

[Chinese Version](#)

[Russian Version](#)

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

```
UPDATE x y z W
```

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

```
QUERY x1 y1 z1 x2 y2 z2
```

Calculate the sum of the values 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).

Function Description

Complete the *cubeSum* function in the editor below.

cubeSum has the following parameters: - *int n: the dimensions of the 3-d matrix - *string operations[m]*: the operations to perform

Returns

- *int[]*: the results of each **QUERY** operation

Input Format

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

For each test case, the first line contains two space-separated integers, *n* and *m*. *n* defines the $n \times n \times n$ matrix.

m defines the number of operations.

The next *m* lines will contain an operation either of these forms:

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

Constraints

$$1 \leq T \leq 50$$

$$1 \leq n \leq 100$$

$$1 \leq m \leq 1000$$

$$1 \leq x1 \leq x2 \leq n$$

$$1 \leq y1 \leq y2 \leq n$$

$$1 \leq z1 \leq z2 \leq n$$

$$1 \leq x, y, z \leq n$$

$$-10^9 \leq W \leq 10^9$$

Sample Input

```
2
4 5
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 1
QUERY 1 1 1 2 2 2
QUERY 2 2 2 2 2 2
```

Sample Output

```
4
4
27
0
1
1
```

Explanation

In the first test case, there is a cube of $4 * 4 * 4$ and there are 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 $23+4 = 27$.