Problem A. Papers

Source file name:	papers.c, papers.cpp, papers.java, papers.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Eddy Cael Mamani Canaviri - New Vision Data - Bolivia`

Eddy had a rectangle paper on his desk, and in order to do some Christmas gifts, he needed to cut the paper in two equal-area parts. One day, her daughter Ailin saw it, and cut an smaller perfect rectangle of the paper. When Eddy arrived to his house, he wanted to do this task, but soon he realized that the task was a bit complicated, because the remain paper does not had a regular shape to easily cut it. Can you help Eddy with this task? In order to simplify the problem we can imagine the first rectangle with sides parallel to the axis X, Y in the Cartesian plane. Please note. She can cut a rotated rectangle... She is a crazy girl...

Input

The input contains several test cases. Each test case is described in 8 lines, and each line contain two real numbers with five decimal digits. The first 8 numbers are the coordinates for the first rectangle. The coordinates are given in the format: $x_1\ y_1\ x_2\ y_2\ x_3\ y_3\ x_4\ y_4$ in counterclockwise order. For the first rectangle the list of points starts from the bottom left point. The next 8 numbers represent the second rectangle also in counter clockwise order. Is guaranteed that the second rectangle is inside the first rectangle and also is guaranteed that the rectangles does not have a empty area.

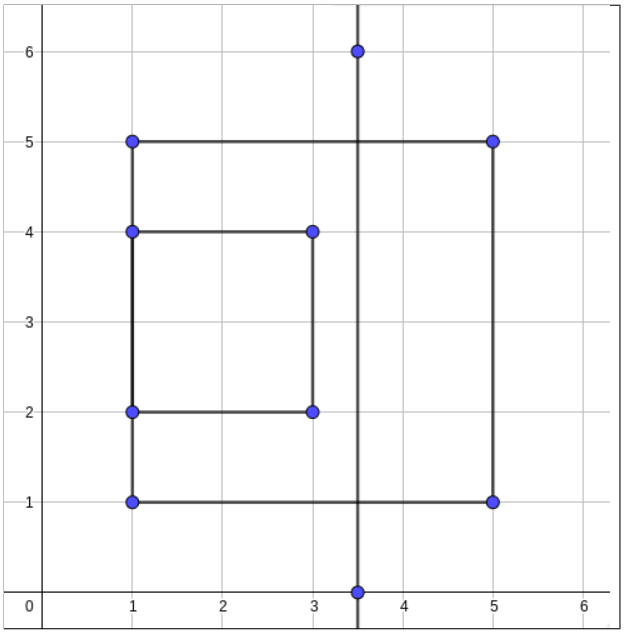
$0 \leq x_i, y_i \leq 1000$

Output

For each test case you must print a line with `NO` if is not possible cut the remain paper in the way explained above. Otherwise print `YES` and then four numbers with 5 decimal places. Those numbers represents two points of the straight line that must cut the paper. In order to evaluate your solution, the difference between the areas delimited by your answer must not differ more that 0.01.

Example

This is a possible solution in the example



Input	Output
1.00000 1.00000 5.00000 1.00000 5.00000 5.00000 1.00000 5.00000 1.00000 2.50000 3.00000 2.50000 3.00000 4.50000 1.00000 4.50000	YES 3.50000 0.00000 3.50000 6.00000

Use fast I/O methods

Problem B. Missing Problemset

Source file name:	missing.c, missing.cpp, missing.java, missing.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Javier Eduardo Ojeda Jorge - Universidad Mayor de San Simón - Bolivia`

As many times, the problemset was printed close to the beginning of the contest but unfortunately for the organizers the printer was broken, so one of the staff printed the problemset in his house, but because there are a lot of teams at the contest site, the staff person needed a wheelbarrow to carry the lot of printed problemsets. In this task we need that you help your partner to carry the wheelbarrow, you must to meet him in the less time possible, assuming that the staff person and you run at some speed in meters per second. Remember the contest will begin soon, so no one will wait the other at some intersection.

Given the city with their respective streets and intersections, the location of the house of the staff guy, the headquarters where you are and the speed of the staff and you, find the minimum possible time to meet at some street of the city.

