Roll Playing Games

Phil Kropotnik is a game maker, and one common problem he runs into is determining the set of dice to use in a game. In many current games, non-traditional dice are often required, that is, dice with more or fewer sides than the traditional 6-sided cube. Typically, Phil will pick random values for all but the last die, then try to determine specific values to put on the last die so that certain sums can be rolled with certain probabilities (actually, instead of dealing with probabilities, Phil just deals with the total number of different ways a given sum can be obtained by rolling all the dice). Currently he makes this determination by hand, but needless to say he would love to see this process automated. That is your task.

For example, suppose Phil starts with a 4-sided die with face values 1, 10, 15, and 20 and he wishes to determine how to label a 5-sided die so that there are a) 3 ways to obtain a sum of 2, b) 1 way to obtain a sum of 3, c) 3 ways to obtain 11, d) 4 ways to obtain 16, and e)1 way to obtain 26. To get these results he should label the faces of his 5-sided die with the values 1, 1, 1, 2, and 6. (For instance, the sum 16 may be obtained as 10 +6 or as 15 +1, with three different "1" faces to choose from on the second die, for a total of 4 different ways.) Note that he sometimes only cares about a subset of the sums reachable by rolling all the dices (like in the previous example).

Input

Input will consist of multiple input sets. Each input set will start with a single line containing an integer n indicating the number of dice that are already specified. Each of the next n lines describes one of these dice. Each of these lines will start with an integer f (indicating the number of faces on the die) followed by f integers indicating the value of each face. The last line of each problem instance will have the form

r m v_1 c_1 v_2 c_2 v_3 c_3 $\cdots$ v_m c_m

where r is the number of faces required on the unspecified die, m is the number of sums of interest, v_1,...,vsm are these sums, and c_1,...,c_m are the counts of the desired number of different ways in which to achieve each of the respective sums.

Input values will satisfy the following constraints: $1 \le n \le 20$, $3 \le f \le 20$, $1 \le m \le 10$, and $1 \le r \le 6$. Values on the faces of all dice, both the specified ones and the unknown die, will be integers in the range 1 ...50, and values for the v_i's and c_i's are all non-negative and are strictly less than the maximum value of a 32-bit signed integer.

The last input set is followed by a line containing a single 0; it should not be processed.

Output

For each input set, output a single line containing either the phrase "Final die face values are" followed by the r face values in non-descending order, or the phrase "Impossible" if no die can be found meeting the specifications of the problem. If there are multiple dice which will solve the problem, choose the one whose lowest face value is the smallest; if there is still a tie, choose the one whose second-lowest face value is smallest, etc.

Example

Input:
1
4 1 10 15 20
5 5 2 3 3 1 11 3 16 4 26 1
1
6 1 2 3 4 5 6
6 3 7 6 2 1 13 1
4
6 1 2 3 4 5 6
4 1 2 2 3
3 3 7 9
8 1 4 5 9 23 24 30 38
4 4 48 57 51 37 56 31 63 11
0

Output:
Final die face values are 1 1 1 2 6
Impossible
Final die face values are 3 7 9 9