GRAN PREMIO 2016 PRIMERA FECHA

Problem A. Acronyms

Source file name:	acronyms.c, acronyms.cpp, acronyms.java
Input:	Standard
Output:	Standard
Author(s):	Manuel Felipe Pineda Loaiza - UTP Colombia

There are several interesting developments in the history that were named in honor of their inventors, for example:

- RSA is made of the initial letters of the surnames of Ron Rivest, Adi Shamir, and Leonard Adleman.
- BPS is made of the initial letters of Bogomolnyi, Prasad, and Sommerfield.

Sometimes, people change the name of those developments with other words (not necessarily meaningful), for example:

- RSA: Randomized Super Algorithm.
- BPS: Beyond Perturbative States.

In this task you must check if a name may have come from certain authors. A name comes from certain authors if the initials of the authors are the same as the initials of the words in the name. Note that the initials should match in the same order that they are given in the input.

Input

The input consists of several test cases. Each test case consists of two lines:

- The name of the authors separated by spaces.
- The name to check separated by spaces.

All the words consist of lowercase english letters.

The number of authors is greater than 1 and less than 10. This number may be different from the number of words in the name to check.

Output

For each test case print "yes", if the name may have come from the given authors. Print "no" otherwise.

Example

Input	Output
rivest shamir adleman	yes
randomized super algorithm	yes
bogomolnyi prasad sommerfield	no
beyond perturbative states	no
arthur jack sean	
amazing jurisprudential victory	
rivest adleman shamir	
randomized super algorithm	

Problem B. Tobby Bones

Source file name:	tobbybones.c, tobbybones.cpp, tobbybones.java
Input:	Standard
Output:	Standard
Author(s):	Santiago Gutiérrez Alzate - UTP Colombia and Google

Tobby is a very intelligent dog who loves to collect bones. In his house Tobby has a large shelf in which he stores his bones. The shelf is divided in N slots and each of those slots can hold at most one bone. The only difference between bones is their size, which is represented by an integer. Tobby is constantly buying new bones to fill his shelf and to replace other bones from his shelf.

Tobby occasionally takes some of his bones for exhibitions around the country. Since Tobby is a lazy dog, he decides which bones to carry to an exhibition in the following way: he chooses two indexes i and j with $i \leq j$ and he carries to the exhibition all bones located between positions i and j of his shelf that have a size of at most S.

Given a log of bones purchases and of exhibitions that Tobby needs to make between purchases, your task is to tell Tobby how many bones he should bring to each exhibition.

Initially all the slots in Tobby's shelf are empty.

Input

The input contains several test cases. Each test case begins with one line, in which there are two positive integers N $(1 \leq N \leq 6 * 10^5)$ and M $(1 \leq M \leq 2 * 10^6)$ that represent, respectively, the number of slots in Tobby's shelf and the number of queries in the log. Then M lines follow, each containing a log entry. Each log entry starts with a number Q:

If Q is 0, then the log entry indicates that Tobby bought a bone. In that case two numbers follow: I $(0 \leq I \leq N - 1)$ and B $(1 \leq B \leq 6 * 10^5)$, indicating the index within the shelf in which the bone will be stored and the size of the bone. If there is already a bone stored in index I, then Tobby just replaces the bone in that index with the new bone.

If Q is 1, then the log entry indicates that Tobby wants to know how many bones he should bring to an exhibition. In that case three numbers follow: I, J $(0 \leq I \leq J \leq N - 1)$ and S $(1 \leq S \leq 6 * 10^5)$, indicating that Tobby wants to make an exhibition with all bones located between I and J (inclusive) of his shelf that have size less than or equal to S.

Output

For each exhibition query in the input, you should print a single line with the number of bones that Tobby should bring to that exhibition.

Example

Input	Output

5 10
0 0 5
0 1 6
0 2 7
1 0 2 5
1 0 2 6
0 3 5
0 4 6
1 0 4 5
0 0 8
1 0 4 7
	1
	2
	2
	4

Problem C. Tobby and Query

Source file name:	tobbyquery.c, tobbyquery.cpp, tobbyquery.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Jhon Jiménez - UTP Colombia`

In his free time Tobby is always searching for interesting things. This time Tobby created the following problem: given a sequence of n integer numbers, Tobby would like to know how many different numbers are in the range $[l,\ r]$ $(r \geq l)$

Input

The input has several test cases. The first line of each test case contains an integer n $(1 \leq n \leq 10^5)$, the size of the sequence of numbers. The next line contains n values a_i $(0 \leq ai \leq 9)$, the numbers in the sequence. The next line contains an integer q $(1 \leq q \leq 10^4)$, the amount of queries. Then there are q lines, each line contains a query: two integers l and r $(1 \leq l,\ r \leq n)$.

Output

For each test case print q integers, representing the amount of different numbers in the range $[l,\ r]$ for each query in the input.

Example

Input	Output

7
0 2 3 3 7 5 2
3
1 1
2 4
2 7
5
7 7 7 7 7
2
4 5
1 5
	1
	2
	4
	1
	1

Problem D. Standard Deviation

Source file name:	deviation.c, deviation.cpp, deviation.java
Input:	`standard`
Output:	`standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

In mathematics, the standard deviation of a set of n integer numbers is defined as:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}$$

where $\overline{x}$ is the average of the set of n integer numbers for which the standard deviation is being calculated. That average is calculated as:

$$\overline{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$

The task is to calculate, in an efficient way, the standard deviation of the first n odd positive integer numbers.

Input

There are several test cases in the input. Each test case consists of a single line containing a positive integer number n ($2 \leq n \leq 10^6$) which indicates the amount of consecutive odd numbers (starting from one) that should be considered when calculating the standard deviation. The last test case has a value of 0, for which you shouldn't generate any response.

Output

For each test case, you should print a single line containing a floating point number: the standard deviation of the first n odd positive numbers. The absolute error of your answer should not be greater than 10^{-6}.

Example

Input	Output
10 100 1000 10000 100000 1000000 0	6.055301 58.022984 577.638872 5773.791360 57735.315593 577350.557865

Use fast I/O methods

Problem E. Tobby and the LED display

Source file name:	tobbyled.c, tobbyled.cpp, tobbyled.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Carlos Arias Londoño - UTP Colombia`

Tobby got his first job! Being a lazy puppy, the work that he has found is not very demanding and it consists of observing a LED display during T minutes and reporting the status of the LED display once this time runs out. The LED display is capable of displaying N characters and the text can move in two directions, Left or Right. In every minute the character that occupies the i_{th} position moves to the $(i-1)_{th}$ or $(i+1)_{th}$ position, depending on the direction in which the text moves in the LED display.

The LED display works in a circular way, therefore, if the character that occupies the $i = 1$ position moves to the left its new position will be $i = N$, moreover, if the character that occupies the $i = N$ position moves to the right its new position will be $i = 1$.

i.e. if Tobby got the board shown below where $N = 10$, $T = 3$ and the direction in which the text moves is Right, the following will happen:

```
Minute 0
+-------------------+
| | | |T|o|b|b|y|!|!|
+-------------------+

Minute 1
+-------------------+
|!| | | |T|o|b|b|y|!|
+-------------------+

Minute 2
+-------------------+
|!|!| | | |T|o|b|b|y|
+-------------------+
```

After 3 minutes, Tobby should report the board shown below.

```
+-------------------+
|y|!|!| | | |T|o|b|b|
+-------------------+
```

As it has been said, Tobby is very lazy and wont spends his time on this boring task, that's why he is willing to give you a bone from his first payment as reward **:)**.

Input

The input consists of several test cases and must be read until EOF.

The first line of each test case contains two integers N, T ($1 \leq N \leq 50000, \quad 1 \leq T \leq 10^{14}$), and one character D ($D =$ 'L' or $D =$ 'R'), here N indicates the number of characters that the LED display can show, T shows the number of minutes that Tobby must wait to report the LED display state and D is the direction in which the LED display will work $L = Left$ and $R = Right$.

Then, there will be 3 lines and each one has $(2 * N) + 1$ characters. The first and third line are the upper edge and the lower edge respectively of the LED display, The second line shows the initial content of the LED display.

Output

For each test case the output must consist of 3 lines each one will have $(2 * N) + 1$ characters. The first and third line are the upper edge and the lower edge respectively of the LED display and the second one will show the LED display state after T minutes.

Example

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>10 3 R
+-------------------+
| | | |T|o|b|b|y|!|!|
+-------------------+</td><td>+-------------------+
|y|!|!| | | |T|o|b|b|
+-------------------+</td></tr>
</table>

Problem F. Triangular Test II

Source file name:	test02.c, test02.cpp, test02.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Alejandro Moreno Agudelo - UTP Colombia`

Hugov Molotov, an evil computer scientist, is obsessed by triangular numbers and how to use them to control the world. Molotov wants to choose a new henchman, so he has prepared a hard task, to test if you are worthy. The task consist in, given a positive integer number N, tell what is the minimum amount of triangular numbers whose sum is equal to N.

Note: a triangular number is given by:

$$\sum_{i=1}^{X} i = 1 + 2 + 3 + 4 + \ldots + X$$

where X is any positive integer number.

Input

The input may contain several test cases. Each test case is presented on a single line, and contains one positive integer N ($1 \leq N \leq 3 \cdot 10^6$). The end of the test cases is given by the end of file (EOF).

Output

For each test case, your program must print one positive integer denoting the minimum amount of triangular numbers, whose sum is equal to N. Each test case must generate one output line.

Example

Input	Output
1 2 6 10	1 2 1 1

Problem G. Tobby and the line game

Source file name:	tobbyline.c, tobbyline.cpp, tobbyline.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Diego Agudelo España - UTP Colombia`

Tobby the pet enjoys a lot going to the circus and he likes to take part in the games available for smart pets like him. This time a new circus has just arrived to Tobby's town and they are promoting a new game called the line game.

The line game is a very simple game played over a rectangular field whose lower left corner is $\mathbf{p}_L = (x_L,\ y_L)$ and its upper right corner is $\mathbf{p}_R = (x_R,\ y_R)$. In one game there are multiple players and each of them have to throw two coins over the game field. The game's goal is to make the line joining the places where the two coins landed as large as possible. The player who manages to make the longest line from his threw coins is the winner of the game and therefore the winner of a big prize as well.

Tobby has been watching some rounds of the line game and he is wondering which are the chances for him to win this game. He knows that he can barely throw the coins with his dog's paws so it can be assumed that all the points over the game field are equally probable for the coins to land and it can also be assumed that the two throws are independent. However, Tobby is so skilled that you can be sure his coins will always land inside the game field or over its boundaries. Tobby wants to compute the expected value of the squared length of the resulting line since this result can help him to make a choice about whether or not he should take part in the line game. Can you help him?

For this problem the coin landing locations can be assumed to be 2D points over a continuous plane. Suppose that the first coin landed at $\mathbf{p}_1 = (x_1,\ y_1)$ and the second at $\mathbf{p}_2 = (x_2,\ y_2)$ so Tobby is interested in computing the expectation over $\|\mathbf{p}_1 - \mathbf{p}_2\|^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$.

Input

The input file contains several test cases. Each test case comprises 4 integer numbers, namely x_L, y_L, x_R and y_R which represent the rectangular field as described above. The end of the input file will be given by EOF.

Constraints:

- $-500 \leq x_L,\ y_L,\ x_R,\ y_R \leq 500$
- $x_L < x_R$
- $y_L < y_R$

Output

Print one number, the required expected value. The absolute error of your answer should not be greater than 10^{-5}.

Example

Input	Output

`-100 -50 23 400`
`0 0 500 1`
	`36271.5`
	`41666.833333333314`

Problem H. Painting the Wall

Source file name:	painting.c, painting.cpp, painting.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Sebastián Gómez - UTP Colombia`

In his years of youth rebellion, Humbertov Moralov decided to paint the walls of this university. But unlike the rest of the revel young people, Moralov invented a robot to paint the wall for him.