Input

Input Format

The input consists of multiple test cases. Each test case begins with two integers I ($1 \leq I \leq 500$) and S ($1 \leq S \leq 65000$), the intersections and the streets respectively. The second line contains two integers T ($1 \leq T \leq 50$) the staff person velocity and Y ($1 \leq Y \leq 50$) your velocity. The next S lines have three integers A ($1 \leq A \leq I$) the origin of the street, B ($1 \leq B \leq I$) the destiny of the street and C ($1 \leq C \leq 10000$) the length of the road in meters. Asumme that $A \neq B$ and each road can used in both ways and every pair of intersections at most have one street joining them. The last line contains two integers H ($1 \leq H \leq I$) the intersection where the staff house is located and Q ($1 \leq H \leq I$) the intersection where the headquarters are located.

Output

Print the minimum time in seconds(rounded to two decimal places) to meet the staff guy from the headquarters.

Example

Input	Output
7 8 2 3 1 2 4 2 5 4 2 4 8 2 3 9 3 6 6 4 6 5 5 6 8 6 7 7 1 7 3 2 2 3 1 2 4 2 3 12 1 3 3 2 2 2 1 2 4 2 3 4 1 3 2 1 2 3 1 2 12 1 2	4.60 3.20 2.00 2.40

Use fast I/O methods

Problem C. Counting Sums

Source file name:	counting.c, counting.cpp, counting.java, counting.py
Input:	Standard
Output:	Standard
Author(s):	Gabriel Gutierrez - Universidad Tecnológica de Pereira - Colombia

In this problem you must calculate the number of ways of writing N as a sum of positive integers, regardless of the order of their summands. For example, 4 can be represented in five ways:
4
$3 + 1$
$2 + 2$
$2 + 1 + 1$
$1 + 1 + 1 + 1$

Input

The input consists of several test cases (not more than 10^4). The input ends with End Of File (EOF). Each test case consists of a single line containing an integer N ($1 \leq N \leq 10^4$).

Output

For each test case, you must print a single line containing the result modulo $10^9 + 7$.

Example

Input	Output
1 2 3 4	1 2 3 5

Use fast I/O methods

Problem D. A shortest path problem

Source file name:	shortest.c, shortest.cpp, shortest.java, shortest.py
Input:	Standard
Output:	Standard
Author(s):	Branimir F. Espinoza Argollo - Universidad Mayor de San Andrés - Bolivia

You are given an undirected graph with weighted edges. You have to find the minimum length of path between vertex 0 and vertex $n-1$ and the number of those paths with minimum length.

Input

There are many several test cases. Each test case begins with two positive integers n and m ($2 \leq n \leq 100000, n-1 \leq m \leq 100000$) - the number of vertices and the number of edges in the graph.

Each of the following m lines describes a corresponding edge. Each one contains three integers u, v and c ($0 \leq u, v < n, 1 \leq c \leq 1000$) - these numbers denote an edge that connects vertices u and v and has weight c.

The graph may contain multiple edges between the same pair of vertices and loops. It is guaranteed that the graph is connected.

Output

For each test case print one line with two numbers separated by a space, the minimum length of path between vertices 0 and $n-1$, and the number of paths between vertices 0 and $n-1$ with minimum length modulo 10^9+7.

Example

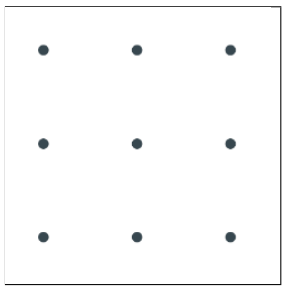
Input	Output
5 12 4 3 3 3 2 1 0 2 2 2 0 2 0 1 4 4 1 2 4 1 5 0 1 2 3 1 1 3 3 1 2 1 1 0 3 1	4 3

Use fast I/O methods

Problem E. Pattern

Source file name:	pattern.c, pattern.cpp, pattern.java, pattern.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Eddy Cael Mamani Canaviri - New Vision Data - Bolivia`

Ailin bought a new phone, so she was using it frequently. She was taking photos and also downloading images and memes, and after a while, she realized that she needs to put a password to her phone. The classic way to lock a phone is setting up a pattern using nine points. for example:



Ailin known this fact, and she was playing with patterns and after some attempts, Ailin realized that she is good making interesting patterns...

