6701 Pirate's Treasure

Long time ago, there was a famous pirate who had collected a lot of treasure. Before he was killed, he revealed this fact to the world. It was said that anyone who found the treasure would become the king of pirates. Many pirates were attracted by the treasure and started to sail on the sea. After that, the world entered the age of pirates. The only clue for the treasure was that the map of the treasure had been divided into n pieces by the famous pirate. He made one additional copy for each piece and put the total $2n$ copies into k boxes. Because he was forgetful, he recorded how he distributed the copies on the bottoms of the boxes. There were m different keys for the boxes. To open a box, it needs at least one key. After many years, no one could find the treasure.

Recently, there is a young pirate, Steven, who gets all the k boxes and all the m keys. However, the keys are eroded so that any key will be broken after it is used to open a box. In order to rebuild the map, Steven must get at least one copy of each piece. Since each key can only be used once now, Steven may lose some pieces of the map if he doesn't choose the boxes to be opened properly.

Please write a program to help Steven to find the treasure. It is possible that there is no way to rebuild the map. If there are more than one ways to rebuild the map, output the way that the sorted sequence of the indices of the selected boxes is lexicographically the smallest. (For example, $(1, 2) < (1, 2, 3) < (1, 3, 4) < (2, 3)$.)

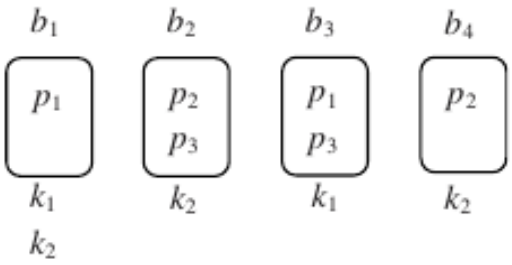

Figure 5: An example.

Consider the example in Figure 5. The map is divided into 3 pieces, p_1, p_2, p_3, and each piece has two copies. The total 6 copies are distributed into 4 boxes, b_1, b_2, b_3, b_4, and there are two keys, k_1, k_2. In this example, if we choose to open b_1 then we can only get p_1. After we open b_1, keys k_1 and k_2 are broken so that we can't open any more boxes. On the other hand, if we open b_3 and b_4, we can get all the 3 pieces to rebuild the map. Another way to rebuild the map is to open b_2 and b_3. In this example, your problem should output $(2, 3)$, since $(2, 3) < (3, 4)$.

Technical Specification

1. There are k boxes $b_1, b_2, \ldots, b_k$, where $2 \leq k \leq 500$.
2. There are n pieces $p_1, p_2, \ldots, p_n$, where $n \leq 10^5$ and $2n \geq k$.
3. There are m keys $k_1, k_2, \ldots, k_m$, where $1 \leq m \leq 10^5$.
4. Each piece has two copies and there are $2n$ copies in total.
5. Each box needs at least one key to open; each key is needed by at most 10 boxes.
6. Each box contains at least one piece; pieces in the same boxes are distinct.

Input

There are at most 20 test cases. The first line of each test case consists of the three integers k, n, and m. The following k lines describe the information of the boxes, where the i-th line (in the k lines) describes the information for box i. Each line consists of two parts. The first part starts with a positive integer $s \leq n$, and the integer is followed by s different integers indicating the pieces contained in the box. The second part starts with an integer $t \leq m$, and the integer is followed by t different integers indicating the keys needed by the box. The last test case will be followed by a line consisting of three integers '`0`'. All the integers in a line are separated by at least one space.

Output

For each case, print the output in a single line. If there is no way to rebuild the map, print '`impossible`'. Otherwise, print the sorted sequence of the solution whose sorted sequence is lexicographically the smallest. Use the format as in the Sample Output.

Sample Input

```
4 3 2
1 1 2 1 2
2 2 3 1 2
2 1 3 1 1
1 2 1 2
4 3 2
1 1 2 1 2
2 2 3 1 2
2 1 3 2 1 2
1 2 1 2
0 0 0
```

Sample Output

```
2 3
impossible
```