Problem A. Archery

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

This is an interactive problem

Recently, a new variant of archery shooting competition was proposed — shooting with an unknown target.

The playfield consists of N targets arranged in a circle. The competition consists of Q rounds.

Before the start of the first round, the shooter can connect some pairs of targets with wires, while no two targets can be directly connected by more than one wire, and no target can be connected to itself. The connection remains unchanged during all rounds.

At the beginning of each round, one of the targets is secretly assigned as the main target, and the others are assigned as fake. The shooter does not know which target is selected as main. The shooter's task is to hit the main target in no more than 10 shots. If the shooter hits the main target, the scoreboard above the stand lights up green, and the round ends. If the shooter hits an fake target that is directly connected to the main target by a wire, the scoreboard lights up yellow. Otherwise, the scoreboard lights up red.

Your task is to write a program that connects the targets so that the target can be hit in no more than 10 shots in each of the Q rounds.

Interaction Protocol

The interaction starts with the jury program printing two integers N and Q — the number of targets and rounds, respectively ($1 \le N, Q \le 200$).

Then your program prints information about the targets connected by wires. First, print a line with a single integer K — the number of wires. Then print K lines, each containing two integers from 1 to N — the numbers of the targets connected by the corresponding wire. Two targets cannot be directly connected by more than one wire, and a target cannot be connected to itself.

After that, your program makes a shot, printing an integer from 1 to N — the number of the next target. The jury program responds with "`green`" if the shot hit the main target (after which you make the first shot of the next round or end the program if it was the last round), "`yellow`" if the shot hit a target directly connected to the main target, and "`red`" otherwise. After any of these two responses, your program continues shooting, choosing the next target. If more than 10 attempts are made in any round, the solution is considered incorrect.

Example

standard input	standard output
3 2 green yellow red green	 2 1 2 1 3 1 1 2 3

Problem B. Backing Up The Password

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

The Gnusmas company has proposed a new way to recall a pin code consisting of four digits. A randomly generated formula is displayed on the screen, in which the digits of the pin code are replaced with `a`, `b`, `c`, and `d`, as well as the value of this formula when the correct values of a, b, c, and d are substituted.

It is assumed that the user will recall the pin code after this. But in some cases, the formula may simply determine the answer unambiguously, which is unsafe - even someone who did not know the password will gain access. Or the formula will always be true - for example, a+b+c+d-a-b-c-d is always equal to zero, regardless of the pin code.

You are an intern at Gnusmas. To evaluate the quality of the hints, the system developers asked you to write a program that, given a hint, outputs the number of pin codes that satisfy it.

Input

The first line of the input contains a formula that the user entered to remember the pin code.

The formula may consist of:

- Variables `a`, `b`, `c`, and `d`, denoting the first, second, third, and fourth digits, respectively.
- Round brackets '`(`' and '`)`'.
- Integers from 1 to 10^4.
- Arithmetic operators '`+`', '`-`', '`*`'.

There are no spaces in the formula.

If several variables are written in a row without signs between them, this denotes a decimal number from the corresponding number of digits. For example, with $a = 0$, $b = 1$, and $c = 2$, *abcab* denotes the number 01201 (or simply 1201, since leading zeros are ignored). This notation is called a *block*. It is guaranteed that the length of each block does not exceed 4. Otherwise, the formula complies with the rules of arithmetic (in particular, in the absence of brackets, multiplication takes precedence over addition and subtraction). Unary minus is absent.

A more formal description of the formula construction rule:

```
<number> :: an integer from 0 to 10000 without leading zeros
<var> :: a | b | c | d
<block> :: <var> | <var><var> | <var><var><var> | <var><var><var><var>
<op> :: + | - | *
<token> :: <block> | <number>
<formula> :: <token> | <sequence> <op> <sequence> | (<sequence>)
```

The formula is not empty, its length does not exceed 200. The formula contains no more than two multiplication signs.

The second line contains an integer N ($1 \le N \le 10^{16}$) - the value of the formula.

Output

Output the number of pin code variants for which the formula value S is equal to the number N. If there is only one variant, then output the pin code itself on the second line.

Examples

standard input	standard output
a+b+c+d 36	1 9999
a+b+c+d 5	56
b+c+d 27	10
abcd+abcd 1024	1 0512
1 2	0
abcd+1083 6006	1 4923

Problem C. Count The Repetitions

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	512 megabytes

Your task is to count the number of repetitions in the entered text. In the context of this task, a *repetition* is defined as a substring of a given string that can be represented as a concatenation of two identical strings. Equal substrings starting from different positions in the string are considered different. For example, in the text `hrhrhrhr` there is one repetition of length 8 (the entire string is a repetition: `hrhr`+`hrhr`), as well as 5 repetitions of length 4 (the substring `hrhr` at positions 1, 3, 5 and the substring `rhrh` at positions 2 and 4), for a total of 6 repetitions.

