NEW BULGARIAN UNIVERSITY

Second inter-university programming contest

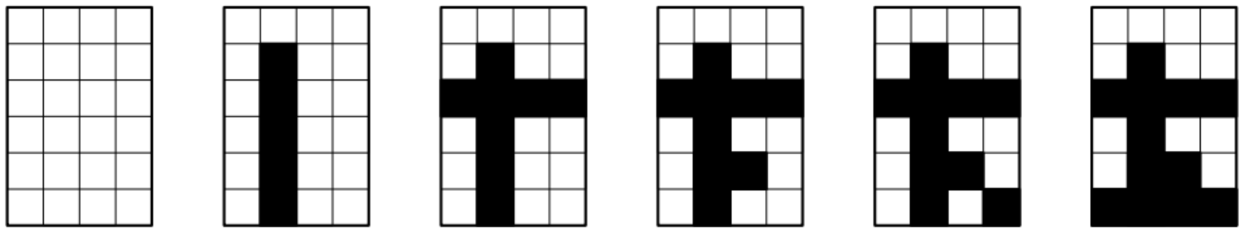
April 1, 2017

A. Strokes

An art pattern is a white sheet of paper that is divided in n x m squares. An artist creates his masterpiece by consecutively placing p horizontal or vertical black strokes. Each stroke begins from a square with coordinates (x_1, y_1), and ends in a square with coordinates (x_2, y_2) ($x_1 = x_2$ *or* $y_1 = y_2$), by changing the color in black of all the squares (x, y), for which $x_1 \le x \le x_2$ and $y_1 \le y \le y_2$.

The beauty of a painting is expressed in the number of regions in the net. Each region is composed from one or more connected white squares. These squares must share a horizontal or a vertical side, but not to be placed in diagonal. The number of such regions we call the beauty of the painting, and initially it is equal to 1.

Your task is to find the beauty of the painting after each painted stroke. This is how it is changing for the test case:



1 регион   3 региона   3 региона   4 региона   3 региона

Input

On the first line of the standard input will be given the number of the test cases. The first line of each of them will contain three positive integers n, m и q ($1 \le n, m \le 1000$, $1 \le q \le 10^4$). After that there will be q lines, each of them with four integers x_1, y_1, x_2 и y_2 ($1 \le x_1 \le x_2 \le n$, $1 \le y_1 \le y_2 \le m$), describing the consecutive stroke. For each stroke $x_1 = x_2$ and/or $y_1 = y_2$.

Output

After each input stroke, output on a new line on the standard output the beauty of the paining, after the stroke has been drawn.

Example input	Example output
1 4 6 5 2 2 2 6 1 3 4 3 2 5 3 5 4 6 4 6 1 6 4 6	1 3 3 4 3

B. Goldbach's hypothesis

Goldbach's hypothesis is one of the oldest, and most famous unsolved problems in the Number's theory, and in mathematics as a whole. The hypothesis states that each even integer that is bigger than 2 can be expressed as the sum of two prime numbers.

Your task is given an integer x ($4 \leq x \leq 32000$) to find all the unique representations of the integer as sum of two prime numbers.

Input

For each test case, on a new line will be given the number x.

Output

For each test case print on a separate line of the standard output the number of the representations p. On the next p lines output the representations themselves, in ascending order of the first addend. It always has to be the smallest of the two addends. Follow the shown output format in the example below, by placing a blank line between each two separate test cases, but not after the last one.

Example input	Example output
4 26 100	4 has 1 representation(s) 2+2 26 has 3 representation(s) 3+23 7+19 13+13 100 has 6 representation(s) 3+97 11+89 17+83 29+71 41+59 47+53

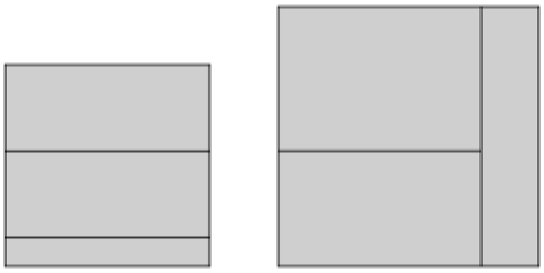
NEW BULGARIAN UNIVERSITY

Second inter-university programming contest

April 1, 2017

C. Rectangles

Given three rectangles, discover whether they can be sticked next to each other in such a way, that they would form a square. For example (the first two test cases):



Input

Each test case is composed by three lines, that contain the height H_i , and the width W_i of the consecutive rectangle *i*. $100 \geq H_i \geq W_i \geq 1$, also $H_1 \geq H_2 \geq H_3$. Between each two test cases there is a blank line.

Output

For each test case output Case #x: YES/NO, depending on whether the given three rectangles can form a square.

Example input	Example output
7 3 7 1 7 3 9 2 7 4 7 5 3 1 3 2 3 3	Case #1: YES Case #2: YES Case #3: NO

D. The Party

You are planning a big party in order to celebrate your birthday. You have *n* friends, and the bar where you are going to celebrate is with capacity *k* persons. Each of your friends would come if certain other of your friends is also attending the party. Of course, you want to maximize the number of people who will come to your party and for that reason you have to write a program that will help you to determine this.

Input

The first line of each test case contains the number of your friends n and the number of seats at the bar k ($1 \le k \le n \le 1000$). The second line contains n numbers, each of which indicates the preference of the i-th of your friends, i.e. he would come to your party only if x_i comes too. End of the entrance is marked with 0.

Output

For each test case display the maximum number of people who attended your party, taking into account their preferences and capacity of the bar.