The robot can execute two simple instructions to draw lines, namely:

- hline r c1 c2: draw a horizontal line in the row r between columns c1 and c2.
- vline c r1 r2: draw a vertical line in the column c between rows r1 and r2.

Now Moralov wants you to write a program that given the piece of rebel art to paint, outputs a program for the robot with the minimum number of instructions to paint the wall.

Hint: Some algorithms that you may find useful to solve this problem may have a theoretical complexity analysis that seems too high for the given input size. Remember that these complexity analysis are usually done thinking on the worst case, and many algorithms run much faster in practice. The input file for this problem was not constructed to break any particular solution, so as long as your solution is polynomial and not too crazy you should give this problem a try!

Input

The input consists of several test cases. Each test case begins with a line with two integers R and C corresponding to the number of rows and columns of the design to paint in the wall. Then R lines follow each with C characters. A character '`*`' means that the cell needs to be painted, and '`.`' means that the cell should be left unpainted.

- $1 \leq R, C \leq 800$

Output

For each test case print in one line a single integer n corresponding to the minimum number of instructions to paint the wall. Then n lines should follow with the instructions to actually paint the wall. If there are multiple solutions output any of them.

Example

Input	Output

`5 7`
`.*...*.`
`.*...*.`
`.*****.`
`.*...*.`
`.*...*.`
	`3`
	`vline 2 1 5`
	`vline 6 1 5`
	`hline 3 2 6`

Problem I. Tobby on Tree

Source file name:	tobbytree.c, tobbytree.cpp, tobbytree.java
Input:	Standard
Output:	Standard
Author(s):	Manuel Felipe Pineda Loaiza - UTP Colombia

Do you remember Tobby? He is a very cute dog who lives in tobbyland.

Tobby lives in a city where any two houses are connected by exactly one path, and that path is bidirectional. All the houses are identified by an integer $0 \leq id < N$, and have another integer indicating its corresponding value.

Tobby found on internet an interesting game called, “the subtraction game” that goes as follow:

Given a sequence of N elements: a_0, a_1, …, a_{n-1}:

```
If all the numbers are equal, the game ends.
Otherwise
    Select two numbers which are unequal
    Subtract the smaller number from the larger number
    Replace the larger number with the result from above.
```

The game **always** ends and no matter how you play, it will always terminate **on the same value**. The result of the game is that value.

The game is so cool for Tobby that he decides to play it as much as possible. For this reason each day Tobby visits two friends who live in the houses u and v and records all the values of the houses in the path from u to v (inclusive). At the end of the day he plays the substraction game with the sequence that he wrote.

The value of a house can change sometimes, but Tobby will always know the new value.

In order to help Tobby, he asks to answer the following types of queries:

1. given u, v: Find the result of the subtraction game on the path from u to v (inclusive).

2. given u, x: Change the value of the house u by x.

Input

The first line contains an integer $N \leq 50000$, the number of houses. The next line contains N integers, the i-th number is the value of the i-th house $1 \leq value \leq 10000$.

The next $N - 1$ lines contain two integers $0 \leq u, v \leq N - 1$ indicating a connection between the houses u and v.

The next line contains an integer $1 \leq q \leq 10000$ followed by q lines each containing a query. The first number of each of those lines is the query type and the rest of the line contains the query parameters.

Output

For each query of type 1, print the result of the subtraction game in the path from u to v.

Example

Input	Output

5
5 15 20 15 9
0 2
0 3
3 1
3 4
3
1 2 1
2 3 3
1 1 4
	5
	3

Problem J. Tobby Stones

Source file name:	tobbystones.c, tobbystones.cpp, tobbystones.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Santiago Gutiérrez Alzate - UTP and Google`

Tobby is an intelligent Boston Terrier who has a garden. In his garden he has N stones which, like Tobby, are either black or white. The stones are organized in a row and numbered from 0 to $N - 1$.

Tobby likes to change his stones quite a lot and he does so by taking a contiguous part of his stones, from integer indexes I to J, and making some changes to all stones located between those two indexes (including the stones at I and J).

Given a list of changes that Tobby makes to his garden, your task is to answer queries about the number of black and white stones in his garden.

Initially all stones in Tobby's garden are white.

Input

The input contains several test cases. Each test case starts with a line in which there are two positive integers N ($1 \leq N \leq 10^6$) and M ($1 \leq M \leq 10^6$) that represent, respectively, the number of stones in Tobby's garden and the number of changes and queries to answer. Then M lines follow, each containing either a change or a query. Each of those lines starts with an integer Q ($0 \leq Q \leq 3$):

If Q is 0, then the line is a query, in that case two numbers follow: I ($0 \leq I < N$) and J ($I \leq J < N$), indicating the start and end positions to consider to answer the query.

If Q is 1, then the line is a change of type reverse, in that case two integers follow: I ($0 \leq I < N$) and J ($I \leq J < N$), indicating that Tobby wants to reverse the color of all stones between those indexes so that the color of stone $I + K$ becomes the color of stone $J - K$ and the color of stone $J - K$ becomes the color of stones $I + K$ for all integer K such that $0 \leq K$ and $I + K < J - K$.

If Q is 2, then the line is a change of type flip color, in that case two integers follow: I ($0 \leq I < N$) and J ($I \leq J < N$), indicating that Tobby wants to change the color of all stones between those indexes so that all white stones become black stones and all black stones become white stones.

If Q is 3, then the line is a change of type set color, in that case three integers follow: I ($0 \leq I < N$), J ($I \leq J < N$) and C ($0 \leq C \leq 1$) indicating that Tobby wants to set the color of all stones between indexes I and J. If C is 0, then all stones between those two indexes should become black, while if it is 1 they should become white.

Output

For each query in the input, you should print a single line with two integers: the number of black stones and the number of white stones between the indexes I and J given in the query.

Example

Input	Output

10 7
0 0 9
3 0 4 0
0 0 4
1 0 9
0 5 9
2 5 9
0 3 9
100 1
0 0 50
	0 10
	5 0
	5 0
	0 7
	0 51

Problem K. Tobby and Seven

Source file name: tobbyseven.c, tobbyseven.cpp, tobbyseven.java
Input: Standard
Output: Standard
Author(s): Jhon Jiménez - UTP Colombia

Tobby has a very special number, when this number is divided by 7 the remainder is 0. Every time that Tobby thinks about this property, he says: WOW, this is a wonderful number!

As everybody knows Tobby is not a common pet, he is smart and curious, so he decides to modify the original number as follows: Tobby writes the number in its binary representation, then he chooses two indexes and make a swap operation of the values of those positions, and then he does several other similar operations for a while. Tobby does not remember the total number of operations that he performed, he just remembers the positions in the binary representation that were swapped at some point and he also remembers that the most significant bit was never swapped.

Now Tobby wants to recover the original number, but Tobby is a busy pet and so he does not have time for that. Can you help him?

Input

The input has several test cases. Each test case has three lines, the first line contains a single integer n ($7 \leq n \leq 2^{60}$), the modified number. The second line contains an integer k ($2 \leq k \leq 20$) representing the amount of positions that were swapped and the third line contains k different integers in increasing order, the positions (relative to the least significant bit) of the number that were swapped (zero-indexed).

Output

Print a single integer: the recovered number which is divisible by 7. If there are several answers, then print the maximum possible value.

Example

Input	Output

79
5
0 1 2 4 5
21
2
0 3
65791
16
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1152921504606846975
5
13 39 40 58 59
	91
	28
	130816
	1152921504606846975

Explication

In the first sample the original number (without swaps) is 91

Tobby remembers that in sometime the positions 0, 1, 2, 4 and 5 were swapped.

Suppose that Tobby did the follow swap operations:

6 5 4 3 2 1 0 = indexes

1 0 1 1 0 1 1 = 91

$\mathsf{s}wap(0,\ 2,\ 1011\overline{0}1\overline{1}) \rightarrow 1\ 0\ 1\ 1\ 1\ 1\ 0 = 94$

$\mathsf{s}wap(4,\ 5,\ 1\overline{0}\ \overline{1}1110) \rightarrow 1\ 1\ 0\ 1\ 1\ 1\ 0 = 110$

$\mathsf{s}wap(0,\ 1,\ 11011\overline{1}\ \overline{0}) \rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 1 = 109$

$\mathsf{s}wap(2,\ 4,\ 11\overline{0}1\overline{1}01) \rightarrow 1\ 1\ 1\ 1\ 0\ 0\ 1 = 121$

$\mathsf{s}wap(0,\ 1,\ 11110\overline{0}\ \overline{1}) \rightarrow 1\ 1\ 1\ 1\ 0\ 1\ 0 = 122$

$\mathsf{s}wap(2,\ 4,\ 11\overline{1}1\overline{0}10) \rightarrow 1\ 1\ 0\ 1\ 1\ 1\ 0 = 110$

$\mathsf{s}wap(0,\ 5,\ 1\overline{1}0111\overline{0}) \rightarrow 1\ 0\ 0\ 1\ 1\ 1\ 1 = 79$

Note that Tobby can swap the same index many times, note also that the position 6 was never swapped.

Problem L. Tobby and Prime Sum

Source file name:	primesum.c, primesum.cpp, primesum.java
Input:	Standard
Output:	Standard
Author(s):	Carlos Arias Londoño - UTP Colombia

Tobby has learned to calculate the sum of the digits of a given number X, but this is very easy for him. Lately he has been studying about prime numbers and he has found a more challenging question:

How many integer numbers in the range from L to R (inclusive) exist such that the sum of their digits is a prime number?.

Tobby is an smart puppy but he can only count to 100, can you help him to solve this problem?

Input

The input consists of several test cases and must be read until EOF. The first line of each test case contains two integers L and R ($1 \leq L \leq R \leq 10^{500}$).

Output

For each test case the output consists of one number X indicating how many numbers in the range L, R meet the property previously mentioned. Because this amount can be very large you also should print the answer modulo $10^9 + 7$.

Example

Input	Output
1 10 20 46	4 11

GRAN PREMIO 2016 SEGUNDA FECHA

Problem A. Advanced multiplications

Source file name:	advanced.c, advanced.cpp, advanced.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Gilberto Vargas Hernández`

Pepito is a really hyperactive kid, and to make it worse you have to take care of him today. You've tried to take him to the arcade machines, the movies and even an amusement park but he can't just stay in peace. You're tired because all the places you've gone so far and you finally came up with a wonderful idea. You're pretty good at math and designed a challenge for him. You remembered that he has been learning multiplication tables in school and asked Pepito:

If you write down all the multiplication tables beginning with 1 and finishing with 10, can you calculate the sum of all the multiplications?

Pepito took a notebook and while he was finishing to write number 1 table he realized about something and told the result: It's 3025. Believe it or not he came up with an answer before you could calculate the result and you want to know he made it.

Input

The input will consist of multiple cases and will finish when you receive a 0, you should not process it. For each test case a line containing N, the limit for the multiplication table.

- $1 \leq N \leq 10^9$

Output

For each test case output one line containing the last ten digits of the result.

Example

Input	Output
1 2 10 0	1 9 3025

Problem B. Board Game

Source file name:	board.c, board.cpp, board.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín Rosas`

Toby has a new board game, this board contains a total of N chambers numbered from 1 to N arranged in a circle on the board, this is, chamber i is connected to chamber $i + 1$ for all $1 \leq i \leq N$ and in order to complete the circle chambers N and 1 are connected.

The board game contains also a sphere that once it's dropped to one of the chambers it will start moving through the chambers following the described connections. For example If you drop the ball on chamber 3 it will move to chamber 4 and then to chamber 5 and so on until it gets to the chamber N to continue it's path through chamber 1. The ball will stop only if it finds a blocked chamber.

