

Problem A. Radio Prize

Source file name: radio.c, radio.cpp, radio.java, radio.py
Input: Standard
Output: Standard



All boring tree-shaped lands are alike, while all exciting tree-shaped lands are exciting in their own special ways. What makes Treeland more exciting than the other tree-shaped lands are the raddest radio hosts in the local area: Root and Leaf. Every morning on FM 32.33 (repeating of course), Root and Leaf of The Full Depth Morning Show serve up the hottest celebrity gossip and traffic updates.

The region of Treeland is made of n cities, connected by $n - 1$ roads such that between every pair of cities there is exactly one simple path. The i th road connects cities u_i and v_i , and has a toll of w_i .

To reward their loyal listeners, The Full Depth Morning Show is giving away a number of travel packages! Root and Leaf will choose $n - 1$ lucky residents from the city that sends them the most fan mail. Each of those residents then gets a distinct ticket to a different city in Treeland.

Each city in Treeland has its own tax on prizes: t_i . Let $d_{u,v}$ be the sum of the tolls on each road on the only simple path from city u to v . For a trip from city u to city v , the cost of that trip is then $(t_u + t_v)d_{u,v}$.

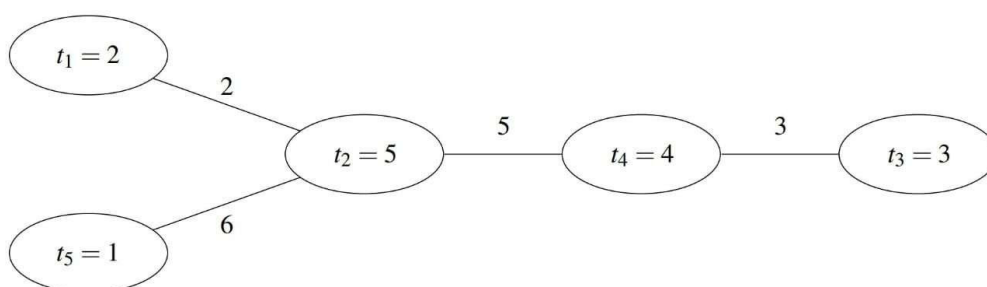


Figure A.1: The map of Treeland corresponding to the first example input.

The shock jocks haven't quite thought through how much their prize is worth. They need to prepare a report to the radio executives, to summarize the expected costs. For each city that could win the prize, what is the total cost of purchasing all the tickets?

Input

The first line of input is a single integer n ($1 \leq n \leq 10^5$). The next line has n space-separated integers t_i ($1 \leq t_i \leq 10^3$), the tax in each city. The following $n - 1$ lines each have 3 integers, u_i , v_i , w_i , meaning the i th road connects cities u_i and v_i ($1 \leq u_i, v_i \leq n$), with a toll of w_i ($1 \leq w_i \leq 10^3$).



Output

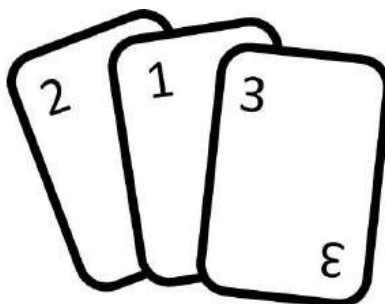
Output n lines. On the i th line, output a single integer: the cost of purchasing tickets if city i wins the contest.

Example

Input	Output
5	130
2 5 3 4 1	159
1 2 2	191
2 4 5	163
4 3 3	171
5 2 6	
6	209
4 3 3 4 3 3	206
1 3 2	232
2 1 1	209
1 4 6	336
4 5 6	232
6 4 2	

Problem B. Perfect Flush

Source file name: flush.c, flush.cpp, flush.java, flush.py
Input: Standard
Output: Standard



You are given a list of integers x_1, x_2, \dots, x_n and a number k . It is guaranteed that each i from 1 to k appears in the list at least once.

Find the lexicographically smallest subsequence of x that contains each integer from 1 to k exactly once.

