

Problem A. Scanner

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

The boss instructed Vitya to scan a large number of documents.

The scanner scans one document in k minutes. Vitya is busy with his main work, so he decides to approach the scanner every t minutes. If the scanner has not finished scanning a document by that time, Vitya leaves; otherwise, he starts the scanning of the next document. Vitya will be scanning documents throughout the entire workday that lasts T minutes.

How many documents can Vitya manage to start scanning?

Input

The first line contains three integers t, k, T — the interval at which Vitya approaches the scanner, the time it takes to scan one document, and the time until the end of the workday, respectively ($1 \leq t, k, T \leq 100$).

Output

Output a single integer — the number of scanned documents.

Examples

standard input	standard output
4 5 17	3
5 5 20	5

Note

If the scanning starts at time L , then the next scanning can be started at time $L + k$ or later.

If the time of the next approach to the scanner coincides with the end of the workday, Vitya first approaches the scanner and then goes home.

Problem B. Patrol

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

On a street represented as a straight line, a police officer patrols a section located between points with coordinates A and B .

The patrolling is carried out under the following rules:

1. At time 0, the police officer is at point A .
2. As soon as the police officer arrives at point A , he starts moving towards point B at a speed of 1 m/s.
3. As soon as the police officer arrives at point B , he starts moving towards point A at a speed of 1 m/s.
4. The change of direction is instantaneous.

At time T , the police officer receives a radio command to stop patrolling and proceed to point C , which is also located on the patrolled street. He immediately stops patrolling and begins to move towards point C at a speed of 1 m/s.

Input

In the first line, the coordinates of the patrolled section A and B are given ($0 \leq A, B \leq 10^5$, $A \neq B$).

In the second line, the moment of receiving the command T ($0 \leq T \leq 10^9$) and the coordinate of point C ($0 \leq C \leq 10^5$) are given.

All numbers in the input are integers. All coordinates are given in meters.

Output

Output the sought number of seconds from the moment of receiving the call to the moment the police officer arrives at point C .

Examples

standard input	standard output
3 8 10 1	2
6 1 7 9	6

Problem C. File Manager

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Vitya is busy at work writing a new file manager called “RAF”. Today, there is supposed to be a demonstration of the alpha version, but currently, “RAF” cannot open directories that contain more than K elements (files or subdirectories).

On the computer where the demonstration will take place, there are currently N files, distributed across various directories. If “RAF” cannot open any directory during the demonstration, the clients will be dissatisfied with Vitya’s work. Therefore, Vitya wants to keep the maximum number of files from those available on the computer while ensuring that the constraint on the number of elements in each directory is met.

To achieve his goal, Vitya may delete some files. If, after a deletion, a directory is left with no elements, that directory is automatically deleted.

Help Vitya and tell him which files he should keep.

Input

The first line contains two integers N and K — the number of files on the computer and the maximum allowed number of elements in a directory ($1 \leq K \leq 100$, $1 \leq N$).

Each of the following N lines contains the full path to a file.

File and directory names consist of lowercase and uppercase Latin letters and the underscore character (“_”, ASCII 95). There are no two elements with the same name in a directory. There are no files or directories with empty names. The path separator is a slash (“/”, ASCII 47).

The total length of the file paths does not exceed 10^6 .

Output

On the first line, output the number of files M that are kept. In each of the next M lines, print the full paths of the kept files. If there are multiple solutions, any of them can be printed.

Example

standard input	standard output
5 2 java/util/List java/time/Instant java/util/stream/Collectors java/util/stream/IntStream java/util/Queue	4 java/util/stream/Collectors java/util/stream/IntStream java/util/Queue java/time/Instant

Problem D. Mosquitoes

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

There is a building with N floors, each having M windows. If a window at position (i, j) has a brightness $F_{i,j}$, then $F_{i,j}$ mosquitoes are attracted to it. Additionally, some mosquitoes attracted by the light of window (i, j) will fly to other windows. If a window has coordinates (a, b) , then the number of mosquitoes that will attempt to settle on it is given by: $\lfloor F_{i,j} / (2^{\max(|a-i|, |b-j|)}) \rfloor$ (the fractional part is discarded).

