

Repetition Statement (**while** loop) in Python and Java

Problem statement: Write a program that computes the hailstone sequence (a.k.a. Collatz sequence) starting with a positive number n . The sequence is computed as follows: if n is even, divide it by two; if n is odd, multiply it by three and add one.

Python solution: hailstone.py

```
1 maximum_number_of_terms = 150
2 num_of_terms = 1
3
4 n = int(input("Enter a positive number: "))
5
6 print("Your hail stone sequence is: ")
7
8 while n > 0 and num_of_terms < maximum_number_of_terms:
9
10     print( int(n), end=", ")
11
12     #compute the next number
13     if n % 2 == 0: #even number
14         n = n / 2
15     else :        #odd number
16         n = 3 * n + 1
17
18     num_of_terms = num_of_terms + 1
19
20 #print the last number
21 print( int(n) )
22
```

Java solution: HailStone.java

```
1 import java.util.Scanner;
2
3 public class HailStone{
4     public static void main ( String [] args ) {
5
6         final int MAXIMUM_TERMS = 150;
7         int numOfTerms = 1;
8
9         Scanner in = new Scanner (System.in) ;
10
11         System.out.print("Enter a positive number: ");
12
13         int n = in.nextInt();
14         System.out.println("Your hail stone sequence is: ");
15
16         while ( n > 0 && numOfTerms < MAXIMUM_TERMS ) {
17             System.out.print( n + ", ");
18             //compute the next number
19             if (n % 2 == 0 ) //even number
20                 n = n / 2;
21             else //odd number
22                 n = 3 * n + 1;
23             numOfTerms++;
24         }
25         //print the last number
26         System.out.print( n + "\n ");
27     }
28 }
29
```

Questions:

1. What is the format/syntax of the **while** repetition statement in each language? Include all the details. How are the two similar and how do they differ?
2. What determines when the repetitions terminate? How the condition is formed?
3. How would the output change if the print statement inside the loop was moved to the last line of the loop body?
4. How would the output change if the second part of the condition was removed?
5. How is the input read from the user?

Write your own code:

Part A: Write your own program that computes and prints all the numbers from 0 to 100 that are not divisible by 3 and are not divisible by 5.

Part B: Compute the sum of all the numbers generated in Part A and print that sum, instead of all the numbers.