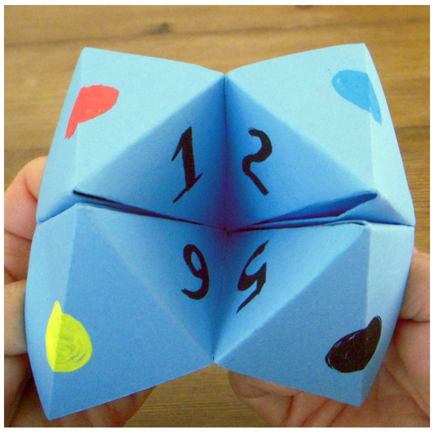
Homework #3 - Make a Cootie Catcher Program



Using a Cootie Catcher:

1. Player1 asks a question. Player 2 holds the Cootie Catcher.
2. Player1 chooses a starting word (usually, a favorite color).
3. Player2 spells out the starting word, opening and closing the Cootie Catcher for each letter. Choosing the word “blue” would spell B-L-U-E, opening and closing the fortune teller 4 times.
4. Player1 picks one of the revealed numbers.
5. Player2 reads the answer to the question.

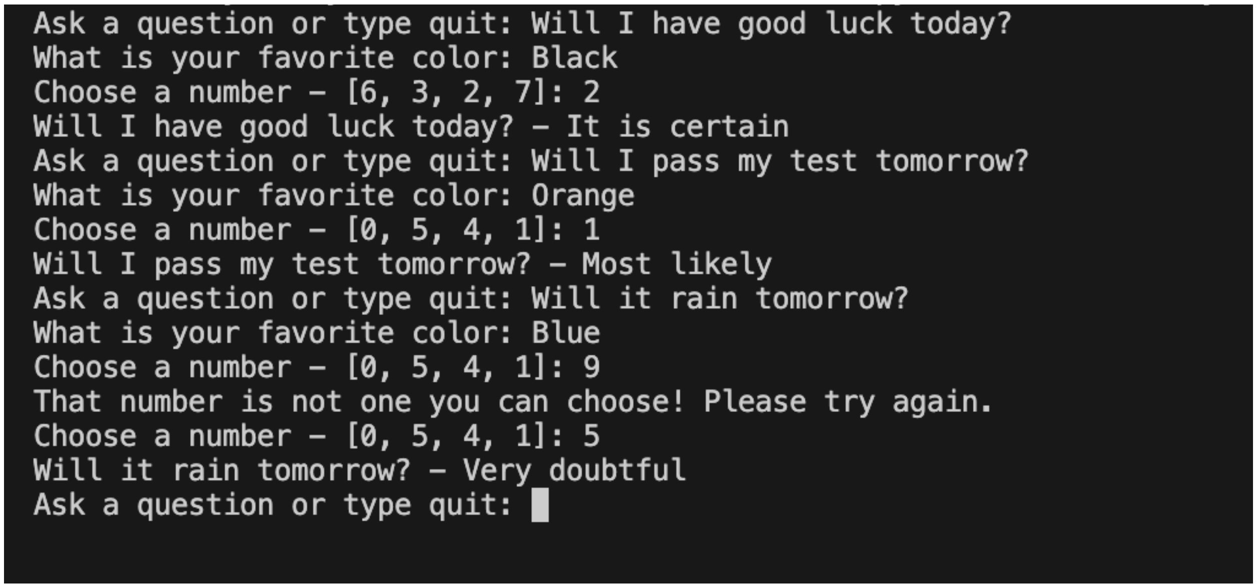
Instructions

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

- **An *__init__(self, answers, num1s, num2s)* method**: This will initialize a new ***CootieCatcher*** object from the 3 passed lists.
    1. Set the attribute ***answers_list*** to the passed ***answers***. This is a list of the eight possible answers a player could receive.
    2. Set the attribute ***num1_list*** to the passed list ***num1s*** with four numbers in the range from 0-7 inclusive. For example, (0, 2, 5, 6).
    3. Set the attribute ***num2_list*** to the passed ***num2s*** with the remaining four numbers in the range from 0-7 inclusive that are not in ***num1s.*** For example (1, 3, 4, 7).
    4. Set the attribute ***questions_history_list*** to an empty list.
    5. Set the attribute ***answers_history_list*** to an empty list.
- ***A __str__(self)* method**: Return a string with all of the answers in ***answers_list*** separated by commas.

