Audit Access Expires 5 de ago de 2020
You lose all access to this course, including your progress, on 5 de ago de 2020.

Problem 6 - Playing a Game

Problem Set due Jul 16, 2020 20:30 -03

Problem 6 - Playing a Game

15/15 points (graded)

A game consists of playing multiple hands. We need to implement one final function to complete our word-game program. Write the code that implements the `playGame` function. You should remove the code that is currently uncommented in the `playGame` body. Read through the specification and make sure you understand what this function accomplishes. For the game, you should use the `HAND_SIZE` constant to determine the number of cards in a hand.

Testing: Try out this implementation as if you were playing the game. Try out different values for `HAND_SIZE` with your program, and be sure that you can play the wordgame with different hand sizes by modifying *only* the variable `HAND_SIZE`.

Sample Output

Here is how the game output should look...

```
Loading word list from file...
   83667 words loaded.
Enter n to deal a new hand, r to replay the last hand, or e to end game: r
You have not played a hand yet. Please play a new hand first!

Enter n to deal a new hand, r to replay the last hand, or e to end game: n
Current Hand: p z u t t t o
Enter word, or a "." to indicate that you are finished: tot
"tot" earned 9 points. Total: 9 points

Current Hand: p z u t
Enter word, or a "." to indicate that you are finished: .
Goodbye! Total score: 9 points.

Enter n to deal a new hand, r to replay the last hand, or e to end game: r
Current Hand: p z u t t t o
Enter word, or a "." to indicate that you are finished: top
"top" earned 15 points. Total: 15 points

Current Hand: z u t t
Enter word, or a "." to indicate that you are finished: tu
Invalid word, please try again.

Current Hand: z u t t
Enter word, or a "." to indicate that you are finished: .
Goodbye! Total score: 15 points.

Enter n to deal a new hand, r to replay the last hand, or e to end game: n
Current Hand: a q w f f i p
Enter word, or a "." to indicate that you are finished: paw
"paw" earned 24 points. Total: 24 points

Current Hand: q f f i
Enter word, or a "." to indicate that you are finished: qi
"qi" earned 22 points. Total: 46 points

Current Hand: f f
Enter word, or a "." to indicate that you are finished: .
Goodbye! Total score: 46 points.

Enter n to deal a new hand, r to replay the last hand, or e to end game: n
Current Hand: a r e t i i n
Enter word, or a "." to indicate that you are finished: inertia
"inertia" earned 99 points. Total: 99 points.

Run out of letters. Total score: 99 points.

Enter n to deal a new hand, r to replay the last hand, or e to end game: x
Invalid command.
```

```
Enter n to deal a new hand, r to replay the last hand, or e to end game: e
```

Hints about the output

Be sure to inspect the above sample output carefully - very little is actually printed out in this function specifically. Most of the printed output actually comes from the code you wrote in `playHand` - be sure that your code is modular and uses function calls to the `playHand` helper function!

You should also make calls to the `dealHand` helper function. You shouldn't make calls to any other helper function that we've written so far - in fact, this function can be written in about 15-20 lines of code.

Here is the above output, with the output from `playHand` obscured:

```
Loading word list from file...
   83667 words loaded.
Enter n to deal a new hand, r to replay the last hand, or e to end game: r
You have not played a hand yet. Please play a new hand first!

Enter n to deal a new hand, r to replay the last hand, or e to end game: n
<call to playHand>

Enter n to deal a new hand, r to replay the last hand, or e to end game: n
<call to playHand>

Enter n to deal a new hand, r to replay the last hand, or e to end game: n
<call to playHand>

Enter n to deal a new hand, r to replay the last hand, or e to end game: x
Invalid command.
Enter n to deal a new hand, r to replay the last hand, or e to end game: e
```

Hopefully this hint makes the problem seem a bit more approachable.

Entering Your Code

Be sure to only paste your definition for `playGame` in the following box. Do not include any other function definitions.