Input

The first line will contain two integers n and k , with $1 \leq k \leq n \leq 2 \cdot 10^5$. The following n lines will each contain an integer x_i with $1 \leq x_i \leq k$.

Output

Write out on one line, separated by spaces, the lexicographically smallest subsequence of x that has each integer from 1 to k exactly once.

Example

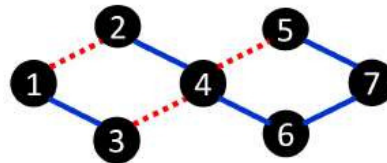
Input	Output
6 3 3 2 1 3 1 3	2 1 3
10 5 5 4 3 2 1 4 1 1 5 5	3 2 1 4 5

Problem C. Coloring Contention

Source file name: contention.c, contention.cpp, contention.java, contention.py

Input: Standard

Output: Standard



Alice and Bob are playing a game on a simple connected graph with N nodes and M edges.

Alice colors each edge in the graph red or blue.

A path is a sequence of edges where each pair of consecutive edges have a node in common. If the first edge in the pair is of a different color than the second edge, then that is a “color change.”

After Alice colors the graph, Bob chooses a path that begins at node 1 and ends at node N . He can choose any path on the graph, but he wants to minimize the number of color changes in the path. Alice wants to choose an edge coloring to maximize the number of color changes Bob must make. What is the maximum number of color changes she can force Bob to make, regardless of which path he chooses?

Input

The first line contains two integer values N and M with $2 \leq N \leq 10^5$ and $1 \leq M \leq 10^5$. The next M lines contain two integers a_i and b_i indicating an undirected edge between nodes a_i and b_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$).

All edges in the graph are unique.

Output

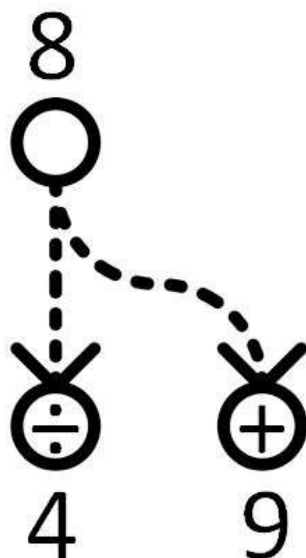
Output the maximum number of color changes Alice can force Bob to make on his route from node 1 to node N .

Example

Input	Output
3 3 1 3 1 2 2 3	0
7 8 1 2 1 3 2 4 3 4 4 5 4 6 5 7 6 7	3

Problem D. Dividing by Two

Source file name: dividing.c, dividing.cpp, dividing.java, dividing.py
 Input: Standard
 Output: Standard



You are given two integers, A and B . You want to transform A to B by performing a sequence of operations. You can only perform the following operations:

- Divide A by two, but only if A is even.
- Add one to A .

What is the minimum number of operations you need to transform A into B ?

Input

The input will be a single line containing two integers A and B with $1 \leq A, B \leq 10^9$.

Output

On a single line write the minimum number of operations required as an integer.

Example

Input	Output
103 27	4
3 8	5

Problem E. Rainbow Strings

Source file name: rainbow.c, rainbow.cpp, rainbow.java, rainbow.py
Input: Standard
Output: Standard



Define a string to be a *rainbow string* if every letter in the string is distinct. An empty string is also considered a *rainbow string*.

Given a string of lowercase letters, compute the number of different subsequences which are *rainbow strings*. Two subsequences are different if an index is included in one subsequence but not the other, even if the resulting strings are identical.

In the first example, there are 8 subsequences. The only subsequences that aren't rainbow strings are **aa** and **aab**. The remaining 6 subsequences are rainbow strings.

Input

The input will consist of a single line with a single string consisting solely of lowercase letters. The length of the string is between 1 and 10^5 (inclusive).

Output

Write on a single line the number of rainbow sequences, modulo the prime 11092019.

Example

Input	Output
aab	6
icpcprogrammingcontest	209952

Problem F. Carny Magician

Source file name: carny.c, carny.cpp, carny.java, carny.py
 Input: Standard
 Output: Standard



Charles and Ada are watching a magician shuffling a deck of thirteen numbered cards, which were originally ordered. The magician spreads the cards out on the table.

