Design and Analysis of Algorithms (CS206)

Practical Exam

U19CS012

1. Sapna is an instructor. She needs to give a few confections to her students. Every student sits in a line and every one of them has a rating score as per their performance.

CONTRAINTS:
Sapna needs to give **at least 1 candy** to every student.
Assuming two kids sit close to one another, the **one with the higher rating should get more confections**.

Sapna needs to limit the complete number of confections she should purchase.

Example:

She gives the students confections in the following minimal amounts: [1,2,1,2,3,1]. She must buy a minimum of 10 candies.

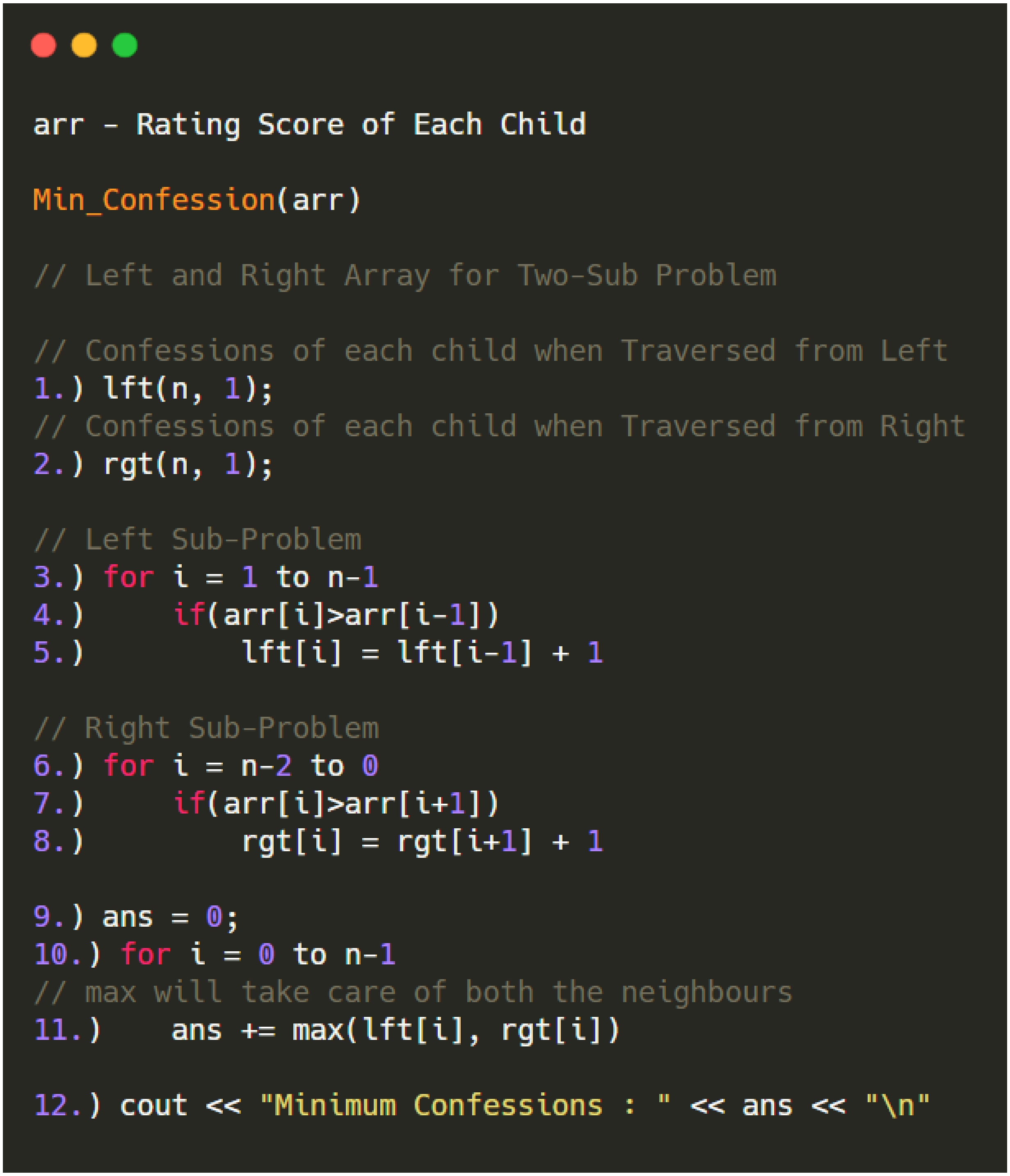
Returns
int: the **minimum** number of confections sapna must buy

Input:
The first line contains an integer, **n**, the size of arr.
Each of the next n lines contains an integer arr[i] indicating the rating of the student at position i.

