

Bingbing has 3 toys: pikachu, wukong and barbie. She didn't know their exact weight, but she knows the interval (in a mysterious weight unit), shown below.

	Pikachu	Wukong	Barbie
Minimum possible weight	1	2	3
Maximum possible weight	3	4	5

Jiajia has an e-mobile, which can tell you the difference between the weights of both sides. The emobile is huge, so you can put as many toys as possible on both sides.

Bingbing asks Jiajia for the mobile, but Jiajia wants to give Bingbing a challenge: she can use each toy at most once on each side (so each toy can be used at most twice in total). Bingbing agreed. She used the mobile twice, as shown below (number D means the left part is D unit heavier than the right part):



Then, Bingbing knows the exact weights of each toy, as shown below:

	Pikachu	Wukong	Barbie
Minimum possible weight	3	4	3
Maximum possible weight	3	4	3

One month later, Bingbing got n new toys and used the mobile m times (obeying the rules above). Could you tell me the tightest bounds of each toy's weight?

Input

There will be at most 20 test cases. Each test case begins with a line containing two integers n and m ($3 \leq n \leq 200$, $1 \leq m \leq 100$). The second line contains $2n$ integers, the $(2i - 1)$ -th and the $2i$ -th integer are the initial lower bound and upper bound of the i -th toy ($1 \leq b \leq c \leq 20000$). There are m lines followed, each for a use of mobile. Each of these lines begins with 3 integers L, R, D ($L, R \geq 0$), that means there are L toys on the left, and R toys on the right. The total weight of the left side is D unit larger than the right side ($D < 0$ means the weight of the left side is $-D$ unit smaller than the right side). The next L integers are the toys on the left side, and the next R integers are the toys on the right side. It is guaranteed that each toy can appear on each side at most once. The input terminates with $n = m = 0$.

Output

For each test case, print $2n$ integers, in the same format as the input. If there is no solution, print a single '-1'.

Sample Input

```
3 2
1 3 2 4 3 5
1 1 -1 1 2
1 1 1 2 3
2 2
1 5 1 5
1 1 0 1 2
1 1 1 2 1
3 1
1 5 2 5 1 3
2 1 1 1 2 3
0 0
```

Sample Output

```
Case 1: 3 3 4 4 3 3
Case 2: -1
Case 3: 1 2 2 3 2 3
```