Ada exclaims, “Odd; ten of the cards are in their original locations!”

Charles thinks for a moment, and says, “Not only that, but it is the forty-second such ordering!”

Can you figure out the order of the cards? Formally, the magician’s cards can be considered as a permutation p_1, p_2, \dots, p_n , that contains each number from 1 to n exactly once. The number of fixed points is the number of indices i such that $p_i = i$.

Given three numbers n , m , and k , find the k th lexicographically smallest permutation of size n that has exactly m fixed points.

Input

The input will be a single line containing the three integers n , m , and k , with $0 \leq m \leq n$, $1 \leq n \leq 50$, and $1 \leq k \leq 10^{18}$.

Output

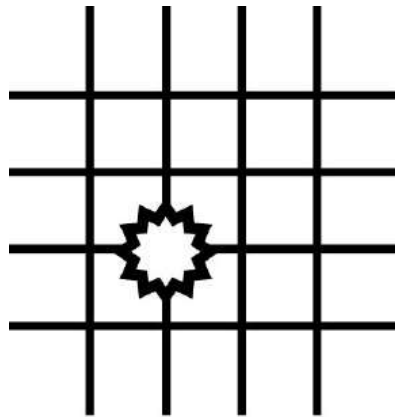
On a single line, write the permutation as a sequence of n space-separated integers. If there are fewer than k permutation satisfying the conditions, then print -1 on a single line.

Example

Input	Output
3 1 1	1 3 2
3 2 1	-1
5 3 7	2 1 3 4 5

Problem G. Glow, Little Pixel, Glow

Source file name: glow.c, glow.cpp, glow.java, glow.py
 Input: Standard
 Output: Standard



An LCD panel is composed of a grid of pixels, spaced 1 alu (“arbitrary length unit”) apart both horizontally and vertically. Wires run along each row and each column, intersecting at the pixels. Wires are numbered beginning with 1 and proceeding up to a panel-dependent maximum. The vertical wire numbered 1 lies along the left edge of the panel, and the horizontal wire numbered 1 lies along the bottom edge of the panel.

A pixel will activate, turning dark, when a current is present along both the vertical and horizontal wire passing through that pixel.

For a period of time, we will send pulses of current down selected wires. The current flows down the wires at a speed of one alu per atu (“arbitrary time unit”). The pulses themselves have a length measured in atus. A pixel activates when current is passing through both intersecting wires at the same time. If the leading edge of a pulse on one wire reaches the intersection at the exact same time that the trailing edge of a pulse on the other wire leaves that intersection, the pixel is not activated.

All pulses in vertical wires start from the bottom of the grid. All pulses in horizontal wires start from the left of the grid. At most one pulse will travel along any one wire.

Given the schedule of pulses to be sent through the wires, determine how many pixels will have been activated by the time all pulses have exited the top and right of the grid.

Input

The first line will contain n , the number of current pulses, with $1 \leq n \leq 2 \cdot 10^5$.

Following this will be n lines, each describing a single pulse. Each such line will contain four elements, separated from one another by a single space:

- A single character that is either ‘h’ or ‘v’, indicating the horizontal/vertical direction of the pulse.
- An integer t , $1 \leq t \leq 2 \cdot 10^5$, denoting the starting time of the pulse. The starting time is considered to be the moment when the leading edge of a vertical [horizontal] pulse crosses horizontal [vertical] wire #1.
- An integer m , $1 \leq m \leq 2 \cdot 10^5$, denoting the length of the pulse.
- An integer a , $1 \leq a \leq 10^5$, denoting the wire number (horizontal or vertical) along which the pulse will travel.



