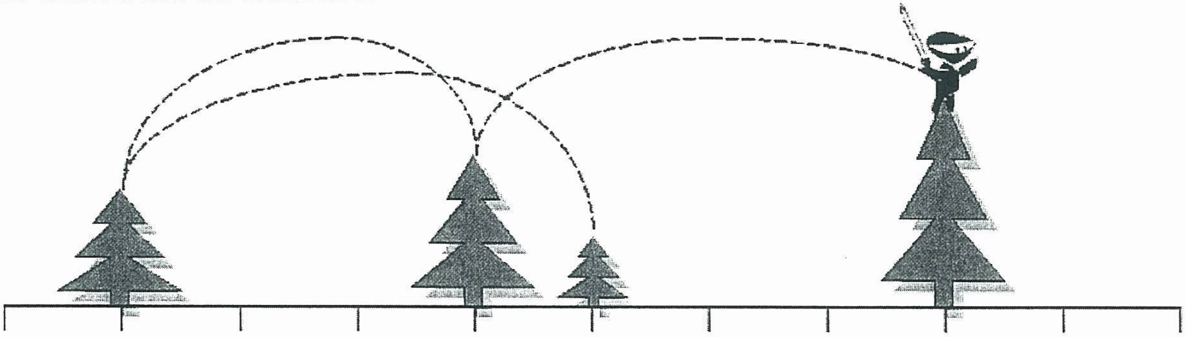
Problem D: The Ninja Way

As we all know, ninjas travel by jumping from treetop to treetop. A clan of ninjas plan to use N trees to hone their tree hopping skills. They will start at the shortest tree and make N-1 jumps, with each jump taking them to a taller tree than the one they're jumping from. When done, they will have been on every tree exactly once, and they will end up on the tallest tree.

The ninjas can travel for at most a certain horizontal distance D in a single jump. To make this as much fun as possible, the Ninjas want to maximize the distance between the planting positions of the shortest and the tallest tree.



The ninjas are going to plant the trees subject to the following constraints:

- All trees are to be planted along a one-dimensional path, which we can regard as the number line.
- Trees must be planted at integer locations along the path, with no two trees at the same location.
- Trees must be arranged so their planted ordering from left to right is the same as their ordering in the input: from $1, 2, \ldots, N$. They must **not** be sorted by height, or reordered in any way. They must be kept in their stated order.
- The Ninjas can only jump so far, so every tree must be planted close to the next tallest tree. In fact, they must be no further than D apart on the ground (the difference in their heights doesn't matter).

Given N trees, numbered $1, 2, \ldots, N$, each with a distinct integer height, help the ninjas figure out the maximum possible distance between the shortest tree and the tallest tree.

Input

Input will consist of multiple datasets. Each dataset begins with a line containing two integers N $(1 \leq N \leq 1000)$ and D $(1 \leq D \leq 10^6)$.

The next N lines each contains a single integer, giving the heights of the N trees.

The last test case is followed by a line with two zeros.

Output

For each test case, output a line with a single integer representing the maximum possible distance between the planted positions of the shortest and tallest tree, subject to the constraints above, or -1 if it is impossible to lay out the trees. Do not print any blank lines between answers.

Example

Given the input

```
4 4
20
30
10
40
5 6
20
34
54
10
15
4 2
10
20
16
13
0 0
```

the output would be

```
3
3
-1
```