Problem A. Star triangles

Input file:	`triangles.in`
Output file:	`triangles.out`
Time limit:	2 seconds
Memory limit:	256 megabytes
Detailed Feedback:	full

Zhomart likes watching the stars and construct a variety of geometric shapes from them. The sky is represented in the form of the Cartesian coordinate system, and the stars are represented by points on it. At this moment Zhomart interested in the question: how many different right triangles whose legs are parallel to the axes, you can create with the help of stars in the sky.

Input

In the first line of the input line you are given N — the number of stars on the sky ($3 \le N \le 300000$). Each of the next N lines contains integer X and Y ($|X, Y| \le 10^9$) — coordinates of the appropriate star.

Output

Print one number – the answer to the question.

Examples

triangles.in	triangles.out
3 0 0 1 0 0 1	1
4 0 0 1 0 0 1 1 1	4

For this problem you will have full feedback.

30% of tests contain $N \le 100$.

Problem B. Hyper-minimum

Input file:	`hyper.in`
Output file:	`hyper.out`
Time limit:	2 seconds
Memory limit:	256 megabytes
Detailed Feedback:	none

There is a 4-dimensional array X, each index of which is in interval from 1 to N. Your task is to construct new 4-dimensional array Y, elements of which can be calculated using the next formula: $Y[i_1, i_2, i_3, i_4] = min(X[j_1, j_2, j_3, j_4])$, where $1 \le i_k \le N - M + 1, i_k \le j_k \le i_k + M - 1$, and M is given.

Input

First line of the input file contains N and M ($1 \le M \le N$). Next lines of the input file contain elements of array X. The number of elements will be not more than 1500000 and elements will be integers not exceeding 10^9 by absolute value. They are given in such order, that the array can be read using following pseudocode:

```
for i = 1 to N:
    for j = 1 to N:
        for k = 1 to N:
            for l = 1 to N:
                read X[i, j, k, l]
```

Output

Output array Y in the same format as the X was given.

Examples

hyper.in	hyper.out
1 1 1	1
3 2 3 1 4 -4 0 4 0 0 -3 0 -2 -5 5 3 5 -4 4 -3 -5 -4 -4 5 -1 0 -3 -2 -1 2 -5 -5 -1 1 1 -4 3 5 3 -3 -3 3 0 1 4 -1 -2 3 -2 5 4 -1 -5 3 -4 0 -3 -1 3 -1 4 4 -1 -5 -3 4 -4 5 1 5 -4 3 2 2 -2 -2 4 2 -4 -3 1 3 1	-5 -5 -4 -3 -5 -5 -4 -5 -5 -5 -5 -5 -4 -5 -4 -5

Problem C. Energetic turtle

Input file:	`turtle.in`
Output file:	`turtle.out`
Time limit:	2 seconds
Memory limit:	256 megabytes
Detailed Feedback:	none

There is a grid with $N + 1$ rows and $M + 1$ columns. The turtle, which is on the cell (0, 0), wants to get into the cell (N, M). The turtle can only go up or right. There are K traps on the grid. If the turtle will get to one of these traps, it will turn up. The turtle has strength to stand up no more than T times. Calculate, how many different ways the turtle can reach the cell (N, M). Since this number can be very large, output the remainder of his division by Z.

Input

The first line contains 5 integers N, M, K, T and Z ($1 \le N, M \le 300000$, $0 \le K, T \le 20$, $1 \le Z \le 1000000000$). Each of the following K lines contains coordinates of a cell with a trap: X, Y ($0 \le X \le N$, $0 \le Y \le M$). It's guaranteed that all traps situated in different cells and there is no trap in cells (0, 0) and (N, M).

Output

Print one number – the answer.

Examples

turtle.in	turtle.out
1 1 1 0 1000 0 1	1
2 2 0 0 10	6

40% of tests contain $N, M \le 1000$

Problem D. Weighting stones

Input file:	`stones.in`
Output file:	`stones.out`
Time limit:	1 second
Memory limit:	256 megabytes
Detailed Feedback:	none

Jack somehow found N stones and arranged them in increasing order of their weights. No two weights are equal. The lightest stone is given the rank 1, the next lightest — 2, and so on, the heaviest stone gets the rank N.

He has a balance scale and decided to put all the stones on it's sides in some order. It's known in which order he is going to put those stones on the scale and on which side each stone gets.

You have to determine the state of scale after each stone is added. Jack doesn't tell the exact weights of those stones.

Input

The first line contains integer number N ($1 \le N \le 100000$).

Each of the next N lines contains two integer numbers: R ($1 \le R \le N$) and S ($1 \le S \le 2$). R is the rank of the next stone which is put on side S. All R's will be distinct.

