12.1 - Code Breakers

Create the game "Code Breakers", which is a guessing game based on the game Mastermind created in the 1970s. In this game, a random code will be generated by the computer which is between 4 and 6 characters long and only contains the digits 0 through 5 (inclusive). Similarly to Wordle, the player attempts to guess the hidden code and is given clues about how close to the correct code they are. After each guess, the program informs the player how many of the digits were of the correct value in the correct location, and how many digits were of the correct value but were in the wrong location. The game ends with a victory message if the player guesses the correct code within 10 guesses. Otherwise, the game ends with a defeat message.

At the start of your program, the player is presented some introductory text followed by a main menu from which they can choose between; 1. `Rules` to view the rules, 2. `New Game` to start a new game, 3. `Load Game` to load a saved game, or 4. `Quit` to exit the program. If the player enters an invalid choice, the program displays an appropriate error message and lets the player try again until a valid selection is made.

Terminal

```
$ python code_breakers_login.py
You are part of Unladened Swallow Society trying to break the infamous Holy
Grail lock.  This lock protects a vault where IU has locked up all the EBEC
grades.  To get your grades you will have to break this lock.  Luckily
those silly IU students messed up when making this lock, and it will give
you hints on what the code is.  However, you don't know the length of the
passcode and only have 10 guesses.  You don't want to run out of these.
Maybe the vault will turn you into a newt!.  Maybe it will destroy the
grades.  What if you have to rewrite time-calculator!

Will you be able to break this lock before your grades are lost forever?

Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice: spam
Please enter 1, 2, 3, or 4.
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice:
```

1 Rules

When the player chooses to view the rules, the program presents the rules as depicted below, and then returns to the main menu.

Terminal

```
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice: 1

Code Breakers Rules:
1. You get 10 guesses to break the lock.
2. Guess the correct code to win the game.
3. Codes can be either 4, 5, or 6 digits in length.
4. Codes can only contain digits 0, 1, 2, 3, 4, and 5.
5. Clues for each guess are given by a number of red and white pins.
   a. The number of red pins in the R column indicates the number of digits
      in the correct location.
   b. The number of white pins in the W column indicates the number of
      digits in the code, but in the wrong location.
   c. Each digit of the solution code or guess is only counted once in the
      red or white pins.

Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice:
```

2 New Game

After the player chooses to start a new game, the program must call a function to produce the game's solution code named `generate_solution`. This function takes two arguments, the code's minimum length, and its maximum length, and *returns the solution code as a string*. Even though the game will only need codes with lengths of 4, 5, or 6 digits, this function is able to generate codes of any length within the bounds of its arguments. Both the length of the solution code, and the digits it contains (0, 1, 2, 3, 4, and/or 5) are determined at random. No other portion of your program will involve random chance.

Once the new solution code is generated, the player is given up to 10 chances to guess the code. Before each guess, the game board, which tracks the status of the game, should be displayed as shown in the examples below. The top row of the board is for the solution. It is initially disguised as 6 "o"s regardless of the solution's length, and is revealed only when the game is complete. The left side of the board tracks the player's guesses. It has space for guesses of up to 6 digits, and its rows are filled from the bottom up. Empty spaces on the board are represented with the letter "o". The right side of the board has columns labeled "R" and "W". These are used to provide clues for each guess according to rules that will be explained shortly.

Terminal

```
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice: 2
+------------------+-----+
| o  o  o  o  o  o | R W |
+------------------+-----+
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
+------------------+-----+
What is your guess (q to quit, wq to save and quit):
```

Beneath the game board, the player is prompted to enter their next guess. The player may enter either "q" to quit the game, "wq" to save and quit, or enter a valid guess of 4 to 6 characters containing only the digits 0 through 5. Invalid guesses are not accepted and do not count. If the player enters a guess that is the wrong length, or contains invalid characters, they are prompted to try again.

