Problem A
A Research Team

Time Limit: 1 second
Memory Limit: 256 megabytes

Phidang wants to build a research team at HCMUS. Thus, he proposes a simple problem to test the applicants in the interview when they want to join his team. The problem is about the total number of HCMUS students.

Phidang tries to estimate the total number of students in the following way in two days. On the first day, Phidang gives every student coming to the campus a sticker to put on their shirts, and he observes that there are n_1 students. On the second day, Phidang counts the number of students coming to the campus. He realizes that there are n_2 students who come to the campus, but only n_{12} students have the stickers. Assuming that the n_1 students on the first day still carry their stickers.

Now, you are a brilliant student in Statistics course and you want to join Phidang's research team, can you help Phidang to estimate the total number of HCMUS students using the Chapman equation below?

$$\hat{N} := \left\lfloor \frac{(n_1 + 1)(n_2 + 1)}{n_{12} + 1} - 1 \right\rfloor$$

Input

The input contains three integers n_1, n_2, n_{12} on a single line.

Constraints:

- $0 \leq n_1, n_2 \leq 10^4$,
- $0 \leq n_{12} \leq \min(n_1, n_2)$

Output

The output contains a single integer $\hat{N}$.

Sample Input	Sample Output
15 18 11	24

Problem B
Rap Songs

Time Limit: 1 second
Memory Limit: 256 megabytes

Phidang likes watching Rap Viet show on TV every Saturday evening. After each show, he usually downloads all the rap songs into his cell phone and listens to them at work. When listening, he likes to turn on crossfading between the rap songs, so during the last seconds of a song, it will slowly fade out while the next one fades in. This happens between any two consecutive songs, but the beginning of the first song and the last song will be played normally.

Given the crossfade time, Phidang wonders how much time does it take him to listen to his whole rap album.

Input

The input consists of:

- Two integers n and $(1 \leq n \leq 100, 1 \leq c \leq 10)$, indicating the number of rap songs in Phidang's album and the crossfade time in seconds.
- In the next n lines, each line contains a string of the form `m:ss` (`0:30` $\leq$ `m:ss` $\leq$ `9:59`), denoting the song length (a single digit for minutes and two digits for the remaining seconds).

Output

Output contains a single string of the form `hh:mm:ss`, giving the total time to listen to the whole album (the first two digits are for the number of whole hours, two digits for the number of remaining whole minutes, and two digits for the remaining seconds).

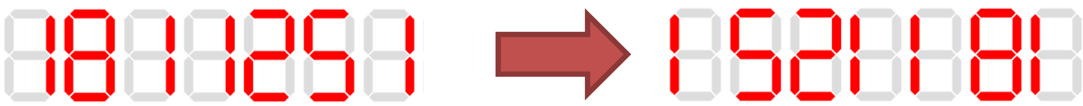
Sample Input	Sample Output
3 5 3:59 0:52 9:40	00:14:21
1 10 6:00	00:06:00

Problem C
Clock and Primes

Time Limit: 1 second

Memory Limit: 1024 megabytes

Last year, Phidang bought a digital alarm clock that looks like the one above. This morning, it went off when Phidang accidentally dropped it. The clock stopped working but it still shows `11:21:80` instead of `08:12:11`. Phidang realized that he was looking at it upside down. It made him think about the numbers showing on the clock the whole day. Some numbers are still numbers after turning them upside down as those displayed on Phidang's clock.



Phidang loves prime numbers. He wants to check whether a number is a prime and still a prime when it is turned upside down.

Noted that there are some digits are no longer valid if they are turned upside down such as 3, 4, and 7; some digits change their value such as 6 and 9; some keep their own value such as 0, 1, 2, 5, and 8.

Input

The input contains a single integer N ($1 \le N \le 10^{16}$) which is the number that needs to be checked. N does not have leading zeros.

Output

Print one line of output containing "`yes`" if the number is prime and still a prime if turned upside down, "`no`" otherwise.

Sample Input	Sample Output
151	yes
23	no
1811	yes

Problem D
Another Problem

Time Limit: 1 seconds

Memory Limit: 1024 megabytes

Phidang thinks that the first problem to test the candidates is too simple and easy. Therefore, he wants to give another problem with numbers. The problem is as follows.

