

BIT: Fenwick/Binary Indexed Tree

```
/*
 * Created by Dipta Das on 23-11-2018
 * Title: Binary Indexed Tree/Fenwick Tree
 * Editorial
 * https://www.topcoder.com/community/competitive-programming/tutorials/binary-indexed-trees/
 * http://www.shafaetsplanet.com/?p=1961&fbclid=IwAR23aI879JfPHbIaW3y93Du6Ql\_68DCTxcUY6euLJUWsLvgtvj\_-b
 * https://www.geeksforgeeks.org/binary-indexed-tree-or-fenwick-tree-2/
 * https://www.youtube.com/playlist?list=PLDV1Zeh2NRsCvoyP-bztk6uXAYoyZg\_U9
 * https://www.hackerearth.com/practice/notes/binary-indexed-tree-or-fenwick-tree/
 * Source Code:
 */

#include <bits/stdc++.h>
#include <stdio.h>
#define fin freopen("input", "r", stdin)
#define whatis(x) cerr << #x << ": " << x << endl;

using namespace std;
using ll = long long;

#define mx 10000
int ar[mx];
int tree[mx];

int read(int idx){
    int sum = 0;
    while (idx > 0){
        sum += tree[idx];
        idx -= (idx & -idx);
    }
    return sum;
}

int readRange (int a, int b) {
    if (a > b) swap(a, b);
    return read(b) - read(a-1);
}

void update(int idx, int val, int n){
    while (idx <= n){
        tree[idx] += val;
        idx += (idx & -idx);
    }
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> ar[i];
        update(i, ar[i], n);
    }
    cout << read(9) << endl;
    cout << readRange(8, 9) << endl;
    update(5, 9, n);
}
```