Count the number of repetitions in the given string.

Input

The input contains a single string s, consisting of lowercase Latin letters. The length of the string is at least 3 and at most 10^5 characters.

Output

Output a single integer — the number of repetitions in the given string.

Examples

standard input	standard output
hrhrhrhr	6
hhhhhhh	12
voronezh	0

Problem D. Distinct Sums

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Given an integer N. Your task is to build the integer sequence a_i of length M such that:

- $1 \le a_i \le N$.
- $a_i \ne a_j$ if $i \ne j$.
- For any $i \ne j$ and $k \ne l$ $a_i + a_j \ne a_k + a_l$ is held, if $(i, j) \ne (k, l)$ and $(i, j) \ne (l, k)$ (i.e. all $M(M-1)/2$ pairwise sums are distinct).
- $M \ge \sqrt{N}/2$.

Input

The input contains one integer N ($1 \le N \le 5 \cdot 10^6$).

Output

In the first line print one integer M – the length of the sequence a_i. In the second line print M pairwise distinct integers between 1 and N, inclusively — the sequence a_i.

If there are more than one solution, any of them will be accepted.

Example

standard input	standard output
10	3 1 3 7

Problem E. Eligibility Test

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

In the Byteland Library was found the ancient book with the stories and legends about the semi-mythical wise queen Bytica and three her children. One of legends says that the Bytica's kingdom has a form of trapezoid of nonzero area with sides a, b, c and d. When the children grow, Bytica decided to split the kingdom to three triangular provinces to look at the children administrative skills as the governors and decide who is eligible to be the next queen or king.

To avoid the conflicts around size and form of the provinces, wise queen decided, that all parts shall be **equal** as the triangles, so two lines on the map was drawn and the children get their quests.

You want to check, if this legend can be true. Of course, you cant go to the archeological expedition during the contest, but you can check, if the situation is possible with the given values of a, b, c and d.

Input

Input contains four integers a, b, c and d, each integer on the new line ($1 \le a, b, c, d \le 1000$).

Output

If the trapezoid with the given length of the sides exists, have non-zero area and the queen can divide the kingdom to 3 equal triangles, print 1. Otherwise print 0.

Examples

standard input	standard output
6 2 3 4	1
1 2 4 8	0

Problem F. Fishing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	512 megabytes

Stef, Klay, Draymond and their friends are fishing now.

They choose $N-1$ meter long river where N fish tanks are placed, one fish tank at each integer coordinate. Some tanks contain white fish, some tanks contain black fish, other tanks are empty.

The fishing is too boring, so Draymond found a funny trick. He can throw (from short distance, of course) the stone into river at any position (either at the some tank or between the tanks), and the wave goes at the same speed in both directions. When the wave reaches the tank with the fish, the fish in the tank jumps out the water for a moment and then falls back in the tank.

All fish have the same reaction time, that is, fish at the same distance from the place where the stone was thrown will jump simultaneously.

Stef and Klay especially likes when both a black and a white fish are jumping simultaneously and call it a combo.

Draymond wants to maximize the number of combos. He can choose the place for the stone and select the color of fish to be placed for the every empty tank (at the moment when stone will be thrown, all tanks shall contain fish). Draymond got W specimens of white fish and B specimens of the black fish to decide among them.

Input

The first line contains three integers N, W, and B ($2 \le N \le 50,000$, $0 \le W, B \le 50,000$), where N is the number of fish tanks, W and B are the number of extra white and black fish respectively.

The second line contains N characters each of which is one of '`W`', '`B`', or '`?`', representing how the fish tanks are originally populated. Here, 'W' means that the tank is occupied by the white fish, 'B' means it is black, and '?' means that you need to place a fish at that position.

The number of positions where Draymond needs to place a fish does not exceed $W + B$.

Output

Print one integer — the maximum number of combos the fishermen can see after Draymond populates all the tanks and throws the stone in the river.

Example

standard input	standard output
7 2 1 BBW?W??	3

Problem G. Great Upgrade

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Your company is planning the great upgrade, so the management is planning to buy new servers.

At the Server Expo, there are N servers presented. Each server costs p_i kilofranklins and have the economic value V_i. There is only one server of each type at the Server Expo.

The CEO is planning use up to K kilofranklins for the upgrade. To decide the final version of the funding, he wants for each positive integer $i \le K$ know the maximal economic value that your company can have while spending no more than i kilofranklins to buy the servers from the Server Expo.

Input

The first line of the input contains two integers N and K — number of the servers on the Server Expo and maximal funding in kilofranklins, respectively ($1 \le N \le 10^6$, $1 \le K \le 5 \cdot 10^4$). Each of the following N lines contains two integers p_i and V_i — the price of the server and its value ($0 \le p_i \le 300$, $0 \le V_i \le 10^9$).

