UMBC
AN HONORS UNIVERSITY IN MARYLAND

CMSC 201 Fall 2015
Homework 3 – Branching

Assignment: Homework 3 – Branching
Due Date: Thursday, September 24th, 2015 by 8:59:59 PM
Value: 4% of final grade

Homework 3 is designed to help you practice branching (one-way and two-way selection structures) in a Python environment.

Remember to enable Python 3 before you run your programs:

```
/usr/bin/scl enable python33 bash
```

Instructions

In this homework, we will be doing a series of exercises designed to make you practice using variables, expressions, and if/else statements. 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 the file header comments. 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()` *as seen in the* `lab2.py` *file in the Lab 2 videos, and as discussed in class. However, you do not need to include the three print statements (for your name, the description, and the instructions) as was described in the Lab 2 videos.*

Details

Homework 3 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.

hw3_part1.py

Read in three floats from the user (you may assume the user will actually enter numbers). Print out the average of these numbers.
(NOTE: This first part of the homework does not use any branching.)

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 hw3_part1.py
Please enter a floating point number: 3.1
Please enter a floating point number: 4.2
Please enter a floating point number: 5.6

The average of the three floats is: 4.3
```

hw3_part2.py

For this part of the homework you will need to create a grade calculator.

Your program should prompt the user for these inputs, **exactly in this order:**

1. Homework weight
2. Homework grade
3. Exam Weight
4. Exam Grade
5. Discussion Weight
6. Discussion Grade

(HINT: A weighted total is computed by multiplying each grade by the corresponding weight and adding them together.)

For these inputs, you can assume the following:

- Weights will be between 0 and 1 (inclusive)
  - You don't need to check that the weights add up to 1
- Grades will be between 0 and 100 (inclusive)

Once you have computed the student's final grade, print out the numerical grade and their letter grade.

- A = 90 – 100 (any grade 90 and above)
- B = 80 – 89.9
- C = 70 – 79.9
- D = 60 – 69.9
- F = 0 – 59.9 (any grade below 60)

Here is some sample output. (Yours does not have to match this exactly.)

```
bash-4.1$ python hw3_part2.py
Please enter the homework weight: 0.3
Please enter the homework grade:  90
Please enter the exam weight: 0.5
Please enter the exam grade:  75
Please enter the discussion weight: 0.2
Please enter the discussion grade:  100

The final numerical grade is: 84.5
This earns you a B in the class.
```

hw3_part3.py

Next you are going to create some medical diagnosis software.
(WARNING: This part of the homework is the most challenging, so budget plenty of time and brain power. And read the instructions carefully!)

Your program can ask the user about four symptoms. Your program should ask the ***minimum*** number of questions needed to diagnosis the patient.
(HINT: Your program should need to ask no less than two questions and no more than three questions to diagnosis a patient.)

- Fever
- Rash
- Stuffy Nose
- Ear hurts

For these inputs, you can assume the following:

- The user will only ever enter either lowercase `y` (for "yes") or lowercase `n` (for "no")

Based on their responses, you must diagnose the user and print out your diagnosis to the screen. Here are the possibilities:

- Don't have a fever and don't have a stuffy nose: Hypochondriac
- Don't have a fever and have a stuffy nose: Head Cold
- Have a fever, don't have a rash, and ear hurts: Ear Infection
- Have a fever, don't have a rash, and ear doesn't hurt: Flu
- Have a fever and have a rash: Measles

*(HINT: Review Lecture 02 (Algorithmic Thinking) and either create a flowchart or write some pseudocode for how you want your program to work. Do **not** start coding this part without having a plan.)*

(See the next page for sample output.)

Here is some sample output for hw3_part3.py, with the user input in blue. (Yours does not have to match this exactly, but it should be similar.)

```
bash-4.1$ python hw3_part3.py
Please enter 'y' for yes and 'n' for no to the following
questions.

Do you have a fever? (y/n) y
Do you have a rash? (y/n)  y

Your diagnosis: You have the Measles.


bash-4.1$ python hw3_part3.py
Please enter 'y' for yes and 'n' for no to the following
questions.

Do you have a fever? (y/n) y
Do you have a rash? (y/n)  n
Does your ear hurt? (y/n)  y

Your diagnosis: You have an Ear Infection.
```

hw3_part4.py

For this part, you are going to ask the user about the state of two switches, and then use this information to determine the state of a generator.

For the input to these two questions, you can assume the following:

- The user will only ever enter either lowercase **`y`** (for "yes") or lowercase **`n`** (for "no")

If the user enters that both switches are in the same state (both on or both off), you should print "The generator is off."
If the user has answered that one switch is on and one switch is off, you should print "The generator is on."

IMPORTANT: You can only use a single `if-else` statement for this part. (The **`if`** statement can have a combination of multiple **`and`**s and **`or`**s.)

Here is some sample output. (Yours does not have to match this exactly.)

```
bash-4.1$ python hw3_part4.py
Please enter 'y' for yes and 'n' for no.

Is the first switch on?  (y/n) y
Is the second switch on? (y/n) y

The generator is off.
```

hw3_part5.py

Finally, we will create a (very simplified) day of the week calculator.

Ask the user to enter a day of the month.

For the input to this question, you can assume the following:

- The user will enter a number.
  - You can assume that the number will be an integer.
  - You cannot assume that the number will be valid!

We will assume the month starts on Monday and has 31 days.
If the day of the month the user entered is not a valid day of the month (less than 1 or greater than 31), print "Invalid day" to the user.
Otherwise print the day of the week that day falls on. For instance, the 2nd would be a Tuesday, the 10th would be a Wednesday, etc.

Here is some sample output. (Yours does not have to match this exactly.)

```
bash-4.1$ python hw3_part5.py
Please enter the day of the month: 0
Invalid day

bash-4.1$ python hw3_part5.py
Please enter the day of the month: 9
Today is a Tuesday!

bash-4.1$ python hw3_part5.py
Please enter the day of the month: 24
Today is a Wednesday!
```

Submitting

Once all five part of your Homework 3 is 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 3 files. To double check this, you can type **`ls`**.

```
linux1[3]% ls
hw3_part1.py  hw3_part3.py  hw3_part5.py
hw3_part2.py  hw3_part4.py
linux1[4]%
```

To submit your files, we use the **`submit`** command, where the class is **`cs201`**, and the assignment is **`HW3`**. Type in (all on one line)
`submit cs201 HW3 hw3_part1.py hw3_part2.py hw3_part3.py hw3_part4.py hw3_part5.py`
and press enter.

```
linux1[4]% submit cs201 HW3 hw3_part1.py hw3_part2.py
hw3_part3.py hw3_part4.py hw3_part5.py
Submitting hw3_part1.py...OK
Submitting hw3_part2.py...OK
Submitting hw3_part3.py...OK
Submitting hw3_part4.py...OK
Submitting hw3_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 HW3`** and hit enter.

And you're done!