Since Toby is not working properly because he is playing a lot with his new game you have taken M blocks that you will use to block the chambers on Toby's board. You don't want Toby to witness you are blocking the roads so you proceed to throw the blocks one by one while he is distracted until each block is blocking one chamber into the board, you are so good throwing blocks that there is no chamber containing more than one block.

Suddenly you realized that maybe the ball will not stop as quick as you want, so you start to see the ball moving on the board and start remembering your math classes. Can you count how many different ways you could have put the blocks on the chambers such that the ball will stop between chambers t_1 and t_2 considering the ball started moving from chamber 1?

Input

The input consist of several test cases. Each test case consists of a line containing the numbers N, M, t_1 and t_2. The end of the test cases is given by the end of file (EOF).

- $1 \leq N \leq 10^6$
- $1 \leq M \leq N$
- $1 \leq t_1 < t_2 \leq N$

Output

For each test case print in one line the requested answer modulo $p = 10^9 + 7$.

Example

Input	Output
5 1 1 2	2
5 2 1 2	7
10 4 6 9	5

Problem C. Counting Strings

Source file name:	counting.c, counting.cpp, counting.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín Rosas`

A Binary string is a string formed only of 0 and 1. For example 00011101 is a binary string while 000201023 is not.

A change on a binary string is defined as follows :

- Let a be the $i-th$ value in the string
- Let b be the $j-th$ value in the string where $j = i + 1$
- If a and b are not the same value, then, there is a change in the string between i and j

In the valid binary string showed bellow (00011101) you can find 3 changes:

- 1^{st} change is between 2 and 3
- 2^{nd} change is between 5 and 6
- 3^{rd} change is between 6 and 7
- Indexes start in 0

To not take much of your time reading this problem we will describe it the fastest we can. Given two numbers N and K, you have to create a computer program that counts how many binary strings with length N exist that have exactly K changes.

Input

The input consists of several test cases. Each test case contains a single line with two numbers N and K separated by a white space. The end of the test cases is given by the end of file (EOF).

- $1 \leq N \leq 10^5$
- $1 \leq K < N$

Output

For each test case print in one line the requested answer modulo $p = 10^9 + 7$.

Example

Input	Output

1 0
3 0
10 3
	2
	2
	168

Problem D. Dividing Hexadecimal Numbers

Source file name:	dividing.c, dividing.cpp, dividing.java
Input:	Standard
Output:	Standard
Author(s):	Gilberto Vargas Hernández

Mr. Homft promised not to let homework to the kids that could accomplish a task in class. Today's lecture was about hex numbers. Hex numbers were invented by the lazy computer scientists from the last century who didn't want to write a lot of zeros and ones, so they synthesized binary into hex numbers. For a binary number it's possible to form groups of 4 bits and replace them by the hex digit. An easy way to convert from hex to binary! A hex digit is represented as a number from 0 to 9 or an uppercase letter from A to F which represents numbers from 10 to 15.

Mr. Homft's class is integrated only by smart kids, so don't get scared by their abilities. The last month they were studying Newton's laws, number theory and some other kind of sorcery tricks.

Well, now that everything has been explained let's go to Mr. Homft's problem. He wrote a hex number N, actually a really huge one, and then he asked if the number was a multiple of 17. Passed no more than 5 minutes, all the kids had answered to the question correctly. You may be wondering if you must have taken the sorcery course last summer to calculate the result but maybe what you need is a different course. Can you answer correctly as the kids? Given a set of hex numbers your task is to say whether they are or not multiples of 17.

Input

You'll have to read until end of file. Each line of input will have the hex number N.

- N will have at most 10^5 hex digits.

Output

For each line of input, write "yes" if the hex number is divisible by 17, write "no" otherwise

Example

Input	Output
9999 11 AA 0 1	yes yes yes yes no

Problem E. Expected Characters

Source file name:	strings.c, strings.cpp, strings.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín Rosas`

A string S is traditionally a variable where a sequence of characters is stored. There are several operations you can do to strings, one of those operations that is of importance to us is rotation: a string $T = uv$ where u and v are also strings is said to be a rotation of S if $S = vu$. For example, if $S = abc$ the string bca is a rotation of abc, where $u = bc$ and $v = a$.

As you can see if a string S has N characters in the sequence then there will be N rotations. For example, if $S = abc$ the 3 rotations are: abc, bca, cab. Toby is a very curious person and after realizing this fact, he found that an $N \times N$ matrix containing all the rotations of S can be created. The Matrix Toby creates is as follows: For the $k-th$ row in the matrix take the substring of S that starts on 0 and ends on $k-1$ as v, and the substring that starts on k and ends on $N-1$ as u then the string $T = uv$ is a rotation of S. An example of this matrix using $S = abc$ is:

```
abc
bca
cab
```

Toby has challenged you to find a quick way to determine given the string S and a Q number of queries where each query asks for the expected character in position i, j of the matrix built of the rotations of S.

Input

The input consists of several test cases. The first line of each test case contains the strings S and the number of queries Q separated by a space. Each of the next Q lines contains the numbers i, and j of the query that needs to be answered. You may assume that S will have no more than 10^6 characters and that $1 \leq Q \leq 10^6$. S contains only lowercase characters and no white spaces. The end of the test cases is given by the end of file (EOF).

Output

For each test case your program should print exactly Q lines, each line containing a single character which is the answer to the query.

Example

Input	Output
abc 3 1 1 2 2 3 3 bca 3 3 1 1 1 2 1 acmicpc 5 2 3 4 5 3 3 1 5 4 2	a c b a b c i a c c c

Problem F. Factorial Divisors

Source file name:	factorial.c, factorial.cpp, factorial.java
Input:	Standard
Output:	Standard
Author(s):	Gilberto Vargas Hernández

The factorial of N is defined as the product of all the numbers between 1 an N and is denoted by the ! symbol. For example, $10! = 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 3628800$. There is a list of numbers that can divide 10!, those are called divisors. For example, the divisors of 4! are $1, 2, 3, 4, 6, 8, 12$ and 24 itself.

There are many ways to count the number of divisors of a number, but since we are talking about factorials you must think one excellent way to do so.

Your task consists in counting how many numbers are that divide $N!$ but not divide $K!$ $(1 \leq K \leq N \leq 10^6)$.

Input

The input consists of several test cases, each one with two numbers N and K separated by one space. The input ends at the end of file (EOF).

- $1 \leq K \leq N \leq 10^6$

Output

For each case you must print a single line with a single number, the result modulo 987654321.

Example

Input	Output
4 2	6

Explication

$4! = 4 * 3 * 2 * 1 = 24$, $2! = 2 * 1 = 2$ Divisors of $4! = 24$ are : $\{1, 2, 3, 4, 6, 8, 12, 24\}$ as stated above. Divisors of $2! = 2$ are : $\{1, 2\}$, then the only 6 divisors of 4! that don't divide 2! are :$\{3, 4, 6, 8, 12, 24\}$

Problem G. Getting Coffee

Source file name:	coffee.c, coffee.cpp, coffee.java
Input:	Standard
Output:	Standard
Author(s):	Juan Pablo Marín Rosas

Toby does not trust anyone. He always enjoys to walk to the nearest coffee store to get his morning coffee, seems like the only one he trusts is the barista in the coffee store since he always trust he is serving his morning latte and not a capuccino. Well, since Toby does not trust in people his morning walk to the coffee store is quite interesting:

- Toby has a list of N places that he likes in the city, the place 1 is his home and place N is the coffee store.
- From these N places he may walk only between places that are connected by one of the M trails he likes.
- To ensure he is not followed or observed by anyone Toby will take a path from home to the coffee store and a different one from the coffee store to home. He may pass more than once on some places on his walk but will never take the same trail.
- Since Toby always leaves his morning yoga video buffering while he goes for his coffee he wants to do his walk as fastest as possible, this is, taking the minimum amount of time since he leaves home until his return.

As Toby is knowing more and more places it is becoming harder for him to plan his morning walks. So he asked you for help to give him a piece of advice on what is the path that he must follow.

Given the preferences Toby have for his walk you have to compute what is the path he can take from home to coffee and coffee to home that takes the less amount of time.

Input

Input contains several test cases. The first line contains two integers separated by a space N and M. Next M lines contains three values separated by space, a, b and t representing the i-th trail connects places a and b and it takes t time for Toby to walk that trail.

- $1 \leq N \leq 100$
- $1 \leq M \leq \frac{(N)(N-1)}{2}$
- $1 \leq t_i \leq 1000$

Note: You may assume there is always a way for Toby to complete his walk.

Output

Your program should output one line per test case containing two integer values. The first one is the less amount of trails Toby have to take on his walk, the second one is the less amount of time that it can take for Toby to go grab coffee and return home.

Example

Input	Output

3 3
1 3 5
2 1 10
3 2 15
	30

Problem H. Hidden string

Source file name:	hidden.c, hidden.cpp, hidden.java
Input:	Standard
Output:	Standard
Author(s):	Juan Pablo Marín Rosas

After several years of research finally someone found the secret of the mystical mexican pyramids. It turns out there is a hidden message in the two strings S and T that are written in the door of the main entrance to the tomb of the red queen.

After this was disclosed to the public it is too obvious how to find this hidden string. The secret message is string of size K that is contained in both string S and T. Now everyone is trying to sell messages saying that is the original message found on the mystical mexican pyramids. As you are a clever person you decided to write a program to verify if it is really possible that what is being sold by people next to the pyramids is the hidden string.

Given S and T, you have to determine if there exist a string of size K that is contained on both strings.

Input

The input consits of several test cases, each test case contains S, T and K separated by a space.

- The length of S and T is the same and is at most 10^5
- $K \leq |S|$ Where $|S|$ denotes the size of the string S

Output

For each test case your program should print a single line with the word "yes" in case there exist at least one string of length K that is contained in both S and T, print "no" otherwise.

Example

Input	Output
abcdef cdefba 2 abcdef bcdabc 5	yes no

Explication

In the first test case the strings $abcdef$ and $cdef$ we are looking for a string of size 2 that is contained in both strings, there are some of them, one is cd that's why the output is yes. In the second test case there are no strings of length 5 that is contained in both strings, that's why the output is no.

Problem I. Inserting a polyominoe

Source file name:	inserting.c, inserting.cpp, inserting.java
Input:	Standard
Output:	Standard
Author(s):	Félix Arreola

Toby is very bored and wants to start a new game with his friends. He loves hexominoes, pentominoes, and tetrominoes, and wants to use them in his new game. His rules are simple, every player has 4 tetrominoes, 5 pentominoes and 6 hexominoes; next, the player choice one polyomino from his stock and tries to put it in the board. The polyomino can be flipped or rotated in any way. The board is formed by a $F \times G$ rectangle. The polyominoes can't overlap with others in the board.

Toby wants to win at any cost, so he contracted the Amazing College Maker (ACM) to write a program that checks, for a given polyomino, if it can fit in the board at any position.

You are an amazing student from this college, and your professor is challenging every student to solve this problem, Can you solve it?

Input

The first line of input contains 2 integers W, H separated by a space will be the size of the polyomino. The next H lines with W integers each one separated by a white space. A 1 represents a block that is part of the piece, a zero means empty space.

Next, there are F, G which represents the size of the board. After that, there are F lines, each one with G integers, every integer separated by a white space. A 1 represents an occupied square in the board. A zero means an empty space

- $1 \leq H,\ W \leq 6$
- $1 \leq F,\ G \leq 20$

Note: All pieces are continuous.

Output

Your program should print a single line with the word "YES" if the polynomino can be put in the board, print "IMPOSSIBLE" otherwise.