There are two extremely large numbers a and b. Phidang asks the candidates to compute the multiplication of all the numbers in the interval $[a, b]$. Let the result of the multiplication is x. Now, the candidates have to do the following process to find the result.

1. Calculate the sum of all digits of x.
2. If the sum is less than 10, print it.
3. Otherwise, repeat the process with x is now the sum of all digits of x.

Input

The input contains two lines corresponding to the two numbers a and $(1 \le a \le b \le 10^{100\,000}$, each lies on one line.

Output

The output contains a single number, which is the result of the above process.

Sample Input / Sample Output

Sample Input	Sample Output
1 5	3
6 8	3
5 68	9

Problem E
Extremums and Extremals

Time Limit: 2 seconds

Memory Limit: 1024 megabytes

Consider an integer sequence $a_1, a_2, a_3, \dots, a_N$, and an element a_i. Phidang gives you some definitions:

- a_i is called a *local maximum* if $a_i > a_{i-1}$ and $a_i > a_{i+1}$
- a_i is called a *local minimum* if $a_i < a_{i-1}$ and $a_i < a_{i+1}$
- a_i is called a *local extremum* if a_i is either a *local maximum* or *local minimum*
- A sequence $p_1, p_2, \dots, p_N$ is called a *permutation* of integer from 1 to N if each of the integers appears in the sequence exactly once.
- A permutation is called *extremal* if each element (except the first and the last) is a local extremum.

Now, you are given an extremal permutation of $1, \dots, N$. Phidang asks you to find the following one in the lexicographical order of all extremal permutations of these elements.

Noted that, in the lexicographical order of a sequence $x_1, x_2, \dots, x_k$ comes before a sequence $y_1, y_2, \dots, y_k$ if and only if the first x_i, which is different from y_i, is less than y_i. In addition, if the given extremal permutation is the last one in the lexicographical order, you have to provide the lexicographically first one.

Input

The first line contains the integer N $(1 \le N \le 100{,}000)$ and the second line contains N integers: the values of $p_1, p_2, \dots, p_N$.

Output

The only line of the output file should contain N integers: the values $q_1, q_2, \dots, q_N$ such that $q_1, q_2, \dots, q_N$ if the permutation immediately following the permutation $p_1, p_2, \dots, p_N$ in the lexicographical order of all extremal permutations of $1, \dots, N$. If the input contains the lexicographically last extremal permutation, print the lexicographically first one as the output.

Sample Input	Sample Output
3 2 3 1	3 1 2
3 3 1 2	1 3 2

Problem F
Encrypt and Decrypt

Time Limit: 1 seconds

Memory Limit: 1024 megabytes

There are many manuscripts in HCMUS that Mr. Do Le wants to encrypt and decrypt. The plaintext is encrypted using a short keyword consisting of lowercase Latin letters. Only Latin letters are replaced during the encryption, all other symbols remain unchanged. The letters are divided into blocks so that all (except for the last one, possibly) have the same length as the key. If a_1 is the position of the first letter of the plaintext in the alphabet and b_1 is the position of the first letter of the key, then the first letter of the text is replaced with the letter at the position $a_1 + b_1$ of the alphabet (if $a_1 + b_1 > 26$, the letter $a_1 + b_1 - 26$ is used instead). The case of the letters of the plaintext is preserved. The following letters of each block are encrypted in the same manner, using the corresponding letter of the key.

For example, let the plaintext be "`crusader`" and the key be "`bow`". The first letter of the ciphertext is '`e`' (the position of the first letter of the plaintext is 3 and the position of the first letter of the key is 2, thus the first letter of the ciphertext must be at the position 5 of the alphabet). The second letter of the plaintext is replaced with '`g`' ($18 + 15 = 33,\ \ 33 - 26 = 7$). Continuing in the same manner, the whole ciphertext turns ou t to be "`egrupagg`". Note that in this case each letter of the plaintext is always represented by the same letter in the ciphertext, but this is a mere coincidence – it does not happen when the distance between the occurrences of a letter in the plaintext is not a multiple of the length of the key!

