UMBC
AN HONORS UNIVERSITY IN MARYLAND

CMSC 201 Spring 2017
Homework 4 – For Loops

Assignment: Homework 4 – For Loops
Due Date: Monday, February 29th, 2015 by 8:59:59 PM
Value: 40 points

Homework 4 is designed to help you practice using **`for`** loops, branching selection structures, strings, and mathematical operators. More importantly, you will be solving problems using algorithms you create and code yourself.

Remember to enable Python 3 before you run your programs:

```
scl enable python33 bash
```

Instructions

In this homework, we will be doing a series of exercises designed to make you practice using **`for`** loops, control statements like **`if/else`**, mathematical operators, and algorithmic thinking. Each one of these exercises should be in a **separate python file**. For this assignment, you may assume that all the input you get will be of the correct type (*e.g.,* if you ask the user for a whole number, they will give you an integer).

For this assignment, you'll need to follow the class coding standards, a set of rules designed to make your code clear and readable. The class coding standards are on Blackboard under "Course Documents" in a file titled "CMSC 201 - Python Coding Standards."
You will lose major points if you do not following the 201 coding standards.

A very important piece of following the coding standards is writing a complete **file header comment block**. Make sure that each file has a comment block at the top (see the coding standards document for an example).

NOTE: **You must use `main()` in each of your files.**

Details

Homework 4 is broken up into five parts. **Make sure to complete all 5 parts.**

NOTE: Your filenames for this homework must match the given ones exactly.

And remember, filenames are case sensitive.

hw4_part1.py **(Worth 5 points)**

For this part of the homework you will create a multiplication table.

Build a program that prompts the user for two integers. The first integer is the number for which you wish to make a multiplication table; the second is how far you want the table to go.

You must use a `for` loop – if you do not, you will lose significant points.

The output of the program should be the number, the multiplication sign, the number you are multiplying it by, the equal sign, and the product.

NOTE: The output should not have any spaces between each piece of text. In other words, your program should print "`3*0=0`" not "`3 * 0 = 0`".
(HINT: Concatenation can help you create strings without spaces.)

Here is some sample output, with the user input in blue.

```
bash-4.1$ python hw4_part1.py
Enter the base number: 3
Enter the max number: 2
3*0=0
3*1=3
3*2=6
bash-4.1$ python hw4_part1.py
Enter the base number: 6
Enter the max number: 4
6*0=0
6*1=6
6*2=12
6*3=18
6*4=24
```

hw4_part2.py **(Worth 6 points)**

For this part of the homework you will write code to output every third letter from a string given to you by the user.

Write code that uses **`for`** loops to create a new string that contains every third character from the original input string. Your new string must be printed on one line.

You must use a **`for`** loop – if you do not, you will lose significant points.

Here is some sample output, with the user input in blue.

```
bash-4.1$ python hw4_part2.py
Please enter a sentence: I am a Chesapeake Bay Retriever.
What is my name?
Original sentence:
I am a Chesapeake Bay Retriever. What is my name?
Every third letter:
Im epkB terWtsya?
```

hw4_part3.py **(Worth 10 points)**

Recently, a group of friends has invited you to participate in the Polar Bear Plunge to raise money and awareness for the Special Olympics. As such, you are required to secure donations for the charity.

For 4 hours, you are going to plunge in and out of the water. Donations work on a per plunge basis: someone pledges to pay a specific amount for each plunge you take. The more plunges, the more money in donations.

Your need to create a program that does the following, **in this exact order**:

1. Get the number of donations
2. Get the value of each donation
3. Get the number of plunges
4. Calculate and display the total amount donated to charity

(WARNING: This part of the homework is the most challenging, so budget plenty of time and brain power. And read the instructions carefully!)