Today she wants to hack the father's cellphone. But the pattern looks a bit complicated, so she decides to make a program in order to verify if the pattern can be unlocked. She only has a screen capture of the father's pattern and now she needs your help.

The points are numbered from 1 to 9, from up to bottom (in rows) and from left to right (in each row). A pattern is a sequence of points that makes a graph. You can use every point only once, and if a line cross over an intermediate points, these points may not be considered, so you can use them later. For example if you go from 1 to 3 in a the grid, the line pass over the point number 2. and the point number 2 is not used. so you can go to the point number 2, and so on. Sometimes the graph description is very small and she needs to complete the missing points. For example: if the graph description has a line from 1 to 7 and from 7 to 9, she knows that the point to be used are: 1, 4, 7, 8, and 9. Please see the input sample for more details.

Input

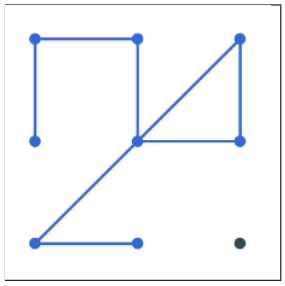
In order to simplify the input the pattern will be represented as a graph (the edges can be repeated). The input file contains several test cases. Each test case starts with a line containing the value of M. Then M lines follow: each line has a pair of values A, B: this means a line from A to B. A line can pass through several points. Remember: a pattern can not finish in the same point that it started. A point can be used only once. $1 \leq A, B \leq 9$

Output

For each case, if the pattern is not valid print `NO`. Otherwise print `YES` and in the next line print the sequence of points that makes that valid pattern. The numbers must be separated by spaces. If there are multiple solutions, print the lexicographically lowest solution. Note that you must use all the points that belong to the input lines.

Explanation

For the first two test cases, the pattern is the same:



Note that the points are numbered:

1	2	3
4	5	6
7	8	9

Example

Input	Output
7 1 2 4 1 2 5 5 6 6 3 3 7 7 8 2 1 3 2 5 4 1 2 2 6 5 6 1 5 2 1 7 7 9 4 1 7 7 9 1 7 8 9	YES 4 1 2 5 6 3 7 8 YES 1 3 2 5 NO YES 1 4 7 8 9 YES 1 4 7 8 9

Use fast I/O methods

Problem F. Pells Numbers

Source file name:	pells.c, pells.cpp, pells.java, pells.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Jorge Terán Pomier - Universidad Mayor de San Andrés - Bolivia`

The Fibonacci sequence can be obtained with the following formula:

$$f_n = f_{n-1} + f_{n-2}$$

A generalization of this formula defines what we call Fibonacci numbers of order n. This numbers are defined as follows

$$f_n = nf_{n-1} + f_{n-2}$$

The Pell numbers are the numbers of the Fibonacci sequence of order 2:

$$f_n = 2f_{n-1} + f_{n-2}$$

The first numbers in the sequence are:

$$0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, 13860, \ldots$$

In this problem we are interested in divisibility of the numbers. For example Pells number 12 that is in the position 4 is divisible by 3, Pells number 70 is divisible by 5.

We are interested in Pell's numbers that can be divided by $(2, 3, 4, 7, 13)$.

Input

The input consists of multiple test cases until end of file is reached. Each test comes in a line containing one numbers $0 \leq a \leq 50000$ that indicates the position in the secuence.

Output

For each test case write a line indicating if the Pell's number in position a is divisible by $(2, 3, 4, 7, 13)$. as shown in the example.

Example

Input	Output
1 4 6 33 24	-1 Pell's number 4 is divisible by 2 Pell's number 4 is divisible by 3 Pell's number 4 is divisible by 4 Pell's number 6 is divisible by 2 Pell's number 6 is divisible by 7 -1 Pell's number 24 is divisible by 2 Pell's number 24 is divisible by 3 Pell's number 24 is divisible by 4 Pell's number 24 is divisible by 7

Use fast I/O methods

Problem G. Password

Source file name:	password.c, password.cpp, password.java, password.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Eddy Cael Mamani Canaviri - New Vision Data - Bolivia`

Ailin wants to use a password for her email, so in order to choose a good password, she takes a string T for her passwords. She knows that the password must be changed frequently, so every month, she takes a number K and get the K-th substring from all the distinct substrings in T.