You are given two fragments of a document: one in the plaintext, the other encrypted using the above system. The lengths of the two fragments are equal and it is known that the first character of the ciphertext is obtained from the first character of the plaintext. It is also known that the number of letters in each fragment is no less than the length of the keyword. It is, however, not known if the fragments start at the beginning of a document!

Based on the given data, derive the shortest possible keyword!

Input

The first line contains N ($1 \le N \le 10^6$) – the number of characters in each fragment.

It is followed by the two text fragments, first the plaintext and then the ciphertext. The plaintext and ciphertext always start in a new line, but each of the fragments may be arbitrarily split across several lines. In the example below, the newlines are marked with the character '•' which will not be present in the actual input.

Output

The output contains only one string, which is the keyword. If there are several possible keywords, output the smallest one in the lexicographical order.

Sample Input	Sample Output
8 Crusader• Egrupagg•	bow
41 To be or not t• o be? What is the question!• Yp ru aw oej ft cu?• Mtfu yi fmf gkqxuyez!•	apple

Problem G
Exotic Convolution

Time Limit: 1 seconds

Memory Limit: 512 megabytes

You are given a binary three-dimensional array $t[n][3][3]$, where $t[i][a][b] = t[i][b][a]$, and two integers u and v where $0 \leq u, v < 3^n$.

In base 3, u and v are represented as:

$$u = a_0 + a_1 \cdot 3^1 + \ldots + a_{n-1} \cdot 3^{n-1}$$

$$v = b_0 + b_1 \cdot 3^1 + \ldots + b_{n-1} \cdot 3^{n-1}$$

where $0 \leq a_i, b_i < 3$.

Let define the function f as follows:

$$f(u, v) = t[0][a_0][b_0] + t[1][a_1][b_1] \cdot 2^1 + \ldots + t[n-1][a_{n-1}][b_{n-1}] \cdot 2^{n-1}$$

It is straight-forward to show that $0 \leq f(u, v) < 2^n$.

Your task is to calculate the array c, defined as follows:

$$c_i = \sum_{f(u,v)=i} a_u \cdot b_v$$

Input

The first line contains n ($0 \leq n \leq 11$).

The next n lines of the input represents the binary array t. The i^{th} line contains 9 integers: $t[i-1][0][0], t[i-1][0][1], t[i-1][0][2], t[i-1][1][0], t[i-1][1][1],$ $t[i-1][1][2], t[i-1][2][0], t[i-1][2][1], t[i-1][2][2]$, separated by spaces.

It is guaranteed that all elements of t are either 0 or 1, and $t[i][x][y] = t[i][y][x]$.

The next line contains 3^n space-separated integers: $a_0, a_1, \ldots, a_{3^n-1}$. It is guaranteed that $0 \leq a_i \leq 10^9$.

The next line contains 3^n space-separated integers: $b_0, b_1, \ldots, b_{3^n-1}$. It is guaranteed that $0 \leq b_i \leq 10^9$.

Output

Print 2^n space-separated integers on a single line: $c_0, c_1, \ldots, c_{2^n-1}$ modulo $10^9 + 7$.

Sample Input	Sample Output
1 0 0 0 0 0 0 0 1 0 0 6 4 8 4	72 24

Problem H
Soldiers

Time Limit: 1 second

Memory Limit: 512 megabytes

Phidang is the leader of N soldiers. One day, the whole team has to prepare for a mock battle. They have to move along the places indexed from $-1{,}000{,}000{,}000$ to $1{,}000{,}000{,}000$. One place is marked as X, where the destination is.

Then, one by one, the i^{th} soldier ($1 \le i \le N$), starting from position a_i, has to move to the destination in the smallest number of turns. A soldier can only move from place y to the place $y + j$ or $y - j$ at the j^{th} turn (if the two places are still in the above range).

Phidang wants to know the total number of turns that all of his soldiers need to move. Given that, a supervisor of Phidang changes the team member positions or the destination X all the time, so please help him to compute the sum of the turns needed after each command from Phidang' supervisor.

Input

The first line contains two integers N ($1 \le N \le 5000$) - the number of soldiers, and X – the initial destination.

