Problem Set 3:

Handed out: Thursday, October 26th, 2023

Due: Thursday, November 2nd, 2023

We will be writing five programs in this problem set. Please create a program for each part in this assignment, and name your programs 'ps3_part<Letter>_name.py'.

Part A:

Write a program that asks the user how many credits they have taken. If they have taken 23 or less, print that the student is a freshman. If they have taken between 24 and 53, print that they are a sophomore. The range for juniors is 54 to 83, and for seniors it is 84 and over.

The program should do the following:

1. Ask the user for the number of credits they've taken.
2. Print out what grade they're in.

Assume that the user always enters a valid integer.

For example, if the user enters 23:

```
Enter how many credits you've taken: 23
You are a freshman.
```

For example, if the user enters 53:

```
Enter how many credits you've taken: 53
You are a sophomore.
```

For example, if the user enters 83:

```
Enter how many credits you've taken: 83
You are a junior.
```

For example, if the user enters 84:

```
Enter how many credits you've taken: 84
You are a senior.
```

Part B:

Write a multiplication game program for kids. The program should give the player three multiplication questions to do. After each question, the program should tell them whether they got it right or wrong and what the correct answer is.

The program should do the following:

1. Ask the user a multiplication question.
2. Print out whether or not they answered correctly.
3. Ask the user a multiplication question.
4. Print out whether or not they answered correctly.
5. Ask the user a multiplication question.
6. Print out whether or not they answered correctly.

Assume that the user always enters a valid number for their answer.

For example, if the user gets every question wrong:

```
What is 6 * 8?: 47
Wrong! The answer was 48.
What is 5 * 13?: 61
Wrong! The answer was 65.
What is 9 * 9?: 76
Wrong! The answer was 81.
```

For example, if the user gets every question correct:

```
What is 6 * 8?: 48
Correct!
What is 5 * 13?: 65
Correct!
What is 9 * 9?: 81
Correct!
```

Part C:

Write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times.

The program should do the following:

1. Ask the user for their name.

2. Ask the user how many times to print their name.
3. Print out the user's name the specified number of times.

Assume that the user only enters a valid number for the number of times to print their name.

For example, if the user enters "justin" and 5:

```
What is your name? Justin
How many times to print? 5
Justin
Justin
Justin
Justin
Justin
```

Part D:

Use a for loop to print a triangle. Allow the user to specify how high the triangle should be.
[Hint: print("*" * 10) prints ten asterisks, print("*" * 4) prints four asterisks, etc]

The program should do the following:

1. Ask the user for how high the triangle should be.
2. Print out the triangle.

Assume that the user always enters a valid integer.

For example, if the user enters 4:

```
How high should the triangle be? 4
*
**
***
****
```

Part E:

Use a for loop to print an **upside-down** triangle. Allow the user to specify how high the triangle should be.

[Hint: print("*" * 10) prints ten asterisks, print("*" * 4) prints four asterisks, etc]

The program should do the following:

1. Ask the user for how high the triangle should be.
2. Print out the triangle.

Assume that the user always enters a valid integer.

For example, if the user enters 4:

```
How high should the triangle be? 4
****
***
**
*
```

Part F:

Use a for loop to write a program that asks the user to enter 5 integers and prints out the maximum of the 5 inputs.

Assume that the user always enter a valid integers.

For example:

```
Enter a number: 4
Enter a number: 27
Enter a number: 7
Enter a number: 28
Enter a number: 5
The maximum number was 28.
```

Part G:

In this program, you will create a number guessing game.

Hint: to generate a random number, first add `import random` to the beginning of your program like so:

Then to generate a random number between 1 and 10:

The program should do the following:

1. Ask the user to guess what the secret number is.
2. If the user guesses incorrectly 3 times, then they lose.
3. If the user guesses correctly, then they immediately win (even if they still had guesses remaining).

Assume that the players always enter valid numbers for the secret number and their guesses.

For example, if 7 was the secret number, and Player B guesses incorrectly 3 times, then the output should look like the following:

```
Enter your guess (1-10)? 9
Enter your guess (1-10)? 1
Enter your guess (1-10)? 3
You lose! The secret number was 7.
```

For example, if 7 was the secret number, and Player B guesses correctly on their first guess, then the output should look like the following:

```
Enter your guess (1-10)? 7
You win! The secret number was 7.
```

Part H (extra credit):

Write a program that lets the user play Rock-Paper-Scissors against the computer. There should be five rounds, and after those five rounds, your program should print out who won and lost or that there is a tie.

Hint: to generate a random computer move, first add `import random` to the beginning of your program like so:

```
import random
```

Then to generate a random move for the computer:

```
random_move = random.choice(["rock", "paper", "scissors"])
```

This code will assign a random choice to the variable random_move.

Assume that the user always enters a "rock", "paper", or "scissors" for their answer.

For example:

```
rock, paper, scissors: rock
You lose!
rock, paper, scissors: rock
You win!
rock, paper, scissors: paper
You tied!
rock, paper, scissors: scissors
You win!
rock, paper, scissors: paper
You win!

You won 3 time(s)
You lost 1 time(s)
You tied 1 time(s)
```

Grading:

Part A - 2 points

Part B - 2 points

Part C - 2 points

Part D - 2 points

Part E - 2 points

Part F - 2 points

Part G - 2 points

Part H - 2 bonus points