Output

Print K integers in one line. i-th of those integers shall denote the maximal value the company can get when the funding for the upgrade is i kilofranklins.

Example

standard input	standard output
5 10 4 2 1 49 4 25 2 73 3 60	49 73 122 122 133 182 182 182 182 207

Problem H. Hypermagic Squares

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Consider the integer matrix $n \times n$ as *magic square*, if it contains pairwise distinct positive integer and the sum of the integers in each horizontal, each vertical and each of two big diagonals is the same.

Consider the matrix $2k+1 \times 2k+1$ *hypermagic square* if:

- It contains all ingeters between 1 and $(2k+1)^2$.
- It is the magic square.
- Any submartix built of integers that have the distance from the martix boundary t or less $(1 \le t \le k-1)$, is the magic square (you may represent those submatrices as the matrices with the same center and odd length of side).

Given the odd integer n, find any hypermagic square $n \times n$.

Input

Input contains one integer n ($5 \le n \le 999$, n is odd).

Output

Print n lines, each consisting n integers — the hypermagic square. If there are more than one solutions, print any of them.

Example

standard input	standard output
5	11 10 21 20 3 4 12 19 8 22 2 9 13 17 24 25 18 7 14 1 23 16 5 6 15

Problem I. Integer Balls

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

...After losing the national championship, the soccer team «Double» decided to hire a famous coach.

However, his pedantry was a surprise even for the club management: seeing several balls with slightly different radii, the coach declared that he would only conduct training with balls of the same radius. As an experienced programmer, you were hired to select the necessary number of balls.

From the aside all balls are completely identical, differing only in their radii. Using a special robot, you have determined four points with integer coordinates on the surface of each ball. Now your task is to select the maximum number of balls with the same radius so that the new coach can conduct training.

Input

The first line contains a single integer N ($1 \le N \le 20$) — the number of balls available to the team. Each of the following N lines contains the coordinates of four points on the surface of each of the N balls in the order $x_1\ y_1\ z_1\ x_2\ y_2\ z_2\ x_3\ y_3\ z_3\ x_4\ y_4\ z_4$ (that is, the coordinates of the first point come first, then the second, then the third, and then the fourth). All $x_i\ y_i\ z_i$ are integers, not less than zero and not greater than 170. It is guaranteed that the tetrahedron with the given four points as vertices has a non-zero volume.

Output

Output a single integer - the maximum number of balls with the same radius that the team can provide to the coach for training.

Example

standard input	standard output
4 1 0 1 2 1 1 1 1 2 1 2 1 3 2 2 2 2 3 2 3 2 2 1 2 6 4 4 4 4 6 4 6 4 4 2 4 2 2 1 1 2 2 2 1 2 0 1 2	2

Problem J. Judging the Mafia game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

This is an interactive problem.

n players are playing «Mafia» game. They are currently voting to eliminate one player from the game. Each player has voted for one of the players (possibly themselves). You do not know who voted for whom, only the game master (in this case, the jury) knows. However, you must determine, in no more than three questions, whether there is a player who has received strictly more than half of the votes.

You can ask the game master about each player, whether it is true that they voted the same way as some other player. You can ask the game master for a sequence of n numbers $a_1, a_2, \ldots a_n$. In response, you will receive a string s, in which the i-th character is equal to one if player i and player a_i voted for the same player, and is equal to zero otherwise.

Interaction Protocol

The first line contains the number T ($1 \le T \le 10^4$) - the number of votes whose results you need to determine.

Each vote begins with an integer n ($1 \le n \le 10^5$). Then you can either ask a question or report an answer. To ask a question, you must output the string "? a_1 a_2 $\ldots$ a_n where $1 \le a_i \le n$. In response, you will receive a string of n characters, in which the i-th character is equal to one if player i and player a_i voted for the same player, and is equal to zero otherwise. Do not forget to flush the output buffer when making a request. The number of questions asked should not exceed 3.

To report the voting result, you must output the string "! X where X is the number of one of the players who voted for the player who received strictly more than half of the votes. That is, there are more than half of the players who voted for the same player as player X, including himself. If there is no such player, X should be equal to -1. After that, flush the output buffer and move on to the next vote, if there is one. The report is not counted as the question.

The sum of n for all votes in one test does not exceed 10^5. It is guaranteed that all votes are fixed at the time the program is launched, i.e. the interactor is not adaptive.

Example

standard input	standard output
2 5 11000 11010 4 1010	 ? 2 1 4 5 3 ? 1 1 1 1 1 ! 4 ? 2 3 4 1 ! -1

Problem K. King of Cabbages

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

In the economics simulator game "King of Cabbages" the player buys factories with the goal of collecting as many cabbages as possible in the shortest time.

