

Problem D: Fuel Stops

Source: `fuel.{c,cpp,java}`

You are required to take a circular tour of a given set of cities: start at a certain city, visit each city once, and return to the city at which you started.

Each city is identified by a number: 1, 2, 3, etc. The numbering of the cities specifies the path you must take, but the starting point is not specified. From the highest numbered city, you proceed to City 1. For example, if there are three cities (numbered 1, 2, 3) you have three choices for completing the tour:

Start at 1, proceed to 2, then to 3, then return to 1.

Start at 2, proceed to 3, then to 1, then return to 2.

Start at 3, proceed to 1, then to 2, then return to 3.

There is a refueling station in each city, with a given quantity of available fuel. The sum of all the fuel supplies at the refuelling stations is equal to the fuel required to make the entire tour. You start with an empty tank at one of the refuelling stations. You will be running out of fuel just as you pull into the starting station upon successful completion of the tour.

You must determine which city (or cities) will qualify as a starting point for the tour without running out of fuel before returning to the starting station. Assume the fuel tank is sufficiently large to handle all of the refuelling operations.

Input

The input will contain data for several test cases. For each test case, there will be three lines of data. The first line will specify the number of cities as a single integer. The second line will specify the quantity of fuel available at each of the refuelling stations, in the order of city numbers: 1, 2, 3, etc. The third line will specify the quantity of fuel needed to get from each station to the next one, in the order of city numbers: from the station at city 1 to the station at city 2, from the station at city 2 to the station at city 3, etc; the last number specifies the quantity of fuel required to get from the highest numbered city's station back to the station at city 1. All fuel quantities are positive integers given in imperial gallons. The sum of the fuel supplies will not exceed the range of signed 32-bit integers. There will be at least two cities and up to 100000 cities. End of input will be indicated by a line containing zero for the number of cities; this line will not be processed.

Output

For each test case, there will be one line of output. After the case number, the output will list the city numbers that work as starting cities for a successful tour, as described above. In case of several possible starting cities, they must be listed in increasing order separated by a single space. Follow the format of the sample output. The Hungarian mathematician L. Lovász proved that there is always at least one possible starting city.

Sample Input

```
3
3 2 2
4 2 1
3
3 2 1
1 3 2
4
3 4 5 2
2 3 8 1
0
```

Sample Output

```
Case 1: 2 3
Case 2: 1
Case 3: 4
```