CMSC 245 Principles of Programming Languages
Lab#6 Python Scripts

In this lab we will learn how to write Python **scripts** in Linux/Bash. A script is a text file containing a sequence of shell commands. The script file can then be executed as a shell command. Upon invocation of a script, all the commands in the script file are carried out. Scrips are useful for automating a sequence of tasks and then using a single command to carry out all of them. We will write a Python program and learn how to convert it into a script.

First, here is the program (stored in a file **`collect.py`**):

```
from random import randint
import math
import sys

def collect(c):
    """Performs one trial of collecting c cards from candy wrappers.
         Returns the number of candy bought to collect c cards.
    """
    nBought = 0
    collected = set()
    while len(collected) < c:
        candy = randint(1,c)
        nBought = nBought + 1
        collected.add(candy)
    return nBought

def main():
    """Input is from the command line in the form:
        $ ./collect.py n t
        n is the number of candy pics to collect
        t is the number of trials.
        Outputs the average # of candies bought to collect n pics after t trials.
    """
    n = int(sys.argv[1])
    trials = int(sys.argv[2])

    sum = 0
    for in in range(trials):
        sum = sum + collect(n)
    print(f"Average for {n} candies is {ceil(float(sum)/trials)}.")

main()
```

This is the program from your Lab#4/Assignment#4. To run the above program, you used the command:

```
$ python3 collect.py 35 1000000
```

The above command runs the program and prints out the results:

```
Average for 35 candies is 146.
```

To convert the above program into a runnable Bash script, you must do the following:

1. Add the *shebang* line specifying where the Python interpreter is (its path).
2. Replace the call to **`main()`** with the top-level conditional expression.
3. Modify file permissions to make it executable.

Let us walk through the details of each.

The *shebang* line

This must be the very first line in your program file. Its purpose is to specify which Python interpreter to use. For example, on our Linux machines, the Python command is called **`python3`** and its path is:

```
/usr/local/bin/python3
```

You can find out the path to Python on your computer by using the command:

```
$ which python3
/usr/local/bin/python3
```

It outputs the path to the Python interpreter. Next, you must place the line below, as the first line in your Python program (**`collect.py`**):

```
#!/usr/local/bin/python3
```

The line begins with the '**`#!`**' characters. These are called the *shebang* (short for *shell bang*). The *shebang* line, as shown above, specifies which Python to use to execute the remainder of the file's contents.

The Top-Level Conditional Expression

Then, you must replace the very last line containing the call to **`main()`** with the following:

```
if __name__ == "__main__":
    main()
```

The above is a Python meta-programming expression. When a Python program is run as a script, its internal meta-variable: `__name__` is set to `"__main__"`. Above, we are checking this and using it to invoke the **`main()`** function. At this point, the file **`collector.py`** should have the following structure:

```
#!/usr/local/bin/python3

<code for the original program goes here>

if __name__ == "__main__":
    main()
```

Executable Files

To run the program **`collect.py`** as a Bash command, you have to make sure its *file permissions* are set to *executable*. You can do this with the command:

```
$ chmod 700 collect.py
```

Once this is done, you are ready to run the program as a script:

```
$ ./collect.py 50 1000
```

Check the output and make sure the program completes successfully. Then you can try other values for **`n`** and **`t`**.

Writing the top-level conditional is useful when you have a Python module that implements some functionality for a larger program (for example, implements an abstract data type). The conditional can then be used to test the module prior to putting it to use in the larger program. This is very similar to the **`main()`** function in Java programs.

Complete the exercise above. Once completed, send an e-mail to your instructor notifying them that you have completed the Lab.