Example

Input	Output

4 4
1 1 1 0
0 1 0 0
0 0 0 0
0 0 0 0
7 7
0 0 0 1 1 1 1
1 1 0 1 1 1 0
1 0 0 1 0 1 0
0 1 1 0 1 1 0
0 0 0 0 0 1 1
1 1 0 0 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
	YES

Problem J. Joining points

Source file name:	joining.c, joining.cpp, joining.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín Rosas`

The annual International Competition for Points Connectors (ICPC) world final is around the corner. a Points Connectors competition is a competition where each contestant is given a board where N points numbered from 1 to N are drawn, to be fair all boards have the same N points drawn. Each contestant will be joining pairs of points with straight lines until all points are connected in such way that you can get from any point to another following only the lines drawn by the contestants, as you can imagine the first contestant that joins all points wins. To make the world finals a more difficult challenge there are two restrictions:

- There is a list of pairs of points that are the only "allowed" moves. This is the final connection that a contestant creates should contain only those that are on this list.
- Each of the allowed moves contain a number of points w_i that will be earned if the pair of points on that allowed move is selected. The grading of a contestant will be the sum of the grades earned by selecting the allowed moves on the list.

Based on these two new restrictions the winner of the world finals will be the contestant that joins the points using only points from the allowed moves list and that his grading is the minimum from all the other contestants.

Last year Toby and Buzz were 1st and 2nd place on the world finals and they will be participating this year again. Toby and Buzz have been training hard connecting the points on all competitions based on the restrictions they will have in the world final, this way they will have less things to worry on the world finals contest.

You are a devoted follower of the Points Connectors contest and after looking to the grades of Toby and Buzz on all the contests that have happen the last 2 years you have found a very interesting pattern: Toby always wins and Buzz is always 2nd place, also, Toby always find the way to connect the points getting the minimum grade possible in the board while Buzz always gets the 2nd minimum grade possible.

Now that everyone is doing bets on who will win the World Finals you will put a difficult bet this time. After the board is disclosed you will bet "Toby will be 1st place getting a grade X, and Buzz will be 2nd getting a grade Y". To do this you have prepared a computer program that given the Board and restrictions will compute what is the grade X Toby will get and the grade Y Buzz will get at the World Finals.

Input

The input consists of several test cases. The first line of each test case contains two numbers N and M, N is the number of points in the board and M is the number of restrictions in the allowed moves list. The next M lines contain three numbers u, v, w separated by a space which represents the points u and v can be connected as an allowed move earning w points. You may assume there is always a way to connect all the points on the board with the given restrictions.

- $1 \leq N \leq 5000$
- $1 \leq M \leq \frac{(N)(N-1)}{2}$

Output

For each test case your program should print two numbers the grade X that Toby will get and the grade Y that Buzz will get on the world finals.

Example

Input	Output

3 3
2 1 67
3 1 46
3 2 75
5 9
2 1 29
3 2 52
4 1 20
5 3 45
2 5 42
2 4 19
1 5 5
5 4 26
4 3 76
	113 121
	89 95

Problem K. King of the bar

Source file name:	king.c, king.cpp, king.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Gilberto Vargas Hernández`

This weekend "el barecito" will finally open. The administrator is planning to place a roulette at midnight and make a game with all the present people. Customer will have a cup numbered from 1 to N. Not two cups will have the same number and all the numbers from 1 to N will appear. Of course the owner of "el barecito" only will give N cups to the first N customers. Next he will spin the roulette, which is numbered from 1 to N too, and a number will be chosen, name it K. The customer with the cup number K will go to the roulette and if he has an empty glass it will be filled, if it was full he will have to drink it up, and the roulette will be spun again. The game will finish when everybody has a filled cup.

The problem is that this game can be delayed forever and this would be mean bankrupt in the opening day. The owner of "el barecito" has contacted you as the king of the bar to help him not to get on bankruptcy he wants you to estimate in average, how many times the roulette will be spun until the end given the number of cups

Input

This problem consists of multiple cases, at most 10. Each test consists of a single line with a single number N the number of cups in the game.

- $1 \leq N \leq 10^5$

Output

For each case print just the average number of times the roulette will be spun until the game ends using modular arithmetic instead of real arithmetic, as the number grow very quickly you have to give the answer modulo $10^9 + 7$.

Example

Input	Output
2 3 4 5	4 10 333333357 666666714

Problem L. Lines for the subway

Source file name:	lines.c, lines.cpp, lines.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Gilberto Vargas Hernández`

This year Guadalajara's government has decided to build another line for the subway. Building a subway line means a lot of work and a headache for the citizens because the traffic. As you will imagine, joining the most important points of the city is the goal, but, unfortunately, they don't have enough money to build all the lines they need. They need your help as an ace coder to solve their problem.

The subway design consists in a set of important points on the city, those are meant to be the stations. There are some tunnels joining two stations. There are no two different paths between two stations and all are connected. Each tunnel has it's own length and hardness. The hardness refers to how difficult it is to be dug and the sharpness needed to break the hardest stones in the tunnel.

The government only can buy a single drill. To calculate the price of building a whole line you have to calculate first the sum of the lengths of all the tunnels between the tow stations and which one is the hardest, so the price is given as the product of both quantities.

Given a scheme of the subway and a set of proposals from the people, can you determine the cost of each one?

Input

There will be only one test per case. In the first line a number N, the number of stations in the plane. The next $N-1$ lines 4 numbers, a, b, h and l which means that the station a is connected with the station b and has a length of l and h the hardness of the terrain. The next line contains a number p, the total of proposals from the people and finally, p lines with 2 numbers, r and s, a proposal of a line starting in the station r and ending in the station s.

- $1 \leq p \leq 10^5$
- $2 \leq a, b, r, s \leq N \leq 10^5$
- $1 \leq l, h \leq 10^6$

Output

p lines with one number on each, the cost of building the i proposal.

Example

Input	Output

8
1 2 3 4
1 3 2 8
3 6 9 7
6 4 12 14
8 6 13 1
4 5 10 15
4 7 7 6
5
1 8
2 7
3 2
5 8
7 3
	208
	468
	36
	390
	324

GRAN PREMIO 2016 TERCERA FECHA

acm International Collegiate Programming Contest 2016 IBM event sponsor

Gran Premio de México & Centroamérica 2016

Tercera fecha

10 de Septiembre de 2016

Libro de problemas

Información general

Este libro contiene un total de 12 problemas; Las páginas están numeradas del 1 al 19, sin contar esta página. Por favor verifica que tu libro esté completo.

A) Sobre el programa
1) Los códigos enviados deberán ser nombrados de la siguiente forma : *código_de_problema*`.c`, *código_de_problema*`.cpp`, *código_de_problema*`.pas`, *código_de_problema*`.java` or *código_de_problema*`.py`, donde *código_de_problema* es la letra mayúscula que identifica al problema. Recuerda que en Java el nombre de la clase principal debe ser el mismo que el nombre del archivo.

B) Sobre la entrada
1) Tu programa deberá leer la entrada desde *entrada standard.*
2) La entrada es un solo caso de prueba, con un número de líneas que dependen del problema.
3) Cuando una línea de entrada contiene mas de un valor, estos valores estarán separados por un espacio en blanco. La entrada no contiene otros espacios en blanco.
4) Cada línea de la entrada, incluyendo la última, contiene exactamente un caracter de fin de línea.
5) El fin de la entrada concuerda con el fin de archivo.

C) Sobre la salida
1) La salida de tu programa deberá ser impresa a la *salida standard.*
2) Cuando una línea de la salida contenga mas de un valor, estos valores deberán ser separados por un espacio en blanco. La salida no debe contener mas espacios en blanco. 3) Cada línea de salida, incluyendo la última, deberá contener exactamente un caracter de fin de línea.

Problema A

Viaje en el tiempo

Supón que tienes una máquina del tiempo que puede usarse a lo más tres veces. Cada vez, puedes elegir entre ir al pasado o ir al futuro. La máquina tiene tres créditos fijados, que son una cantidad de años. Puedes viajar, una, dos o tres veces, y cada crédito puede ser usado solo una vez. Por ejemplo, si los créditos fueran $5, 12$ y 9, si decidieras viajar 2 veces : ir 5 años al futuro y luego regresar 9 años al pasado, de esta manera podrías regresar 4 años al pasado y terminar en 2012. También podrías viajar al futuro 3 veces usando los créditos en orden terminando en el 2042.

En este problema, dados los valores de los 3 créditos, tu programa debe decidir si es posible o no viajar en el tiempo y regresar al presente, usando al menos un crédito y a lo máximo tres creditos. Siempre usando cada uno de los tres créditos solo una vez.

Entrada

La unica línea de entrada contiene tres números enteros A, B y C ($1 \leq A, B, C \leq 1000$), representando los valores de los créditos.

Salida

La salida contiene una única línea con la letra "`S`" si es posible realizar el viaje y volver al presente, o "`N`" en caso de que no sea posible.

Ejemplos

Entrada	**Salida**
`22 5 22`	`S`

Entrada	**Salida**
`31 110 79`	`S`

Entrada	**Salida**
`45 8 7`	`N`

Problema B

Papa caliente

La papa caliente es un juego popular entre los niños de las escuelas. El juego es simple: en cada turno, el niño que tiene la papa la pasa a otro niño. En algún punto, el maestro, el cual no está viendo al juego dirá que el juego terminó. Cuando esto sucede, el niño que tiene la papa pierde el juego.

Una variación de este juego, jugado en la cafetería es propuesta por el maestro. Los niños son numerados del 1 al N acorde a la posición en la que se encuentran en la fila, donde el niño con el número 1 es el primero en la fila. Cada niño recibe un papel con un número y durante el juego cuando el niño recibe la papa él la tendrá que pasar al niño en la posición de el número que tiene anotado su papel. Si la papa llega a una posición menor o igual a X en la fila, donde X es elegido al inicio del juego, el juego termina y el maestro ganará el juego. Si esto nunca sucede, el juego nunca terminará y los niños ganarán el juego: Al siguiente dia todos ellos tendrán un descuento en la cafetería.

El maestro inicia el juego pasando la papa a algúno de los niños en la fila. Como su vista no es muy buena, el puede garantizar que pasará la papa a un niño en el rango $L \dots R$ en el juego con una misma probabilidad. El maestro está considerando diferentes rangos para el inicio del juego. Para esto, él quiere saber, para cada uno de estos rangos, cual valor de X debe elegir de modo que el juego sea lo más justo posible, esto es, que la probabilidad de que el juego termine es lo más cercana posible a la probabilidad de que el juego no termine.

Tu tarea es ayudar al maestro a evaluar las propuestas. Dados los papeles que tiene cada niño en la fila y diferentes rangos posibles, debes responder, para cada rango, el valor X que hace el juego lo más justo posible. Si hay más de un valor de X que satisfaga esta condición, responde la X mas cercana al frente de la fila.

Entrada

La entrada contendrá varios casos de prueba. En cada caso, La primer línea de entrada contiene dos números enteros, N y Q ($2 \leq N \leq 50000$, $1 \leq Q \leq 10^5$). La siguiente línea contiene N enteros separados por un espacio $p_1, p_2 \dots p_N$ ($1 \leq p_i \leq N$), el número en el papel que recibe cada niño. Las siguientes Q líneas, contienen dos enteros cada una L y R ($1 \leq L \leq R \leq N$), representando cada una de estas un rango en las propuestas del maestro.

Salida

Para cada caso de prueba en la entrada, Debes imprimir exactamente Q líneas, cada una respondiendo para el rango considerado por el maestro, el número X que debe ser elegido para que el juego sea lo más justo posible.