Terminal

```
What is your guess (q to quit, wq to save and quit): 123
Your guess was "123". This is too short.
Guess lengths must be between 4 and 6.
What is your guess (q to quit, wq to save and quit): 123123123
Your guess was "123123123". This is too long.
Guess lengths must be between 4 and 6.
What is your guess (q to quit, wq to save and quit): 4567
Your guess was "4567". It must be only numbers 0 through 5.
What is your guess (q to quit, wq to save and quit): spam
Your guess was "spam". It must be only numbers!
What is your guess (q to quit, wq to save and quit):
```

2.1 Clues

After the player inputs a valid guess, a clue is displayed on the right side of the game board as a number of red pins (the “R” column), and a number of white pins (the “W” column). The number of red pins indicates how many digits in the guess are in the correct position and have the correct value. The number of white pins indicates how many digits in the guess are in the solution code, but are not in the correct position. Each digit of the solution code or guess can only be counted once in the red or white pins.

For example, given the solution code 550022 and the guess 153455, the correct clue is 1 red pin and 1 white pin. The matching 5s in the 2nd position earn the red pin. Since only one other 5 remains in the code, only one white pin is assigned for the two 5s at the end of the guess. Similarly, if the code contains more of a value than the guess, only as many of that value as are in the guess can contribute to the pin count. For example in the case of code 550033 and guess 24311 the 3 in the guess only earns one white pin even though the solution code contains two 3s.

Terminal

```
What is your guess (q to quit, wq to save and quit): 153455
+------------------+-----+
| o  o  o  o  o  o | R W |
+------------------+-----+
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| 1  5  3  4  5  5 | 1 1 |
+------------------+-----+
What is your guess (q to quit, wq to save and quit):
```

2.2 Ending the Game

The game ends when the player makes a correct guess or after their 10th incorrect guess. In either case the game board is displayed one more time with the solution code revealed in the top row. The game will also end if the player enters “q” to quit or “wq” to save and quit instead of a guess. After the game ends, the player is returned to the main menu.

Terminal

```
What is your guess (q to quit, wq to save and quit): 550022
+------------------+-----+
| 5  5  0  0  2  2 | R W |
+------------------+-----+
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| 5  5  0  0  2  2 | 6 0 |
| 1  5  3  4  5  5 | 1 1 |
+------------------+-----+
Congratulations, you broke the lock!
The grades are safe!
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice:
```

Terminal

```
What is your guess (q to quit, wq to save and quit): 550122
+------------------+-----+
| 5  5  0  0  2  2 | R W |
+------------------+-----+
| 5  5  0  1  2  2 | 5 0 |
| 5  5  1  1  2  3 | 3 0 |
| 5  4  3  2  1  0 | 1 2 |
| 5  5  4  4  3  3 | 2 0 |
| 1  1  2  2  3  3 | 0 2 |
| 1  2  3  1  2  3 | 1 1 |
| 1  1  1  2  2  2 | 2 0 |
| 5  4  3  2  1  0 | 1 2 |
| 0  1  2  3  4  5 | 0 3 |
| 1  5  3  4  5  5 | 1 1 |
+------------------+-----+
You hear a machine yell OUT OF TRIES!
  ...
Is that burning you smell?
  ...
OH, NO! It looks like IU has destroyed all the EBEC grades!

Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice:
```

Terminal

```
What is your guess (q to quit, wq to save and quit): q
Ending Game.
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice:
```

2.3 Saving Games

When the player enters “wq” to save the game, the program displays a save game menu. This menu allows the player to choose one of three save slots in which to save the game. Each slot displays the name, date, and time of the last save to that slot. If a slot has never been used, it displays “empty”. The player selects a save slot by its number (1, 2, or 3). If the player enters “c” the save is cancelled, they are returned to the active game. If the player enters an invalid choice, they are prompted to try again. After selecting a valid save slot, the player is prompted to enter their name. Names containing any characters other than letters, numbers, and spaces are rejected, and the player is prompted to try again. Once a valid name is entered, the game state is saved such that saved games are persistent between program runs.

Hint:

- The date and time can be captured using Python’s `datetime` module. For example: `datetime.datetime.now().isoformat(timespec="seconds")`.