A Cool Trick about 'print'

A cool trick about `print`: you can make two or more print statements print to the same line! Try out the following code. It will separate the first and second line with a space, and the second and third line with a "?" rather than putting each on a new line.

```
print('Hello', end = " ")
print('world', end="?")
print('!')
```

```
def playGame(wordList):
    """
    Allow the user to play an arbitrary number of hands.

    1) Asks the user to input 'n' or 'r' or 'e'.
      * If the user inputs 'n', let the user play a new (random) hand.
      * If the user inputs 'r', let the user play the last hand again.
      * If the user inputs 'e', exit the game.
      * If the user inputs anything else, tell them their input was invalid.

    2) When done playing the hand, repeat from step 1
    """
    hand ={}
    while True:
        escolha = input('Enter n to deal a new hand, r to replay the last hand, o
```

Press ESC then TAB or click outside of the code editor to exit

Correta

Test results

Hide output

CORRECT

Function call: playGame(<edX internal wordList>)

Test 1: Playing a single game, then quitting.

Output:

```
Enter n to deal a new hand, r to replay the last hand, or e to
end game:n
Hand passed to playHand:  a b c <playHand execution not shown
for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:e
None
```

Function call: playGame(<edX internal wordList>)

Test 2: Playing three games, then quitting.

Output:

```
Enter n to deal a new hand, r to replay the last hand, or e to
end game:n
Hand passed to playHand:  a z <playHand execution not shown for
grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:n
Hand passed to playHand:  q i <playHand execution not shown for
grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:n
Hand passed to playHand:  o d <playHand execution not shown for
grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:e
None
```

Function call: playGame(<edX internal wordList>)

Test 3: Replaying a hand.

Output:

```
Enter n to deal a new hand, r to replay the last hand, or e to
end game:n
Hand passed to playHand:  a t o b <playHand execution not shown
for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
Hand passed to playHand:  a t o b <playHand execution not shown
for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:e
None
```

Function call: playGame(<edX internal wordList>)

Test 4: Replaying a hand.

Output:

```
Enter n to deal a new hand, r to replay the last hand, or e to
end game:n
Hand passed to playHand:  l e e b t t f o z <playHand execution
not shown for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
Hand passed to playHand:  l e e b t t f o z <playHand execution
not shown for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
Hand passed to playHand:  l e e b t t f o z <playHand execution
not shown for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
Hand passed to playHand:  l e e b t t f o z <playHand execution
not shown for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:e
None
```

`Function call: playGame(<edX internal wordList>)`

Test 5: Nothing should break if I call 'r' first - you should just print a message to the user if they do this. User should be able to enter 'r' endlessly and the message should always display. (Hint: use a loop for this!)

Output:

```
Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
You have not played a hand yet. Please play a new hand first!

Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
You have not played a hand yet. Please play a new hand first!

Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
You have not played a hand yet. Please play a new hand first!

Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
You have not played a hand yet. Please play a new hand first!

Enter n to deal a new hand, r to replay the last hand, or e to
end game:r
You have not played a hand yet. Please play a new hand first!

Enter n to deal a new hand, r to replay the last hand, or e to
end game:n
Hand passed to playHand:  a a t e e p r <playHand execution not
shown for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:e
None
```

Function call: playGame(<edX internal wordList>)

Test 6: Invalid input test. If the input is invalid, a message - 'Invalid command.' - should print out.

Output:

```
Enter n to deal a new hand, r to replay the last hand, or e to
end game:n
Hand passed to playHand:  q s a a a t j v i i <playHand
execution not shown for grading brevity>
Enter n to deal a new hand, r to replay the last hand, or e to
end game:x
Invalid command.
Enter n to deal a new hand, r to replay the last hand, or e to
end game:y
Invalid command.
Enter n to deal a new hand, r to replay the last hand, or e to
end game:z
Invalid command.
Enter n to deal a new hand, r to replay the last hand, or e to
end game:k
Invalid command.
Enter n to deal a new hand, r to replay the last hand, or e to
end game:s
Invalid command.
Enter n to deal a new hand, r to replay the last hand, or e to
end game:w
Invalid command.
Enter n to deal a new hand, r to replay the last hand, or e to
end game:e
None
```

Hide output

Note: the `input` function on Spyder may print an extra newline. That's ok. Do not try to move text backwards using `end='\b'` in a print statement

Enviar You have used 1 of 30 attempts

✔ Correct (15/15 points)

Problem 6 - Playing a Game

Ocultar discussão

Topic: Problem Set 4 / Problem 6

Show all posts por atividade recente

Exception 1

Used an exception! Came in very handy...

? 'main' variable

Anyone on elaborating this statement: "if __name__ == '__main__':" ? I looked it up but found the expl... 5

A Couple of Things That Tripped Me Up

You should only assign HAND_SIZE a value on your own computer. The grader provides its own test ... 3