Project 4 – Loops

Loops — statements that repeat

Keep in mind

❶ **Before:** What happens **before** the loop starts

❷ **During:** What happens **during** the loop

❸ **After:** What happens **after** the loop ends.

Two classic problems:

1. My loop **never ends**!
   You must make sure your statements during the loop include something to update the variable that gets checked for exit.

2. My loop **never gets started**.
   Make sure you prepare for the loop:
   Zero totals and counters
   Do you need to get an initial value before you enter the loop?

Overview

P4_repeat 4 points
Ask the user for a number
Repeat a block of statements as many times as the user requested.

P4_for 6 points
Use each item from a list;
No change to item.

Modify the contents of a list
Examine each item in a list using an index to get the item by number from the list.

P4_while1 7 points
Debug a loop that does not work correctly.

P4_while2 8 points
Create a loop that adds items until some exit condition is met.

Notes on loops are available in Canvas > Modules > Week4 > **Loop_Notes.pdf**

P4_repeat

P4_repeat.py **4 points**

Create two variables
n The number of times to repeat an action
x A number you will use to calculate some answers

Set n to a number from 10 to 20 a user provides
Set x to 7

1 point
Ask for a number from 10 to 20, such as 15; assign that value to **n**.

1 point
Use a for count in range(n) to repeat some statements n times.

1 point
In the body of your loop, calculate answer as the result of raising x to the count power. print x, the power to which it is raised, and the result of raising x to a power.

1 point
Print x, the power, the answer.

Output:

```
Enter a number from 10 - 20: 15
Repeat 15 times
Powers of 7

7 ** 0 is 1
7 ** 1 is 7
7 ** 2 is 49
7 ** 3 is 343
7 ** 4 is 2401
7 ** 5 is 16807
7 ** 6 is 117649
7 ** 7 is 823543
7 ** 8 is 5764801
7 ** 9 is 40353607
7 ** 10 is 282475249
7 ** 11 is 1977326743
7 ** 12 is 13841287201
7 ** 13 is 96889010407
7 ** 14 is 678223072849
```

P4_for.py 6 points:

1 point
Create a list top_movies with the names of your top four films.

1 points
Use a for loop of the "for item in my_list:" style to print the names of your top movies, like this

```
Good movies

Star Wars
Avatar
Wizard of Oz
Gone with the Wind
```

1 point
Create this list of numbers

```
numbers = [7, 11, 30, 19, 22, -3]
```

The list has some even numbers such as 30, and some odd numbers such as 7.

If num is an integer (whole number, no decimals), you can tell if it's even this way

```
if num % 2 == 0:    # even
     print(num, "is even")
```

Your task: modify the numbers list, **doubling all the odd numbers**.

If the list was [3, 4, 5], after your program ran, it would be [6, 4, 10].

2 points
Use a for loop that uses an index that ranges from 0 up to but not including the length of the numbers list.

1 point
When finished, print the numbers list.

You'll see this:

```
Modified list
[14, 22, 30, 38, 22, -6]
```

P4_while1.py 7 points

This program is buggy. Find the bugs, fix them and run the corrected program.

```
# p4_while1.py

# Create a list of countries,
# then print each country on a new line

# find and fix the bugs

hint = "Enter name of country or 'quit': "

country_list = [ ]

nation = "None"

while country != "quit"
    nation = input(hint)
    nation = nation.title()
    country_list.append(nation)

print()
print("Countries")
for country in countries:
    print(country)

print()
print("Finished")
```

When you have fixed the bugs (keep testing!), you will get results like this when running the fixed program:

```
Enter name of country or 'quit': panamá
Enter name of country or 'quit': Peru
Enter name of country or 'quit': Monaco
Enter name of country or 'quit': Vatican City
Enter name of country or 'quit': Cayman Islands
Enter name of country or 'quit': quiT

Countries
Panamá
Peru
Monaco
Vatican City
Cayman Islands

Finished
```

P4_while2.py 8 points

Here you will keep a list of rolls of 2 dice.

1 point
Starting with an empty roll_list, keep adding 2 dice to the list; your last 2 in the list are double-6.

2 points
After you roll 6-6, add it to your list and stop.

Although there is no guarantee that your program will ever roll a double 6, each such roll has about 1 chance in 36 of coming up double-6.

After 24 rolls, the odds are about 50-50 that you have **not** rolled a double-6.

After 100 rolls, there about a 6 % chance that you have **not** yet rolled a double 6.

Eventually, well before 20,000 rolls of pairs of dice, you are almost 100% sure of getting a pair of 6's

1 point
Your list **must end** with **6 6**.

2 points
When you finally roll a double-6, print the length of the list, and print the dice list too.

1 point
Remember your first non-comment statement must be
`import random`

To roll a die

```
roll_1 = random.randint(1, 6)
```

1 point
Roll 2 dice to randomly use numbers 1 through 6.

Note that rolling
1 6
6 5 will not end your dice rolling
3 4
6 6 This ends the dice rolls

Your program will look more or less like this:

```
Rolled 20 dice as 10 pairs of dice

[4, 2, 6, 3, 1, 2, 4, 5, 1, 2, 4, 4, 3, 4, 4, 3, 4, 5, 6, 6]
```

Another approach is possible:

Store each pair of rolls as a single 2-digit number such as 23 or 66.

Doing so can simplify the testing in your while loop.

Your program output would look somewhat like this:

```
Rolled 144 dice as 72 pairs of dice

[46, 31, 24, 36, 56, 54, 12, 36, 41, 63, 41, 45, 33, 15, 26, 11, 65, 26, 63, 42,
14, 46, 63, 21, 12, 43, 46, 44, 14, 61, 15, 35, 36, 46, 21, 14, 34, 41, 11, 62, 3
5, 23, 24, 45, 62, 34, 35, 31, 43, 53, 16, 32, 24, 61, 41, 64, 21, 26, 12, 61, 13
, 32, 43, 63, 26, 61, 41, 14, 53, 61, 52, 66]
```

Note that these sequences of rolls all must end with double 6'

Bonus 1 point

Re-write your while loop as a **`while True:`** loop, with a **break.**