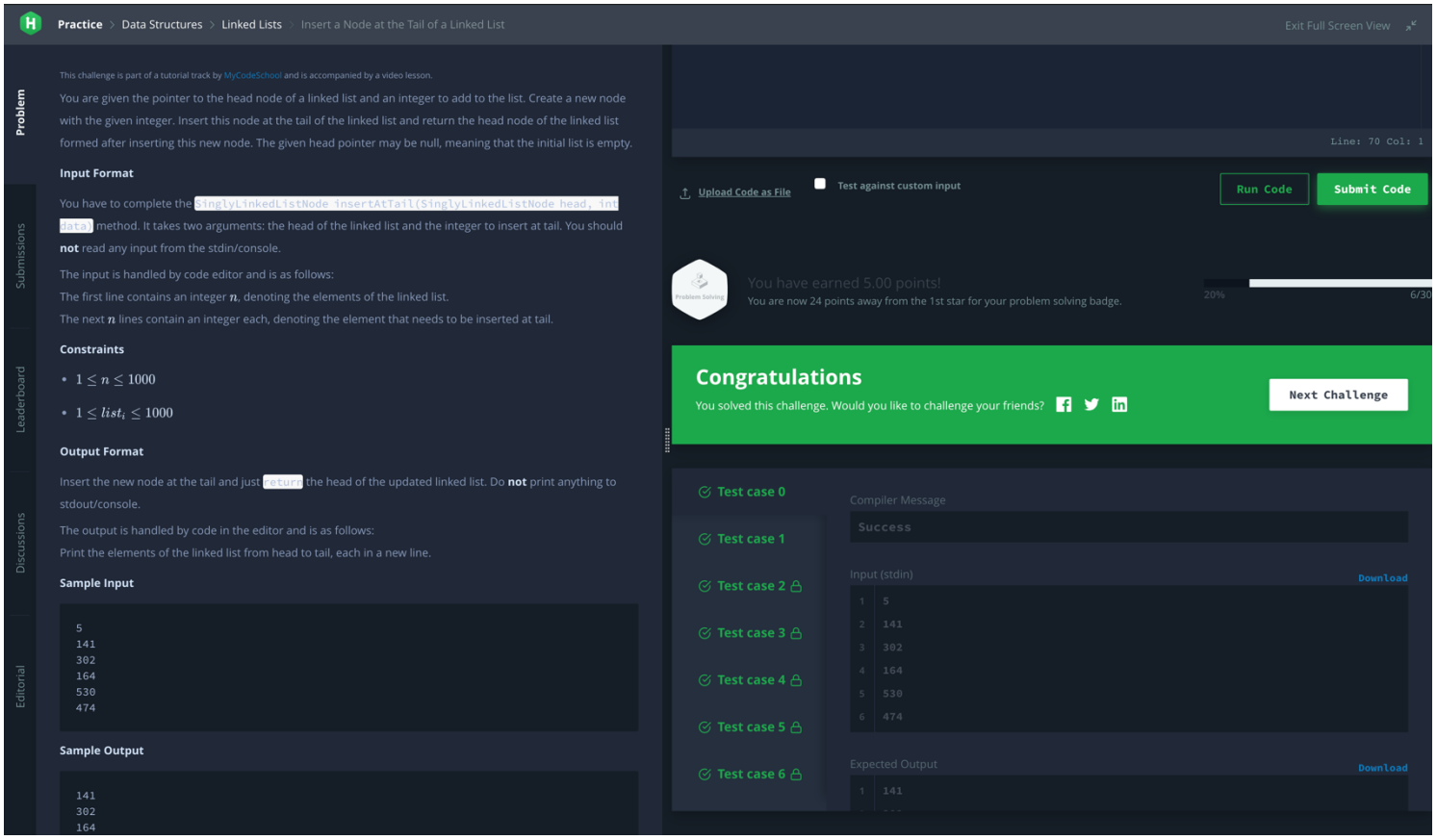
Alex Vargas A00824525

Reto Semestral – Ronda 1

Problema 1- fácil

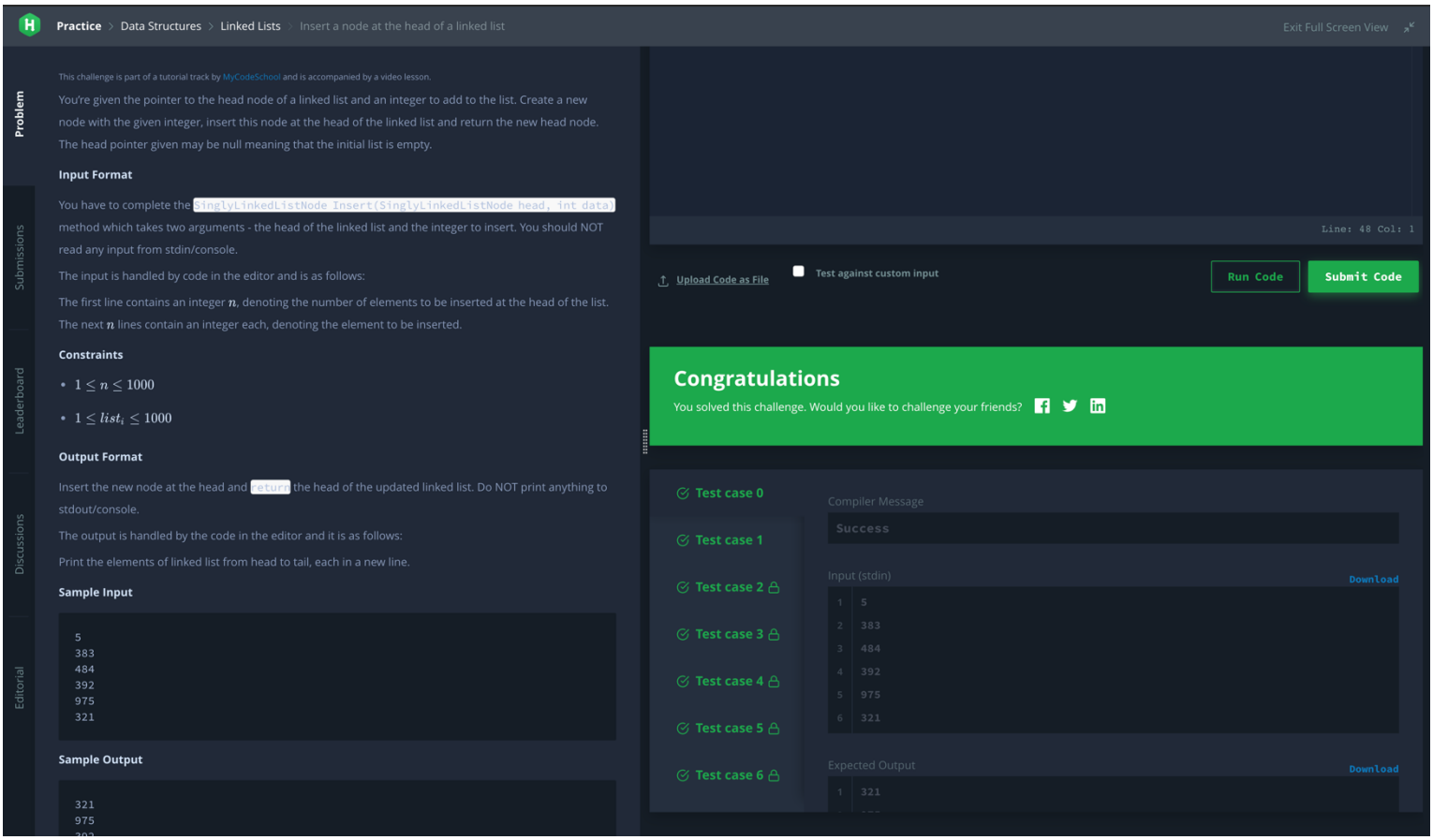
```cpp
#include <bits/stdc++.h> ⋯

// Complete the insertNodeAtTail function below.

/*
 * For your reference:
 *
 * SinglyLinkedListNode {
 *     int data;
 *     SinglyLinkedListNode* next;
 * };
 *
 */
SinglyLinkedListNode* insertNodeAtTail(SinglyLinkedListNode* head, int data) {
    if (head == NULL){
        head = new SinglyLinkedListNode(data);
    }else{
        SinglyLinkedListNode* curr = head;
        while(curr->next != NULL){
            curr = curr->next;
        }
        curr->next = new SinglyLinkedListNode(data);
    }
    return head;
}

int main() ⋯
```