The second line contains N integers, the integer a_i ($1 \le a_i \le 10^6$) is the initial position of the i^{th} soldier.

The third line contains a single integer M ($1 \le M \le 10^6$) – The number of time that Phidang' supervisor commands.

In the next M lines, each line can be one of two following formats:

- $1\ x\ y$: change a position of soldier x to y ($1 \le x \le N, 1 \le y \le 10\text{^}6$) (assign $a_x = y$)
- $2\ Y$: change the destination to Y ($1 \le Y \le 10^6$) (assign $X = Y$)

Output

The output contains M integers, line i presents the total number of turns needed after each command.

Sample Input	Sample Output
2 1 9 11 2 2 10 1 1 12	2 4

Explanation:

After the first command, $X = 10$. The first soldier moves from 9 to 10, the second soldier moves from 11 to 10. The total number of moves is 2.

After the second command, $a_1 = 12$. The first soldier moves from 12 to 11 then to 13 and finally reaches 10. The second soldier moves from 11 to 10. The total number of moves is 4.

Problem I
Simple Regular Expression

Time Limit: 1 second

Memory Limit: 64 megabytes

Mr. Do Le is learning about regular expression. He has just learned about the notation "*" (star). This character can be used for representing any string including an empty string.

Mr. Do Le considers that a string ***T*** matches a pattern ***P*** if and only if there is a way to replace the stars in ***P*** with (possibly empty) sequences of lowercase letters so that the result equals ***T***. For example, the string `aadbc` matches the pattern `a*b*c`, as we can obtain the string from the pattern by replacing the first star in the pattern with `ad` and the second one with the empty string. On the other hand, the string `abcbcb` does not match this pattern.

Given a non-empty string ***S***, Mr. Do Le wants to know the number of cyclic shifts of ***S*** that match the given pattern ***P***.

Note: The cyclic shift is defined as the string ***S*** can be split into two non-empty parts ***X*** + ***Y*** and in one operation we can transform ***S*** to ***Y*** + ***X*** from ***X*** + ***Y***.

Input

The first line of input contains the pattern P (no more than 100 characters).

The second line contains the string S (no more than 10^5 characters).

Output

The output consists of a single integer, the number of cyclic shifts of S that match the pattern P.

Sample Input	Sample Output
aaaa aaaa	4
a*a aaaaaa	6
*a*b*c* abacabadabacaba	15

Problem J
Multicolored Cycles

Time Limit: 2 seconds

Memory Limit: 512 megabytes

You are given an undirected graph with n vertices and m edges. The graph may contain self-loops (edges connect one vertex to itself) or parallel edges (edges that connect the same pair of vertices). A customer wants you to color all edges in this graph with either **blue** or **red**. Because the customer is a very weird person, he wants every cycle of the graph to have edges of both colors, i.e. there cannot be a cycle with edges only in red or blue. Your task is to find out if you can accomplish the problem.

Input

The first line contains two integers n and $(1 \leq n \leq 2000, 1 \leq m \leq 4000)$. Each of the next n lines contains two integers u and v denoting an edge connecting the u^{th} vertex to the v^{th} vertex $(1 \leq u, v \leq n)$.

Output

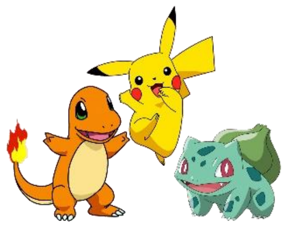
Output "`Yes`" if it is possible to color the edges satisfying the customer's condition, or "`No`" if it is impossible to do so.

Sample Input	Sample Output
2 2 1 2 1 2	Yes
2 3 1 2 1 2 1 2	No
5 4 1 2 2 3 1 5 2 4	Yes
5 8 1 2 2 3 3 1 1 4 4 5 5 1 2 4	Yes

Problem K
Pokemons

Time Limit: 1 seconds

Memory Limit: 64 megabytes



Phidang is on his way to becoming a Pokemon Master. Recently, he has passed Kanto, Johto, Hoenn, and now he is in Sinoh. In order for him to beat all the Gym Masters in Sinoh, he needs to capture many stronger, faster, higher status-point pokemons in this 4^{th} region.

