

C. DZY Loves Sequences

time limit per test 1 second
memory limit per test 256 megabytes
input standard input
output standard output

DZY has a sequence a , consisting of n integers.

We'll call a sequence a_i, a_{i+1}, \dots, a_j ($1 \leq i \leq j \leq n$) a subsegment of the sequence a . The value $(j - i + 1)$ denotes the length of the subsegment.

Your task is to find the longest subsegment of a , such that it is possible to change at most one number (change one number to any integer you want) from the subsegment to make the subsegment strictly increasing.

You only need to output the length of the subsegment you find.

Input

The first line contains integer n ($1 \leq n \leq 10^5$). The next line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Output

In a single line print the answer to the problem — the maximum length of the required subsegment.

Examples

input

[Copy](#)

```
6
7 2 3 1 5 6
```

output

[Copy](#)

```
5
```

Note

You can choose subsegment a_2, a_3, a_4, a_5, a_6 and change its 3rd element (that is a_4) to 4.

```
1 //> http://codeforces.com/contest/447/problem/C
2 #include<bits/stdc++.h>
3 using namespace std;
4 #define MAXN 100005
5 int n,a[MAXN],lft[MAXN],rgt[MAXN];
6 void preprocess()
7 {
8     lft[1]=0;
9     for(int i=2; i<=n; i++){
10         if(a[i]>a[i-1])lft[i]=lft[i-1]+1;
11         else lft[i] = 0;
12     }
13
14     rgt[n]=0;
15     for(int i=n-1; i>=0; i--){
16         if(a[i]<a[i+1])rgt[i]=rgt[i+1]+1;
17         else rgt[i] = 0;
18     }
19 }
20 int solve()
21 {
22     preprocess();
23
24     int ans = 1;
25     for(int i=1; i<=n; i++){
26         ans = max(ans,lft[i]+rgt[i]+1);
27         if(a[i]<=a[i-1]&&i>1){
28             int v = a[i-1]+1;
29             int cnt = lft[i-1]+1;
30             cnt++;
31             if(i<n && v<a[i+1]){
32                 cnt += rgt[i+1]+1;
33             }
34             ans = max(ans,cnt);
35         }
36         if(a[i]>=a[i+1]&&i<n){
37             int v = a[i+1]-1;
38             int cnt = rgt[i+1]+1;
39             cnt++;
40             if(i>1 && v>a[i-1]){
41                 cnt += lft[i-1]+1;
42             }
43             ans = max(ans,cnt);
44         }
45     }
46     return ans;
47 }
48 int main()
49 {
50     ios::sync_with_stdio(false); cin.tie(0);
51     cin>>n;
52     for(int i=1; i<=n; i++)cin>>a[i];
53
54     cout << solve() << endl;
55
56     return 0;
57 }
```