→ Pre computation (1D ARRAY)

Q) Given array of a of N integers. Given Q queries and in each query L and R, Print sum of array elements from index L to R (L, R included).

Constraints:
$1 \le N \le 10^5$
$1 \le a[i] \le 10^9$
$1 \le Q \le 10^5$
$1 \le L, R \le N$

```
#INCLUDE <BITS/STDC++.H>
USING NAMESPACE STD;
CONST INT N = 1E5+10;
INT A[N];

INT MAIN()
{
    INT N;
    CIN >> N;
    FOR(INT I=1; I<N; I++)
    {
```

```
        CIN >> A[i];
    }

    INT Q;
    CIN >> Q;
    WHILE (Q--)
    {
        INT L, R;
        CIN >> L >> R;
        LONG LONG SUM = 0;
        FOR (INT i = L; i <= R; i++)
        {
            SUM += A[i];
        }
        COUT << SUM << ENDL;
    }

    // Time complexity -> O(N) + O(Q*N) == O(N^2)
                          == 10^10
    So, it will give TLE.

    RETURN 0;
}

// We can prevent it from TLE by using prefix sum method.
Prefix sum is storing values beforehand testing our test cases.

-> Optimized Solution:
```

```
#include <bits/stdc++.h>
using namespace std;
const int N= 1e5+10;
int A[N];
int pf[N];

int main()
{
    int n;
    cin >> n;
    for(int i=1; i<=n; i++)
    {
        cin >> A[i];
        pf[i] = pf[i-1] + A[i];
    }

    int q;
    cin >> q;
    while(q--)
    {
        int l, r;
        cin >> l >> r;
        cout << pf[r] - pf[l-1] << endl;
    }

    // Time complexity -> O(N) + O(Q) == O(N^2) == 10^5
    It won't give TLE.

    return 0;
}
```