To capture the pokemons, Phidang came to the pokeshop to buy many different types of poke-balls (e.g, Poke Ball, Great Ball, Ultra Ball, Master Ball, Safari Ball, Lure Ball, Moon Ball, Friend Ball, Love Ball, Heavy Ball, Fast Ball, etc.). Each poke-balls type has a particular size (poke-balls of the same type have the same size), and for each type, he bought a certain amount for later use.

Unfortunately, during the battle in the first Gym in Sinoh, Phidang dropped his bag, and all of his poke-balls (the balls are perfect spheres) were broken into two halves. Hence, he has to use his tape to repair their poke-balls.

Given the length of the tape that Phidang has, the radius and the number of the poke-balls of each poke-ball type, Phidang wants to know the maximum number of poke-balls that he can fix regardless of the types.

Input

- The first line contains an integer t $(1 \le t \le 10^4)$, the length of the tape in centimeters.
- The second line contains an integer b $(1 \le b \le 100)$, the number of poke-ball types.
- In the next b lines, line i contains two integers s $(1 \le s \le 10^3)$ – the number of poke-balls of the i^{th} type, and d $(1 \le d \le 10^3)$ – the radius (in centimeters) of the poke-balls of the i^{th} type.

Output

- The output contains a single integer indicating the maximum number of poke-balls that Phidang can fix using his tape.

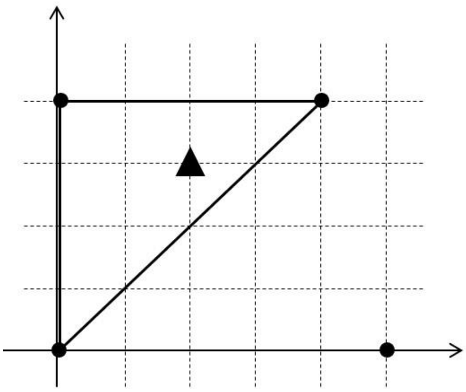
Sample Input	Sample Output
1000 2 2 30 3 20	5
2000 4 20 30 1 20 1 15 20 25	13

Problem L
Treasure Protection

Time Limit: 2 seconds

Memory Limit: 1024 megabytes

Recently, the very famous VNU Museum has been attracting many students and visitors from around the world. Thus, the treasures in the museum could be in danger and we have to protect them. In order for the managers of the museum to guard the treasure against stealing, they decided to install fences around all the treasures.



We can consider the museum is a convex polygon whose vertices are the corners, and the treasures are points inside the polygon. The fences that will be installed form 1 or 2 triangles (noted that the common area of these triangles is 0), and the vertices of these triangles must be the vertices of the polygon.

The managers of the museum hope that the fences cause the least inconvenience for the visitors, so the total areas of the two triangles should be as small as possible.

In other words, given n vertices on Oxy plane, which are the corners of the museum, and the position of the m treasures in the museum, find the minimum total areas of the guard regions.

Input

- The first line contains an integer t – the number of test cases. Each test case has the following structure:
  - The first line contains two integers n, m ($3 \le n \le 2000; 1 \le m \le 2000$) – the number of the corners, and the number of treasures, corresponding.
  - In the next n lines, the i^{th} line contains two integers x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$), the coordinate of the i^{th} corner. They are listed in counter-clockwise direction.
  - In the next m lines, the j^{th} line contains two integers x_j, y_j $\left(-10^9 \le x_j, y_j \le 10^9\right)$, the coordinate of the j^{th} treasure.

- It is guaranteed that all the treasures are inside the museum, and there are no three collinear points among the given $n + m$ points.
- The total number of vertices in all test cases are not greater than 2000.

Output

- Print -1 if there is no possible solution.
- Otherwise, print the minimum total areas of the two guard regions. The output must contain exact 3 digits after the floating point.

Sample Input	Sample Output
3 4 1 0 0 5 0 4 4 0 4 2 3 5 3 0 0 6 -6 11 0 8 4 3 4 3 2 7 3 8 -2 8 4 -4 -4 0 -7 4 -4 6 0 4 4 0 7 -4 4 -6 0 -2 -5 2 -5 3 2 -3 2	8.000 -1 54.000