There are a total of N different types of factories in the game.

The number of cabbages f_0 increases every second first by C times, and then the number of cabbages produced by the factories of the type 1 (basic) is added. A factory of the first type produces K_1 cabbages every second, so if at time t you have f_0 cabbages and f_1 factories of the type 1, at time $t+1$ you will have $C \cdot f_0 + K_1 \cdot f_1$ cabbages.

The factories of type $i > 1$, however, are producing not the cabbages directly. They are producing factories!

The number of factories of type i is also increased every second first by C times, and then the number of factories of type i produced by factory of type $i+1$ is added. Factories of type $i+1$ (where i>0) every second produce $(i+1) \cdot K_{i+1}$ factories of type i, so if at time t you have f_i factories of type i and f_{i+1} factories of type $i+1$ at time $t+1$ you will have $C \cdot f_i + (i+1) \cdot K_{i+1} \cdot f_{i+1}$ factories of type i.

Given the amount of cabbages (f_0) and factories of each type at the time $t = 0$ and the values of C and K_i, determine the total number of cabbages at time $t = T$ modulo 998 244 353.

Input

The first line of the input contains three integers N ($1 \le N \le 2 \cdot 10^5$) — the number of the production types (i.e. cabbages and $N-1$ factory), the target time moment T ($1 \le T \le 10^9$) and the constant C ($1 \le C \le 10^9$).

The second line contains N integers — the initial values of f_0, f_1, …, f_{N-1}, in order ($0 \le f_i \le 10^9$). The third line contains N integers K_0, K_1, …, K_{N-1}, ($0 \le K_i \le 10^9$).

Output

Print one integer — the number of the cabbages after second T modulo 998 244 353.

Example

standard input	standard output
3 3 2 1 1 1 1 1 1	32

Problem L. Lake Fishless

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

The map of Lake Fishless with its shores is represented by a rectangle $R \times C$.

Each cell of this rectangle is either a water area or a land area (with all the boundary cells of the rectangle being land areas). All land areas from which it is possible to reach the boundary of the rectangle only by traversing land cells (moving each time to a cell that shares a side with the current cell) form the *shore* of the lake. Land areas that do not belong to the boundary form islands.

For each water area, there is a defined direction of water flow — up, down, right, or left.

As the name suggests, initially there are no fish in Lake Fishless. Ecologists have decided to populate Lake Fishless with a population of fish. In one action, fish can be put to any cell that is part of the shore, then the fish populate any cell that contains water and shares a side with the starting cell. After the release, the fish populate the cells according to the direction of water flow, but no more than the first L cells (including the starting cell). If the flow leads to land or to a cell already populated by the fish, the process stops. If the flow leads to a cell where fish are not yet present, that cell is populated with the fish.

What is the minimum number of releases required to populate Lake Fishless entirely (i.e., all its water areas)?

Input

The first line of the input contains three integers R, C, and L - the number of rows and columns on the map of the lake, and the maximum distance that fish can swim, respectively ($3 \le R, C \le 2000$, $1 \le L \le 10^9$).

The following R lines describe the flow map of Lake Fishless. Each of these lines has a length of C.

The map consists of the characters '`#`', '`L`', '`R`', '`U`', or '`D`', representing land, water flow direction to the left, right, up, and down, respectively. The characters are listed in natural order (rows from top to bottom, columns from left to right).

It is guaranteed that the first and last rows, as well as the first and last columns, consist entirely of the character '`#`'.

Output

Output the minimum number of releases of fish into the lake in order to have the fish present in every area of the lake, or -1 if it is not possible to achieve this with any number of releases.

Examples

standard input	standard output
3 11 3 ########### #RRRRLL#DL# ###########	3
6 6 3 ###### ##DLD# #DL#D# #D#DL# #RLL## ######	5
17 14 5 ############### ############DU# ########DLLRU# ########RDUL## #######RURUU## ######RUDLLL## #####RRDDDLLL# ####RDRDRDRLL# ####DRURULLL## ###DDRLLLLL### ###DRU#LDL#### ###RD##URL#### ##RDRRRUL##### ##URRRLLL##### #RRRDULL###### #ULLRRRR###### ###############	22
17 14 1000000000 ############### ############DU# ########DLLRU# ########RDUL## #######RURUU## ######RUDLLL## #####RRDDDLLL# ####RDRDRDRLL# ####DRURULLL## ###DDRLLLLL### ###DRU#LDL#### ###RD##URL#### ##RDRRRUL##### ##URRRLLL##### #RRRDULL###### #ULLRRRR###### ###############	18
5 7 2 ####### ##RRDD# #D#DLD# #RRRLL# #######	-1