Entrada	**Salida**

`9 4`
`2 3 4 5 6 7 4 9 5`
`1 3`
`3 5`
`2 8`
`7 9`
	`1`
	`3`
	`3`
	`1`

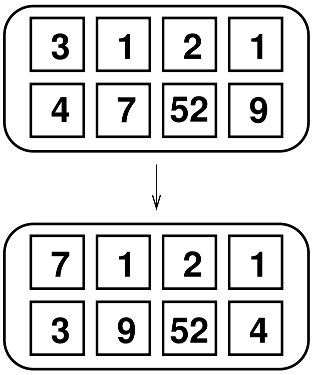
Entrada

```
3 3
1 3 3
1 1
1 2
2 3
```

Salida

```
1
1
2
```

Problema C
Contenedores



El CBS–Contenedores Balanceados por Sistema necesita ser actualizado de modo que trabaje una nueva clase de barcos, el "dos por cuatro", el cual son barcos que pueden cargar hasta 8 contenedores grandes dispuestos en dos líneas y cuatro columnas, exactamente como es mostrado en la imagen. Estos barcos tienen una grua fija que les permite realizar solo un tipo de movimiento: Tomar hasta dos contenedores adyacentes en una fila o columna y cambiar su posición. Para acelerar las cosas, durante el proceso de carga en el puerto los ocho contenedores son acomodados en cualquiera de las 8 posiciones, definiendo así una configuración inicial. Cuando el barco sale del puerto, la grua necesita mover los contenedores de modo que sean acomodados en una configuración final predefinida.

El problema es que el costo de la grua para realizar un movimiento es igual a la suma de los pesos de los dos contenedores adyacentes que están siendo cambiados de posición. Dados los pesos de los contenedores en ambas configuraciones la inicial y la final, el CBS debe computar cual es el costo total mínimo posible para llevar los contenedores de la posición inicial a la final a travez de una secuencia de movimientos.

Entrada

La entrada consiste de 4 líneas conteniendo, cada una, cuatro enteros entre 1 y 1000 inclusivos. Las primeras dos lines definen los pesos de la configuración inicial y las últimas dos líneas definen los pesos de la configuración final. Siempre hay una solución, dado que los contenedores en la configuración inicial y final son los mismos, posiblemente en diferentes posiciones.

Salida

Imprime una línea conteniendo un entero, representando el costo total mínimo posible para una secuencia de movimientos que llega de la configuración inicial a la final.

Ejemplos

Entrada	**Salida**

`3 1 2 1`
`4 7 52 9`
`7 1 2 1`
`3 9 52 4`
	`81`

Entrada	**Salida**

`1 2 3 4`
`5 10 7 8`
`1 2 3 4`
`5 8 7 10`
	`50`

Entrada	**Salida**

34 5 6 998
4 17 77 84
34 5 6 998
4 17 77 84
	0

Problema D

Divisores

Piensa en un número entero positivo n. Ahora dime un divisor A de n. Ahora dame otro número B que no es un divisor de n. Ahora un múltiplo C y un no múltiplo D. El número que pensaste es....

Parece magia, pero, ¡son matemáticas!! Será que, dados los valores de A, B, C y D, ¿puedes encontrar cuál es el número n? Note que puede existir más de una solución.

En este problema, dados los valores de A, B, C y D debes escribir un programa que determine cuál es el número n más pequeño que se pudo haber pensado ó concluir que nno existe un valor posible para n.

Entrada

La entrada contendrá varios casos de prueba. En cada caso, 4 números enteros A, B, C, D, como mencionado anteriormente ($1 \leq A, B, C, D \leq 10^9$).

Salida

Para cada caso de prueba en la entrada, En caso de que exista al menos un número n para el cuál los valores de A, B, C y D hagan sentido, entonces tu programa deberá imprimir el número más pequeño posible de n, de otra forma imprime -1.

Entrada	**Salida**
`2 12 8 2`	`4`

Entrada	**Salida**
`3 4 60 105`	`6`

Problema E

Estadística Hexa

Dada una secuencia de enteros positivos en su representación hexadecimal, por ejemplo, $S = [\texttt{9af47c0b}, \texttt{2545557}, \texttt{ff6447979}]$, definimos $\texttt{sum}(S)$ como la suma de todos los elementos en S. Ahora, considere una permutación de los 16 digitos hexadecimales, por ejemplo:
$p = [\texttt{4}, \texttt{9}, \texttt{5}, \texttt{a}, \texttt{0}, \texttt{c}, \texttt{f}, \texttt{3}, \texttt{d}, \texttt{7}, \texttt{8}, \texttt{b}, \texttt{1}, \texttt{2}, \texttt{6}, \texttt{e}]$, Iniciando con la secuencia base S, podemos definir una secuencia transformada $S^{[4]}$, obtenida al eliminar todas las ocurrencias de el digito hexadecimal `4` de todos los números en S, $S^{[4]} = [\texttt{9af7c0b}, \texttt{255557}, \texttt{ff67979}]$. Después podríamos eliminar el digito `9` y obtener $S^{[4,9]} = [\texttt{af7c0b}, \texttt{255557}, \texttt{ff677}]$. Siguiendo el orden de los digitos en la permutacion p, podemos definir de esta manera 16 sequencias: $S^{[4]}, S^{[4,9]}, S^{[4,9,5]}, \ldots, S^{[4,9,5,\texttt{a},0,\texttt{c},\texttt{f},3,\texttt{d},7,8,\texttt{b},1,2,6,\texttt{e}]}$. Lo que nos interesa sabes es la suma de los elementos de estas 16 secuencias:

$$\texttt{total}(S,p) = \texttt{sum}(S^{[4]}) + \texttt{sum}(S^{[4,9]}) + \texttt{sum}(S^{[4,9,5]}) + \cdots + \texttt{sum}(S^{[4,9,5,\texttt{a},0,\texttt{c},\texttt{f},3,\texttt{d},7,8,\texttt{b},1,2,6,\texttt{e}]})$$

Claramente, este valor depende de la permutación p usada en el proceso de eliminación. Dada la secuencia de los N enteros positivos en su representación hexadecimal, debes computar, considerando todas las posibles permutaciones de los 16 digitos hexadecimales: el mínimo total, el máximo total y la suma de los totales de todas las permutaciones. Para la suma de los totales, imprime el resultado modulo `3b9aca07` ($10^9 + 7$ on base 10).

Entrada

La primer línea de entrada contiene un entero N, $\texttt{1} \leq N \leq \texttt{3f}$, representando el tamaño de la secuencia. Las siguientes N líneas contienen, cada una, un entero positivo P, $\texttt{0} \leq P \leq \texttt{ffffffff}$, definidiendo la secuencia inicial S de enteros. Todos los números en la entrada son representados en hexadecimal, con letras minúsculas.

Salida

Imprime una línea conteniendo tres enteros positivos, en hexadecimal, con letras minúsculas, representando el minimo total, el maximo todal y la suma de los totales considerando todas las posibles permutaciones de los 16 digitos hexadecimales.

Ejemplos

Entrada	**Salida**

`3`
`9af47c0b`
`2545557`
`ff6447979`
	`1312c99c b4e87e9387 5bb5fc`

Entrada	**Salida**

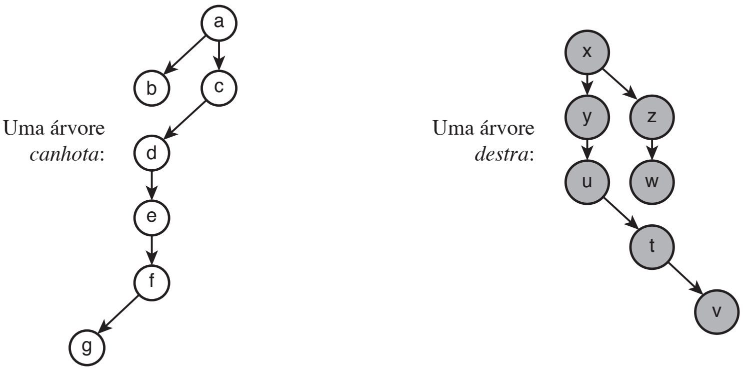
`1`
`ffffffff`
	`0 efffffff1 15dac189`

Problema F

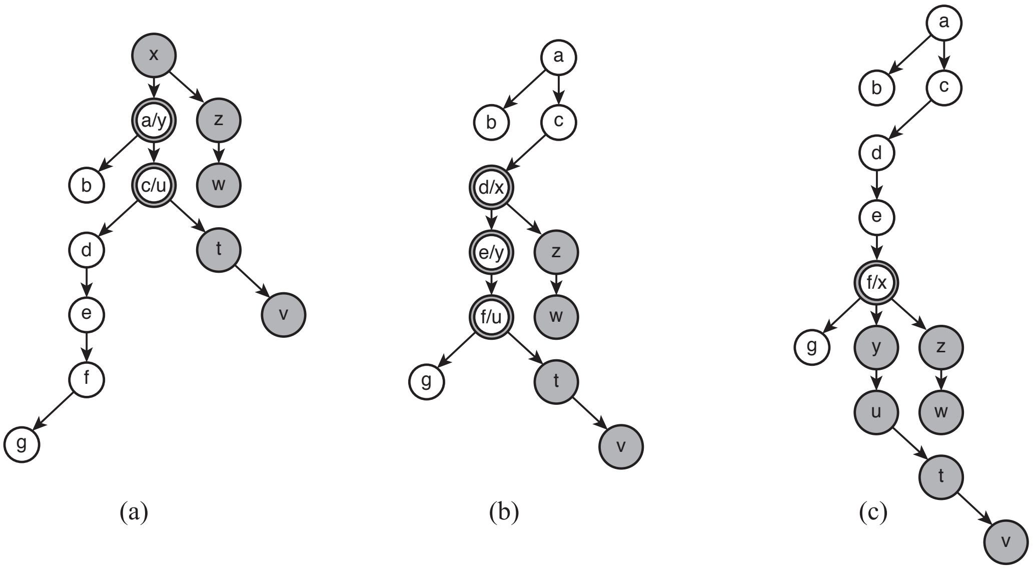
Fusionando árboles

En computación, los árboles son objetos extraños: ¡La raiz está arriba y las hojas abajo!. Un árbol es una estructura compuesta de N vertices conectados por $N-1$ aristas de modo que es posible llegar de un vértice a cualquier otro vértice siguiendo las aristas. En un árbol *enraizado*, cada arista conecta a un vértice *padre* y un vértice *hijo*. Un único vértice que no tiene padre es llamado *raiz*. Entonces, es posible llegar desde la raiz a cualquier otro vértice del árbol siguiendo las aristas en direccion de padre a hijo.

En un árbol *ternario* cada vértice puede tener hasta tres vértices hijo, llamadaos *izquierdo*, *central*, y *derecho*. Un árbol ternario *zurdo* es un árbol ternario enraizado en el cual ningún vértice tiene un hijo derecho. Un árbol ternario *diestro* es un árbol ternario enraizado en el cual ningún vértice tiene un hijo izquierdo. La raíz de un árbol ternario siempre es un vértice *central*. La siguiente imagen muestra ejemplos de un árbol ternario zurdo y de un árbol ternario diestro.



Una *superposición* S de un árbol zurdo C y un árbol diestro D es un árbol ternario enraizado donde la raiz es o la raiz de C, la raiz de D o las raíces de C y D superpuestas, y su estructura contiene a ambos árboles traslapados. La figura siguiente muestra algunos árboles formados por la superposición de el árbol zurdo y el árbol diestro de la imagen anterior.



(a) (b) (c)

Note que en la figura (a) la raiz es un vértice x (de el árbol diestro) y los pares de vértices (a, y) y (c, u) están superpuestos. En la figura (b) la raiz es el vértice a (de el árbol zurdo) y los pares de vértices (d, x), (e, y) y (f, u) están superpuestos. En la figura (c) la raiz es también el vértice a (de el árbol zurdo) y el par de vértices (f, x) están superpuestos.

Dados un árbol zurdo y un árbol diestro, tu tarea es determinar el número mínimo de vértices necesario para formar un árbol ternario que es una superposición de los árboles dados.

