ICPC Teams

Rachael is the coach of her university's ICPC team. She wants to form teams of students to participate in the upcoming contest. She has been tracking each student's performance and has given each student an integer score[1]. As a reminder, in ICPC, each team must have exactly three students, and each student must be on exactly one team. The strength of a team is the sum of the scores of its three students.

Before forming the teams, Rachael asks the students if they have any requests for teammates. For every request between two students, those two students must be placed on the same team regardless of their scores.

There may be many arrangements that satisfy all of the requests. Of these, Rachael prefers the arrangement whose strongest team is as strong as possible. If several arrangements have equally strong strongest teams, she compares their second-strongest teams, then their third-strongest teams, and so on. At the first position where the team strengths differ, Rachael prefers the arrangement with the stronger team at that position.

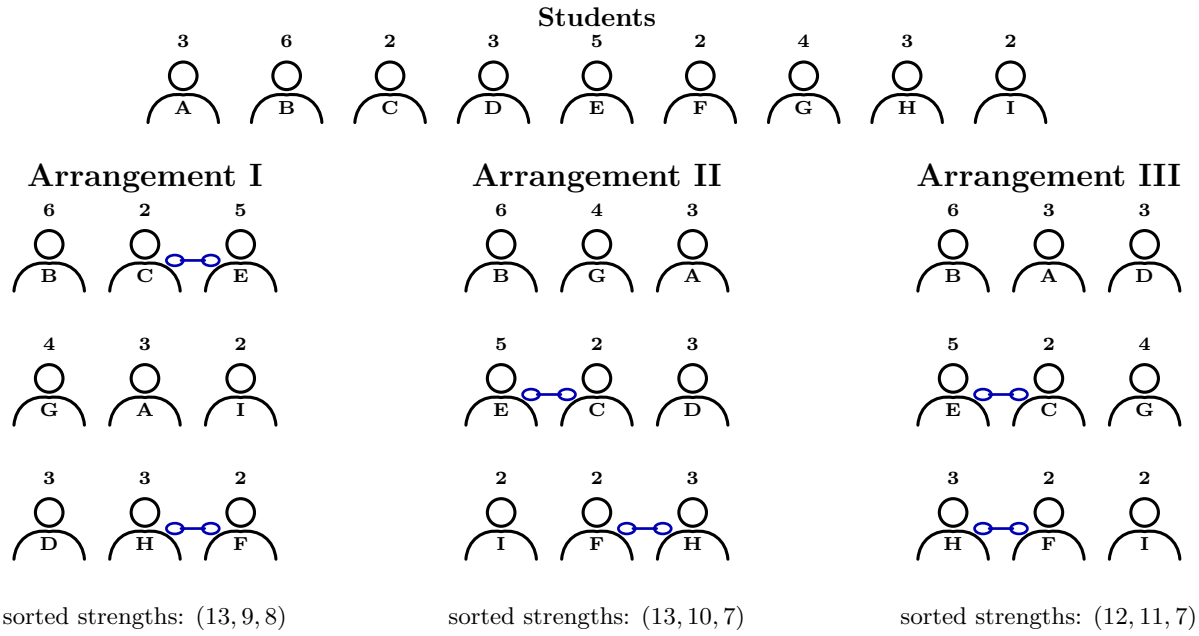


Figure 1: Say there are two requests: students C and E want to be on the same team, and students H and F want to be on the same team. Here are three possible arrangements of the 9 students. Arrangement II is preferred to Arrangement I because their strongest teams have the same strength, but Arrangement II has a stronger second-strongest team. Arrangement III is not preferred to either of the other two because its strongest team is weaker.

[1]The score takes into account things important for ICPC participants: problem-solving skills, coding speed, teamwork, reading the whole problem (including footnotes), etc.

Given the scores of all students and the requests made, find an arrangement that satisfies all requests and maximizes the list of team strengths when sorted from strongest to weakest.

Input

The first line of input contains two integers K ($1 \leq K \leq 100$), which indicates that there are $3K$ students, and R ($0 \leq R \leq 1\,000$), which is the number of requests for teammates. The second line of input contains $3K$ integers $s_1, s_2, \ldots, s_{3K}$ ($1 \leq s_i \leq 1\,000$), which are the scores of the students. Students are indexed $1, 2, \ldots, 3K$.

The next R lines describe the requests. Each of these lines contains two integers x and y ($1 \leq x < y \leq 3K$), indicating that student x and student y want to be on the same team. No two requests are the same.

Output

If it is not possible for Rachael to form an arrangement that satisfies all the requests, output `Impossible`. Otherwise, output `Possible` followed by K lines describing an arrangement that maximizes the list of team strengths when sorted from strongest to weakest.

Each line should contain a single team. Each team should be represented by three integers, the (1-based) indices of the students on that team. You may output the teams in any order, and you may output the students within a team in any order.

If there are multiple optimal solutions, any will be accepted.

Sample Input 1

```
3 2
3 6 2 3 5 2 4 3 2
3 5
6 8
```

Sample Output 1

```
Possible
2 7 1
5 3 4
9 6 8
```

Sample Input 2

```
2 3
3 1 4 1 5 9
1 2
3 4
5 6
```

Sample Output 2

```
Impossible
```

Sample Input 3

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

Sample Output 3

```
Possible
1 3 6
4 5 2
```

Sample Input 4

```
3 2
8 8 8 8 8 8 8 8 8
4 7
3 7
```

Sample Output 4

```
Possible
1 2 6
3 7 4
8 5 9
```

Sample Input 5

```
2 5
2 7 1 8 2 8
1 2
1 3
1 4
1 5
1 6
```

Sample Output 5

```
Impossible
```