# 4178   Dezider's Fence

Our robotic friend Dezider is given a very important job — fixing fences. A typical fence has n fence posts and, ideally, they should be distributed along the fence line in regular intervals. Unfortunately, Dezider's fences are very messed up and his job is to fix them as much as possible — he wants to minimize the maximum distance between any two neighboring fence posts.

Dezider knows how to do things properly — he first marks the positions of all posts. He is happy to report that all posts are perpendicular to the ground and their positions (counted from the left end of the fence line) are integers. Dezider also noticed that it is possible, starting at the left and proceeding right from one post to the next, to move each post by at most k units to the left or right, as long as there is no other post in the way and as long as each post still stays on the fence line. The posts always have to be at integral positions, even after they are moved. Finally, proceeding strictly from left to right, every post — except for the first and last one — can be moved at most once.
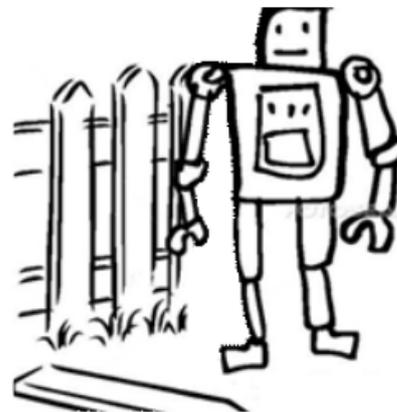
Input

Input consists of positive integers separated by white space. The first line contains n and k, where n is the number of fence posts, $1 < n < 1000$, and k is the amount by which each post can be moved. The second input line contains n numbers — the positions of the posts, from left to right; the fence line itself starts at position 0 and is never more than 10000 units long. The positions form an increasing sequence and cannot leave the fence line.

Output

The output contains a single line with a single integer, which is the maximum distance between any two neighboring fence posts after Dezider is done moving posts.

Note for the sample: As one possible solution is to leave the second post at 3, to move the third post right from 4 to 9 and the fourth post left from 19 to 14; the first and last post must not be moved. From left to right the distances between the posts are now 3, 6, 5, and 6; the maximum distance is 6.

Sample Input

```
5 5
0 3 4 19 20
```

Sample Output

```
6
```