Sample input	Sample output
4 4 1 2 3 4 12 3 2 3 4 5 6 7 4 7 8 8 12 12 5 4 2 3 1 5 4 0	4 2 3

E. Cameras

On the long straight street are located n houses, numbered from 1 to n. The houses are equidistant from each other and on k of them are placed cameras. Find the minimum number of cameras that need to be placed so that each sequence of r houses has at least two cameras.

Input

On the first line of the standard input will be set the number of test cases. The first row of each of the test cases will contain the integers n ($2 \le n \le 100000$), k ($0 \le k \le n$) and r ($2 \le r \le n$). The next k rows contain the numbers of the houses (their locations), where the cameras are placed. There is no house with more than one camera.

Output

For each test case, displayed on a separate line the number of searched cameras.

Sample input	Sample output
1 15 5 4 2 5 7 10 13	3

F. Dice

Two players take turns throwing dice. The dice of both players are like your familiar hexagonal dice, with the exception that some of the numbers might be repeated.

First of course is the first player after him is the second. Player whose dice shows a large number, written on top of the dice wins. When numbers are equal there is no winner.

By set of numbers written on the walls of the dice for both players, determine the probability the first player to win.

Input

On the first line of standard input will be assigned the number of test cases. The first line of each of the test cases contains six numbers written on the sides of the die of the first player, the second line contains those of the second player. It is guaranteed that these numbers are between 1 and 6 inclusive.

Output

For each test case, display on a separate line the probability the first player to win. Results must output with accuracy at least 5 decimal places.

Sample input	Sample output
2 1 2 3 4 5 6 6 5 4 3 2 1 4 1 4 4 1 4 3 3 3 3 3 3	0.50000 0.66667

G. Interesting numbers

We call an interesting number a number, of which at least half of the digits are the same. Any leading zeros do not count! For example 3223 and 110 are interesting numbers, while 97791 and 123 are not.

Write a program, which finds the count of all interesting numbers between X ($100 \le X \le 10^{18}$) и Y ($X \le Y \le 10^{18}$) inclusive.

Input

Each test case is set on a separate line of the standard input, and consists of natural numbers X and Y, separated by each other with an interval.

Output

For each test displayed one number in a row – the count of interesting numbers between X and Y inclusive.

Sample input	Sample output
`110 133`	`14`

The interesting numbers between 110 and 133 are 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 121, 122, 131 и 133

H. Intervals

N ($1 < N < 100000$) integers in the interval [0, 1000000000] are given and also Q ($1 < Q < 100000$) queries of type “How many are the integers greater than or equal to A and less than or equal to B?”.

Input

The program should operate with a lot of samples. On the first row of any sample N integers will be given. On the second row of the sample the number of the queries Q will be given. Any of the following Q rows of the sample contains two integers A and B ($0 \le A \le B \le 1000000000$).

Output

For every query print the number of integers in the sequence greater than or equal to A and less than or equal to B.

Sample input	Sample output
3 2 7 5 6 2 3 2 4 2 5 2 7 4 6 8 10	2 2 3 4 1 0

NEW BULGARIAN UNIVERSITY

Second inter-university programming contest

April 1, 2017

I. Cocktails

A barman is a great fan of vodka and he can drink it clean or to prepare a variety of cocktails in which it is the primary alcohol. He can experiment with N alcoholic and soft drinks, numbered from 1 to N, combining them in any way to obtain a new cocktail. Some of the drinks is not good to be blended as the resulting cocktail will be disgusting. For example, it would not be a good idea to add brandy to vodka.

Write a program to determine the maximum number of different types of cocktails that can be made by combining the available N drinks in such a way that no one of the cocktails contains two or more incompatible drinks. Two cocktails are different, if one has an additive that is lacking in the other.

Input

Any test case starts with two integers N and M, $1 \le N \le 20$, $1 \le M \le 400$. Any of the following M rows contains pairs of two integers a and b, $1 \le a, b \le N, a \ne b$ – identifiers of two incompatible drinks. Some of the pairs could be repeated in the input data. The end of the input is marked with -1.

Output

For any test case, the number of possible cocktails should be printed on a different row of the standard output. Vodka without any added drinks also be considered as a cocktail..

Sample input	Sample output
3 2 2 1 3 2 3 0 3 3 1 2 3 1 2 3 -1	5 8 4

NEW BULGARIAN UNIVERSITY

Second inter-university programming contest

April 1, 2017

J. Searching

You should find the number of ways a given matrix can be recognized as a submatrix of another given matrix.

Input

The first row of any task contains four positive integers h_p w_p h_m w_m – the height and the width of the pattern matrix and the height and the width of the matrix that it's going to be searched for. Any of the following h_p rows contains w_p symbols 'o' or 'x'. After that it follows h_m rows, any of which contains w_m symbols of the same type.

Output

For any of the samples print the answer to a different row of the standard output.

Restrictions:

$1 \le h_p, w_p \le 2\ 000$
$1 \le h_m, w_m \le 2\ 000$
$h_p \le h_m$
$w_p \le w_m$

Sample input	Sample output
4 4 10 10 oxxo xoox xoox oxxo xxxxxx**oxxo** **oxxo**oo**xoox** **xoox**xx**xoox** **xoox**xx**oxxo** **oxxo**xxxxxx ooooxxxxxx xxx**oxxoxxo** ooo**xooxoox** ooo**xooxoox** xxx**oxxoxxo**	4

The four submatrices corresponding to the pattern are given in bold.