# E - Choices of Life

**Run-time Limit:** 3 seconds
**Memory Limit:** 64 MB

## DESCRIPTION

In one moment of life, people must decide in one of some choices, if he takes one of it, the other possibilities of choices may available to him, not taking a choice is also a choice. Before he can take a choice, he must fulfill the requirement, it can be either 0 or more.
This often makes people feel frustrated, insecure, because of the uncertainties of the future. Everyone wants to have as many good choices so they can happy, but what's most important is they have a good ending in his final time. Good ending is when people take a good choices in his final moment of his life.
Luckily, this is just a problem that appear in Vocomfest 2018, where the future is known and defined. Each of the choices that he can take has a score, positive or negative. And the possibilities that may appear are also known.
Surita wants to be a good example of how people live his life, as his full name Suri Tauladan, given by his parents has meaning 'role models'.
Help Surita calculate the best sum of score of the choices that he can make. But remember, it must have a good ending, meaning that the choice that he take in the final moment of his life is considered good. It's better to have a low sum of score but has a good ending, rather than high score but has a bad ending.

## INPUT FORMAT

The first line contains the number of test cases **T** ($1 <= T <= 10$)
For each case, the first line contains two integers **N** ($1 <= N <= 15$) denoting total of choices that may appear, and **A** ($1 <= A <= 10$) denoting Surita's age when he has final moment.
The next N lines, each of the k-th line (k=1,2,...N) will contains **A** integers **Sk,j** (j=1,2,...,A) separated by space, with the value of either 0 or 1. The value 1 indicates that the k-th choice will appear at moment j, and 0 meaning it's not.
The next N lines, each of the k-th line (k=1,2,...,N) will contains requirements that is needed so that the k-th choice can be taken. Starting with the number of requirements **X** ($0 <= X < N$) and followed **X** integers **Rk,j** separated by space (j=1,2,...,X; $1 <=$ **Rk,j** $<= N$). It is guaranteed that the requirement of a choice will not refer to itself (e.g. the requirements for choice 2 will not contains 2)
The next line will contains **N** integers **Dk** (k=1,2,...,N; $-100{,}000 <= Dk <= 100{,}000$)**,** each denoting the value of the k-th choice.
The last line will contains an integer **V** ($0 <= V <= N$), followed by V integers denoting the choice that are considered good.

## OUTPUT FORMAT

For each case, output "Case #X: Y", where X is the case number starts from 1, and Y the output for each case, that is the best sum of Surita's life choices. Output the string 'impossible' (without quotes) if he can't get good ending.

**INPUT EXAMPLE**

```
3
3 3
0 0 1
1 1 1
1 0 0
1 3
1 3
0
30 -5 -3
3 1 2 3
3 3
1 0 1
1 1 1
1 0 0
0
0
0
30 -5 -10
2 1 2
3 3
1 0 1
1 1 1
1 0 0
0
0
0
30 -5 -10
0
```

**OUTPUT EXAMPLE**

```
Case #1: 27
Case #2: 30
Case #3: impossible
```

**EXPLANATION**

In the first case, Surita can choose choice 3 at the first moment, and get the score of (-3). Then at the second moment, he can choose to not take any choice. At the third moment, he can choose choice 1, a good ending, which its requirements (choice 3) has been chosen, and get 30 score. His sum of score would be (-3) + 30 = 27.
In the second case, Surita can choose choice 1 (which has no requirements) at the first or third moment, and do not take any other choice. That way he can get a good ending, with a total score of 30.
In third case, none of the choices are considered good, thus it is impossible to get a good ending. So we output ‘impossible’.

Notes : a choice can be taken multiple times, but the score will be calculated only once.