Output

Output N lines — one for each added stone. If after adding the corresponding stone side 1 is heavier, output "<". If side 2 is heavier, output ">". If it's not clear in which state the scale will be, output "?".

Examples

stones.in	stones.out
5 1 2 3 1 2 1 4 2 5 1	< > > ? >

Problem E. Odd-even

Input file:	`oddeven.in`
Output file:	`oddeven.out`
Time limit:	2 seconds
Memory limit:	256 megabytes
Detailed Feedback:	full

You are given an increasing sequence of positive integers $1, 2, 4, 5, 7, 9, 10, 12, 14, 16, 17, ...$ It is formed by taking one odd integer, then two even integers, then three odd integers and so on. Output N-th element of this sequence.

Input

One positive integer N ($1 \leq N \leq 10^{100}$).

Output

Output one integer N-th element of the sequence.

Examples

oddeven.in	oddeven.out
1	1
4	5

50% of tests contain $N \leq 10^{18}$.

Problem F. Skyline

Input file:	`skyline.in`
Output file:	`skyline.out`
Time limit:	2 seconds
Memory limit:	256 megabytes
Detailed Feedback:	none

You want to have in your city a beatiful skyline. You have decided to build N skyscrapers in a straight row. The i-th of them should have exactly $h[i]$ floors.

You have got offers from different construction companies. One of them offers to build one floor in any of the skyscrapers for 3 Million Euros. The other one offers to build one floor in each of two neighbouring skyscrapers for 5 Millions in total. Note that it doesn't matter whether these floors are on the same height or not. The third one can build one floor in each of three consecutive skyscrapers for only 7 Millions.

You can build the floors in any order you want. Calculate the minimal possible total amount of money needed to finish the construction.

Input

The first line contains integer number N ($1 \le N \le 300$). The second line contains space separated N integer numbers, $h[1]$, $h[2]$, ..., $h[N]$, $1 \le h[i] \le 200$.

Output

Output one integer number: the amount of money, in Millions.

Examples

skyline.in	skyline.out
3 2 2 2	14
4 1 3 1 1	15

Problem G. Collider

Input file:	collider.in
Output file:	collider.out
Time limit:	2 seconds
Memory limit:	256 megabytes
Detailed Feedback:	none

Physicians are investigating particles of three types: x, y and z. They load a numbered row of n particles into collider. During the experiment an exposure on a concrete particle is having place, after which the particle disappears from i-th position of the row and instantly appears on position j. After disappearance of the particle numbers of particles to the right are decreased by 1 and after the appearance number of particles to the right of that place are increased by 1. After a number of exposures scientists want to know, which particle is on place k. Write program, which will help them.

Input

The first line of the input file contains two integer numeber: n — number of particles and m — total number of exposures and queries ($1 \le n \le 1000000$, $1 \le m \le 15000$).

In the second line there is a sequence of characters x, y and z of length n. Each of the next m lines contains exposure or query description. Line, containing an exposure, starts with character a and space and contains two integer number from interval $[1; n]$. First number is start position of the particle during the exposure and the second one is finish position. Line, describing a query, starts with character q and space and contains one number from interval $[1; n]$ — position, which scientists are interested in.

Output

Output one line for each question from input file. Line number i must contain the answer to the question i — name of the corresponding particle x, y or z.

Examples

collider.in	collider.out
15 6 xzxyyzxxzxyyzyx a 2 10 a 15 4 q 3 a 12 2 q 14 q 2	y z y

Note. Sequence after the first exposure — xxyyzxxzxzyyzyx, after the second — xxyxyzxxzxzyyzy, after the third — xyxyxyzxxzxzyzy.

Problem H. K-th path

Input file:	kthpath.in
Output file:	kthpath.out
Time limit:	2 seconds
Memory limit:	256 megabytes
Detailed Feedback:	none

Suppose you have a table of N rows and M columns. Each cell of the table contains a single lowercase english letter. Consider any path from the top-left to the bottom-right cell of the table, if you are only allowed to move right and down. Letters in the cells met along the path form a string. This string is said to be the value of the path. Now consider all such possible paths and sort them by their values in alphabetical order. Your task is to find the value of the K-th path in this sorted list.

Input

The first line of the input file contains two integer numbers N — the number of rows and M — the number of columns of the given table ($1 \le N, M \le 30$). Each of the next N lines contains exactly M lowercase english letters. The last line of the input file contains a single integer K ($1 \le K \le 10^{18}$). It is guaranteed that the answer exists for the given K.

Output

The first and only line of the output file must contain a single string – the answer to the problem.

Examples

kthpath.in	kthpath.out
3 4 abcd efdg hijk 4	abfdgk

abcdgk, abcdgk, abcdjk, abfdgk, abfdjk, abfijk, aefdgk, aefdjk, aefijk, aehijk