Alex Vargas A00824525

Problema 2- fácil

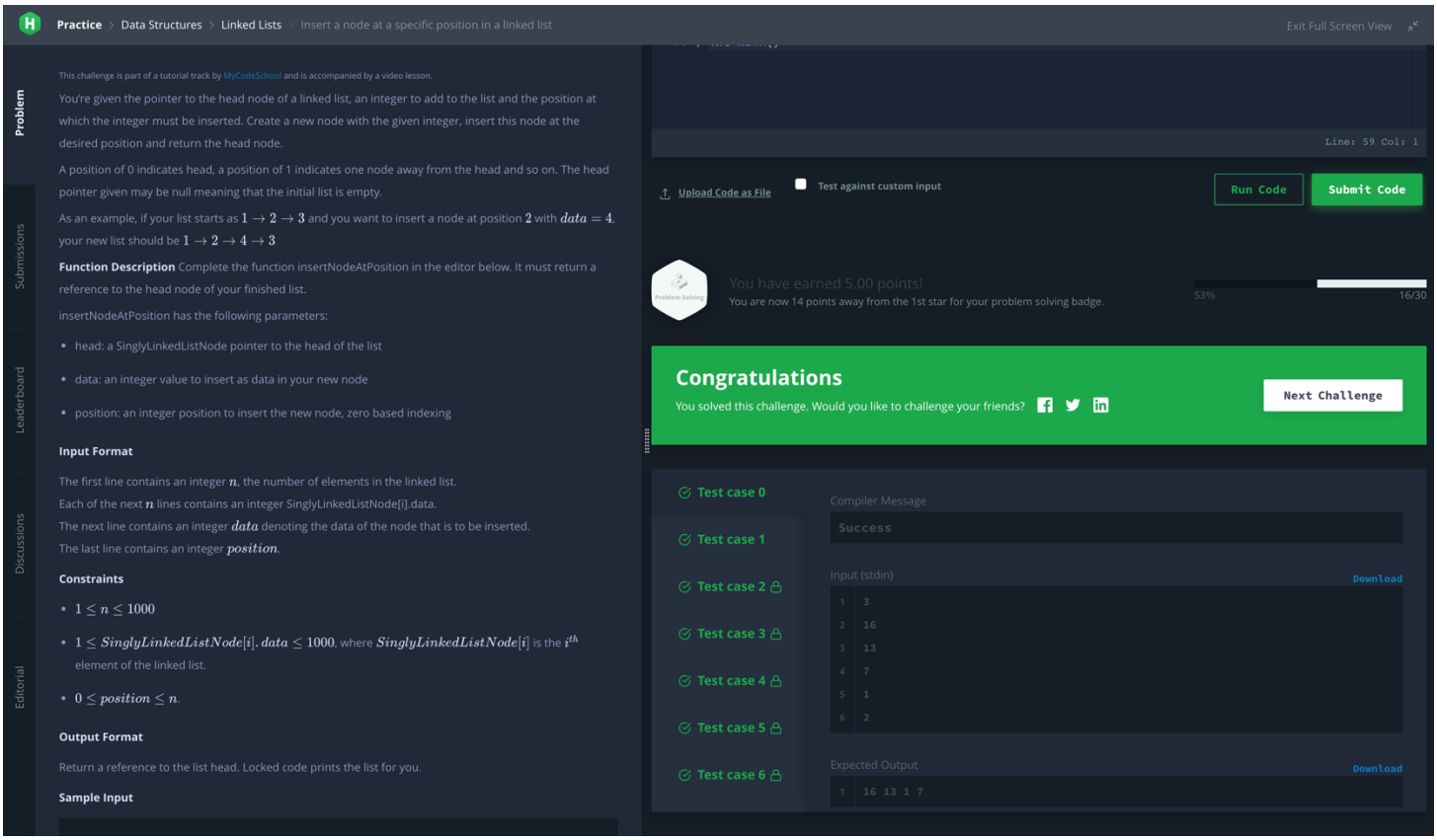
```
1 > #include <bits/stdc++.h> …
48
49  // Complete the insertNodeAtHead function below.
50
51  /*
52   * For your reference:
53   *
54   * SinglyLinkedListNode {
55   *     int data;
56   *     SinglyLinkedListNode* next;
57   * };
58   *
59   */
60  SinglyLinkedListNode* insertNodeAtHead(SinglyLinkedListNode* llist, int data) {
61      if (llist == NULL){
62          llist = new SinglyLinkedListNode(data);
63      }else{
64          SinglyLinkedListNode *new_node = new SinglyLinkedListNode(data);
65          new_node->next = llist;
66          llist=new_node;
67      }
68      return llist;
69  }
70
71 > int main() …
```