Output

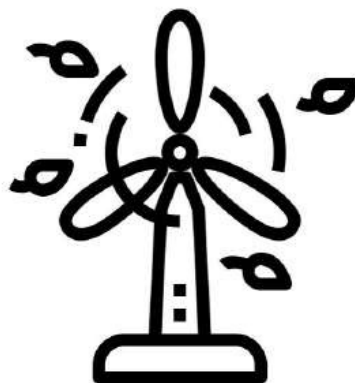
Print on a single line the number of pixels that will have activated by the time the last pulse of current has left the grid.

Example

Input	Output
4 h 1 4 1 v 2 4 2 h 10 2 2 v 11 2 3	2
4 h 1 10 1 h 5 10 2 v 1 10 1 v 5 10 3	4
7 v 1 3 1 v 1 15 2 h 4 5 1 h 5 5 2 h 6 5 3 h 7 5 4 h 8 5 5	5

Problem H. Pivoting Points

Source file name: pivoting.c, pivoting.cpp, pivoting.java, pivoting.py
Input: Standard
Output: Standard



Consider a set of points P in the plane such that no 3 points are collinear. We construct a “windmill” as follows:

Choose a point p in P and a starting direction such that the line through p in that direction does not intersect any other points in P . Draw that line.

Slowly rotate the line clockwise like a windmill about the point p as its pivot until the line intersects another point p' in P . Designate that point p' to be the new pivot (call this “promoting” the point p'), and then continue the rotation.

Continue this process until the line has rotated a full 360 degrees, returning to its original direction (it can be shown that the line will also return to its original position after a 360 degree rotation).

During this process, a given point can be promoted multiple times. Considering all possible starting pivots and orientations, find the maximum number of times that a single point can be promoted during a single 360 degree rotation of a line.

Input

The first line of the input will be a single integer n with $2 \leq n \leq 2000$. Following this will be n lines, each with two integers x_i and y_i with $-10^4 \leq x_i, y_i \leq 10^4$.

Output

On one line, write an integer with the largest number of times any particular point can be a pivot when an arbitrary starting line does a full rotation as described above.

**Example**

Input	Output
3 -1 0 1 0 0 2	2
6 0 0 5 0 0 5 5 5 1 2 4 2	3

Problem I. Error Correction

Source file name: correction.c, correction.cpp, correction.java, correction.py
Input: Standard
Output: Standard



You are given W , a set of N words that are anagrams of each other. There are no duplicate letters in any word. A set of words $S \subseteq W$ is called “swap-free” if there is no way to turn a word $x \in S$ into another word $y \in S$ by swapping only a single pair of (not necessarily adjacent) letters in x . Find the size of the largest swap-free set S chosen from the given set W .

Input

The first line of input contains an integer N ($1 \leq N \leq 500$). Following that are N lines each with a single word. Every word contains only lowercase English letters and no duplicate letters. All N words are unique, have at least one letter, and every word is an anagram of every other word.

Output

Output the size of the largest swap-free set.



Example

Input	Output
6 abc acb cab cba bac bca	3
11 alerts alters artels estral laster ratels salter slater staler stelar talers	8
6 ates east eats etas sate teas	4

Problem J. Interstellar Travel

Source file name: interstellar.c, interstellar.cpp, interstellar.java, interstellar.py
Input: Standard
Output: Standard



You are planning to travel in interstellar space in the hope of finding habitable planets. You have already identified N stars that can recharge your spaceship via its solar panels. The only work left is to decide the orientation of the spaceship that maximizes the distance it can travel.

Space is modeled as a 2D plane, with the Earth at the origin. The spaceship can be launched from the Earth in a straight line, in any direction. Star i can provide enough energy to travel T_i distance if the spaceship is launched at an angle of a_i with the x -axis. If the angle is not perfectly aligned, then the spaceship gets less energy. Specifically, if the launch direction makes an angle of a with the x -axis, then it gets enough energy to travel distance of

$$\max(0, T_i - s_i \cdot \text{dist}(a_i, a))$$

from star i , where $\text{dist}(a, b)$ is the minimum radians needed to go from angle a to b . The distance that the spaceship can travel is simply the sum of the distances that each star contributes. Find the maximum distance T that the starship can travel.