For example, if $T = ababa$ the distinct substrings ordered are: $a, ab, aba, abab, ababa, b, ba, bab, baba$, so the 5-th substring is: $ababa$. We can assume that if K is greater that the total number of distinct substrings the K-th substring will be an asterisk symbol $*$.

But there is a problem. Her friend Emily saw her notebook and knows the number K! That is why she need to add an order in the alphabet, so the K-th substring will depend of that order. Only she knows the alphabet, and in this way, her passwords are secure.

In the previous example the alphabet order was:

$$abcdefghijklmnopqrstuvwxyz$$

. Now if we change the alphabet order to

$$zyxwvutsrqponmlkjihgfedcba$$

the distinct ordered substrings for $T = ababa$ will be: $b, ba, bab, baba, a, ab, aba, abab, ababa$.

In order to build her password, she needs the substring defined by the alphabet in the way described and also the number of occurrences of that string.

Input

The input contain several test cases. Each test case starts with the string T of lowercase English letters. The next line contains the alphabet order of 26 lowercase English characters. All the characters in the alphabet are distinct. The next line contains the number of queries Q. Each of the next Q lines contains a number K.

$1 \leq |T| \leq 100000$

$1 \leq Q \leq 20$

$1 \leq K \leq 10^{18}$

Output

For each query print two lines with the previously described answers

Example

Input	Output
ababa abcdefghijklmnopqrstuvwxyz 3 1 5 9 ababa zyxwvutsrqponmlkjihgfedcba 3 1 5 9 ababa abcdefghijklmnopqrstuvwxyz 3 15 2 10	a 3 ababa 1 baba 1 b 2 a 3 ababa 1 * 0 ab 2 * 0

Use fast I/O methods

Problem H. How Many Digits does N have? I

Source file name:	howmany.c, howmany.cpp, howmany.java, howmany.py
Input:	`standard`
Output:	`standard`
Author(s):	`Hugo Humberto Morales - Universidad Tecnológica de Pereira - Colombia`

Recently, Professor *Humbertov Moralov* was organizing his workplace, he needed space to be able to locate his Christmas decorations. For this reason, he began selecting documents to throw in the trash or to recycle. In this process he found some freehand notes with the puzzle: How many digits does $N = 2^{2003} \cdot 5^{2007}$ have?

Using this interesting puzzle as our starting point, the problem you are asked to solve now is: Given two nonnegative integers a, b ($0 \leq a,\ b \leq 10^5$), find out the total of digits of $N = 2^a \cdot 5^b$.

Note: it is guaranteed that $|a - b| \leq 10^4$.

Input

Input begins with an integer t ($1 \leq t \leq 5 \times 10^5$), the number of test cases, followed by t lines, each line contains two nonnegative integers a, b ($0 \leq a,\ b \leq 10^5, \quad |a - b| \leq 10^4$).

Output

For each test case, you should print a single line containing one integer, denoting the total of digits of N.

Example

Input	Output
8 10 8 8 10 2003 0 0 2007 2003 2007 1000 100 10000 10000 20000 10000	9 10 603 1403 2006 371 10001 13011

Use fast I/O methods

Problem I. Mersenne Test

Source file name:	mersenne.c, mersenne.cpp, mersenne.java, mersenne.py
Input:	Standard
Output:	Standard
Author(s):	Jorge Terán Pomier - Universidad Mayor de San Andrés - Bolivia

Your friend Lucas is trying to proof a new mathematical idea. He wants to know if he multiply two numbers and subtract 1, the answer is prime. A Mersenne number is one that is one that has the form $2^n - 1$.

In this problem you will be given two numbers a, b and the expected answer is if $(2^a \cdot 2^b - 1)$ is a prime number.

Input

The input consists of multiple test cases. Each test case comes in one line containing two integers $(2 \leq a, b \leq 1000)$. The input ends when reach the end of file.

Output

For each test case write a line in the output with the sentence "is prime" or "not prime" as shown in the sample.

