CMPSCI 119
LAB #1
Professor William T. Verts

The goal of this first Python programming assignment is to successfully create, enter, debug, and run a simple program in the JES environment.

If you have not done so already, you must first install the JES environment on your own personal computer from the download site (`https://github.com/gatech-csl/jes/releases`). Please set up a special folder on your desktop or in your documents folder to hold all your Python projects. I also recommend that you buy a flash drive (SanDisk recommended) and put a folder for your Python projects there as well – the flash drive can then be used for backup and for bringing code in for us to help you debug.

Math Background

The *Hailstone Numbers* are a sequence of integers, where the next value in the sequence is generated from the current value by a simple set of rules: if the current integer is even then divide it by 2, but if it is odd then multiply it by 3 and add 1. The sequence continues until the current value becomes exactly 1, at which point we stop. For example, if we start at 7, the Hailstone Numbers are generated as follows:

7 (odd)	becomes 22
22 (even)	becomes 11
11 (odd)	becomes 34
34 (even)	becomes 17
17 (odd)	becomes 52
52 (even)	becomes 26
26 (even)	becomes 13
13 (odd)	becomes 40
40 (even)	becomes 20
20 (even)	becomes 10
10 (even)	becomes 5
5 (odd)	becomes 16
16 (even)	becomes 8
8 (even)	becomes 4
4 (even)	becomes 2
2 (even)	becomes 1
1 (stop)	

A Python function called `Hailstones`, given one integer parameter `N`, would then return a list of the integers in the Hailstone sequence starting at `N`. Using our example above, calling `Hailstones(7)` would return `[7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1]` as its result.

The Assignment

The purpose of this assignment is to write a program that computes and saves to file the Hailstone Numbers for all integers starting from 1 up through and including 25.

In the JES environment, type in the following program code, except for the comments in the bodies of the functions, and save it in your Python folder with `Lab1.py` as the filename. Where you see my name, replace it with *your own name*.

```
# William T. Verts - Lab 1

def Hailstones (N):
    # Your code for the Hailstones function
    # goes here.

def Main():
    # Your main program code
    # goes here.
    return
```

Task #1

Complete the `Hailstones` function so that for any input value in `N`, it computes and returns a list containing all Hailstone values starting at `N` and continuing until `1` is reached. That is:

`Hailstones(1)` should return `[1]`
`Hailstones(2)` should return `[2, 1]`
`Hailstones(3)` should return `[3, 10, 5, 16, 8, 4, 2, 1]`
`Hailstones(4)` should return `[4, 2, 1]`
`Hailstones(5)` should return `[5, 16, 8, 4, 2, 1]`
etc.

When you finish writing the Hailstones function, save your file and then click the Load Program button. Fix any syntax errors. Every time you change the program you must re-save the source code (File-Save, or Ctrl`S` on Windows, or ⌘`S` on Mac), then click the Load Program button. Be very careful about indentation (please use exactly four spaces per level throughout the program) and capitalization (for example, variables `frog`, `FROG`, and `Frog` are all different). Places where errors can creep in are through missing or extra quotes, parentheses, colons, commas, etc., along with incorrect indentation levels.

Test your function in the interactive area of Python, and make certain it works before proceeding.

Task #2

Now we need to complete the main program, which is to call the `Hailstones` function for all values from 1 through 25 (inclusive) and write the results to a text file called `Hailstones.txt`. The sequence of steps for you to complete in `Main` is as follows:

Step 1: Using `pickAFolder`, ask the user for the name of the folder to store the results and save that string in a variable.

Step 2: Create a file name variable by combining the result of step 1 with the string `"Hailstones.txt"`.

Step 3: Open the file for writing using the file name generated in step 2.

Step 4: For each integer from 1 through 25, get the appropriate Hailstone sequence for that integer, convert the resulting list to a string, and write the string to the file.

Step 5: Close the file.

Step 6: Return

If everything works correctly when you type `Main()` followed by Enter↵, you will get asked for the name of the destination folder, and then the program will run to completion. In the specified folder there will now be a `Hailstones.txt` file, where the first few lines of that 25-line file will be:

```
[1]
[2, 1]
[3, 10, 5, 16, 8, 4, 2, 1]
[4, 2, 1]
[5, 16, 8, 4, 2, 1]
…
```

If the file does not contain the correct information, debug your program and run it again. Repeat the process until the resulting file does contain the correct information.

Finishing Up

When you are finished and everything runs correctly, go to the class site and click on the link for submitting lab assignments. In JES select all the text, copy it to the clipboard, in the Web page paste the text into the program area of the submission form, fill in your name and ID number and the lab number in the appropriate slots, and then submit the assignment for grading.

The graders will score your program by running it to see if it correctly creates the `Hailstones.txt` file. You will also be graded on the completeness of each sequence (does each list start with the value of `N` and end in `1`?), on the set of starting values (does your main program call `Hailstones` for all starting values between 1 and 25?), program completeness (do you accomplish all needed steps?), and efficiency (do you use loops correctly?).