day25 Rock Paper Scissors
Due: 6th period: Thursday 10/6
7th period: Friday 10/7

Today's task is to create the game "Rock Paper Scissors" in Python.

If you don't remember how to play, review the game here:

https://en.wikipedia.org/wiki/Rock%E2%80%93paper%E2%80%93scissors

My requirements for you:

- Create a repl named day 25 RPS.
- Put your name in a comment at the top.
- Your game welcomes the user with text that explains we're going to play Rock Paper Scissors, then goes into a loop and plays rock paper scissors until the user leaves the entry blank.
- Your game keeps score and reports the cumulative score at the start of every turn.
- Your game plays properly (that is, computer choices are made using random.choice(), the computer's choices are displayed, and the correct side wins as appropriate.)
- When the user chooses to quit, the final score is displayed and the user is congratulated (if the user won), taunted (if the computer won) or left with a generic whatever sort of comment (if it was a tie.)
- If the user enters something that is not r, p, or s, the game tells them that they must enter one of those characters to make a valid play and to try again. This will be an if statement, not a try/except. You can say if entry not in "rps": to do this. That makes sure that their entry was either an r, a p, or an s.

Put "import random" at the top of your program, and then use the following line deep in your program to make a random choice for the computer:

```
c = random.choice(["r","p","s"])
```

That line randomly chooses "r", "p", or "s" and stores it in the variable c. You do this for the computer's choice of play. The user should enter their choice before the computer's choice is displayed.

You should implement the game to a point where you can check that it is working correctly with one user move (for example rock) before you try to make the program work for all plays. What I mean is, make it so that you always enter rock as the user and make sure that you see the computer play all three possible values and that the correct thing happens when you do (that is, rock and rock is a tie, rock and paper you lose, rock and scissors you win). Only after you get everything working with the user playing rock each time should you move to add support for the user playing scissors and paper.

Check out my sample output on the next page.

Here is the help video (from two years ago, but even though the assignment number is different, the assignment is the same.)

Good luck, have fun!

(continued on next page)

Sample output:

```
Welcome to rock-paper-scissors.
You play the computer and we'll see who wins!

The score is: you: 0 computer: 0

Make your move: rock(r), paper(p), scissors(s)
(leave blank to quit): r
Computer also chose rock, it's a tie

The score is: you: 0 computer: 0

Make your move: rock(r), paper(p), scissors(s)
(leave blank to quit): r
Computer chose scissors, you win!

The score is: you: 1 computer: 0

Make your move: rock(r), paper(p), scissors(s)
(leave blank to quit): hello
Not a valid move, please try again.

The score is: you: 1 computer: 0

Make your move: rock(r), paper(p), scissors(s)
(leave blank to quit): r
Computer chose paper, you lose!

The score is: you: 1 computer: 1

Make your move: rock(r), paper(p), scissors(s)
(leave blank to quit): r
Computer chose scissors, you win!

The score is: you: 2 computer: 1

Make your move: rock(r), paper(p), scissors(s)
(leave blank to quit): s
Computer chose paper, you win!

The score is: you: 3 computer: 1

Make your move: rock(r), paper(p), scissors(s)
(leave blank to quit):

The final score was: you: 3 computer: 1

Congratulations, you won!
```