12.1 - Pyword

Create a variation of the game Jotto (a predecessor to the hit game Wordle published in 1955). This is a word guessing game where the player attempts to guess a secret word. After each guess, the program will reveal which letters from the guess that are in the secret word and in the correct position, which are in the secret word but in the wrong position, and which are not in the secret word.

At the start of your program, the player should be presented with a main menu from which they can choose between starting a new game, viewing the hall of fame, or exiting the program. If the player enters an invalid choice, the program should display an appropriate error message and let the player try again until a valid selection is made. The program should end if the player chooses to exit.

A new game will consist of three rounds, with each round using a different secret five letter word. The secret words should be selected from the words in the provided `words.txt` file. To make sure the words in each round are different, write a function named `pick_game_words` to choose the three secret words in the game. This function will take a list of all of the words in the `words.txt` file as its only argument and should return a list of three of those words chosen at random. The returned words must all be lowercase and must be used in the same order as they appear in the returned list. No other portion of your program should involve random chance.

In each round, the player will then be given up to a maximum of six turns to try to guess the secret word. Each guess can be any combination of five letters. If the player enters an invalid guess, the program should display an invalid input message and let the player try again without losing a turn.

After each valid guess, the program will mark the letters of the guess with one of three characters. Letters that are in the secret word *and* in the correct position should be marked with an "`!`", letters that are in the secret word but are in the wrong position should be marked with a "`?`", and letters that not in the secret word should be marked with an "`X`" (Note that a letter can be marked with a "`?`" more times than that letter appears in the secret word). The program will also provide a marked alphabet where each letter is marked with either a blank space if not yet used in any guess, an `!` if used in any guess in the correct location, a `?` if used in any guess in the wrong location *and* never used in the correct location, or an `X` if used an any guess but the letter is not in the secret word.

A round ends when the player successfully guesses the secret word or they run out of turns. At the end of each round, the player will earn points based on how many turns they used to guess the word. The formula for determining the number of points earned is

$$\text{points} = 2^{(\,\text{maximum\_turns} - \text{turns\_used}\,)}.$$

If the player runs out of turns, the program should display a failure message and reveal the secret word. However, if the player guesses the secret word, the program should display an accolade and the number of points they earned. A table showing the appropriate accolade and number of points earned based on the number of turns remaining is shown below in Table 1. Note that guessing correctly on the sixth turn earns the player 1 point. If the player fails to guess correctly after six turns, they will earn zero points.

Turns Left	Accolade	Points
6	Impossible	64
5	Genius	32
4	Magnificent	16
3	Impressive	8
2	Splendid	4
1	Great	2
0	Phew	1

Table 1: Accolades and points per turn remaining.

No matter how the round ends, once it is over, a summary of the marking hints provided for each guess should be displayed. One line of marks for each guess. This format makes it easy for a player to share their performance with others without revealing the word they were trying to guess.

Round two starts immediately after the end of round one, and round three starts immediately after the end of round three. At the end of the third round, the player's total score for the game will be the sum of their scores from each of the three rounds. If this score is larger than the lowest score in the Hall of Fame, the players score and name should be added at the appropriate position in the Hall of Fame. Then a message should be displayed that congratulates the player and shows the scores from the Hall of Fame as a table with columns for rank, score, and player name. The table should be sorted from highest to lowest score. If multiple players have the same score, they should be listed in the order that their scores were achieved (i.e. the first player to score 20 points should appear above any subsequent players who score 20 points). Once the game is complete, the program returns to the main menu.

From the main menu, the player can also choose to view the Hall of Fame. This option displays the same table presented to the user when they achieve a high score. Hall of fame records should be saved in a file named `hall_of_fame.txt`. Each line in the file should be a player's score followed by a comma, a space, and the player's name. Only the top ten scores should be kept. The data from the file will be loaded when your program begins, and should be updated with any new high scores before your program ends.

Other than the function `pick_game_words` 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 below. Your output should exactly 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 `pyword_login.py`, where `login` is your Purdue login and then submit it. Screenshots are **not** required.