Output:
Minimber number of confections n.

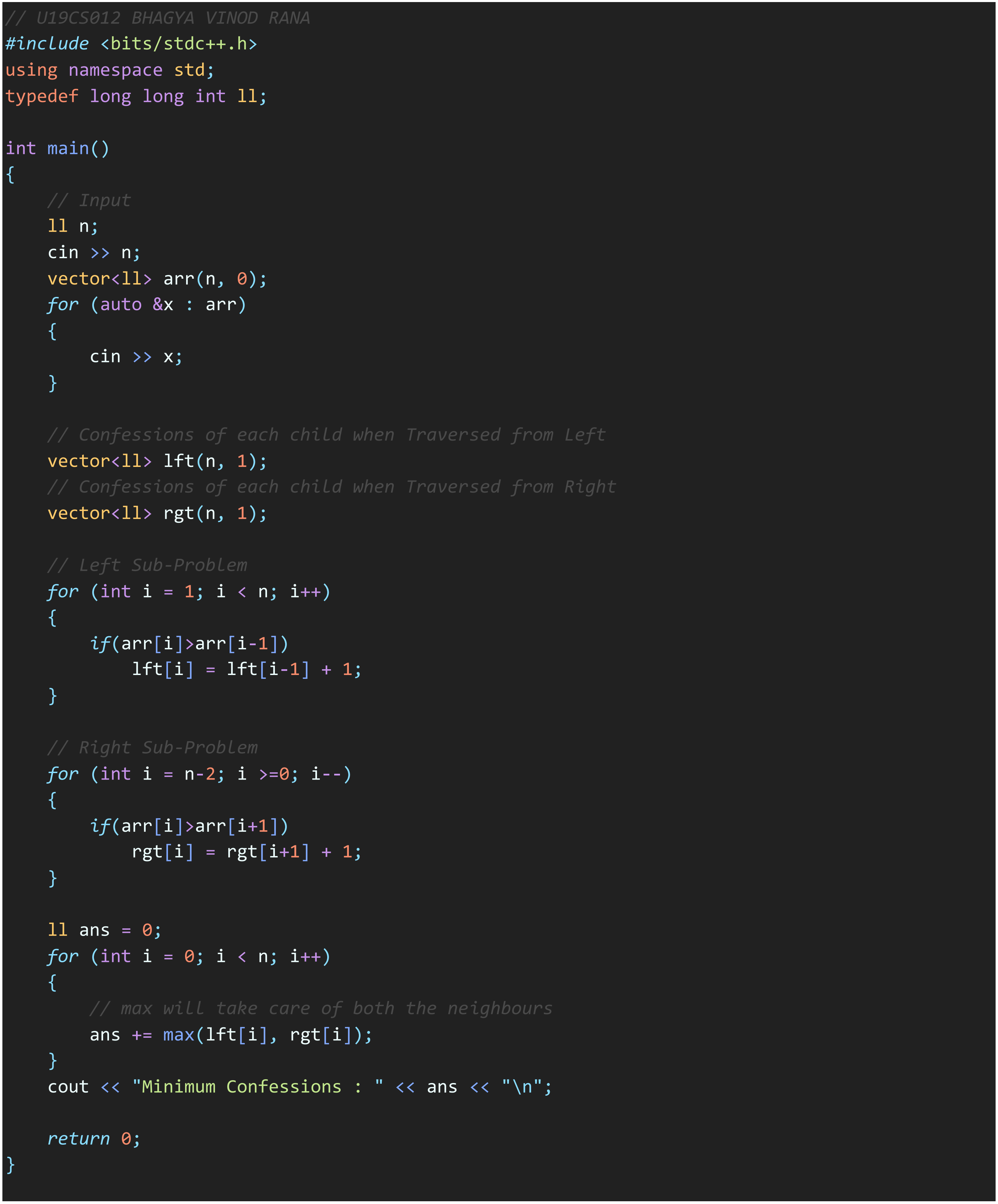
Key Observations:
1.) Sorting Approach Won't Work! [Change the Arrangement]
2.) Finding the Highest and Lowest Won't Work!