- ***An ask (self, question)*** **method**: The method takes a question and first checks if the question is already in the ***questions_history_list***. If so, it returns a string, **`"I've already answered that question."`** Otherwise, it adds the question to the ***questions_history_list*** and returns the result from the ***get_fortune*** method.
- ***A get_fortune(self)*** **method** that:
    1. Prompts the user with **`"What is your favorite color: "`** If the length of the string for their favorite color is even, use ***num1_list*** in the next step, if it is odd, else use ***num2_list***.
    2. Prompts the user to **`"Pick a number - <numbers from appropriate list here>: "`**
        a. Example prompt: **`"Pick a number - [0, 1, 2, 3]: "`**
        b. If the user enters a number that is not in the list, print **`"That number is not one you can choose! Please try again."`** then re-prompt the user for input
    3. Uses the selected number to access an answer from ***answers_list***
    4. Returns the answer to the player's question and add the index to ***answers_history_list.***
- ***A print_questions_history_list(self)*** **method**: If there are no items in the ***answers_history_list***, it prints **`"None yet"`**. Otherwise, the method prints **`"<number> <question> - <answer>" for each item on the`** each on a separate line.
- A ***main()*** **function**: Loops until the user types "quit", gets a question from the user, calls the ***ask*** method, and prints the question and response from ***ask*** as **`"<question> - <answer>"`** as shown below.

Sample output from the main method (your output will depend on what you use when you create your Cootie Catcher object**).**


```
Ask a question or type quit: Will I have good luck today?
What is your favorite color: Black
Choose a number - [6, 3, 2, 7]: 2
Will I have good luck today? - It is certain
Ask a question or type quit: Will I pass my test tomorrow?
What is your favorite color: Orange
Choose a number - [0, 5, 4, 1]: 1
Will I pass my test tomorrow? - Most likely
Ask a question or type quit: Will it rain tomorrow?
What is your favorite color: Blue
Choose a number - [0, 5, 4, 1]: 9
That number is not one you can choose! Please try again.
Choose a number - [0, 5, 4, 1]: 5
Will it rain tomorrow? - Very doubtful
Ask a question or type quit: 
```

Grading Rubric - Total of 60 points

- 5 points - the ***__init__*** method sets the object's ***answers_list, num1_list,*** and ***num2_list*** attributes correctly to the passed arguments, sets both the object's ***questions_history_list*** and ***answers_history_list*** attributes to an empty list
- 5 points - the ***__str__*** method returns a string with all answers in ***answers_list*** separated by commas
  - Correct answers for a list "**`["Definitely", "Most likely", "It is certain", "Maybe", "Cannot predict now", "Very doubtful", "Don't count on it", "Absolutely not"]`**"
- 5 points - the ***ask*** method returns "**`I've already answered that question`**" if the question has already been asked
- 5 points - the ***ask*** method calls the ***get_fortune*** method and returns the answer when the user asks a new question and adds the passed question to the ***questions_history_list***.
- 5 points - the ***get_fortune*** prompts the user for their favorite color and prompts the user to input a number from either ***num1_list*** or ***num2_list***
- 5 points - if the user provides a number that was not present in ***num1_list*** or ***num2_list,*** display "**`That number is not one you can choose!`**

`Please try again."` and re-prompt the user for input

- 5 points - the ***get_fortune*** method returns the appropriate answer and saves the index to the ***answers_history_list***
- 5 points - the ***print_questions_history_list*** method prints `"None Yet"` when there are no items in ***answers_history_list***
- 10 points - ***print_questions_history_list*** prints `"<number> <question> - <answer>"` for each of the questions in the ***questions_history_list*** and ***answers_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.