

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.

	OUTPUT:	STACK DIAGRAM:
	-----	-----
1 <code>def listZeroer(L, maxval):</code>		
2 <code>count = 0</code>		
3 <code>for i in range(len(L)):</code>		
4 <code>if L[i] &gt; maxval:</code>		
5 <code>L[i] = 0</code>		
6 <code>count += 1</code>		
7		
8 <code># draw stack here</code>		
9 <code>return count</code>		
10		
11 <code>def main():</code>		
12 <code>reds = [210,50,300,177,290]</code>		
13 <code>rmax = 255</code>		
14 <code>nchanged = listZeroer(reds,rmax)</code>		
15 <code>print("reds = %s" % str(reds))</code>		
16 <code>print("number changed = %d" % (nchanged))</code>		
17		
18 <code>main()</code>		

**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
```