Cross Mountain Climb

Somewhere in the neighborhood we have a very nice mountain that gives a splendid view over the surrounding area. There is one problem though: climbing this mountain is very difficult, because of rather large height differences. To make more people able to climb the mountain and enjoy the view, we would like to make the climb easier. To do so, we will model the mountain as follows: the mountain consists of n adjacent stacks of stones, and each of the stacks is hi high. The successive height differences are therefore $h_{i+1} – h_i$ (for 1 <= i <= n – 1). We would like all absolute values of these height differences to be smaller than or equal to some number d.

We can do this by increasing or decreasing the height of some of the stacks. The first stack (the starting point) and the last stack (the ending point) should remain at the same height as they are initially. Since adding and removing stones requires a lot of effort, we would like to minimize the total number of added stones plus the total number of removed stones. What is this minimum number?

Input

On the first line one positive number: the number of testcases, at most 100. After that per testcase:

- One line with two integers n (2 <= n <= 100) and d (0 <= d <= 10^9): the number of stacks of stones and the maximum allowed height difference.
- One line with n integers hi (0 <= h_i <= 10^9): the heights of the stacks.

Output

Per testcase:

One line with the minimum number of stones that have to be added or removed or “impossible” if it is impossible to achieve the goal.

Example

Input:

```
3
10 2
4 5 10 6 6 9 4 7 9 8
3 1
6 4 0
4 2
3 0 6 3
```

Output:

```
6
impossible
4
```