**How many pizzas ?**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)

You are given two stacks of pizzas, but you can only take those pizzas from one stack that are also present in the other stack, **in the same order**, the pizzas that you can take need not be continuous.

Print the maximum amount of pizzas that you can take. (You can only take pizzas from one of the two stacks)

**INPUT**

The first line contains only one integer, t, the number of test cases.

For each case, there are two lines:

**Line 1:** 10 integers, which represent radii of the pizzas in the first stack.

**Line 2:**10 integers, each of which represent radii of pizzas in the second stack.

**OUTPUT:**

For each case, a single integer that gives the maximum amount of pizzas that you can take.

**Constraints:**

1<=T<=100

1<=radii<=1000

**Example:**

**Input:**

1  
891 424 945 741 897 514 692 221 678 168  
702 952 221 614 69 753 821 971 318 364

**Output:**

1

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=1200>

#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

int a[10];

for(int i =0; i<10; i++) {

scanf("%d", &a[i]);

}

int b[10];

for(int i =0; i<10; i++) {

scanf("%d", &b[i]);

}

int ans =0;

for(int i =0; i<10; i++) {

for(int j =0; j<10; j++) {

if(a[i] == b[j]) {

ans++;

}

}

}

printf("%d\n", ans);

}

return 0;

}