Entrada

La primer línea de un caso de prueba contiene un entero N representando el número de vértices de el árbol zurdo ($1 \leq N \leq 10^4$). Los vértices en este árbol están identificados por los números del 1 al N y su raiz es el vértice con el número 1. Cada una de las siguientes N líneas contiene tres números I, L, y K, representando el identificador del vértice I, el identificador para el hijo izquierdo L de I y el identificador del hijo central K de I ($0 \leq I, L, K \leq N$). La siguiente línea contiene un entero M representando el número de vertices en el árbol diestro ($1 \leq M \leq 10^4$). Los vértices de este árbol estan identificados por los números del 1 al M y su raiz es el vértice con el número 1. Cada una de las siguientes M líneas contiene tres numeros enteros P, Q, y R, representando el identificador para el vértice P, el identificador para el hijo central Q de P y el identificador para el hijo derecho R de P. ($0 \leq P, Q, R \leq N$). Un valor de 0 representa un vértice que no existe (Usado cuando un vértice no tiene alguno de sus hijos).

Salida

Imprime el mínimo número de vertices de un árbol ternario que es una superposición de los dos árboles dados en la entrada.

Entrada

```
7
1 2 3
2 0 0
3 4 0
4 0 5
5 0 6
6 7 0
7 0 0
7
1 2 3
2 4 0
3 5 0
4 0 6
5 0 0
6 0 7
7 0 0
```

Salida

```
11
```

Entrada

```
5
1 2 3
2 4 5
3 0 0
4 0 0
5 0 0
3
1 2 3
2 0 0
3 0 0
```

Salida

```
6
```

Entrada

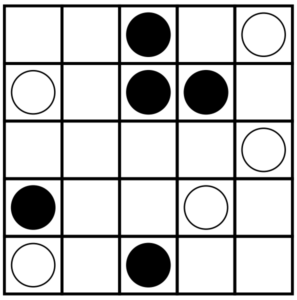
```
3
3 0 2
2 0 0
1 0 3
2
2 0 0
1 2 0
```

Salida

```
3
```

Problema G

Go−−



Go−− es un juego similar al tradicional juego de Go, ¡pero es mucho más simple! Es jugado en una matriz cuadrada de dimension N inicialmente vacia, donde dos jugadores, uno jugando con piedras negras y el otro con blancas alternan turnos poniendo una piedra a la vez en alguna celda que aún no esta ocupada. El juego termina cuando cada jugador ha puesto P piedras en la matriz. Considere todos las posibles submatrices cuadradas de dimensiones entre 1 y N. Una submatriz cuadrada pertence a el jugador con las piedras negras si contiene al menos una piedra negra y ninguna piedra blanca. Una submatriz cuadrada pertenece al jugador con las piedras blancas si contiene al menos una piedra blanca y ninguna piedra negra. Note que algunas submatrices no pertenecen a ningun jugador, aquellas que no tienen piedras o que tienes piedras de los dos colores.

En este problema, dada la configuración final de la matriz, tu programa debe computar cuantas submatrices cuadradas pertencen a cada jugador, de modo que podamos saber quien gano el juego. En la figura, el jugador de las piedras negras tiene 12 submatrices (5 de dimension 1, seis de dimension 2 y una de de dimension 3). El jugador con las piedras blancas, que perdió el juego, solo tiene 10 submatrices.

Entrada

La primer línea de entrada contiene dos números enteros N y P, $2 \leq N \leq 500$, $1 \leq P \leq 500$ y $P \leq N^2/2$, representando las dimensiones de la matriz y el número de piedras que cada jugador ha puesto respectivamente. Cada una de las siguientes P líneas contiene dos números enteros X y Y, definiendo las coordenadas de las piedras negras. Las siguientes P líneas contienen dos números enteros X y Y definiendo las coordenadas de las piedras blancas. Todas las piedras están en celdas diferentes.

Salida

Imprime una línea con dos números enteros indicando cuántas submatrices cuadradas pertenecen al jugador que juega con las piedras negras y cuántas al que juega con las blancas.

Ejemplos

Entrada	**Salida**

`2 1`
`1 1`
`2 2`
	`1 1`

Entrada	**Salida**

5 5
1 3
2 3
2 4
4 1
5 3
1 5
2 1
3 5
4 4
5 1
	12 10

Entrada	**Salida**

500 3
500 498
500 499
500 500
120 124
251 269
499 498
	4 12463784

Problema H
huaauhahhuahau

Es común entre los jóvenes usar cadenas de letras en los chats las cuales pueden llegar a parecer aleatorias para representar risas. Algunos ejemplos comunes son:

```
huaauhahhuahau
hehehehe
ahahahaha
jaisjjkasjksjjskjakijs
huehuehue
```

Claudia es una jóven programadora a quien la ha intrigado el sonido de la "risa digital". ¡Ella no puede incluso pronunciar algunas de ellas! Pero, se ha dado cuenta de que algunas de ellas representan mejor la risa que otras. La primer cosa de la que se dio cuenta es que en realidad las consonante no interfieren in como la risa digital transmite el sentimiento de risa. La segunda cosa que identifico es que las risas digitales mas graciosas son aquellas en que las vocales se leen de la misma forma en su orden natural (de izquierda a derecha) y en su orden inverso (de derecha a izquierda), ingorando las consonantes. Por ejemplo, "hahaha" y "huaauhahhuahau" son de las risas digitales mas graciosas mientras que "riajkdhhihhjak" y "huehuehue" no lo son.

Claudia está muy ocupada con el análisis estadístico de la risa digital y te ha pedido ayuda para escribir un programa que, para una risa digital, determine si es de las risas digitales mas graciosas o no.

Entrada

La entrada contiene una secuencia con no mas de 50 caracteres, conteniendo solo letras minúsculas sin acentuación y al menos una vocal. Las vocales son las letras '`a`','`e`','`i`','`o`','`u`'.

Salida

Tu programa deberá imprimir "`S`", en caso de que la risa sea de las más graciosas, or"`N`" en caso contrario.

Entrada	**Salida**
`hahaha`	`S`

Entrada	**Salida**
`riajkjdhhihhjak`	`N`

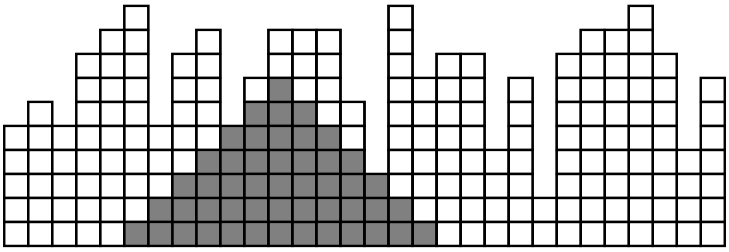
Entrada	**Salida**
`a`	`S`

Entrada	**Salida**
`huaauhahhuahau`	`S`

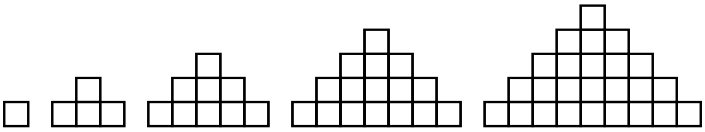
Problema I

Isósceles

Dos hermanos estaban jugando con bloques de madera tratando de construir una pared que seguía incompleta teniendo columnas de diferentes alturas como se muestra en la siguiente imagen.



Han decidido quitar bloques de la pared, siempre de arriba de las columnas, de modo que dejen un triángulo al final. Ellos solo pueden quitar bloques, no pueden moverlos de una columna a otra y el triángulo debe ser completo. La siguiente imagen ilustra los primeros cinco triángulos, del tipo que ellos están interesados con alturas $1, 2, 3, 4$ y 5 respectivamente.



Dada la secuencia de las alturas de las columnas en la pared, tu programa debe ayudar a los hermanos a encontrar cual es la altura mas grande posible para un triángulo que puedan hacer. Para la pared en la primer imagen, con 30 columnas de bloques, el triangulo mas alto posible tiene una altura de siete.

Entrada

La primer línea de entrada contiene un entero N, $1 \leq N \leq 50000$, representando el número de columnas en la pared. La segunda línea contiene N enteros A_i, $1 \leq A_i \leq N$, para $1 \leq i \leq N$ indicando la altura de cada columna.

Salida

Para cada caso de prueba en la entrada, El programa deberá imprimir una sola línea, la cual contiene Un entero H,, que representa la altura más grande que puede tener un triángulo al final..

Entrada	**Salida**
16 5 6 5 8 9 10 5 8 9 5 7 9 9 9 6 3	6

Entrada	**Salida**
8 5 1 1 1 1 1 1 3	1

Problema J

Juegos Olímpicos

Un grupo de inversionistas piensa invertir su dinero en los atletas de la delegación Brasileña después de los juegos olímpicos de Rio. Para esto ellos han estado observando a N atletas y se han dado cuenta que algunos estan en declive y otros en ascenso. Particularmente, el grupo observa dos atributos de cada atleta: su cansancio y su habilidad. Ellos han anotado los valores de estos atributos de cada atleta justo al finalizar los juegos olímpicos. Después el grupo estimó el factor con el que cada atleta gana o pierde habilidad y el factor con el que cada atleta se va cansando conforme pasa el tiempo, se dieron cuenta que estos factores son constantes para ambos atributos.

Aquellos que saben de apuestas habran notado que la información dada puede permitir definir lo que han decidido llamar el atleta de oro: Un atleta que, en un perido de tiempo es el menos cansado y con más habilidad. Se ha decidido que las inversiones serán hechas solo en atletas de oro. Encuentra cuantos atletas de los que han sido observados recibirán una inversion. Considere que el tiempo $t = 0$ es el momento de las olimpiadas de Rio: Ningún atleta que se haya convertido en atleta de oro antes de este tiempo recibirá inversión. También considere cualquier momento despues de las olimpiadas sin importar que tan grande sea este. Un atleta que es atleta de oro al momento $t = 0$ recibirá inversión.

Entrada

La entrada contendrá varios casos de prueba. En cada caso, La entrada inicia con una línea con un solo número entero N ($1 \leq N \leq 10^5$), el número de atletas. Las siguientes N líneas, cada una contiene 4 números enteros: M_i, H_t, C_i, C_t ($-10^6 < H_i, H_t, C_i, C_t \leq 10^6$, $H_t, C_t \neq 0$): La habilidad inicial de el i-ésimo atleta, su factor de variación de habilidad, su cansancio inicial y su factor de variación de cansancio.

Salida

Para cada caso de prueba en la entrada, Imprime una solo línea con un número entero, el número de atletas que recibirán inversión del grupo.

Entrada	**Salida**

3
3 2 1 2
2 2 2 2
1 2 3 2
	1

Entrada	**Salida**

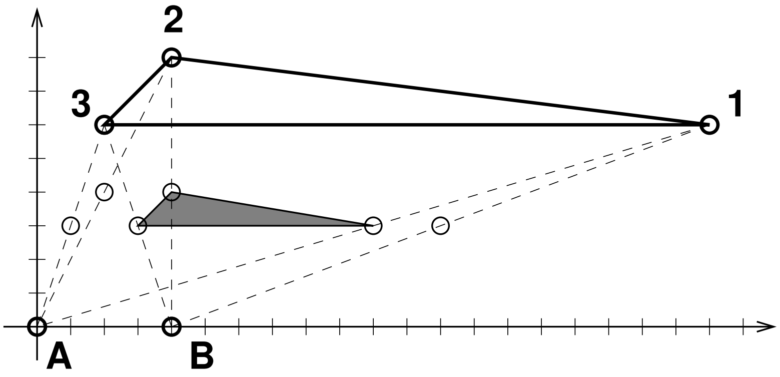
6
1 10 5 8
8 7 12 -5
10 -2 -3 8
-3 -5 -8 -12
0 1 10 2
8 3 9 -3
	0

Problema K

Kit de encogimiento de polígonos

Un *kit de encogimiento de polígonos* es una herramienta que se usa mucho en las clases de geometría mágica en Nlogonia. . El *kit* consiste de dos puntos, A y B en el plano cartesiano. Considere un polígono convexo dado por los vértices $1, 2...N$, en ese orden. Para encoger este polígono usando el *kit* se deben seguir algunas reglas. Cada vértice x de el polígono debe moverse una vez: ya sea al punto medio del segmento Ax o al punto medio del segmento Bx. El procedimiento de encogimiento debe producir un nuevo polígono convexo que mantiene el orden de los vértices del polígono original. En otras palabras, considerando todas las posibles formas de aplicar el kit, solo aquellas donde la secuencia de vértices $1, 2...N$ representa al polígono convexo resultanto son validas. Nótese que los vértices del polígono convexo original pueden estar en dirección del sentido de las manecillas del reloj y una operación de encogimiento válida puede producir un polígono convexo con los vértices en dirección contraria a las manecillas del reloj. Solo el orden relativo de los puntos es importante, no la dirección de estos.