Input

The first line contains the value N , $1 \leq N \leq 10^5$. Following this are N lines each containing three real numbers T_i , s_i , and a_i , with $0 < T_i \leq 1000$, $0 \leq s_i \leq 100$, and $0 \leq a_i < 2\pi$. All real numbers in the input have at most 6 digits after the decimal point.

Output

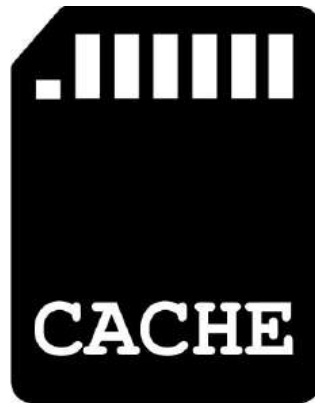
On a single line output the maximum distance the spacecraft can travel. Your answer is considered correct if it has an absolute error of at most 10^{-6} .

**Example**

Input	Output
2 100 1 1 100 1 1.5	199.500000
4 100 1 0.5 200 1 1 100 0.5 1.5 10 2 3	405.500000

Problem K. Computer Cache

Source file name: computer.c, computer.cpp, computer.java, computer.py
 Input: Standard
 Output: Standard



Your computer has a cache consisting of n different addresses, indexed from 1 to n . Each address can contain a single byte. The i^{th} byte is denoted as a_i . Initially all cache bytes start off with the value zero. Formally, the cache can be modeled by a byte array of length n that is initially all zeros.

You have m different pieces of data you want to store. The i^{th} piece of data is a byte array x_i of length s_i .

You are going to do q different operations on your computer. There are three types of operations:

- 1 i p** Load data i starting at position p in the cache. Formally, this means set $a_p = x_{i,1}$, $a_{p+1} = x_{i,2}, \dots$, $a_{p+s_i-1} = x_{i,s_i}$, where $x_{i,k}$ represents the k th byte of the array x_i . This overwrites any previously stored value in the cache. It is guaranteed that this is a valid operation (e.g. $s_i + p - 1 \leq n$). It is possible for multiple versions of some data to be loaded in multiple positions at once.
- 2 p** Print the byte that is stored in address p
- 3 i l r** Increment the l^{th} through r^{th} bytes in the i^{th} piece of data, modulo 256. Formally, this means to set $x_{i,k} = (x_{i,k} + 1) \bmod 256$ for $l \leq k \leq r$. This does not affect values that are already loaded in the cache and only affects future loads.

Input

The first line of input consists of three numbers n , m , and q .

The following m lines consist of descriptions of the data, one per line. The following q lines consist of descriptions of operations, one per line.

It is guaranteed there is at least one type 2 print query operation in the input. Additionally:

$$1 \leq n, m, q \leq 5 \cdot 10^5$$

$$\sum_i s_i \leq 5 \cdot 10^5$$

$$s_i \geq 1$$

$$0 \leq x_{i,j} \leq 255$$



Output

Your program must output the results for each type 2 operation, one integer value per line.

Example

Input	Output
5 2 10	0
3 255 0 15	255
4 1 2 1 3	1
2 1	255
1 2 2	0
1 1 1	3
2 1	
2 4	
3 1 1 2	
2 1	
1 1 2	
2 2	
2 5	

Explanation

- 2 1 Nothing has been put into the cache, so print 0
- 1 2 2 The cache is now [0, 1, 2, 1, 3]
- 1 1 1 The cache is now [255, 0, 15, 1, 3]
- 2 1 Print the first value of the cache which is 255
- 2 4 Print the fourth value of the cache which is 1
- 3 1 1 2 The first piece of data becomes [0, 1, 15]. The cache is still [255, 0, 15, 1, 3]
- 2 1 Print the first value of the cache which is 255.
- 1 1 2 The cache becomes [255, 0, 1, 15, 3].
- 2 2 Print the second value of the cache which is 0.
- 2 5 Print the fifth value of the cache which is 3.