Alex Vargas A00824525

Problema 3- fácil

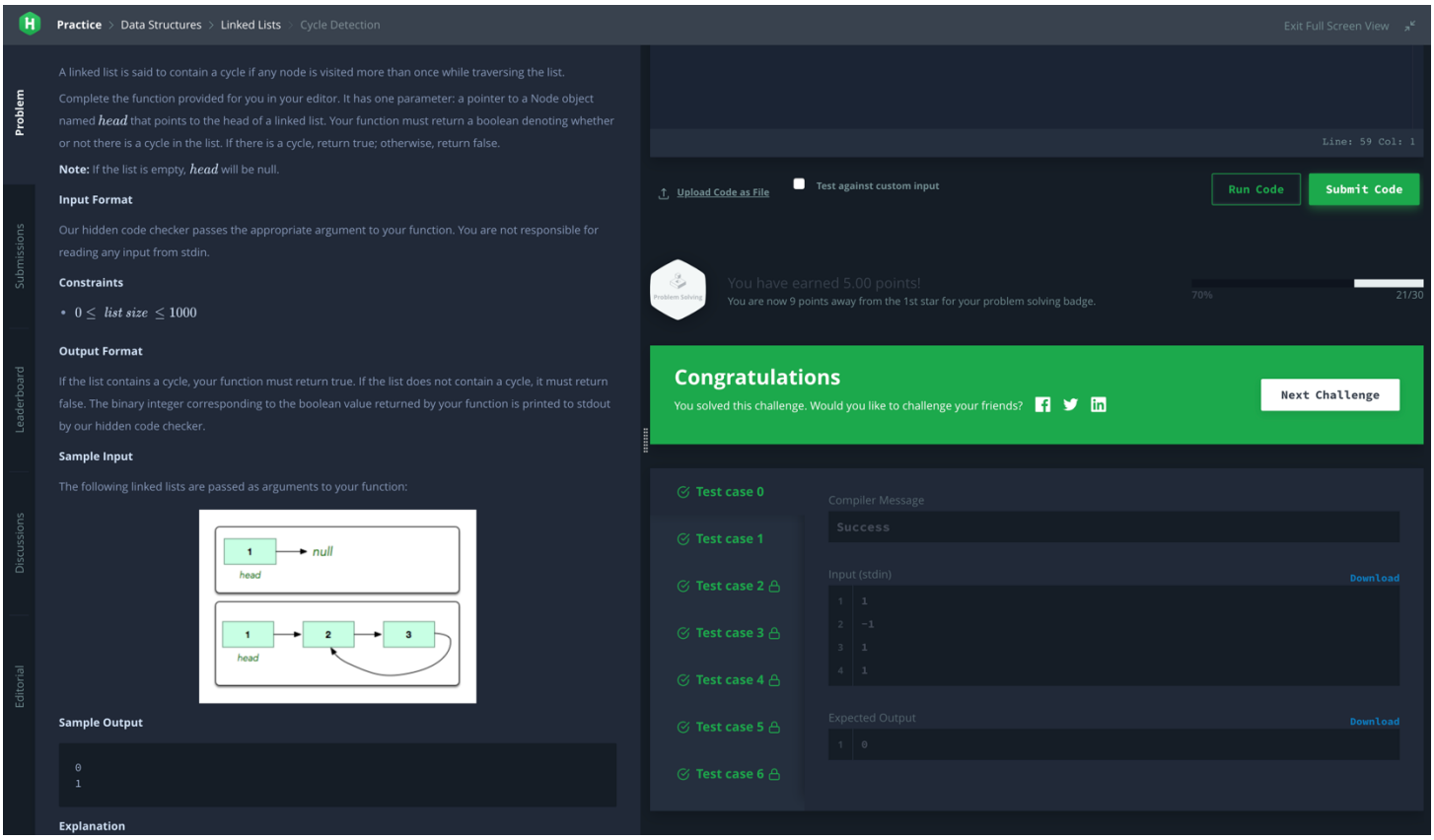
```
1 > #include <bits/stdc++.h> ⋯
59
60  // Complete the insertNodeAtPosition function below.
61
62  /*
63   * For your reference:
64   *
65   * SinglyLinkedListNode {
66   *     int data;
67   *     SinglyLinkedListNode* next;
68   * };
69   *
70   */
71  SinglyLinkedListNode* insertNodeAtPosition(SinglyLinkedListNode* head, int data, int
    position) {
72      if (position == 0){
73          SinglyLinkedListNode* new_node = new SinglyLinkedListNode(data);
74          new_node->next = head;
75          new_node=head;
76      }else{
77          SinglyLinkedListNode *curr = head;
78          for (int i = 0;i<position-1;i++){
79              curr=curr->next;
80          }
81          SinglyLinkedListNode* new_node = new SinglyLinkedListNode(data);
82          new_node->next = curr->next;
83          curr->next = new_node;
84      }
85      return head;
86  }
87
88 > int main() ⋯
```