Se sabe que la geometría mágina no es el fuerte de la mayoria de los estudiantes. El maestro les ha pedido que usando el kit engojan un polígono convexo para obtener el polígono de menor área posible. Un amigo te ha pedido que le ayudes a resolver el problema. Responde con el area mas pequeña posible que se puede obtener para el polígono dado.



La imagen de arriba muestra usos válidos del kit, donde el polígono sombreado es el de menor área posible que preserva la secuencia dada de vértices. Los puntos A y B corresponden a los puntos de el kit. Note que, a pesar del nombre *encogimient* a veces se podría usar el kit para incrementar el area del polígono. ¡Qué difícil es la geometría!

Un punto o línea no se consideran polígonos. Por lo tanto, si el kit se usa para producir como resultado algo que no es un polígono convexo entonces no es un uso válido de este.

Entrada

La entrada contendrá varios casos de prueba. En cada caso, La primer línea de entrada contiene un número entero N ($3 \leq N \leq 10^5$), el número de vértices en el polígono.Las siguientes N líneas, cada una contiene dos números enteros x, y ($-10^6 \leq x, y \leq 10^6$), los vértices del polígono. La última línea de entrada contiene cuatro números enteros, A_x, A_y, B_x e B_y ($-10^6 \leq A_x, A_y, B_x, B_y \leq 10^6$), las coordenadas x y y para el punto A y las coordenadas x y y para el punto B, respectivamente. Los puntos en la entrada serán dados en el orden que aparecen en el polígono, en sentido de las manecillas del reloj on en contra de las manecillas del relog. No habrá puntos repetidos en el polígono convexo.

Salida

Para cada caso de prueba en la entrada, Imprime un número real con 3 décimas de presición representando la menor área posible por el polígono creado usando el *kit.*

Entrada	**Salida**

3
20 6
4 8
2 6
0 0 4 0
	3.500

Entrada	**Salida**

3
0 4
4 4
0 0
3 -2 -3 -2
	1.000

Entrada	**Salida**

3
0 4
4 4
0 0
2 -2 -2 -2
	2.000

Problema L

Ladrillos

Avelino tiene un mosaico en una de las paredes de su casa. Es un mosaico antiguo compuesto de pequeños ladrillos de colores. Como es un mosaico antiguo al pasar el tiempo algunos ladrillos se han caido formando huecos.

Avelino quiere restaurar el mosaico cubriendo los huecos con nuevos ladrillos, sin embargo, para ahorrar un poco de dinero el quiere comprar todos los ladrillos con los que cubrirá los huecos de un mismo color. El comprará los ladrillos ya sea de el color original o de un color que no está en el mosaico.

Siendo un mosaico, no se quiere tener areas grandes con un mismo color. Avelino decidió que elegirá el color de los ladrillos de modo que el tamaño mas pequeño de un area monocromática sea lo mas pequeña posible, así el mosaico tendrá mas detalle. Notese que puede haber mas de un color posible para lograr esto. Un área monocromática es un área donde todos los ladrillos en ella son del mismo color. Dos ladrillos adyacentes pertenecen a la misma area tendrán el mismo color y dos ladrillos son adyacentes si comparten un lado.

Mira el primer ejemplo, hay tres areas de color 1 (una de tamaño 3, dos de tamaño 2), un área de color 2 (tamaño 3) y un area de color 3 (tamaño 7). Una respuesta posible sería elegir el color 2, de esta manera el area monocromática mas pequeña sería 2. Si elegimos el color 1 el área monocromática mas pequeña sería 3.

Crea un programa que imprima el tamaño de el área monocromática mas pequeña posible.

Entrada

La primer línea de entrada contiene dos números H y L, el alto y ancho del mosaico respectivamente $1 \leq H \leq 200$ and $1 \leq L \leq 200$. Las siguientes H líneas contienen cada una L números esteros separados por un espacio representando el color de cada ladrillo. Un valor de 0 representa un hueco en el mosaico y un valor $i \neq 0$ representa un ladrillo de color i $1 \leq i \leq 40000$

Salida

Deberás imprimir una sola línea con un entero el tamaño del área monocromática mas pequeña posible.

Entrada	**Salida**

3 8
3 3 3 1 1 0 0 0
3 1 1 0 2 2 0 1
3 3 3 0 0 2 1 1
	2

Entrada	**Salida**

3 7
1 1 0 2 2 1 1
1 1 0 2 2 1 1
1 1 0 0 3 3 3
	3

Entrada	**Salida**

3 6
2 2 2 2 0 2
2 2 2 0 2 2
2 2 2 2 0 2
	1

GRAN PREMIO 2016 CUARTA FECHA

The 2016 ACM-ICPC Caribbean Local Contests
Gran Premio de México & Centroamérica - Cuarta Fecha

Real Contest Problem Set

Document composed by 13 pages (including this cover)

Problem authors and refiners:

Carlos Joa Fong (INTEC, Dominican Republic)
Humberto Díaz Suárez (UPR-MY, Puerto Rico)
Óscar Dávalos Orozco (UP-B, Mexico)
Alfredo Fundora Rolo (UM, Cuba)
Rainel Estrada Montejo (UM, Cuba)
Gregorio Ferrer Córdova (UCI, Cuba)
Yonny Mondelo Hernández (UCI, Cuba)
Carlos J. Toribio (Dominican Republic)
Daniel Otero Baguer (UH, Cuba)
Norge Vizcay Zaldívar (UH, Cuba)
Oreste Abrahan Nillar Cambara (UCI, Cuba)
José Daniel Rodríguez Morales (UCLV, Cuba)
Frank Rafael Arteaga Salgado (ULT-PSNIC, Cuba)
Jorge Enrique Moreira Broche (Alphabet Inc., USA)
José Carlos González Fernández (Kueski, Mexico)

October 1st, 2016.

Problem A – Strongest Digit

Description

Alex is a very smart kid who is studying for a side event of the Computing Olympiad in his school. The side event is called "Seeking the Strongest Digit" and is about the following: a list of N integers A_1, A_2, ..., A_N will be given to participants and then a computer will ask Q queries that the participants must execute. The computer can ask the following kind of queries:

- 0 *i j* : The participants must find which is the most repeated digit (0..9) inside the interval [*i, j*] in the current list. In case of multiple digits having the same frequency, the answer must be the smallest of them.
- 1 *i j m* : This operation replaces each number inside the given interval [*i, j*] by the number *m*.

Contestants must answer each operation of the first kind taking into account the operations of the second kind performed previously and the changes in the initial list. The contestant who answer first any operation awards one point, and finally at the end of all operations the contestant with the highest amount of point is the winner of the event. Alex wants to win the competition so he needs your help to write a program that allows him to process those operations as quickly as possible. ¿Can you help Alex?

Input specification

The first line of input has two integer numbers N ($1 \leq N \leq 10^5$) and Q ($1 \leq Q \leq 10^4$) representing the size of the list and the total amount of operations respectively. Each of the following N contains an integer number Ai ($1 \leq Ai \leq 10^{101}$) representing an element of the list. The elements will be given from the first one to the last one in that order. The following Q lines will contains one of the operations that the computer can ask: 0 *i j* ($1 \leq i \leq j \leq N$) or 1 *i j m* ($1 \leq i \leq j \leq N$), ($1 \leq m \leq 10^{101}$). The operations must be processed in the order of entry.

Output specification

For each operation of the first kind, you must output a line containing a digit (0..9) representing the corresponding answer.

Sample input

```
7 6
41
7745980
96771000
7338885
36
83764
665582563
0 1 2
0 2 5
0 3 5
1 2 6 98
0 2 4
1 1 7 3694161
```

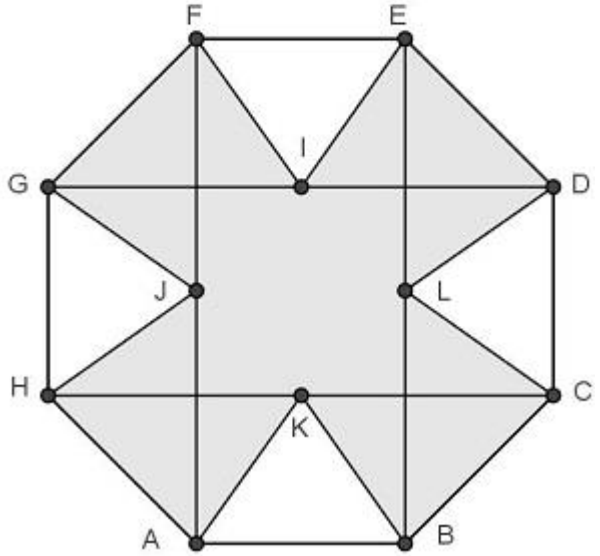
Sample output

```
4
7
0
8
```

Problem B – Area of the Polygon

Description

In the next figure, ABCDEFGH is a regular octagon and the measure of each of their sides is *x* units. I is the midpoint of the segment GD, J is the midpoint of the segment FA, K is the midpoint of the segment HC and L is the midpoint of the segment BE. The area of the polygon AKBCLDEIFGJH is *a+b*sqrt(c)*, where *a*, *b*, and *c* are integers and *sqrt(c)* cannot be simplified further. Your task consists in finding the value of *a + b + c*.



Input specification

The first line contain a integer number *t* ($1 \le t \le 100$) representing the number of test cases. The next *t* lines contains a integer *x*, $0 < x < 10000$.

Output specification

For each case, you must output a line with the value of *a + b + c*.

Sample input

```
1
3
```

Sample output

```
29
```

Problem C – Max of the K

Description

Given a list of N integer numbers A_1, A_2, ..., A_N and a positive integer number $K \le N$, you must output $N\text{-}K\text{+}1$ numbers: maximum ($A_1, A_2, ..., A_K$), maximum ($A_2, A_3, ..., A_{K+1}$), maximum ($A_3, A_4, ..., A_{K+2}$), ..., maximum ($A_{N-K+1}, A_{N-K+2}, ..., A_N$).

Input specification

The first line of input contains an integer T ($0 < T \le 10^3$) denoting the number of test cases. Each case is composed by two lines. The first line of each case contains two space-separated integer numbers N and K ($0 < K \le N \le 10^6$) respectively. And the second line contains N space-separated integer numbers whose absolute values are less than or equal to 2^{31} - 1. You can safely assume that sum of all values for N in a single file of input is lower or equal to 10^6.

Output specification

For each case, you must output a line containing $N\text{-}K\text{+}1$ numbers: maximum ($A_1, A_2, ..., A_K$), maximum ($A_2, A_3, ..., A_{K+1}$), maximum ($A_3, A_4, ..., A_{K+2}$), ..., maximum ($A_{N-K+1}, A_{N-K+2}, ..., A_N$).

