N meetings in one room

Submissions: 16320 (/problem_submissions.php?pid=443) Accuracy: 51.24% Difficulty: Easy (https://practice.geeksforgeeks.org/Easy/0/0/) Marks: 2

Associated Course(s): Geeks Classes (/courses/geeks-classes/) Sudo Placement (/courses/sudo-placement/)
More

Show Topic Tags

Company Tags | Amazon (/company/Amazon/) | Flipkart (/company/Flipkart/) | MakeMyTrip (/company/MakeMyTrip/) | Morgan Stanley (/company/Morgan Stanley/) | Visa (/company/Visa/)

Problems

There is **one** meeting room in a firm. There are **N** meetings in the form of (S[i], F[i]) where S[i] is start time of meeting i and F[i] is finish time of meeting i.

What is the *maximum* number of meetings that can be accommodated in the meeting room?

Input:
The first line of input consists number of the test cases. The description of T test cases is as follows:
The first line consists of the size of the array, second line has the array containing the starting time of all the meetings each separated by a space, i.e., S [i]. And the third line has the array containing the finishing time of all the meetings each separated by a space, i.e., F [i].

Output:
In each separate line print the order in which the meetings take place separated by a space.

Constraints:
$1 \le T \le 70$
$1 \le N \le 100$
$1 \le S[i], F[i] \le 100000$

Example:
Input:
2
6
1 3 0 5 8 5

2 4 6 7 9 9
8
75250 50074 43659 8931 11273 27545 50879 77924
112960 114515 81825 93424 54316 35533 73383 160252

Output:
1 2 4 5
6 7 1

Explanation:
Testcase 1: Four meetings can held with given start and end timings.

** For More Input/Output Examples Use 'Expected Output' option **

Author: madhuradlakha (https://auth.geeksforgeeks.org/user/madhuradlakha/practice/)

(/problem_submissions.php?pid=443) (/problem_submissions.php?pid=443&isSolved=ALL&lang=ALL&user=Self)

My Submissions | All Submissions

Editorial (/editorial.php?pid=443)

Monokai ▾ C++ (g++ 5.4) ▾

(https://auth.geeksforgeeks.org/edit-profile.php)

```
#include<bits/stdc++.h>
using namespace std;
struct meeting{
    int start;
    int end;
    int pos;
};
bool comparator(meeting m1,meeting m2){
    return(m1.end<m2.end);
}
int main()
 {
    int t,n;
    cin>>t;
    while(t--){
        cin>>n;
        struct meeting meet[n];
        for(int i=0;i<n;i++){
            cin>>meet[i].start;
        }
        for(int i=0;i<n;i++){
            cin>>meet[i].end;
            meet[i].pos=i+1;
        }

```