Problem PM
Task scheduler

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

A simple task scheduler schedules tasks for an embedded system. Each task has a task ID, priority value, and a function pointer to its corresponding callback function. The task ID is represented by an integer ranging from 0 to 99. The priority value is represented by an integer ranging from 0 to 255. The callback function is responsible for printing the associated task ID.

Tasks are added statically by developers. After N tasks are added, the task scheduler starts to schedule the tasks. The task with the smallest priority value will be scheduled first. If multiple tasks have the same priority value, the task added to the system earliest will be scheduled first. Now, let's see what the system prints.

Input Format

The first line contains an integer T, representing the number of test cases. Each test case below includes three lines.

For each test case, the first line is an integer $N \leq 100$, representing the number of the following two lines of integers. The second line contains N integers, each integer representing the ID of a task. The integer closer to the beginning of the second line indicates that its corresponding task was added to the system earlier. The third line also contains N integers, representing the priority value. The $i-th$ number in the third line is the priority value of the task represented by the $i-th$ number in the second line.

Output Format

Each test case outputs N integers, separated by a space. These integers are the IDs of tasks, and the order in which they are printed corresponds to the order in which they are scheduled.

Technical Specification

- $1 \leq T \leq 10$.
- $1 \leq N \leq 100$.
- $0 \leq taskID \leq 99$.
- $0 \leq priority\ value \leq 255$.

Sample Input 1

```
4
1
0
5
3
0 1 2
1 0 0
4
0 1 2 3
6 4 2 2
8
0 1 2 3 4 5 6 7
10 12 12 3 1 3 1 1
```

Sample Output 1

```
0
1 2 0
2 3 1 0
4 6 7 3 5 0 1 2
```