Problem K: Recombinant Lock

Base Program Constraints: 2s, 256 MB

Clear Reward: +1 Point

Splitting off with Polly, Calvin gets to work on cracking the Cardinal Vault! The Vault Wing contains a series of locks and traps before Calvin can access the vault, so he'll need to clear each of them in order to move on.

The first door to the vault wing is guarded by a **combination lock** with n digits, which is regularly changed by an internal system. Luckily, from her research, Olivia knows that the combination lock is formed via a predictable pattern.

The system behind the lock uses two **base combinations** to form the main lock's password. Starting with the **first** combination, the system chooses exactly 2 non-overlapping subarrays of length m. Then, the system swaps those subarrays with the subarrays in the same position in the **second** combination. This "spliced" combination is then used as the password to the main lock.

Olivia knows that on this particular day, the combination lock chosen by the system will have the **greatest possible sum of digits**. Additionally, of the passwords with the maximal sum of digits, today's password is the one that occurs **last lexographically** (i.e. is the largest when written left-to-right as an n-digit number).

Find today's password.

Input

The first line contains an integer t $(1 \leq t \leq 10^3)$ - the number of test cases.

The first line of each test case contains an integer n $(1 \leq n \leq 2 * 10^5)$, the number of digits in the combination lock.

The next two lines each contain n space-separated integers, where each integer is between 1 and 9, representing the first and second combinations used by system, respectively.

It is guaranteed that the sum of n across all test cases does not exceed $2 * 10^5$.

Output

For each test case, output today’s password as n space-separated integers.

Sample Test Cases

Sample 1 - Input

```
4
4 1
1 2 3 4
9 9 9 9
4 1
9 9 9 9
1 2 3 4
6 2
1 2 3 4 5 6
6 5 4 3 2 1
6 2
1 1 1 1 1 1
9 9 9 9 9 9
```

Sample 1 - Output

```
9 9 3 4
9 9 3 4
6 5 4 3 5 6
9 9 9 9 1 1
```

Notes

In the third test case, a maximal sum is obtained by swapping the subarray with the first two characters and the subarray with the next two characters.

In the fourth test case, note that choosing any two subarrays will result in a maximal sum, but 999911 is the code that is lexographically last.