PROGRAMMING ASSIGNMENT 1

DUE: Tuesday November 2, 11:59 PM. No late submissions allowed.
DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read http://www.student.cs.uwaterloo.ca/~cs341 for general instructions and policies.

Problem Description

Problem Statement

For this assignment, you are asked to implement an algorithm to solve Problem 2 of Assignment 4. Let us recall the problem:

Suppose you have $n$ people lined up waiting for consultations. Person $i$ needs $q_i$ minutes for their consultation, where $q_i$ is a positive integer. There are $t$ rooms in which the consultations can be held and each room can be used for a maximum of $L$ minutes. The goal is to assign the people to the rooms so that the first $n_1$ go to room 1, the next $n_2$ go to room 2, $\ldots$, the last $n_t$ go to room $t$. Because you want to minimize contact during the pandemic, you want to leave gaps between successive people using the same room. For example, if $L = 10$ and room 1 will be used for $i = 1, 2, 3$ with $q_1 = 4, q_2 = 1, q_3 = 2$ then since $q_1 + q_2 + q_3 = 7$ there are 3 spare minutes, and you would have a gap of 1.5 minutes between person 1 and person 2, and a gap of 1.5 minutes between person 2 and person 3.

The objective function is to maximize the minimum gap between any two people. In other words, let $G$ be the smallest gap between any pair of people that are scheduled consecutively in the same room. The goal is to choose $n_1, n_2, \ldots, n_t$ so that $G$ is maximized.

Hints: You might pre-compute $G(i, j) =$ the gap value obtained by placing people $i, i+1, \ldots, j$ into one room. After that, use dynamic programming and aim for $O(nt)$ subproblems.
Input Format
The first line of input consists of integers \( n \), \( t \), and \( L \), each separated by one space.
The second line of input consists of \( n \) integers \( q_1, q_2, \ldots, q_n \), each separated by one space.

Output Format
If there does not exist any valid room assignment with \( G \geq 0 \), output IMPOSSIBLE.
Otherwise, output two lines. On the first line, output the optimal value of \( G \). On the second line, output \( t \) non-negative integers \( n_1, n_2, \ldots, n_t \), separated by one space, so that assigning the first \( n_1 \) people to room 1, the next \( n_2 \) people to room 2, and so on, is a valid room assignment that attains the optimal minimum gap.

The checker is case-sensitive, so please output IMPOSSIBLE in all caps. End each line with no trailing spaces and one newline character.

IMPORTANT: On a given test, if there exists a valid room assignment with \( G \geq 0 \), your output will be considered correct if and only if:

- The smallest gap you find has an absolute or relative error at most \( 10^{-5} \). Specifically, let \( G_{\text{out}} \) be the optimal value of \( G \) you output and \( G_{\text{ans}} \) be the correct optimal value of \( G \), then
  \[
  \frac{|G_{\text{out}} - G_{\text{ans}}|}{\max(1, G_{\text{ans}})} \leq 10^{-5};
  \]
  and

- The output room assignment has smallest gap \( G_{\text{ans}} \). Again, we accept absolute or relative error up to \( 10^{-5} \), so if your room assignment has smallest gap \( G \), it should hold that
  \[
  \frac{|G - G_{\text{ans}}|}{\max(1, G_{\text{ans}})} \leq 10^{-5}.
  \]

In particular, if there is more than one room assignment that attains the optimal \( G \), you can output any of them.

Constraints
For all test cases, \( 1 \leq n \leq 1000, 1 \leq t \leq 50, 1 \leq L \leq 10^6, 1 \leq q_i \leq 1000. \)
In addition, it holds that \( t < n \) (so that the optimal \( G \) will not be infinity).

Sample Input 1
6 2 120
10 30 40 20 80 10
Sample Output 1
6.6666666667
4 2

Sample Input 2
5 3 50
30 30 30 30 30

Sample Output 2
IMPOSSIBLE

Submission Instructions

- Submit your solution on Marmoset.
- You can choose to code in either C++ or Python.
- Name your program `prog1.cpp/prog1.py`.
- **Time limit:** 2 seconds (C++) / 6 15 seconds (Python) for each test case.
- Compilation command for C++: `g++ -std=c++14 prog1.cpp -o prog1`.
- Execution command for Python: `python3 prog1.py`.
- Read from standard input and write to standard output.
- There will be several test cases, worth a total of 30 points. The public tests are worth 8 points and the secret tests are worth 22 points. The public tests (input only, not the answer) will be made available under a separate file.
- We will take the submission with the highest score. Please, however, refrain from excessive submissions.
- General collaboration policy applies. Please acknowledge your collaborator(s) by adding a comment in the beginning of your code.
- FAQ and updates will be posted on Piazza when necessary.

(More) Hints

- The last public test is a large test. Use it to help gauge your solution’s efficiency and identify errors like memory error, stack overflow, etc.
- The format string for outputting a floating point number corrected to [p] decimal places is `%.[p]f`.