It is known that no more than S mosquitoes can settle on a single window, and all mosquitoes that cannot find space will fall to the ground.

Given the brightness of each window, calculate the total number of mosquitoes that settle on all the windows of the building, as well as how many fall to the ground.

Input

In the first line, the number of floors N ($1 \leq N \leq 500$), the number of windows on each floor M ($1 \leq M \leq 500$), and the maximum number of mosquitoes that can fit on one window S ($1 \leq S \leq 10^9$) are given.

Each of the following N lines contains M integers representing the brightness of the windows $F_{i,j}$ ($0 \leq F_{i,j} \leq 500$).

Output

Output two integers: the total number of mosquitoes that successfully settled on the windows and the number of mosquitoes that fell to the ground.

Examples

standard input	standard output
7 7 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	94 3
4 5 6 9 0 0 0 0 1 2 0 0 0 0 0 0 0 2 0 0 4 0 0	73 6

Note

In the first test case, the number of mosquitoes on the windows will be as follows:

```
1 1 1 1 1 1 1
1 2 2 2 2 2 1
1 2 4 4 4 2 1
1 2 4 6 4 2 1
1 2 4 4 4 2 1
1 2 2 2 2 2 1
1 1 1 1 1 1 1
```

In the second test case:

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

Problem E. Net

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

You are given a convex polyhedron. Find its net.

A net of a polyhedron is an arrangement of non-overlapping edge-joined polygons in the plane which can be folded (along edges) to become the faces of the polyhedron.

Input

The first line contains a single integer N — the number of vertices of the polyhedron ($4 \leq N$).

In each of the following N lines, there are three integers x_i, y_i, z_i — the coordinates of the vertices of the polyhedron ($-100 \leq x_i, y_i, z_i \leq 100$).

It is guaranteed that no four points lie on the same plane. It is guaranteed that the number of faces of the polyhedron does not exceed 20. It is guaranteed that a net exists.

Output

In the first line, output the number of vertices in the net M .

In each of the following M lines, output three numbers — the index of the corresponding vertex of the polyhedron (numbered from 1) and the coordinates in the order of traversal. The coordinates must be printed with at least 12 decimal places, and they must not exceed 10^6 in absolute value.

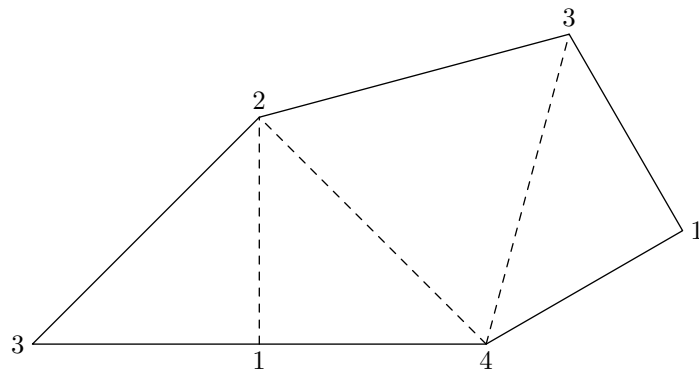
If multiple valid outputs exist, you may print any of them. A mirrored net may also be an acceptable output.

Examples

standard input	standard output
4 0 0 0 1 0 0 0 1 0 0 0 1	6 3 0.0000000000000000 0.0000000000000000 1 1.0000000000000000 0.0000000000000000 4 2.0000000000000000 0.0000000000000000 1 2.866025403784438 0.5000000000000000 3 2.366025403784438 1.366025403784438 2 1.0000000000000000 1.0000000000000000
5 0 1 0 2 0 0 0 0 3 3 4 5 4 5 3	8 1 0.0000000000000000 0.0000000000000000 4 -6.5269924715350 -0.6311650152892 5 -6.1481704595758 1.7888543819998 4 -6.1636910319861 4.2382949531981 2 0.0000000000000000 2.2360679774998 4 5.8022851187880 5.1228460085045 3 3.1304951684997 0.4472135955000 4 4.5129010182472 -4.7574913977330

Note

The net from the first example is illustrated in a picture below.



