

## Hashing

$$\Theta = 10^5$$

number = 1

int f(number, arr)

{

    cnt = 0

    for ( i=0 ; i < n ; i++ ) →  $(5 \times O(N))$

}

        if ( arr [ i ] == number )

            cnt = cnt + 1 ;

}

1	2	1	3	2
---	---	---	---	---

$$\Theta \rightarrow \begin{cases} 1 \rightarrow 2 \\ 3 \rightarrow 1 \\ 4 \rightarrow 0 \\ 2 \rightarrow 2 \\ 10 \rightarrow 0 \\ \vdots \\ 12 \rightarrow 0 \end{cases}$$

$$O(\Theta \times N)$$



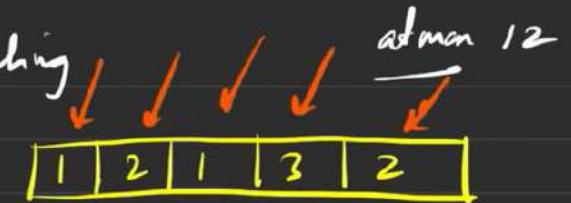
$\text{cnt} = \text{cnt} + 1; \quad O(\Theta \times n)$

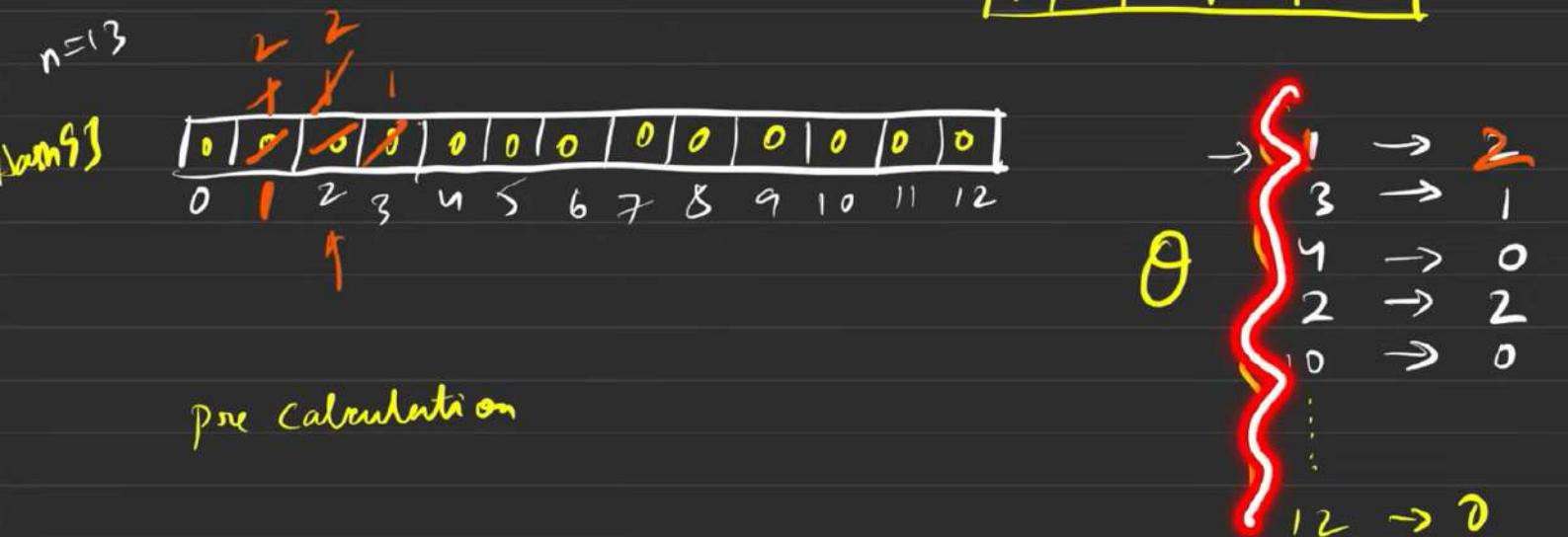
}

$O(10^5)$

return  $\text{cnt}$ ;

}

Hashing → pre storing / fetching 



```
#include<bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cin >> n;
    int arr[n];
    for(int i = 0;i<n;i++) {
        cin >> arr[i];
    }

    // precompute
    int hash[13] = {0};
    for(int i = 0;i<n;i++) {
        hash[arr[i]] += 1;
    }

    int q;
    cin >> q;
    while(q--) {
        int number;
        cin >> number;
        // fetch
        cout << hash[number] << endl;
    }
    return 0;
}
```

```
1 5
2 1 3 2 1 3
3 5
4 1
5 4
6 2
7 3
8 12
```

≡ output.txt ×  
≡ output.txt  
1 2
2 0
3 1
4 2
5 0
6

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cin >> n;
    int arr[n];
    for(int i = 0;i<n;i++) {
        cin >> arr[i];
    }

    // precompute
    int hash[13] = {0};
    for(int i = 0;i<n;i++) {
        hash[arr[i]] += 1;
    }

    int q;
    cin >> q;
    while(q--) {
        int number;
        cin >> number;
        // fetch
        cout << hash[number] << endl;
    }
}

return 0;
```

```
3
4 int main() {
5     string s;
6     cin >> s;
7
8     //pre compute
9     int hash[26] = {0};
10    for(int i = 0;i<s.size();i++) {
11        hash[s[i] - 'a']++;
12    }
13    int q;
14    cin >> q;
15    while(q--) {
16        char c;
17        cin >> c;
18        // fetch
19        cout << hash[c-'a'] << endl;
20    }
21    return 0;
22 }
```

1	abcdabehf
2	5
3	a
4	g
5	h
6	b
7	c

1	2
2	0
3	1
4	2
5	1
6	

```
int main() {
    string s;
    cin >> s;

    //pre compute
    int hash[26] = {0};
    for(int i = 0;i<s.size();i++) {
        hash[s[i] - 'a']++;
    }

    int q;
    cin >> q;
    while(q--) {
        char c;
        cin >> c;
        // fetch
        cout << hash[c-'a'] << endl;
    }
    return 0;
}
```