**Tablets**

Attempted by: **5094**

/

Accuracy: **90%**

/

Maximum Points: **20**

/

34 Votes

/

**Share**

Dynamic Programming

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

[**DISCUSSIONS**](https://www.hackerearth.com/practice/algorithms/dynamic-programming/introduction-to-dynamic-programming-1/practice-problems/algorithm/tablets/discussion/)NEW

Therasa is a Nurse. She wants to give some tablets to the patients in her practice. All the patients sit in a line and each of them has a rating score according to his or her health score. Therasa wants to give at least 1 tablet for each patient. Patients get jealous of their immediate neighbors, so if two patients sit next to each other then the one with the higher rating must get more tablets. Therasa wants to save money, so she wants to minimize the total number of tablets.

**Input**  
The first line of the input is an integer N, the number of patients in Therasa’s practice. Each of the following N lines contains an integer indicates the health score of each patient.

**Output**  
Output a single line containing the minimum number of tablets Therasa must give.

**Constraints**  
1 <= N <= 100000  
1 <= health score <= 100000

**SAMPLE INPUT**

3

1

2

2

**SAMPLE OUTPUT**

4

**Explanation**

Here 1, 2, 2 is the health score. Note that when two patients have equal health score they are allowed to have different number of tablets. Hence optimal distribution will be 1, 2, 1.

#include<bits/stdc++.h>

#define int long long int

#define pb push\_back

#define pii pair<int,int>

#define vec vector<int>

#define rep(i,a,b) for(i=a;i<b;i++)

using namespace std;

main()

{

int n;

cin>>n;

int a[n],i;

for(i=0;i<n;i++)

cin>>a[i];

int t[n];

t[0]=1;

for(i=1;i<n;i++)

{

if(a[i]>a[i-1])

t[i]=t[i-1]+1;

else

t[i]=1;

}

for(i=n-2;i>=0;i--)

{

if(a[i]>a[i+1] && t[i]<=t[i+1])

t[i]=t[i+1]+1;

}

int sum=0;

for(i=0;i<n;i++)

sum+=t[i];

cout<<sum;

}