Problem 1. Sum of powers

(Time Limit: 2 seconds)

Problem Description

A schoolboy would like to calculate the sum

$$S_k(n) = \sum_{i=1}^{n} i^k$$

for some fixed natural number k and different natural number n. He observed that calculating i^k for all i ($1 \le i \le n$) and summing up the results is too slow to do it, because the number of required arithmetical operations increases as n increases. Fortunately, there is another method which takes only a constant number of operations regardless of n. It is possible to show that the sum $S_k(n)$ is equal to some polynomial of degree k+1 in the variable n with rational coefficients, i.e.,

$$S_k(n) = \frac{1}{M}\left(a_{k+1}n^{k+1} + a_k n^k + \cdots + a_1 n + a_0\right)$$

for some integer numbers $M, a_{k+1}, a_k, \ldots, a_1, a_0$.

We require that integer M must be positive and as small as possible. Under this condition the entire set of such numbers (i.e. $M, a_{k+1}, a_k, \ldots, a_1, a_0$) will be unique for the given k. You have to write a program to find such set of coefficients to help the schoolboy make his calculations faster.

Input Format

The first line contains an integer t indicating the number of the test cases. For each test case, there a positive integers k ($1 \le k \le 20$) in one line.

Output Format

For each test case, output the integer numbers $M, a_{k+1}, a_k, \ldots, a_1, a_0$ sequentially in a line. Numbers should be separated by one space.

Example

Sample Input:	**Sample Output:**

1 2	6 2 3 1 0

Problem 2. OOXX

(Time Limit: 1 second)

Problem Description

HH just learnt that how to play Tic-tac-toe. He wrote many Os and Xs on everywhere he can write. His big brother, HHH was annoyed by these things. Thus HHH want to ask HH a hard question. If HH answers incorrectly, he cannot write Os or Xs in HHH's room anymore.

As you may have guessed already, the question is also about Os and Xs! The question is that what's the maximum number of Os and/or Xs HH can write, if HH can only write Os and/or Xs on a straight line following these rules?

- At most M_O Os can be written in total.
- At most M_X Xs can be written in total.
- At most C_O Os can be written consecutively.
- At most C_X Xs can be written consecutively.

For example, if $M_O = M_X = 3$ and $C_O = C_X = 2$, then

- OXOO is valid
- OOXOXX is valid and is one of the longest valid solution
- OOXOO is invalid, since it contains 4 Os.
- OXXXO is invalid, since it contains 3 consecutive Xs.

Please help HH to answer this question!

Technical Specification

- The number of test case $T \leq 1000$
- $0 \leq M_O, M_X, C_O, C_X \leq 2000000000$

Input Format

The first line contains an integer T indicating the number of the test cases. For each test case, there are four integers M_O, M_X, C_O, C_X in one line.

Output Format

For each test case, output the maximum number of Os and/or Xs HH can write if HH must conform above rules.

Example

Sample Input:	**Sample Output:**

2
3 3 2 2
1 9 2 3
	6
	7

Problem 3. Approximate Palindrome

(Time Limit: 3 seconds)

Problem Description

A palindrome is a string that is the same when you read it from either direction. For example, “ada”, “bb” and “abba” are palindromes. However, some strings that are not exactly palindromes may not be too far away from this. For example, consider the string “abaa”. You can change the “b” to “a” in order to make this string a palindrome. This edit operation is called a *substitution*. You can also remove the last “a” and the resultant string is also a palindrome. This is called a *deletion* operation. Moreover, you can also add an “a” in front of this string. This is called an *insertion* operation. Your job is as follows. Given a string of length *n*, please evaluate the minimum number of edit operations, including insertions, deletions and substitutions, that are required in order to turn this string into a perfect palindrome.

Technical Specification

- The alphabet of input strings is the set of lowercase English letters.
- $1 \le n \le 5000$.

Input Format

The first line of the input contains an integer indicating the number of test cases. Each test case has a single line which specifies the input string for this test case.