Terminal

```
$ python pyword_login.py
Welcome to PyWord.

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? quit

Invalid choice. Please try again.

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? spam

Invalid choice. Please try again.

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? #$%

Invalid choice. Please try again.

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? 3
Goodbye.
```

Note that incorrect input is handled by displaying an error message and then presenting the main menu again.

Terminal

```
$ python pyword_login.py
Welcome to PyWord.

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? 2

--- Hall of Fame ---
 ## : Score : Player
  1 :    64 : Graham
  2 :    52 : Graham
  3 :    40 : Graham
  4 :    38 : Michael
  5 :    25 : Michael

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? 3
Goodbye.
```

Note that the hall of fame can of less than 10 entries, but it cannot of more than 10 entries.

Terminal

```
$ python pyword_login.py
Welcome to PyWord.

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? 1
Enter your player name: Eric

Round 1:
1? ESHzA
   XXXX?     ?   X  X          X      X
   eshza     abcdefghijklmnopqrstuvwxyz
```

Terminal

```
2? briar
   !!!!!     !!  X  X!        !X      X
   briar     abcdefghijklmnopqrstuvwxyz
Magnificent! You earned 16 points this round.
Round 1 summary:
   XXXX?
   !!!!!

Round 2:
1? sbooC
   XX??X      XX           ?   X
   sbooc     abcdefghijklmnopqrstuvwxyz
2? ss<%5

Invalid guess. Please only enter letters.

2? ];|{;^~

Invalid guess. Please enter exactly 5 characters.

2? rRipZ
   XX?XX      XX     ?     ?X XX      X
   rripz     abcdefghijklmnopqrstuvwxyz
3? movie
   !!!!!      XX !   !   ! !X XX  !   X
   movie     abcdefghijklmnopqrstuvwxyz
Impressive! You earned 8 points this round.
Round 2 summary:
   XX??X
   XX?XX
   !!!!!
```

Note that invalid guesses can occur any number of times and on any turn and that they should not count as a turn.

Terminal

```
Round 3:
1? qlika
   XXX??     ?       X ?X    X
   qlika     abcdefghijklmnopqrstuvwxyz
2? vrnpi
   XXXXX     ?       X ?X X XXX   X
   vrnpi     abcdefghijklmnopqrstuvwxyz
```

Terminal

```
3? peawh
   XX!??     !   X  ?X ?X X XXX   X?
   peawh     abcdefghijklmnopqrstuvwxyz
4? aragk
   ?X!X!     !   X X?X !X X XXX   X?
   aragk     abcdefghijklmnopqrstuvwxyz
5? achaa
   ?????     ! ? X X?X !X X XXX   X?
   achaa     abcdefghijklmnopqrstuvwxyz
6? rhahk
   X!!?!     ! ? X X!X !X X XXX   X?
   rhahk     abcdefghijklmnopqrstuvwxyz
You ran out of tries.
The word was whack.
Round 3 summary:
   XXX??
   XXXXX
   XX!??
   ?X!X!
   ?????
   X!!?!

Way to go Eric!
You earned a total of 24 points and made it into the Hall of Fame!

--- Hall of Fame ---
 ## : Score : Player
  1 :    64 : Graham
  2 :    52 : Graham
  3 :    40 : Graham
  4 :    38 : Michael
  5 :    25 : Michael
  6 :    24 : Eric

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? 3
Goodbye.
```

Note that the game ends after three rounds and that the Hall of Fame is persistent between program runs.

Terminal

```
6? whelp
   !!!!!     X  X!XX!   !   !  X   !
   whelp     abcdefghijklmnopqrstuvwxyz
Phew! You earned 1 points this round.

You earned a total of 1 points.

----- Main Menu -----
1. New Game
2. See Hall of Fame
3. Quit

What would you like to do? 3
Goodbye.
```

Note that if the player does not earn enough points to make it into the Hall of Fame, it is not shown when the game ends.