Problem F. Equity

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Given a sequence A consisting of N integers.

In one operation, you can select a contiguous subsequence $A_i A_{i+1} \dots A_j$ ($i \leq j$) that consists of the same element and increase all its elements by 1. Your task is to equalize all elements of the sequence with the minimum number of operations.

Input

The first line contains the length of the sequence N ($1 \leq N \leq 10^5$).

The second line contains the elements of the sequence A ($0 \leq A_i \leq 10^9$).

Output

Output the minimum number of operations required to equalize all elements of the sequence.

Example

standard input	standard output
4 3 1 2 4	3

Note

Consider the example for the problem:

In the first operation, we increase the second element, resulting in $(3; 2; 2; 4)$. In the second operation, we increase the elements from the second to the third, resulting in $(3; 3; 3; 4)$. In the third operation, we increase the elements from the first to the third, resulting in $(4; 4; 4; 4)$.

Problem G. Robot Vacuum Cleaner

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

The room has a rectangular shape; the floor of the room is tiled with N rows of M square tiles in each row.

Before cleaning, a number P , consisting of exactly four distinct digits: 1 (“north” $(r, c) \rightarrow (r - 1, c)$), 2 (“east” $(r, c) \rightarrow (r, c + 1)$), 3 (“south” $(r, c) \rightarrow (r + 1, c)$), and 4 (“west” $(r, c) \rightarrow (r, c - 1)$) must be entered into the memory of the robot vacuum cleaner. The order of the digits in the number P describes the sequence of checking neighboring tiles during cleaning (the changes in row and column numbers are provided in parentheses).

At the start of the cleaning process, the robot is placed on the 1st tile in the 1st row and turned on. The robot always cleans the tile it is currently on, after which it remembers that it has already cleaned that tile. Next, according to the number P , it checks the neighboring tiles until it finds an uncleaned one to which it can move. If, instead of a tile, the robot encounters a wall or a door, it considers that tile to be cleaned. After each check, regardless of the result, the number P shifts cyclically to the left (the 1st digit becomes the 4th, and the positions of the others decrease by 1) — thus, the robot always checks the tile corresponding to the first digit (the most significant digit) of the number P at the current moment. If the robot cannot move to an uncleaned cell after 4 consecutive checks, it stops cleaning and turns off.

Your task is to find the smallest number P for which the robot can clean all the tiles in the room, or say that it is impossible.

Input

In the first line, the dimensions of the room N and M are given ($1 \leq N \leq M \leq 1000$).

Output

Output the smallest number P for which the robot can clean all the tiles in the room. If there is no suitable number, output “NO” (without quotes).

Examples

standard input	standard output
2 5	1324
4 4	NO

Note

In the first test case, the sequence of tile visits by the robot is:

1 4 5 8 9
2 3 6 7 10

Problem H. Amphitheater

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

In an amphitheater, there are N rows. The first row has K seats. Each subsequent row has 2 more seats than the previous one, and the second and penultimate seats in each row are positioned directly behind the first and last seats of the previous row, respectively.

If a person in the audience begins to applaud, then after 1 second, the audience members sitting in the adjacent seats (to the left, right, front, and back) will also begin to applaud, and this continues with their neighbors, and so on.

You and your acquaintances bought Q tickets. Each ticket specifies the coordinates of a seat in the auditorium: the row number R_i and the seat number C_i . For each ticket, calculate how many seconds T_i after the applause starts at the seat indicated on the ticket, it will take for all the spectators in the auditorium to begin applauding, assuming that there are no empty seats.

Input

The first line contains two integers — the number of rows N ($1 \leq N \leq 10^6$) and the number of seats in the first row K ($1 \leq K \leq 10^6$).

The second line contains the number of tickets Q ($1 \leq Q \leq 10^5$).

The following lines describe the tickets: R_i — the row number, C_i — the seat number in the row. Rows and seat numbers are numbered starting from 1. It is guaranteed that the seat (R_i, C_i) exists in the auditorium.

