Consider the following picture of the pillars and cable that provide the structure for a cable car. Successive pillars are equidistant. The height of each pillar is indicated above it. Continuous pillars make up for a good ride if the slope does not change over the corresponding section. Here the longest good ride is between the pillar of height 25 and the pillar of height 28; it is depicted in blue. It has a length of 3 (units, whatever it is).

true

If one gets rid of the black pillars and keeps only the remaining (red) ones, then one can use use the latter and create a new structure (with successive pillars being equidistant) which makes up for a perfect ride, that is, good from first to last pillar:

true

It is the longest perfect ride one can obtain this way; so 5, the number of black pillars, is the minimal number of pillars to remove to build a perfect ride from the rest.

Write a program, stored in a file named cable\_car.py, that performs the following task.

•

The program prompts the user to input a file name. If there is no file with that name in the working directory, then the program outputs an error message and exits.

•

The contents of the file should consist of a strictly increasing sequence of at least two strictly positive integers. Consecutive numbers are separated by at least one space, and there can be extra spaces, including empty lines, anywhere. If that is not the case then the program outputs another error message and exits. These numbers are meant to represent the heights of pillars to build a cable car structure. The program then outputs:

1.

whether or not this structure makes up for a perfect ride;

2.

the length of the longest good ride;

3.

how many pillars to remove to build a perfect ride from the remaining ones.

代码部分：

import os.path

import sys

from collections import defaultdict