

# Rectangles in a Square

In two fundamental branches of modern science -- electronics and telecommunication -- progress is so marked that it may be perceived nearly as a natural power, controlling the fate of people and companies and transforming human life. Mainframes, computers, LAN, internet, built-in systems, Wi-Fi - generation upon generation of technology has sprung up within a time interval shorter than that of human life. Progress has its own life cycle, and periods of growth of semiconductor device production are interleaved with periods of decline, approximately once every five years. Experts believe that the main reason for such decline is the lack of new tools for Electronic Design Automation (EDAs), which can take full advantage of the latest technological achievements.

You are employed by the designers of a modern EDA and you have been asked by your boss to solve one of the stages of the design process. More specifically, you are to present a piece of software which, given a square-shaped board and a list of rectangular semiconductor devices, tries to place them on the board. No element may lie outside the board (even partially) or overlap with another element.

You are rather vague about the details of your task, and so (surprisingly) is your boss. "Just make sure the guys from Marketing can feature <<Efficient Automated Semiconductor Placement>> in our sales brochure" -- he says, and leaves you to it.

Eventually, you decide to pack as many of the listed chips as possible on the given board (leaving out those that simply won't fit in), and go off for the evening to the local whisky bar, wondering whether the next recession in the technological cycle won't come sooner than in 5 years' time...

## Input

$t$  – number of test cases, then  $t$  tests follow. [ $t \leq 500$ ]

In the first line of each test there is an integer  $N$ , and in the second line an integer  $K$ .  $N$  is the length of the side of the square [ $2 \leq N \leq 1000$ ], and  $K$  is the number of available rectangles [ $1 \leq K \leq 10000$ ]. Then exactly  $K$  lines follow, with 3 numbers on each of them:  $w_i$ ,  $h_i$ ,  $l_i$ .  $w_i$  - length of rectangle [ $w_i \leq N$ ],  $h_i$  - height of rectangle [ $h_i \leq N$ ],  $l_i$  - number of rectangles of this type [ $l_i \leq 200000$ ]. You may rotate a rectangle by a multiple of 90 degrees.

## Output

For each test case output integer  $R$  - number of used rectangles, and then exactly  $R$  lines. On each of these lines output integer coordinates of opposite corners of rectangle  $x_{i1}$ ,  $y_{i1}$ ,  $x_{i2}$ ,  $y_{i2}$ . Solution will be accepted if all rectangles won't intersect with each other and won't overrun the bounds of square.

## Score

The score awarded to your program is the sum of scores for individual test cases. For individual test case you will receive points equals to area cover with rectangles divided by area of square. For test in which square doesn't have empty area, you will receive 4 points. If score = xxx.xxxaaa,

aaa means the number of test cases with fully covered square.

## Example

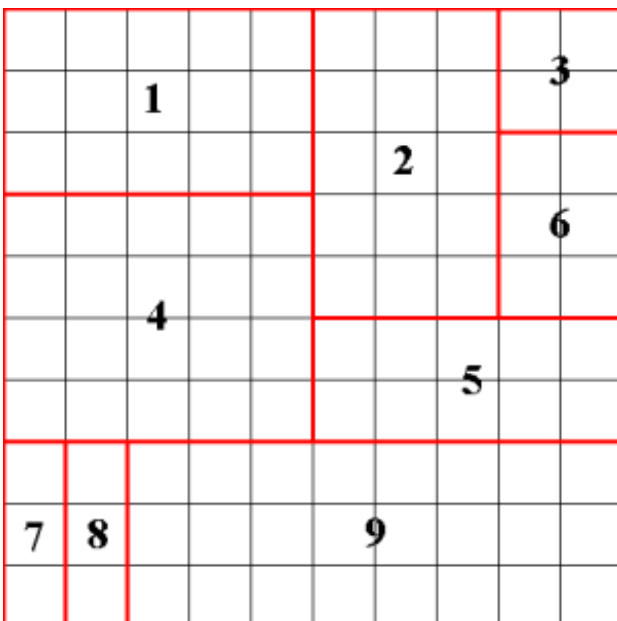
### Input :

```
1
10
8
3 5 2
2 2 1
2 3 1
2 5 1
4 5 1
1 3 2
3 8 1
1 1 1
```

### Output :

```
9
1 1 5 3
6 1 8 5
9 1 10 2
1 4 5 7
6 6 10 7
9 3 10 5
1 8 1 10
2 8 2 10
3 8 10 10
```

### Example explanation :



On the figure rectangles marked with numbers in accordance with position in example output. For this test case you will receive 4.000001 points, because square fully covered with rectangles.