CS 115 - Introduction to Programming in Python

Lab 02

Lab Objectives: Strings, Loops, Nested Loops.

Q1: Write a program `Lab02_Q1.py` that does the following:

a. Uses a **for** loop to calculate and display the first 10 powers of 5.

b. Inputs a lower bound and upper bound from the user. Then, it uses a **while** loop to find and display the powers of 5 in the user specified range. Assume that lower bound and upper bound inputs are valid positive integers.

Sample Run 1:

```
The first 10 powers of 5: 1, 5, 25, 125, 625, 3125, 15625, 78125, 390625, 1953125

Enter lower bound: 100
Enter upper bound: 10000
Powers of 5 between 100 and 10000:
5 ** 3  is 125
5 ** 4  is 625
5 ** 5  is 3125
```

Sample Run 2:

```
The first 10 powers of 5: 1, 5, 25, 125, 625, 3125, 15625, 78125, 390625, 1953125

Enter lower bound: 1000000
Enter upper bound: 1000000000
Powers of 5 between 1000000 and 1000000000:
5 ** 9  is 1953125
5 ** 10  is 9765625
5 ** 11  is 48828125
5 ** 12  is 244140625
```

Q2: Write a program `Lab02_Q2.py` to input a string from the user, and creates a new string that replaces each non-alphabetic character in the original string with an asterisk (*). You should solve this problem in 2 ways, first use a for loop that iterates through each character in the string, the second should use a for loop that iterates through a range. ***Hint:*** You can invoke the `is_alpha()` function on a string, and it will return `True` if the string is alphabetic, `False` if not. For example `'4'.isalpha()` returns `False`, but `'y'.isalpha()` returns `True`.

Sample Run:

```
Enter a string: abc?xyz43.klm,uvw   AB12
New string: abc*xyz***klm*uvw***AB**
New string: abc*xyz***klm*uvw***AB**
```

Q3: Write a program, `Lab02_Q3.py`, which prompts the user to enter two integers until the user enters 0 for both. It inputs a lower bound and upper bound from the user and finds and displays the number and average of all numbers between the lower and upper bound (inclusive) that are divisible by 7, but that are not a multiple of 5. Your program should validate that the lower bound is less than the upper bound.

Sample Run:

```
Enter lower bound: 0
Enter upper bound: 100
12 values are divisible by 7 but not 5 between 0 and 100 , inclusive
Average of those values:  52.5

Enter lower bound: 2000
Enter upper bound: 4000
229 values are divisible by 7 but not 5 between 2000 and 4000 , inclusive
Average of those values:  2996.886462882096

Enter lower bound: 0
Enter upper bound: 100
12 values are divisible by 7 but not 5 between 0 and 100 , inclusive
Average of those values:  52.5

Enter lower bound: 100
```

```
Enter upper bound: 500
45 values are divisible by 7 but not 5 between 100 and 500 , inclusive
Average of those values:  301.93333333333334

Enter lower bound: 500
Enter upper bound: 100
Invalid range

Enter lower bound: 200
Enter upper bound: 0
Invalid range


Enter lower bound: 200
Enter upper bound: 400
23 values are divisible by 7 but not 5 between 200 and 400 , inclusive
Average of those values:  301.9130434782609


Enter lower bound: 0
Enter upper bound: 10000
1143 values are divisible by 7 but not 5 between 0 and 10000 , inclusive
Average of those values:  5000.627296587926

Enter lower bound: 0
Enter upper bound: 0
bye!
```