Terminal

```
What is your guess (q to quit, wq to save and quit): wq
Files:
 1: empty
 2: Tim the Enchanter - Time: 2024-03-26T11:45:52
 3: empty
What save would you like to overwrite (1, 2, 3, or c to cancel):
```

Terminal

```
What save would you like to overwrite (1, 2, 3, or c to cancel): c
cancelled
+------------------+-----+
| o  o  o  o  o  o | R W |
+------------------+-----+
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| 5  0  4  0  o  o | 2 1 |
| 5  0  4  1  o  o | 1 1 |
| 4  4  5  5  o  o | 0 2 |
| 1  2  3  1  2  3 | 1 1 |
| 3  4  5  2  o  o | 0 2 |
| 1  2  3  4  5  o | 0 2 |
+------------------+-----+
What is your guess (q to quit, wq to save and quit):
```

Terminal

```
What save would you like to overwrite (1, 2, 3, or c to cancel): 4
That is an invalid selection.
What save would you like to overwrite (1, 2, 3, or c to cancel): spam
That is an invalid selection.
What save would you like to overwrite (1, 2, 3, or c to cancel):
```

Terminal

```
What save would you like to overwrite (1, 2, 3, or c to cancel): 3
What is your name (no special characters): $pam
That is an invalid name.
What is your name (no special characters): King Arthur
Game saved in slot 3 as King Arthur.
Ending Game.
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice:
```

3 Load Game

The load game menu option takes the player to a menu similar to the save game menu. This menu allows the player to choose to load a saved game from one of three save slots. Typing “c” in this menu takes the player back to the main menu. Other entries besides “1”, “2”, and “3” result in an error message and the player is prompted to choose again. If the player chooses an empty slot, they are informed and prompted to try again. Once a valid save slot is selected, the game is loaded, displaying the game board as it appeared when saved. This game resumes with the same solution code and all previous guesses and clues. Note that a new solution must NOT be generated when a game is loaded.

Terminal

```
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice: 3
Files:
 1: empty
 2: Tim the Enchanter - Time: 2024-03-26T12:01:25
 3: King Arthur - Time: 2024-03-26T12:10:15
What save would you like to load (1, 2, 3, or c to cancel):
```

Terminal

```
What save would you like to load (1, 2, 3, or c to cancel): c
cancelled
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice:
```

Terminal

```
Files:
 1: empty
 2: Tim the Enchanter - Time: 2024-03-26T12:01:25
 3: King Arthur - Time: 2024-03-26T12:10:15
What save would you like to load (1, 2, 3, or c to cancel): 4
That is an invalid selection.
What save would you like to load (1, 2, 3, or c to cancel): spam
That is an invalid selection.
What save would you like to load (1, 2, 3, or c to cancel): 1
That file is empty!
What save would you like to load (1, 2, 3, or c to cancel): 3
+------------------+-----+
| o  o  o  o  o  o | R W |
+------------------+-----+
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| o  o  o  o  o  o | 0 0 |
| 5  0  4  0  o  o | 2 1 |
| 5  0  4  1  o  o | 1 1 |
| 4  4  5  5  o  o | 0 2 |
| 1  2  3  1  2  3 | 1 1 |
| 3  4  5  2  o  o | 0 2 |
| 1  2  3  4  5  o | 0 2 |
+------------------+-----+
What is your guess (q to quit, wq to save and quit):
```

4 Quit

The program should only end if the player chooses to quit from the main menu.

Terminal

```
Menu:
  1: Rules
  2: New Game
  3: Load Game
  4: Quit
Choice: 4
Goodbye
```

Other Requirements

Other than the function `generate_solution` described above, you are free to design your program however you like. You may use object-oriented programming techniques but are not required to do so.

Test your program thoroughly. Format your program to match the samples. Your output should closely match the sample output, character for character, including all white space and punctuation. User input in the sample has been highlighted in Pappy's Purple to distinguish it from the program's output, but your user input does not need to be colored. Save your program as `code_breakers_login.py`, where `login` is your Purdue login, and then submit it. Screenshots are **not** required.