**Mobile phones**

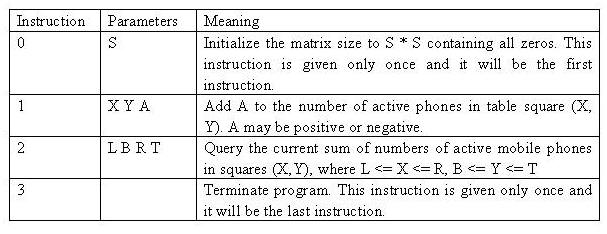
总时间限制: 5000ms 内存限制: 65536kB

**描述**

Suppose that the fourth generation mobile phone base stations in the Tampere area operate as follows. The area is divided into squares. The squares form an S \* S matrix with the rows and columns numbered from 0 to S-1. Each square contains a base station. The number of active mobile phones inside a square can change because a phone is moved from a square to another or a phone is switched on or off. At times, each base station reports the change in the number of active phones to the main base station along with the row and the column of the matrix.  
  
Write a program, which receives these reports and answers queries about the current total number of active mobile phones in any rectangle-shaped area.

**输入**

The input is read from standard input as integers and the answers to the queries are written to standard output as integers. The input is encoded as follows. Each input comes on a separate line, and consists of one instruction integer and a number of parameter integers according to the following table.



The values will always be in range, so there is no need to check them. In particular, if A is negative, it can be assumed that it will not reduce the square value below zero. The indexing starts at 0, e.g. for a table of size 4 \* 4, we have 0 <= X <= 3 and 0 <= Y <= 3.  
  
Table size: 1 \* 1 <= S \* S <= 1024 \* 1024  
Cell value V at any time: 0 <= V <= 32767  
Update amount: -32768 <= A <= 32767  
No of instructions in input: 3 <= U <= 60002  
Maximum number of phones in the whole table: M= 2^30

**输出**

Your program should not answer anything to lines with an instruction other than 2. If the instruction is 2, then your program is expected to answer the query by writing the answer as a single line containing a single integer to standard output.

**样例输入**

0 4

1 1 2 3

2 0 0 2 2

1 1 1 2

1 1 2 -1

2 1 1 2 3

3

**样例输出**

3

4

**来源**

IOI 2001

**Codes:**

二维树状数组，有了前面题目的基础，其实二维很容易就出来了。

#include<iostream>

#include<stack>

#include<map>

#include<list>

#include<queue>

#include<cstdio>

#include<string.h>

#include<string>

using namespace std;

#define lowbit(x) (x&(-x))

int s;

//int num[1028][1028] = { 0 };

int C[1028][1028] = { 0 };

int sum(int x, int y) {

int nx = x, adds = 0;

while (nx) {

int ny = y;

while (ny) {

adds += C[nx][ny];

ny -= lowbit(ny);

}

nx -= lowbit(nx);

}

return adds;

}

void update(int x,int y,int d) {

int nx = x;

//num[x][y] += d;

while (nx <= s) {

int ny = y;

while (ny <= s) {

C[nx][ny] += d;

ny += lowbit(ny);

}

nx += lowbit(nx);

}

}

int main() {

int instr;

int l, b, r, t;

scanf("%d %d", &instr, &s);

while (1) {

scanf("%d", &instr);

if (instr == 3)

break;

else if (instr == 1) {

scanf("%d %d %d", &l, &r, &t);

update(l + 1, r + 1, t);

}

else if (instr == 2) {

scanf("%d %d %d %d", &l, &b, &r, &t);

printf("%d\n", sum(l, b) + sum(r+1, t+1) - sum(l, t+1) - sum(r+1, b));

}

}

system("pause");

return 0;

}