# **Cube Summation**



#### **Problem Statement**

Chinese Version Russian Version

You are given a 3-D Matrix in which each block contains 0 initially. The first block is defined by the coordinate (1,1,1) and the last block is defined by the coordinate (N,N,N). There are two types of queries.

UPDATE x y z W

updates the value of block (x,y,z) to W.

QUERY x1 y1 z1 x2 y2 z2

calculates the sum of the value of blocks whose x coordinate is between x1 and x2 (inclusive), y coordinate between y1 and y2 (inclusive) and z coordinate between z1 and z2 (inclusive).

#### **Input Format**

The first line contains an integer T, the number of test-cases. T testcases follow.

For each test case, the first line will contain two integers N and M separated by a single space.

N defines the N \* N \* N matrix.

M defines the number of operations.

The next M lines will contain either

1. UPDATE x y z W 2. QUERY x1 y1 z1 x2 y2 z2

# **Output Format**

Print the result for each QUERY.

### **Constrains**

$$1 \le M \le 1000$$

$$1 \le x1 \le x2 \le N$$

$$1 \le y1 \le y2 \le N$$

$$1 \le z1 \le z2 \le N$$

$$1 <= x,y,z <= N$$

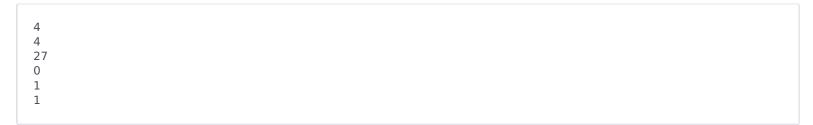
$$-10^9 \le W \le 10^9$$

# **Sample Input**

```
2
45
UPDATE 2 2 2 4
QUERY 1 1 1 3 3 3
UPDATE 1 1 1 23
QUERY 2 2 2 4 4 4
QUERY 1 1 1 3 3 3
2 4
UPDATE 2 2 2 1
QUERY 1 1 1 1 1
```



# **Sample Output**



# **Explanation**

First test case, we are given a cube of 4 \* 4 \* 4 and 5 queries. Initially all the cells (1,1,1) to (4,4,4) are 0. UPDATE 2 2 2 4 makes the cell (2,2,2) = 4

QUERY 1 1 1 3 3 3. As (2,2,2) is updated to 4 and the rest are all 0. The answer to this query is 4.

UPDATE 1 1 1 23. updates the cell (1,1,1) to 23. QUERY 2 2 2 4 4 4. Only the cell (1,1,1) and (2,2,2) are non-zero and (1,1,1) is not between (2,2,2) and (4,4,4). So, the answer is 4.

QUERY 1 1 1 3 3 3. 2 cells are non-zero and their sum is 23+4=27.