Loops and Functions

Andrew Rosen

For each of the following problems, write a function that solves the problem. Demo each function you write by calling it.

Hints 1: Creating a Function in Python

You can create your own functions in Python. Let's look at the following program.

```
def manyHellos(n):
        for i in range(n):
                print("Hello")


hellos = int(input("How many hellos do you want?"))
manyHellos(hellos)
```

This program will print out "Hello" a number of times equal to the user's input. The first line defines a new function (command) for your python program, called `manyHellos`. The `manyHellos` function requires the user to pass in a number as an argument in order for it to work. By the way, you shouldn't use `hellos` as a variable in your program; that will tell me you're just copy/pasting. Also note that the input statement is outside the function; this is ideal place to put it since it can easily be replaced by a literal for easy testing without changing the function.

If you need to review how to write functions, refer to the linked readings: `https://runestone.academy/runestone/assignments/doAssignment?assignment_id=42928`

Hints 2: Printing with sep and end

There are three special things we can do with print in python.

Printing will multiple arguments

The first is giving a print function call multiple arguments:

```
print("Hello",'World', 1)
```

This will print `Hello World 1`. *By default*, your arguments are printed with a single space between them.

sep

We can use the `sep` parameter to override the default behavior of printing multiple arguments. Let learn by looking at a few examples:

```
print("Hello",'World', 1)                 # will print Hello World 1
print("Hello",'World', 1, sep='_')        # will print Hello_World_1
print("Hello",'World', 1, sep='taco')     # will print HellotacoWorldtaco1
print("Hello",'World', 1, sep='')         # will print HelloWorld1
print("Hello",'World', 1, sep=' ')        # the default behavior
                                          # will print Hello World 1
```

end

We can change how print ends a line by using the `end` parameter.

```
for i in range(5):
        print(i,end="") # will print 01234
print() #goes to a new line of output
```

You can combine `sep` and `end` in the same statement if needed.

1 99 Bottles of Beer

Write a function that uses a for loop to print out the the lyrics of the infamous "99 Bottles of Beer on the Wall" drinking song. However, this function should take in an `int` as a parameter and start the lyrics from there. For example, if the function is called with 10 as the parameter, the output should be:

```
10 bottles of beer on the wall, 10 bottles of beer
Take one down, pass it around, 9 bottles of beer on the wall

9 bottles of beer on the wall, 9 bottles of beer
Take one down, pass it around, 8 bottles of beer on the wall

... (output continues in the same pattern) ...

1 bottles of beer on the wall, 1 bottles of beer
Take one down, pass it around, 0 bottles of beer on the wall
```

2 Multiplication Table

Write a function which, given an integer n as an input, prints out an $n \times n$ multiplication table.

If n is 4, print out a 4×4 multiplication table like below

```
1       2       3       4
2       4       6       8
3       6       9       12
4       8       12      16
```

If n is 5, you want to print out:

```
1       2       3       4       5
2       4       6       8       10
3       6       9       12      15
4       8       12      16      20
5       10      15      20      25
```

3 Summation of squares

Write a function which, given an integer n, uses a for loop to print out the sum of all numbers squared from 1 to n. For example, if the given integer is 5, the program should print out 55, as $1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$.

4 Hourglass

Write a function that creates the following figure of an hourglass. This function takes no inputs.

```
|""""""""""|
 \::::::::/
  \::::::/
   \::::/
    \::/
     ||
    /::\
   /::::\
  /::::::\
 /::::::::\
|""""""""""|
```

Use for loops to print out spaces and colons. I highly recommend writing 2 separate loops, one for the top part and one for bottom part.

5 Slash Figure

Write a function, which given an `int` n, prints out a slash-based ASCII art of size n. Below is an example of what the output looks like at size 4:

```
!!!!!!!!!!!!!!
\\!!!!!!!!!!//
\\\\!!!!!!////
\\\\\\!!//////
```

And size 6

```
!!!!!!!!!!!!!!!!!!!!!!
\\!!!!!!!!!!!!!!!!!!//
\\\\!!!!!!!!!!!!!!////
\\\\\\!!!!!!!!!!//////
\\\\\\\\!!!!!!////////
\\\\\\\\\\!!//////////
```

And size 7:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!
\\!!!!!!!!!!!!!!!!!!!!!!//
\\\\!!!!!!!!!!!!!!!!!!////
\\\\\\!!!!!!!!!!!!!!//////
\\\\\\\\!!!!!!!!!!////////
\\\\\\\\\\!!!!!!//////////
\\\\\\\\\\\\!!////////////
```

6 Grading

Each problem is worth 20 points, broken down as follows:

12 points The problem is solved as directed. Partial credit may be given for partial solutions at the grader's discretion.

3 points The code is properly indented and easy to read.

5 points The problem is in a function.