Alex Vargas A00824525

Problema 4- medio

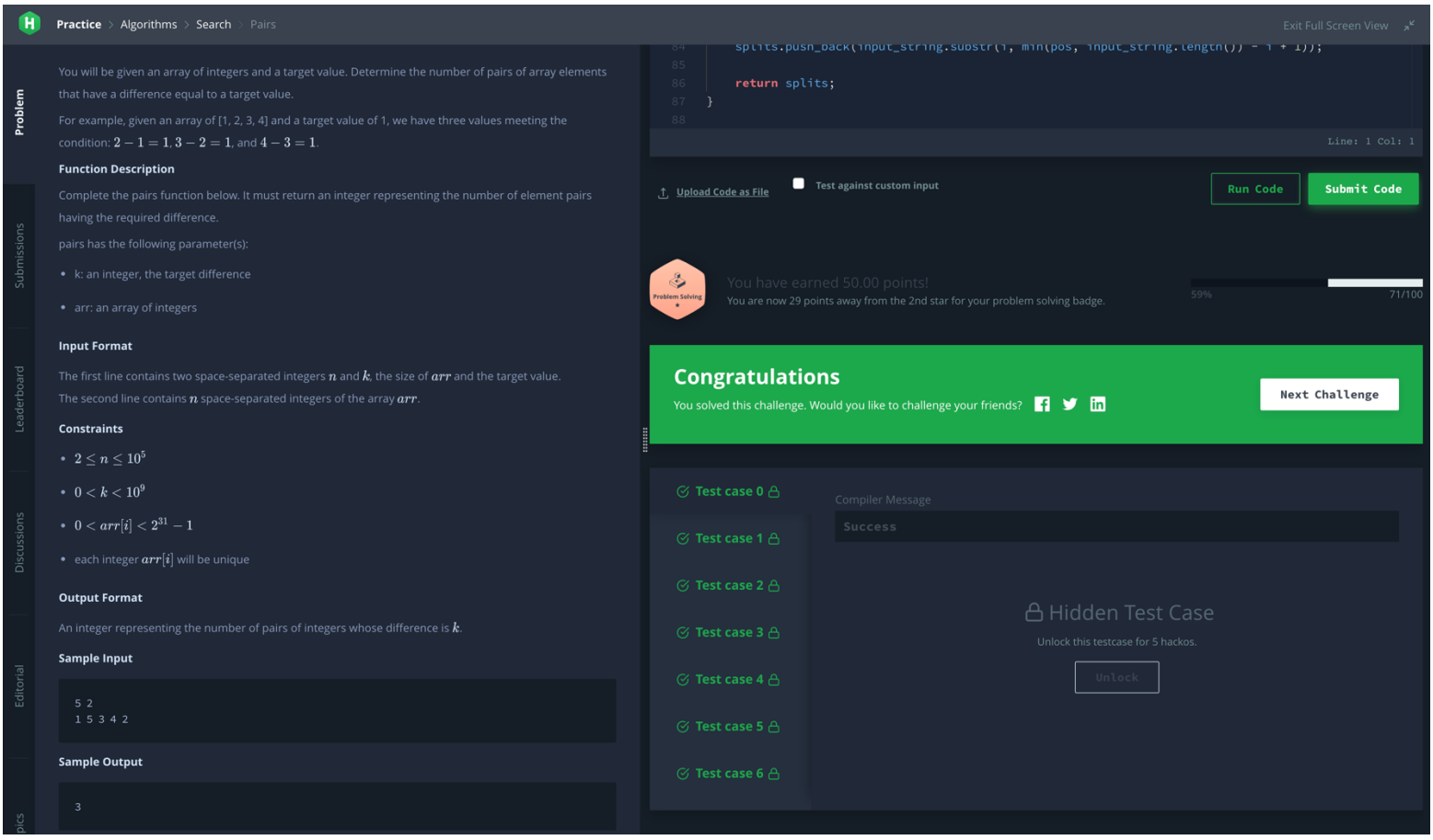
```cpp
#include <bits/stdc++.h> …

// Complete the has_cycle function below.

/*
 * For your reference:
 *
 * SinglyLinkedListNode {
 *     int data;
 *     SinglyLinkedListNode* next;
 * };
 *
 */
bool has_cycle(SinglyLinkedListNode* head) {
    SinglyLinkedListNode* slow = head;
    SinglyLinkedListNode* fast = head;

    while (slow!=NULL && fast!=NULL && fast->next != NULL){
        slow=slow->next;
        fast=fast->next->next;
        if (slow == fast){
            return 1;
        }
    }

    return 0;
}

int main() …
```



Alex Vargas A00824525

Problema 5- medio

```cpp
#include <bits/stdc++.h>

using namespace std;

vector<string> split_string(string);

// Complete the pairs function below.
int pairs(int k, vector<int> arr) {
    unordered_set <int> set;
    for (int i=0;i<arr.size();i++){
        set.insert(arr[i]);
    }

    long int iR=0;

    for (int j=0;j<arr.size();j++){
        if (set.find(arr[j]-k) != set.end()){
            iR++;
        }
    }

    return iR;
}
```



Alex Vargas A00824525

Problema 6- difícil

```cpp
#include <bits/stdc++.h>

using namespace std;

vector<string> split_string(string);

// Complete the arrayManipulation function below.
long arrayManipulation(int n, vector<vector<int>> queries) {
    vector <int> arr(n+1);
    long int one,two,three,max=0,temp=0;

    for (int i =0;i<queries.size();i++){
        one = queries[i][0];
        two = queries[i][1];
        three = queries[i][2];

        arr[one]+=three;
        ((two+1)<=n) ? arr[two+1]-=three:0;
    }

    for(int j=1;j<=n;j++){
        temp=temp+arr[j];
        (temp>max) ? max=temp:0;
    }

    return max;
}
```