Stopwatch Review Lab

If you do not make it to lab today email me as soon as you see this to get instructions.

The first step of this assignment do on your own.

Step 1:

Write a function definition

Str stopwatch(int [])

Robin just received a stopwatch from her grandfather. Robin's stopwatch has a single button. Pressing the button alternates between stopping and starting the stopwatch's timer. When the timer is on, the displayed time increases by 1 every second.

Initially the stopwatch is stopped and the timer reads 0 seconds. Given a sequence of times that the stopwatch button is pressed, determine what the stopwatch's timer displays.

Input

The function takes in a list of integers, the first integer, N, is between 1 and 1000 which is the number of times the stopwatch was pressed.
The next N integers describe the times the stopwatch's button was pressed in increasing order. Each integer after the first is between 0 and 10^6, which is the time the button was pressed. No two button presses happen on the same second.

Output

Display `still running` if the stopwatch's timer is still running after all button presses were made. Otherwise display the number of seconds displayed on the stopwatch's timer.

input [2,7,11]	returns 4
input [5,2,5,9,10,17]	returns still running
input [4,0,2,104,117]	returns 15

Once written, remove all extra code leaving just your comments, import statements, and function definition. Save it as stopwatch1.py and submit it to moodle. Email your function to your partner and cc me(bdonoven1@mtech.edu) on it.

Stopwatch Review Lab

Step 2:

Once you receive your partners program, add a new “header” below their program and you will write unit tests for the function. Program a unit test for the 3 test cases above, and you create 2 more. Each test case should print Passed or Failed if the function returns the correct value. Your output format should match the example below.

Test 1: Passed

Test 2: Failed

Test 3: Passed

Test 4: Passed

Test 5: Failed

Add comments about what tests passed and failed and suggestions on formatting, variable names, comments, readability.

Save the program as stopwatch2.py, submit it to moodle, and email it to your partner with me cc’d again.

Step 3:

Make corrections on your program to pass all test cases, if any are needed. Clean up any formatting/readability issues, if any.

Save the program as stopwatch3.py and submit it to moodle.

Grade Item	Points Possible	Points Earned
Step one Compiles and Runs	3	
Step one Comments	2	
Step two Compiles and Runs	3	
Step two Comments	2	
Step two Good Cases	2	
Step two Tests Work	3	
Step three test cases pass	3	
Step three style comments	2	
Total	20	

Stopwatch Review Lab

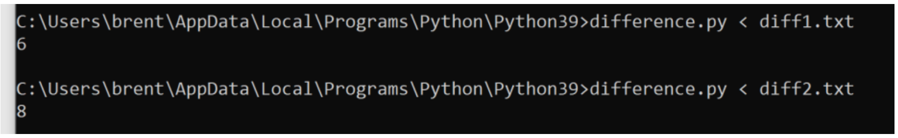
Extra Credit up to 4 points

Create an account on Kattis https://open.kattis.com/

Follow their instructions to read from standard input (stdin) and output to standard output (stdout) https://open.kattis.com/help/python3

Rewrite your program to read from stdin and write to stdout and submit your program https://open.kattis.com/problems/stopwatch. If your program gets accepted take a screen shot and submit it moodle. Save your program as stopwatchEC.py and submit it to moodle.

To test your Extra Credit program you will need to send standard input into it. To do that, run your program from the command line and use the "<" character to pipe in a text file as shown below. This example program and text files are provided on katie.



```
C:\Users\brent\AppData\Local\Programs\Python\Python39>difference.py < diff1.txt
6

C:\Users\brent\AppData\Local\Programs\Python\Python39>difference.py < diff2.txt
8
```