

Ranking Books

There are N books in a shelf which are already arranged in non-increasing order of number of pages in the books. Each book will be given a rank starting from 1 depending on the number of pages. The book with the highest number of pages will be ranked 1. You are writing a new book and it will be completed in M days. Everyday after adding some pages in your book, you want to find the rank of the book in the shelf. Please note that book's with same number of pages will be given same rank. Write an algorithm to find the rank of your book in the shelf for each of the M days.

Input Format

First line of input contains T - number of test cases. Its followed by 4T lines, the first line contains N - number of books in the shelf and the second line contains N integers, which denotes the number of pages in i^{th} book. The third line contains M - number of days to complete your new book and the fourth line contains M integers, which denotes the number of pages in your new book at the end of i^{th} day.

Constraints

30 points

$1 \leq T \leq 100$

$1 \leq N, M \leq 10^2$

$1 \leq arr[i] \leq 10^6$

70 points

$1 \leq T \leq 200$

$1 \leq N, M \leq 10^4$

$1 \leq arr[i] \leq 10^6$

Output Format

For each testcase, print M integers denoting the rank of your book at the end of each of the M days, separated by a new line.

Sample Input 0

```
2
7
100 100 50 40 40 20 10
4
5 25 50 120
3
45 30 12
6
5 12 25 35 45 80
```

Sample Output 0

```
6 4 2 1
4 3 3 2 1 1
```

Explanation 0

Test Case 1

When your book has 5 pages, it will be ranked 6: 100 100 50 40 40 20 10 "5"

When your book has 25 pages, it will be ranked 4: 100 100 50 40 40 "25" 20 10

When your book has 50 pages, it will be ranked 2: 100 100 50 "50" 40 40 20 10

When your book has 120 pages, it will be ranked 1: "120" 100 100 50 40 40 20 10