Predicting a Position

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

An ascending sorted array of unique integer keys x_i and some integer non-negative constant ε is given.

It is necessary to split the original key array into a minimum number of sub-arrays, so that on each sub-array $[l \dots r]$ there is some linear function $f(x) = k \cdot x + b$, predicting for the key x_i its position on this sub-array — it is equal to $i - l$ — with an error not exceeding ε.

Formally, for the j-th of the sub-array, the coefficients k_j and b_j must exist (not necessarily integers) so that for all the keys x_i on that sub-array $i \in [l_j \dots r_j]$ is true:

$$|f_j(x_i) - (i - l_j)| \le \varepsilon$$
$$f_j(x) = k_j \cdot x + b_j$$

Input

In the first line of the input data, two integers are separated by a space: n, ε — the number of keys and the maximum allowable error, respectively ($2 \le n \le 10^6$, $0 \le \varepsilon \le n$).

The second line of input data contains the unique keys x_i separated by a space in ascending order ($-2 \cdot 10^9 \le x_1 < x_2 < \ldots < x_n \le 2 \cdot 10^9$).

Output

In a single line of the output data, print a single integer m — the minimum number of sub-arrays into which you were able to split the original array of keys so as to meet the requirements.

Example

standard input	standard output
8 0 1 2 3 4 7 10 13 16	2

Note

In the example, you can split the specified array of keys into two: $[1, 2, 3, 4]$ and $[7, 10, 13, 16]$. In the first case, the key position is predicted exactly by the function $f(x) = x - 1$. In the second case, the key position is accurately predicted by the function $f(x) = (x - 7)/3 = 1/3 \cdot x - 7/3$.