**Easy one**

Attempted by: **356**

/

Accuracy: **80%**

/

Maximum Score: **20**

/

7 Votes

Tag(s):

Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

**Aurora** is the Annual Cultural Festival of IIITM Gwalior which also happens to be Central India's largest Cultural Festival...! This year the promotion team of Aurora is distributing free passes for the fest if one can successfully answer the following question :

Given an array **'A'** of **'N'** non-negative integers and **'Q'** queries. Each query consist of a single integer **'X'**, you have to simply tell the no. of occurrences of integer **'X'** in the given array **'A'**.

**Input :**

* First line consists of an integer **'T'** i.e. the no. of test cases.
* First line of each test case consists of two space separated integers **'N'** and **'Q'** i.e. the no. of elements in the array and no. of queries.
* Next line contains **'N'** space separated non-negative integers.
* Following **'Q'** lines contains a single non-negative integer **'X'**.

**Output :**

* For each query output the no. of occurrences of 'X' in the given array.

**Constraints :**

* 1<=T<=10
* 1<=N<=105
* 1<=Q<=105
* 0<=array element Ai<=109
* 0<=X<=109

**SAMPLE INPUT**

1

5 3

5 8 1 4 5

1

5

3

**SAMPLE OUTPUT**

1

2

0

**Explanation**

Self Explanatory.

**Time Limit:**2.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**Bash, C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), TypeScript, Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Swift-4.1, Visual Basic

<https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/easy-one-8/description/>

1. // Sample code to perform I/O:
3. #include <iostream>
4. #include <stdio.h>
5. #include <map>
6. #define ll long long int
7. using namespace std;
9. int main() {
11. int t;
12. scanf("%d", &t);
14. while(t--)
15. {
17. int n, queryes;
18. scanf("%d %d", &n, &queryes);
20. map<ll, int> hash;
21. for(int i =0; i<n; i++) {
22. ll elem;
23. //cin >> elem;
24. scanf("%ld", &elem);
25. hash[elem]++;
26. }

29. while (queryes--)
30. {
31. ll q;
32. cin >> q;
33. if(hash.find(q ) != hash.end()) {
34. cout << hash[q] << endl;
35. }
36. else{
37. cout << "0" << endl;
38. }
40. }
41. }
43. }