Output Format

For each test case, please output the minimum number of edit operations (i.e. insertions, deletions and substitutions) that can make this string a palindrome.

Example

Sample Input:	Sample Output:
3 a abaa abcccda	0 1 1

Problem 4. Shrinking Lands

(Time Limit: 3 seconds)

Problem Description

You are a farmer on an $M*N$ grid land, in each week you have to select exactly one rectangular area in the grid to grow tomatoes. On the grid cell locating at the i-th row and the j-th column of the land can grow $A_{i,j}$ kilograms of tomato per week if that piece of the land is chosen within the area. (Note that the chosen rectangles should contain several complete grid cells and hence they must form rectangles with sides parallel or perpendicular to the sides of the grid.)

But it seems that the quality of the land becoming worse and worse. At the end of each week, someone will buy exactly one 1*1 grid cell of the land to build their house. So your rectangular area must NOT contain any of these houses. After $M*N$ weeks, you have no land to crop so till then you will decide to change your job to code farmer who will be producing codes in the virtual world.

Given another matrix $B_{i,j}$, indicating that at exactly end of $B_{i,j}$-th week, that piece of the land will be bought and become unavailable. How many kilograms of tomato at the end you can get if you choose the rectangular area wisely in each week?

Technical Specification

- M, N, $1 \leq M, N \leq 100$
- $1 \leq A_{i,j} \leq 10000$.
- $1 \leq B_{i,j} \leq M*N$, all numbers will be distinct.

Input Format

The first line of the input contains an integer T ($1 \leq T \leq 20$) indicating the number of test cases. Each test case start with two integers M and N, then matrix A_{ij} and B_{ij} follows.

Output Format

For each test case, please output the kilograms of tomato you can get.

Example

Sample Input:	**Sample Output:**

2
2 3
1 1 1
1 1 1
1 2 3
4 5 6
1 6
10 10 10 10 10 10
6 3 5 2 1 4
	19
	160

Problem 5. Lightning Strike

(Time Limit: 5 seconds)

Problem Description

There are n trees on a plane, ordered 1, …, n in terms of their height. Tree 1 is the shortest, and tree n is tallest. Assume tree i would release carbon dioxide of amount c_i if it is burnt. After some period of time, there were q lightning strikes. Every time a lightning strikes it will affect a rectangular area of $[L, R]\times[B, T]$. If there exists a tree in that area, then the lightning would hit the tallest tree among them and the tree hit would burn. What is the total carbon emission?

Technical Specification

- Number of test cases T: $1 \le T \le 20$
- Number of trees n: $1 \le n \le 30000$
- Number of lightning strikes q: $1 \le q \le 15000$
- Coordinates of tree (x, y): $0 \le x, y \le 1000000$
- Amount of carbon emission c of each tree: $0 \le c \le 50000$
- The boundaries L, R, B, T are signed 32-bit integers.

Input Format

The first line of the input contains an integer T indicating the number of test cases. The first line of each test case contains two integers n, q, where n is the number of trees on this plane and q is the number of lightning that struck this area. For the next n lines there will be three integers x, y, c each. On the i-th line of these n lines, x, y represents the coordinate of the i-th shortest tree (i.e., the tree is located at (x, y)), and it will release c amount of carbon dioxide if it is burnt. The q lines following would each contain four signed 32 bit integers L, R, B, T revealing the information about the lightning, where the j-th line indicated that the j-th lightning strike affects the area $[L, R] \times[B, T]$.

Output Format

For each test case, please output a number representing the total carbon dioxide emission.

Example

Sample Input:	**Sample Output:**

2
10 5
1 1 1
2 2 2
3 3 4
4 4 8
2 1 16
3 2 32
4 3 64
5 4 128
3 1 256
4 2 512
1 2 2 3
1 2 3 4
3 4 1 2
2 4 1 3
1 7 1 4
1 1
1 1 1
3 3 4 4
	898
	0