Quiz 4 – Name: ______________________

**Question 1.** For the following program, show the output and draw the stack as it would look, just before line 9 (the `return count` statement). You may ignore the loop variable `i` in your stack diagram.

```python
def listZeroer(L, maxval):
    count = 0
    for i in range(len(L)):
        if L[i] > maxval:
            L[i] = 0
            count += 1

    # draw stack here
    return count

def main():
    reds = [210, 50, 300, 177, 290]
    rmax = 255
    nchanged = listZeroer(reds, rmax)
    print("reds = %s" % str(reds))
    print("number changed = %d" % nchanged)
main()
```

**Question 2.** Write a function called `getInput` that asks the user to enter a choice (rock, paper, scissors, or quit) and returns the user's choice. Your function should only allow one of those 4 choices, but ignore case (i.e., Rock, ROCK, and RoCk are OK). If an invalid choice is made, print an error message and ask again, until you get a valid choice. Here is a short example (user input in **bold**):

rock/paper/scissors/quit: hello
Please enter rock, paper, scissors, or quit!
rock/paper/scissors/quit: pony
Please enter rock, paper, scissors, or quit!
rock/paper/scissors/quit: rock
Question 3. Write a function called `winner` that has two parameters: `player1` and `player2`. Each of these contains one of the following strings: "rock", "paper", or "scissors". Your function should return: 1 if `player1` is the winner, 2 if `player2` is the winner, and 0 if it is a tie. For this game, paper beats rock, scissors beats paper, and rock beats scissors. Hint: to make this function shorter, check for a tie, first.

Here are a few examples of calling this function, and what it would return:

```
winner("rock", "rock") would return 0  # tie
winner("rock", "paper") would return 2  # player2 wins
winner("paper", "rock") would return 1  # player1 wins
```

Question 4. Write a `main()` function that uses the above functions and plays the rock/paper/scissors game. The game should keep going until the user enters "quit". Each time the user enters rock/paper/scissors, the computer should choose randomly, and your program should display the results. Here is a small example of the game:

```
rock/paper/scissors/quit:  rock
I chose scissors
You win...
rock/paper/scissors/quit:  ROCK
I chose scissors
You win...
rock/paper/scissors/quit:  rock
I chose paper
*I* win!!!
rock/paper/scissors/quit:  paper
I chose paper
it's a tie...
rock/paper/scissors/quit:  quit
```