Problem L. Carry Cam Failure

Source file name: carry.c, carry.cpp, carry.java, carry.py
 Input: Standard
 Output: Standard



**A Cam from a
Babbage Analytical Engine**

“Drat!” cursed Charles. “This stupid carry bar is not working in my Engine! I just tried to calculate the square of a number, but it’s wrong; all of the carries are lost.”

“Hmm,” mused Ada, “arithmetic without carries! I wonder if I can figure out what your original input was, based on the result I see on the Engine.”

Carryless addition, denoted by \oplus , is the same as normal addition, except any carries are ignored (in base 10). Thus, $37 \oplus 48$ is 75, not 85.

Carryless multiplication, denoted by \otimes , is performed using the schoolboy algorithm for multiplication, column by column, but the intermediate additions are calculated using *carryless addition*. More formally, Let $a_m a_{m-1} \dots a_1 a_0$ be the digits of a , where a_0 is its least significant digit. Similarly define $b_n b_{n-1} \dots b_1 b_0$ be the digits of b . The digits of $c = a \otimes b$ are given by the following equation:

$$c_k = a_0 b_k \oplus a_1 b_{k-1} \oplus \dots \oplus a_{k-1} b_1 \oplus a_k b_0$$

where any a_i or b_j is considered zero if $i > m$ or $j > n$. For example, $9 \otimes 1234$ is 9876, $90 \otimes 1234$ is 98760, and $99 \otimes 1234$ is 97536.

Given N , find the smallest positive integer a such that $a \otimes a = N$.

Input

The input consists of a single line with an integer N , with at most 25 digits and no leading zeros.

Output

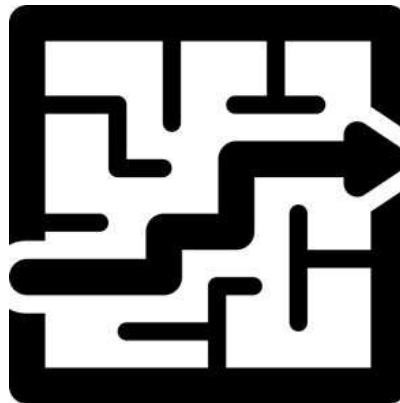
Print, on a single line, the least positive number a such that $a \otimes a = N$. If there is no such a , print ‘-1’ instead.

Example

Input	Output
6	4
149	17
123476544	11112
15	-1

Problem M. Maze Connect

Source file name: maze.c, maze.cpp, maze.java, maze.py
Input: Standard
Output: Standard



Given an orthogonal maze rotated 45 degrees and drawn with forward and backward slash characters (see below), determine the minimum number of walls that need to be removed to ensure it is possible to escape outside of the maze from every square of the (possibly disconnected) maze.

```

/\
\/

```

This maze has only a single square fully enclosed. Removing any wall will connect it to the outside.

```

/\..
\.\.
.\/\
..\\/

```

This maze has two enclosed areas. Two walls need to be removed to connect all squares to the outside.

Input

The first line has two numbers, R and C , giving the number of rows and columns in the maze's input description. Following this will be R lines each with C characters, consisting only of the characters '/', '\', and '.'. Both R and C are in the range $1 \dots 1000$.

Define an odd (even) square as one where the sum of the x and y coordinates is odd (even). Either all forward slashes will be in the odd squares and all backslashes in the even squares, or vice versa.

Output

Output on a single line an integer indicating how many walls need to be removed so escape is possible from every square in the maze.



Example

Input	Output
2 2 /\n \\	1
4 4 /\n. \\. .\n\ ..\\	2
2 2 \\n /\n	0
8 20 /\n/\n/\n/\n/\n/\n/\n/\n \\../\n.\n.\n.\n.\n/\n/\n/\n /.\n.\n.\n.\n/\n.\n.\n.\n/\n \\n/\n.\n.\n.\n.\n/\n.\n.\n.\n/\n /\n.\n/\n/\n.\n.\n.\n.\n.\n/\n \\.\n.\n.\n.\n/\n.\n.\n.\n/\n /...\n/\n.\n.\n.\n.\n.\n/\n \\n/\n/\n/\n/\n/\n/\n/\n/\n	26