Sample input

```
1
5 3
8 3 1 4 5
```

Sample output

```
8 4 5
```

Problem D – Eleven Sum

Description

Given a number *S*, find the smallest positive integer number *n* that is evenly divisible by 11 and the sum of its digits equals *S*.

Input specification

Input consists of several test cases (no more than 101), one per line. Each line contains the integer number *S* ($1 \leq S \leq 10^6$). The last test case is followed by a line containing a 0 (zero), which should not be processed.

Output specification

For each input line, output the smallest positive integer number divisible by 11 whose digits add up to *S*, or -1 if no such number exists.

Sample input

```
4
0
```

Sample output

```
22
```

Problem E – Rubik Packages

Description

In the warehouse of some toy factory, there are *N* Rubik's Cubes with a shape of a cube with side 1, which should be transported to the market. For this task, it is required to store all Rubik's Cubes inside cubic boxes of equal size. Nevertheless, no matter the size of the boxes, it must be guaranteed that all boxes are completely filled with Rubik's Cubes. Can you determine the total number of possible ways to package all the Rubik's Cubes using boxes of equal size? All cubes are indistinguishable, so two ways are different only if the size of the boxes is different.

Input specification

Input consists of several test cases (no more than 101), one per line. Each case is composed by a line with a integer number N ($1 \le N \le 10^8$). The last test case is followed by a line containing a 0 (zero), which should not be processed.

Output specification

For each case, you must output a line with an integer number representing the total number of possible ways to package all the Rubik's Cubes using boxes of equal size.

Sample input

```
8
0
```

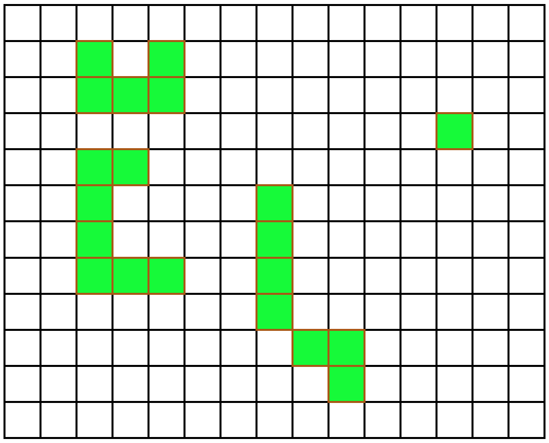
Sample output

```
2
```

Problem F – Connect the Islands

Description

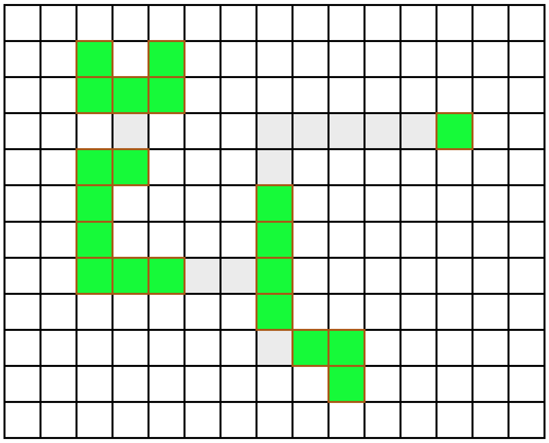
You are given a rectangular map of *RxC* cells showing a number of small islands surrounded by sea. Each cell contains either land or sea water (but not both). Two cells are considered connected if they share an edge or you may travel between those two cells using other connected cells. An island is a maximal group of land cells that are connected. For instance, the following 12x15 map has 5 islands (painted in green):



To facilitate inter-travel, the governments of these island nations wish to build some number of bridges to connect the islands. Each bridge connects exactly two islands. The cost of building a bridge is equal to the number of cells that it occupies on the map. As all bridges have distinct heights, they do not intersect each other.

What is the minimum cost to build the bridges required to connect all the islands?

For the example above, here is a way to build bridges (denoted by the grey squares) that minimizes the total cost:



Input specification

In the first line of input, we have a single integer *T* (1 ≤ *T* ≤ 10), which corresponds to the number of test cases.

Each test case consists of two lines:

1. Line 1 contains exactly two space-separated integers *R* (3 ≤ *R* ≤ 50), *C* (3 ≤ C ≤ 50).
2. Lines 2 through *R*+1 contain *C* characters each, where the *jth* character in the *ith* row corresponds to cell (*i, j*) in the map. Each character is either the '.' or 'X', representing sea and land respectively. It is guaranteed that cells in the border of the map (ie, in the first row, last row, first column or last column) contain water.

For each test case, the number of islands is at least 2 and at most 300.

Output specification

For each test case, in the order given in the input, print the minimum cost to build bridges that connect all islands.

Sample input

```
2
12 15
...............
..X.X..........
..XXX..........
............X..
..XX...........
..X....X.......
..X....X.......
..XXX..X.......
.......X.......
........XX.....
.........X.....
...............
5 5
.....
.X...
...X.
.X...
.....
```

Sample output

```
10
3
```

Problem G – Count Odd Increasing Subsequences

Description

Given a sequence of *N* integers A_1, A_2, ..., A_N, count the number of strictly increasing subsequences whose sum is odd.

A subsequence of sequence *A* is obtained by selecting some positions of the sequence *A*, keeping the elements at those positions in the same order, and removing all elements at positions that were not selected. A strictly increasing subsequence is a subsequence where the elements at the selected indices are increasing. Two subsequences are different if there is a position selected in one subsequence but not the other.

More formally, a sequence S_1, ..., S_k is called a strictly increasing subsequence of sequence A_1, ..., A_N, if there is a sequence of indices $1 \leq i_1 < ... < i_k \leq N$ such that $S_1 = A_{i1}$, ..., $S_k = A_{ik}$ and $S_1 < S_2 < ... < S_k$. A subsequence *S* is different than subsequence *T* if there exists an index $j \in \{i_1, \ldots, i_k\}$ that is selected by *S* but not by *T*.

Input specification

In the first line of input, we have a single integer *C* ($1 \leq C \leq 100$), which corresponds to the number of test cases to process.

Each test case consists of two lines:

1. Line 1 of each test case contains a single integer *N* ($1 \leq N \leq 10^5$), the length of the sequence *A*.
2. Line 2 contains a sequence of *N* space separated integers *Ai* ($0 \leq Ai \leq 10^9$ for $1 \leq i \leq N$).

The sum of all *N* over all test cases does not exceed $2 * 10^5$.

Output specification

For each test case, in the order given in the input, count the number of strictly increasing subsequences whose sum is odd. As this number may be too large, print out its remainder after dividing it by $10^9 + 7$.

Sample input

```
2
3
1 2 2
5
5 1 4 3 5
```

Sample output

```
3
7
```

Hint(s)

In the first sample test case, the 3 subsequences are: (1), (1, 2), (1, 2)
In the second sample test case, the 7 subsequences are: (5), (1), (1, 4), (1, 3, 5), (4, 5), (3), (5)

Problem H – My Longest Palindrome

Description

Given a string S of uppercase letters, you may perform any number of these operations on it (and in any order you wish):

- Delete a character at any position
- Swap any two characters in adjacent positions

¿What is the longest palindrome you can get? A palindrome is a sequence of characters that reads the same backward or forward.

Input specification

The first line of input consists of a single integer T ($1 \le T \le 100$) denoting the number of test cases to process.

Each test case consists of two lines:

1. Line 1 contains a single integer N ($1 \le N \le 1000$), the length of the string S.
2. Line 2 consists of string S having exactly N uppercase letters.

Output specification

For each test case, in the order given in the input, output the longest palindrome you can get after applying any number of the described operations. If there are many solutions, output the lexicographically smallest of them. A string S is lexicographically smaller than string T if the character at position i in S is smaller than the character in the same position i in T where i is the first position that the two strings differ.

Sample input

```
4
4
NOON
5
MADAM
3
ABC
8
XXXZZZYY
```

Sample output

```
NOON
AMDMA
A
XYZXZYX
```

Problem I – Fair Funding

Description

An investment firm is working on providing funding for several entrepreneurs. The funding is assigned by awards, to which certain entrepreneurs are eligible. An award must be given to all eligible recipients or to none of them.

Each entrepreneur is classified as regular or risky. A regular entrepreneur will be eligible for two awards at most and must receive exactly one. A risky entrepreneur will be eligible for three awards and must receive all or none of them. They are not guaranteed to receive an award, unlike a regular.

Given a description of who is eligible for each award, determine the minimum number of awards that must be given out, or whether it's impossible to comply with the above constraints.

Input specification

The input will begin with a line containing an integer T ($1 \le T \le 20$) denoting the number of test cases. Each case will begin with a line containing two integers N and M ($1 \le N, M \le 1000$), separated by a space, denoting the number of awards and entrepreneurs, respectively. The entrepreneurs will be numbered 1 through M.

The next N lines will each describe an award. A description will begin with an integer P ($1 \le P \le M$), indicating the number of entrepreneurs eligible for that award. Then a sequence of integers C_1 through C_P will follow on the same line, listing the entrepreneurs eligible for that award. It's guaranteed that no entrepreneur will be listed twice for the same award. There will be a space between every pair of consecutive integers.

Output specification

The output will consist of one line for each test case indicating the minimum number of awards that must be given out. If it's impossible to provide awards while following all of the funding rules, then the result must be "impossible". The order of the results must follow the same order in which the test cases are provided.

Sample input

```
4
4 2
1 1
1 1
2 1 2
1 2
4 6
2 1 2
2 1 3
2 2 4
4 3 4 5 6
3 2
1 1
2 1 2
2 1 2
2 2
1 2
1 2
```

Sample output

```
1
2
impossible
impossible
```

Hint(s)

In the first case, entrepreneur 1 is risky and entrepreneur 2 is regular. The fourth award would fund entrepreneur 2, while entrepreneur 1 must be left without funding. That is the minimum solution. In the second case, the first and fourth awards are enough to provide everyone with exactly one award. In the third case, it's impossible to satisfy both entrepreneurs at the same time. In the fourth case, it's impossible to satisfy entrepreneur 1, who is not eligible for any awards.

Problem J – Crazy Circle

Description

Carla recently started assisting with her school's math club. She loves teaching them about math and logic problems. The best problems are those that she can turn into a game to entertain students. This week, Breanna has brought a problem about kangaroos in a circle. Everyone is eager to play along.

A group of students sits in a circle. Carla numbers them 1 through *N* in clockwise order. She covers her eyes and the students have a moment to move around by swapping positions. Two students can swap if they are adjacent and none of them has moved before. The swaps are done one by one, so two swaps cannot occur simultaneously. After some time, Carla must figure out which students moved.

Sometimes she has a little trouble figuring out who moved. She chooses some student (not necessarily 1) as a starting point, then writes down the sequence of their numbers in clockwise order to ask you. Please, give Carla a hint by telling her the number of students who moved.

Input specification

The input will begin with a line containing an integer *T* (1 ≤ *T* ≤ 1000) denoting the number of test cases. Each case will consist of one line. The line will begin with an integer *N* (3 ≤ *N* ≤ 1000). Exactly *N* integers will follow on that line, indicating the sequence of student numbers recorded by Carla. These will be a permutation of the integers 1 through *N*. There will be a space between each pair of consecutive integers. You may assume that every student follows the rules of the game.

Output specification

The output will consist of one line for each test case indicating the number of students who moved. There may be more than one way in which students can move and Carla can write their order that gives the same sequence, but all of them will use the same number of moves. The order of the results must follow the same order in which the test cases are provided.

Sample input

```
3
6 1 3 2 5 4 6
5 3 5 4 1 2
7 7 2 3 5 4 6 1
```

Sample output

```
4
2
4
```

Hint(s)

In the first case, students 2 and 3 swapped positions, as did students 4 and 5. In the second case, students 4 and 5 swapped positions. In the third case, students 1 and 7 swapped positions, and students 4 and 5 also swapped.