→ HASHing :

Q) Given array a of N integers. Given Q queries in each query given a number X, Print count of that number in array.

Constraints :
$1 \leq N \leq 10^5$
$1 \leq a[i] \leq 10^7$
$1 \leq Q \leq 10^5$

```
→ # INCLUDE <BITS/STDC++.h>
USING NAMESPACE STD;

INT MAIN()
{
    INT N;
    CIN >> N;
INT A[N];
FOR ( INT i=0; i < n; i++ )
{
    CIN >> A[i];
}


INT Q;
CIN >> Q;
WHILE (Q--)
{
    INT X;
    CIN >> X;
    INT CT = 0;
    FOR ( INT i=0;  i<N; i++ )
    {
        IF (a[i] == X)
        {
            CT++;
        }
    }
    COUT << CT << "\n";
}

// It's time complexity is :
   O(N) + O(Q*N) == O(N²) = 10¹⁰
   So, it will give TLE.

RETURN 0;
}

// We can prevent our solution from TLE,
```

by using hashing.
Hashing is storing values beforehand testing our test cases.

→ Optimized solution:

```
# INCLUDE <BITS/STDC++.h>
USING NAMESPACE STD;
CONST INT N = 1e7+10;
INT HSH[N];   // We don't need to initialize it
                with 0, global arrays are by
                default initialized with zero.
INT MAIN()
{
    INT n;
    CIN >> n;
    INT A[n];
    FOR ( INT i=0;  i<n;  i++)
    {
        CIN >> A[i];
        HSH [A[i]]++;
    }

    INT Q;
    CIN >> Q;
    WHILE(Q--)
    {
        INT X;
        CIN >> X;
        COUT << HSH[X] << "\n";
    }
```

Page No.

Date

// It is time complexity is :
$O(N) + O(Q) \approx O(N) = 2 * 10^5$
It won't give TLE now.

RETURN 0;
}