(HINT: You will need to use the **`range`** *function for this exercise!)*

Here is some sample output, with the user input in blue.
(Yours does not have to match this exactly, but it should be similar.)

```
bash-4.1$ python hw4_part3.py
How many pledges did you get? 3
What is the value of donation 1: 0.85
What is the value of donation 2: 1.25
What is the value of donation 3: 1.05
How many plunges did you do? 22
Based on your 22 plunges you earned $69.30000000000001 for
the charity.
```

hw4_part4.py **(Worth 8 points)**

Next, you are going to create a simple program that counts the number of times vowels appear in a string. For our purposes, the letters that count as vowels are **`a e i o u y`**.

First, you must ask the user to enter a string to search through. Then, you need to output how many total vowels appear in the string. (You do not need to count how many "**`a`**"s, "**`e`**"s, etc. – just the total number of vowels.)

IMPORTANT: Your program must not be case sensitive! It must count every instance of vowels in the string, whether they appear in upper or lower case.

(HINT: How you write the code to check for the vowels is up to you! But don't forget that you can nest one **`for`** *loop inside another, and that the Boolean operators "***`or`***" and "***`and`***" exist.)*

You must use **at least** one **`for`** loop for this question.

You may **not** use any built-in Python methods to check for vowels in the string. Remember, you should only be using material we have covered in lecture and lab. If you aren't sure if you're allowed to use something, ask a TA or instructor.

Here is some sample output, with the user input in blue.
(Yours does not have to match this exactly, but it should be similar.)

```
bash-4.1$ python hw4_part4.py
Please enter a string: curious quandaries
There are 9 vowels in the string 'curious quandaries'

bash-4.1$ python hw4_part4.py
Please enter a string: Y M C A
There are 2 vowels in the string 'Y M C A'
```

hw4_part5.py **(Worth 6 points)**

Write a program that prints the numbers from 1 to 100 (inclusive), one per line. However, there are three special cases where instead of printing the number, you print a message instead:

1. If the number you would print is **divisible by 3**, print the message:
   Better three hours too soon than a minute too late.
2. If the number you would print is **divisible by 5**, print the message:
   Where do you see yourself in five years?
3. If the number you would print is **divisible by 3 and 5**, instead print out:
   In the future, everyone will be world famous for 15 minutes.

Make sure to print the exact strings given above!

Here is some sample output, up to the number 22.

```
bash-4.1$ python hw4_part5.py
1
2
Better three hours too soon than a minute too late.
4
Where do you see yourself in five years?
Better three hours too soon than a minute too late.
7
8
Better three hours too soon than a minute too late.
Where do you see yourself in five years?
11
Better three hours too soon than a minute too late.
13
14
In the future, everyone will be world famous for 15 minutes.
16
17
Better three hours too soon than a minute too late.
19
Where do you see yourself in five years?
Better three hours too soon than a minute too late.
22
```

Submitting

Once all five parts of your Homework 4 are complete, it is time to turn them in with the **`submit`** command.

Don't forget to complete the header block comment for each file! Make sure that you updated the header block's file name and description for each file.

You must be logged into your GL account, and you must be in the same directory as the Homework 4 files. To double check this, you can type **`ls`**.

```
linux1[3]% ls
hw4_part1.py  hw4_part3.py  hw4_part5.py
hw4_part2.py  hw4_part4.py
linux1[4]%
```

To submit your files, we use the **`submit`** command, where the class is **`cs201`**, and the assignment is **`HW4`**. Type in (all on one line)
`submit cs201 HW4 hw4_part1.py hw4_part2.py hw4_part3.py hw4_part4.py hw4_part5.py`
and press enter.

```
linux1[4]% submit cs201 HW4 hw4_part1.py hw4_part2.py
hw4_part3.py hw4_part4.py hw4_part5.py
Submitting hw4_part1.py...OK
Submitting hw4_part2.py...OK
Submitting hw4_part3.py...OK
Submitting hw4_part4.py...OK
Submitting hw4_part5.py...OK
linux1[5]%
```

If you don't get a confirmation like the one above, check that you have not made any typos or errors in the command.

You can double-check that all five homework files were submitted by using the **`submitls`** command. Type in **`submitls cs201 HW4`** and hit enter.

And you're done!