Pseudo Code:



```
arr - Rating Score of Each Child

Min_Confession(arr)

// Left and Right Array for Two-Sub Problem

// Confessions of each child when Traversed from Left
1.) lft(n, 1);
// Confessions of each child when Traversed from Right
2.) rgt(n, 1);

// Left Sub-Problem
3.) for i = 1 to n-1
4.)     if(arr[i]>arr[i-1])
5.)         lft[i] = lft[i-1] + 1

// Right Sub-Problem
6.) for i = n-2 to 0
7.)     if(arr[i]>arr[i+1])
8.)         rgt[i] = rgt[i+1] + 1

9.) ans = 0;
10.) for i = 0 to n-1
// max will take care of both the neighbours
11.)    ans += max(lft[i], rgt[i])

12.) cout << "Minimum Confessions : " << ans << "\n"
```

Code:



```cpp
// U19CS012 BHAGYA VINOD RANA
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

int main()
{
    // Input
    ll n;
    cin >> n;
    vector<ll> arr(n, 0);
    for (auto &x : arr)
    {
        cin >> x;
    }

    // Confessions of each child when Traversed from Left
    vector<ll> lft(n, 1);
    // Confessions of each child when Traversed from Right
    vector<ll> rgt(n, 1);

    // Left Sub-Problem
    for (int i = 1; i < n; i++)
    {
        if(arr[i]>arr[i-1])
            lft[i] = lft[i-1] + 1;
    }

    // Right Sub-Problem
    for (int i = n-2; i >=0; i--)
    {
        if(arr[i]>arr[i+1])
            rgt[i] = rgt[i+1] + 1;
    }

    ll ans = 0;
    for (int i = 0; i < n; i++)
    {
        // max will take care of both the neighbours
        ans += max(lft[i], rgt[i]);
    }
    cout << "Minimum Confessions : " << ans << "\n";

    return 0;
}
```

Time Complexity = **O(n)**

Test Cases:

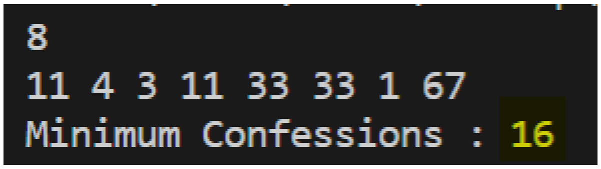
1) [11, 4, 3, 11, 33, 33, 1, 67]

Left: [1, 1, 1, 2, 3, 1, 1, 2]

Right: [3, 2, 1, 1, 1, 2, 1, 1]

Answer: [3, 2, 1, 2, 3, 2, 1, 2]

Minimum Confessions: 16



```
8
11 4 3 11 33 33 1 67
Minimum Confessions : 16
```

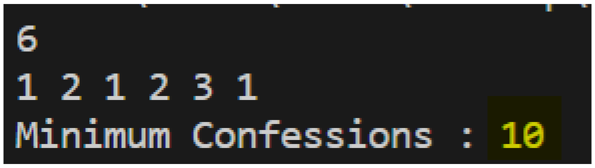
2) [1, 2, 1, 2, 3, 1] <Sample Test Case>

Left: [1, 2, 1, 2, 3, 1]

Right: [1, 2, 1, 1, 2, 1]

Answer: [1, 2, 1, 2, 3, 1]

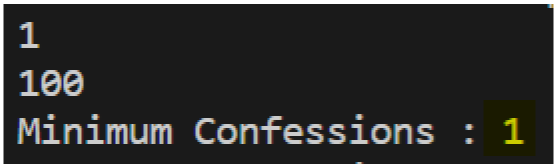
Minimum Confessions: 10



```
6
1 2 1 2 3 1
Minimum Confessions : 10
```

3) [100] <Trivial Case>

Minimum Confessions: 1



```
1
100
Minimum Confessions : 1
```

SUBMITTED BY:

U19CS012

BHAGYA VINOD RANA