Example

Input	Output
2 2	2^4-1 not prime
2 5	2^7-1 is prime
3 3	2^6-1 not prime
3 4	2^7-1 is prime
20 31	2^51-1 not prime

Use fast I/O methods

Problem J. The Producer Trouble

Source file name:	producer.c, producer.cpp, producer.java, producer.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Javier Eduardo Ojeda Jorge - Universidad Mayor de San Simón - Bolivia`

The producer has a list of melodies for a song. He has the hard task to make a song with melodies such that the most of the people like.

The people are splitted in groups according to their musical preferences, if a song has a melody that some group likes, the group will like the song.

The producer knows this fact, but you cannot put all the melodies in some song, expecting that all people will like it, a song can be ruined if you do that because some melodies are incompatibles with other melodies.

Given the list of mutually incompatible melodies and the musical preferences of the people groups, find the maximum number of groups that will like the song and the minimum number of the melodies used for this purpose.

Input

The input file contains several test cases. Each test case starts with a line containing two integer values M ($1 \leq M \leq 20$) and G ($1 \leq G \leq 20$), the number of melodies and the number of groups, respectively. The next line contains an integer C ($0 \leq C \leq M$),the number of incompatible melodies groups. The next C lines contains first an integer I ($1 \leq I \leq M$) the number of mutually incompatible melodies in the group C_i, followed by I integers separated by a space, the melodies that belong to the group, it's guarantee that no one melody belong to more than one group. The next G lines contains first an integer L ($1 \leq L \leq M$) the number of melodies that the group G_i likes, followed by L integers separated by a space, the melodies that the group G_i likes.

Output

For each case, print the maximum number of groups that will like the song, and the minimum number of melodies used for this purpose.

Example

Input	Output
5 5 2 2 1 2 3 3 4 5 1 2 2 1 2 1 3 3 3 4 5 1 5 5 3 1 5 1 2 3 4 5 2 1 2 2 3 4 3 3 2 4 5 3 5 1 1 1 2 1 3 1 4 1 5 2 1 2 2 3 4 3 3 2 4 5 3 0 2 1 2 2 3 4 3 3 2 4	4 2 2 1 3 2 3 2

Use fast I/O methods

Problem K. Tri Lucas

Source file name:	trilucas.c, trilucas.cpp, trilucas.java, trilucas.py
Input:	`standard`
Output:	`standard`
Author(s):	`Eddy Cael Mamani Canaviri - New Vision Data - Bolivia`

Ailin learned Lucas numbers the past year. Do you remember? Eddy asked her again about the Tri lucas numbers and she said: "The problems about lucas numbers are so easy.. :) " and started solving the problem. But she could not solve the problem!

The statement for this year problem is: Given an array of N numbers you must to answer Q two-type queries.

1) Given two numbers L_i, R_i you need to calculate the value of S given by the following:

$$S = \sum_{k=L_i}^{R_i} L(a_k)$$

2) Given two numbers L_i, R_i you need to increase the value of all numbers by d in the range defined by these two numbers. (L_i, R_i inclusive).

By the way, remember that the Tri Lucas sequence is defined as:

$$L(0) = 0$$

$$L(1) = 2$$

$$L(2) = 3$$

$$L(n) = L(n-1) + L(n-2) + L(n-3) \qquad \forall \qquad n > 2$$

Input

The first line on the input contains the number of test cases. For each test case you have a line with the number N, the next line contains the numbers a_i. Next line contain the number of queries. Each query is defined in the format defined above. The first number is the query's type: 1 for the first type, and then values for L_i, R_i. Otherwise the query is the second type, so the next values will be $L_i, R_i md$

$1 \leq N, Q \leq 50000$

$1 \leq a_i \leq 100000 \qquad \forall \qquad 1 \leq i \leq N$

$0 \leq d \leq 100$

$L_i \leq R_i$

Output

For each query of the type 1, output the value of S defined above. Print this modulo $10^9 + 7$.

Example

Input	Output
1 10 2 5 1 2 4 2 2 3 1 6 6 1 8 10 2 6 9 2 1 1 5 1 6 9 2 8 9 1 1 6 9	40 36 43 63

Use fast I/O methods