```cpp
#include <bits/stdc++.h>
#define ll long long
#define inf (int)1e9
using namespace std;
```

Sparse table for (min / max / gcd) query in 1D-Array:

```cpp
#define log2(x) (31^__builtin_clz(x))

vector<vector<int>> st(20, vector<int>(100100)); //change the log when change the size
vector<int> a(100100);
int n;

//to build and answer the query:
void build()
{
    for (int i = 0; i < n; i++)
        st[0][i] = a[i]; // to change
    for (int j = 1; 1 << j <= n; j++)
        for (int i = 0; i + (1 << j) <= n; i++)
            st[j][i] = temp(st[j - 1][i], st[j - 1][i + (1 << j - 1)]); // to change
}

//get query in O(1)
int inline get(int l, int r)
{
    int k = log2(r - l + 1);
    return temp(st[k][l], st[k][r - (1 << k) + 1]); // to change
}
```

Sparse table for sum query in 1D-Array:

```cpp
vector<vector<int>> st(20, vector<int>(100100)); //change the log when change the size
vector<int> a(100100);
int n;

void build()
{
    for (int i = 0; i < n; i++)
        st[0][i] = a[i];
    for (int j = 1; 1 << j <= n; j++)
        for (int i = 0; i + (1 << j) <= n; i++)
            st[j][i] = st[j - 1][i] + st[j - 1][i + (1 << j - 1)];
}

//get query in O(log(n))
ll inline get(int l, int r)
{
    ll sum = 0;
    for (int j = 20; j >= 0; j--)
    {
        if ((1 << j) <= r - l + 1)
        {
            sum += st[j][l];
            l += 1 << j;
        }
    }
    return sum;
}
```