Output

For each ticket, output the number T_i in a separate line — how many seconds it takes for everyone in the auditorium to be clapping after applause starts at the seat i .

Example

standard input	standard output
4 1	6
3	5
1 1	4
4 2	
3 3	

Problem I. Conflagration

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

On a field of size $n \times n$, there is a chess knight. Some cells in the field have caught fire. The fire spreads to neighboring cells at the end of each knight's move, and if the knight is caught by the fire, it will burn. The fire cannot spread beyond the field boundaries, while the knight can escape.

Your task is to determine the number of starting positions for the knight from which it can escape the fire moving outside of the field boundaries.

Input

The first line contains an integer n ($1 \leq n \leq 500$).

Each of the following n lines contains n characters representing the initial state of the field. “#” indicates fire, while “.” indicates an empty cell.

Output

Output a single integer — the number of cells starting from which the knight can escape from the fire.

Example

standard input	standard output
6 ##### ####.. ...#.. .##... ##### #####	10

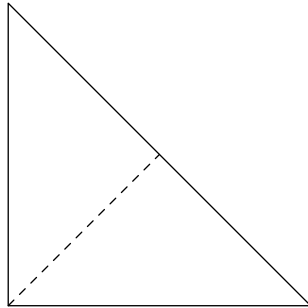
Note

A chess knight moves two squares horizontally and then one square vertically, or one square horizontally and then two squares vertically.

Problem J. Origami

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Vitya cut out an isosceles right triangle from paper. Then he folded it in half, as shown in the picture.



Then he folded it in half again, and again, and again... In total, he folded the paper triangle n times. How many squares formed by the crease lines will Vitya see when he unfolds the paper triangle?

Input

The first line contains an integer n ($1 \leq n \leq 40$).

Output

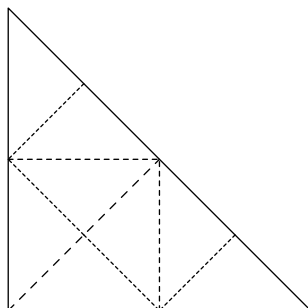
Output the integer — the number of resulting squares.

Example

standard input	standard output
3	3

Note

The paper folds from the example are shown in the picture.



Problem K. Magic Chess Piece

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Given a grid consisting of n rows and m columns. Each cell (r, c) contains a number determined by the following formula (where r is the row number, c is the column number, and both rows and columns are numbered starting from 1):

$$(n + m + c + \lfloor \frac{r^2}{c} \rfloor + \lfloor \frac{c^3}{r^2} \rfloor) \bmod 4;$$

Here, $\lfloor x \rfloor$ denotes rounding x down to the nearest integer, and $x \bmod y$ is the remainder of the division of x by y .

In the cell with coordinates (r_1, c_1) , there is a magical chess piece. The way the magical chess piece moves depends on the cell it is moving from. If the cell contains the number 0 — the piece moves like a knight, 1 — like a bishop, 2 — like a rook, and 3 — like a queen.

Determine the number of ways to move the piece from the starting cell (r_1, c_1) to the final cell (r_2, c_2) in no more than 3 moves. Two ways are considered different if they involve a different number of moves, or if in one way the piece moves to a different cell at any specific move. The final cell may be visited multiple times in a single way.

Input

The first line contains the integers n and m ($1 \leq n, m \leq 10^5$). The second line contains the integers r_1, c_1, r_2, c_2 ($1 \leq r_1, r_2 \leq n; 1 \leq c_1, c_2 \leq m$). The coordinates (r_1, c_1) and (r_2, c_2) are distinct.

Output

Output a single integer — the answer to the problem.

Example

standard input	standard output
5 5 2 4 5 5	19

Note

In the first example, the grid looks as follows:

```
10020
30032
00331
30321
00221
```

The chess pieces move according to the following rules:

- The knight moves two squares horizontally and then one square vertically, or one square horizontally and then two squares vertically.
- The bishop moves to any other cell that is on the same diagonal.
- The rook moves to any other cell that is in the same row or column.
- The queen can move as either a bishop or a rook.