Arrays Challenge-Longest Arithmetic Subarray (Google kickstart)

Problem

An arithmetic array is an array that contains at least two integers and the differences between consecutive integers are equal. For example, [9, 10], [3, 3, 3], and [9, 7, 5, 3] are arithmetic arrays, while [1, 3, 3, 7], [2, 1, 2], and [1, 2, 4] are not arithmetic arrays.

Sarasvati has an array of **N** non-negative integers. The i-th integer of the array is $\mathbf{A}_i$. She wants to choose a contiguous arithmetic subarray from her array that has the maximum length. Please help her to determine the length of the longest contiguous arithmetic subarray.

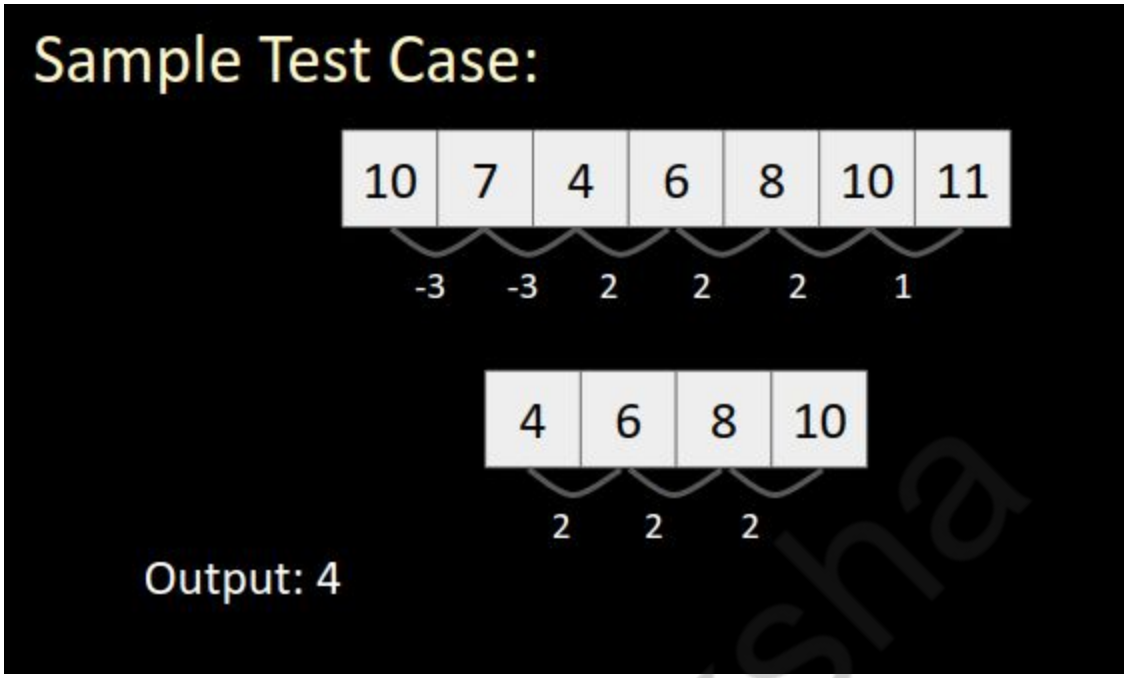
Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case begins with a line containing the integer **N**. The second line contains **N** integers. The i-th integer is $\mathbf{A}_i$.

Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is the length of the longest contiguous arithmetic subarray.

Constraints
Time limit: 20 seconds per test set.
Memory limit: 1GB.
$1 \leq \mathbf{T} \leq 100$.
$0 \leq \mathbf{A}_i \leq 10^9$.
Test Set 1
$2 \leq \mathbf{N} \leq 2000$.
Test Set 2
$2 \leq \mathbf{N} \leq 2 \times 10^5$ for at most 10 test cases.
For the remaining cases, $2 \leq \mathbf{N} \leq 2000$.



Solution

Constraints Analysis

1 sec = 10^8 operations

20 sec = $2x10^9$ operations

Intuition: We have to loop over the array and find the answer.

Steps

1. While iterating in the array we need to maintain the following variables,
    a. Previous common difference (pd) - To compare it with current common difference (a[i] - a[i-1]).
    b. Current arithmetic subarray length (curr) - It denotes the arithmetic subarray length including a[i].
    c. Maximum arithmetic subarray length (ans) - It denotes the max. Arithmetic subarray length till a[i].
2. While iterating, there will be two cases,
    a. pd = a[i] - a[i-1]
        i. Increase curr by 1
        ii. ans = max(ans, curr)
3. After loop ends, output the answer. (stored in ans variable).

Code

```cpp
int main()
{
    int n;
    cin >> n;

    int a[n];

    for(int i=0; i<n; i++)
        cin >> a[i];

    int ans = 2;
    int d = a[1]-a[0];
    int j=2;
    int curr=2;
    while(j<n)
    {
        if(a[j]-a[j-1] == d)
            curr++;
        else
        {
            d = a[j]-a[j-1];
            curr=2;
        }
        ans = max(ans,curr);
        j++;
    }
    cout << ans << endl;
    return 0;
}
```