Code Jam 2015 - Round 3

Log Set

Problem

The *power set* of a set S is the set of all subsets of S (including the empty set and S itself). It's easy to go from a set to a power set, but in this problem, we'll go in the other direction!

We've started with a set of (not necessarily unique) integers S, found its power set, and then replaced every element in the power set with the sum of elements of that element, forming a new set S'. For example, if S = {-1, 1}, then the power set of S is {{}, {-1}, {1}, {-1, 1}}, and so S' = {0, -1, 1, 0}. S' is allowed to contain duplicates, so if S has N elements, then S' always has exactly 2^N elements.

Given a description of the elements in S' and their frequencies, can you determine our original S? It is guaranteed that S exists. If there are multiple possible sets S that could have produced S', we guarantee that our original set S was the *earliest* one of those possibilities. To determine whether a set S_1 is earlier than a different set S_2 of the same length, sort each set into nondecreasing order and then examine the leftmost position at which the sets differ. S_1 is earlier iff the element at that position in S_1 is smaller than the element at that position in S_2.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each consists of one line with an integer **P**, followed by two more lines, each of which has **P** space-separated integers. The first of those lines will have all of the different elements E_1, E_2, ..., $E_{\mathbf{P}}$ that appear in S', sorted in ascending order. The second of those lines will have the number of times F_1, F_2, ..., $F_{\mathbf{P}}$ that each of those values appears in S'. That is, for any i, the element E_i appears F_i times in S'.

Output

For each test case, output one line containing "Case #x: ", where x is the test case number (starting from 1), followed by the elements of our original set S, separated by spaces, in nondecreasing order. (You will be listing the elements of S directly, and not providing two lists of elements and frequencies as we do for S'.)

Limits

Memory limit: 1 GB.
$1 \le \mathbf{T} \le 100$.
$1 \le \mathbf{P} \le 10000$.
$\mathbf{F_i} \ge 1$.

Small dataset

Time limit: 240 seconds.
S will contain between 1 and 20 elements.
$0 \le$ each $E_i \le 10^8$.

Large dataset

Time limit: 480 seconds.
S will contain between 1 and 60 elements.
$-10^{10} \le$ each $E_i \le 10^{10}$.

Sample

Sample Input	Sample Output
5 8 0 1 2 3 4 5 6 7 1 1 1 1 1 1 1 1 4 0 1 2 3 1 3 3 1 4 0 1 3 4 4 4 4 4 3 -1 0 1 1 2 1 5 -2 -1 0 1 2 1 2 2 2 1	Case #1: 1 2 4 Case #2: 1 1 1 Case #3: 0 0 1 3 Case #4: -1 1 Case #5: -2 1 1

Note that Cases #4 and #5 are not within the limits for the Small dataset.

In Case #4, S = {-1, 1} is the only possible set that satisfies the conditions. (Its subsets are {}, {-1}, {1}, and {-1, 1}. Those have sums 0, -1, 1, and 0, respectively, so S' has one copy of -1, two copies of 0, and one copy of 1, which matches the specifications in the input.)

For Case #5, note that S = {-1, -1, 2} also produces the same S' = {-2, -1, -1, 0, 0, 1, 1, 2}, but S = {-2, 1, 1} is earlier than {-1, -1, 2}, since at the first point of difference, -2 < -1. So `-1 -1 2` would **not** be an acceptable answer. `1 -2 1` would also be unacceptable, even though it is the correct set, because the elements are not listed in nondecreasing order.