A. Weird Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Yaroslav, Andrey and Roman can play cubes for hours and hours. But the game is for three, so when Roman doesn't show up, Yaroslav and Andrey play another game.

Roman leaves a word for each of them. Each word consists of $2 \cdot n$ binary characters "0" or "1". After that the players start moving in turns. Yaroslav moves first. During a move, a player must choose an integer from 1 to $2 \cdot n$, which hasn't been chosen by anybody up to that moment. Then the player takes a piece of paper and writes out the corresponding character from his string.

Let's represent Yaroslav's word as $s = s_1 s_2 \dots s_{2n}$. Similarly, let's represent Andrey's word as $t = t_1 t_2 \dots t_{2n}$. Then, if Yaroslav choose number k during his move, then he is going to write out character s_k on the piece of paper. Similarly, if Andrey choose number r during his move, then he is going to write out character t_r on the piece of paper.

The game finishes when no player can make a move. After the game is over, Yaroslav makes some integer from the characters written on his piece of paper (Yaroslav can arrange these characters as he wants). Andrey does the same. The resulting numbers can contain leading zeroes. The person with the largest number wins. If the numbers are equal, the game ends with a draw.

You are given two strings s and t. Determine the outcome of the game provided that Yaroslav and Andrey play optimally well.

Input

The first line contains integer n $(1 \le n \le 10^6)$. The second line contains string s — Yaroslav's word. The third line contains string t — Andrey's word.

It is guaranteed that both words consist of $2 \cdot n$ characters "0" and "1".

Output

Print "First", if both players play optimally well and Yaroslav wins. If Andrey wins, print "Second" and if the game ends with a draw, print "Draw". Print the words without the quotes.

Sample test(s)

input

```
2
0111
0001
```

output

```
First
```

input

```
3
110110
001001
```

output

```
First
```

input

```
3
111000
000111
```

output

```
Draw
```

input

```
4
01010110
00101101
```

output

```
First
```

input

```
4
01100000
10010011
```

output

```
Second
```

B. Distinct Paths

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a rectangular $n \times m$-cell board. Some cells are already painted some of k colors. You need to paint each uncolored cell one of the k colors so that any path from the upper left square to the lower right one doesn't contain any two cells of the same color. The path can go only along side-adjacent cells and can only go down or right.

Print the number of possible paintings modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains three integers n, m, k $(1 \le n, m \le 1000, 1 \le k \le 10)$. The next n lines contain m integers each — the board. The first of them contains m uppermost cells of the board from the left to the right and the second one contains m cells from the second uppermost row and so on. If a number in a line equals 0, then the corresponding cell isn't painted. Otherwise, this number represents the initial color of the board cell — an integer from 1 to k.

Consider all colors numbered from 1 to k in some manner.

Output

Print the number of possible paintings modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input

```
2 2 4
0 0
0 0
```

output

```
48
```

input

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

output

```
0
```

input

```
5 6 10
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

output

```
3628800
```

input

```
2 6 10
1 2 3 4 5 6
0 0 0 0 0 0
```

output

```
4096
```

C. Cube Problem

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Yaroslav, Andrey and Roman love playing cubes. Sometimes they get together and play cubes for hours and hours!

Today they got together again and they are playing cubes. Yaroslav took unit cubes and composed them into an $a \times a \times a$ cube, Andrey made a $b \times b \times b$ cube and Roman made a $c \times c \times c$ cube. After that the game was finished and the guys left. But later, Vitaly entered the room. He saw the cubes and wanted to make a cube as well. But what size should the cube be? Of course it should be a large cube with the side of length $a + b + c$. Besides, Vitaly decided to decompose the cubes built by Yaroslav, Andrey and Roman and compose his own large cube out of them. However, it turned out that the unit cubes he got from destroying the three cubes just weren't enough to make a large cube. We know that Vitaly was short of exactly n cubes. Vitaly got upset, demolished everything and left. As he was leaving, he met Petya and told him that there had been three cubes in the room and that he needed another n unit cubes to make his own large cube.

Petya entered the room and saw the messily scattered cubes. He wanted to make it neat and orderly again. But he only knows that there had been three cubes, made of small unit cubes and that Vitaly needed n more unit cubes to make a large one! Help Petya understand, how many ways of sizes a, b, c are there to restore Yaroslav's, Andrey's and Roman's cubes.

Input

The single line of the input contains integer n ($1 \le n \le 10^{14}$). We know that all numbers a, b, c are positive integers.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Output

In the single line print the required number of ways. If it turns out that there isn't a single way of suitable sizes of a, b, c, print 0.

Sample test(s)

input
```
24
```
output
```
1
```

input
```
648
```
output
```
7
```

input
```
5
```
output
```
0
```

input
```
93163582512000
```
output
```
39090
```

D. Ksusha and Square

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ksusha is a vigorous mathematician. She is keen on absolutely incredible mathematical riddles.

Today Ksusha came across a convex polygon of non-zero area. She is now wondering: if she chooses a pair of distinct points uniformly among all integer points (points with integer coordinates) inside or on the border of the polygon and then draws a square with two opposite vertices lying in the chosen points, what will the expectation of this square's area be?

A pair of distinct points is chosen uniformly among all pairs of distinct points, located inside or on the border of the polygon. Pairs of points p, q $(p \neq q)$ and q, p are considered the same.

Help Ksusha! Count the required expectation.

Input

The first line contains integer n $(3 \leq n \leq 10^5)$ — the number of vertices of Ksusha's convex polygon. Next n lines contain the coordinates of the polygon vertices in clockwise or counterclockwise order. The i-th line contains integers x_i, y_i $(|x_i|, |y_i| \leq 10^6)$ — the coordinates of the vertex that goes i-th in that order.

Output

Print a single real number — the required expected area.

The answer will be considered correct if its absolute and relative error doesn't exceed 10^{-6}.

Sample test(s)

input

```
3
0 0
5 5
5 0
```

output

```
4.6666666667
```

input

```
4
-1 3
4 5
6 2
3 -5
```

output

```
8.1583333333
```

input

```
3
17 136
859 937
16 641
```

output

```
66811.3704155169
```

E. Close Vertices

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You've got a weighted tree, consisting of n vertices. Each edge has a non-negative weight. The length of the path between any two vertices of the tree is the number of edges in the path. The weight of the path is the total weight of all edges it contains.

Two vertices are close if there exists a path of length at most l between them and a path of weight at most w between them. Count the number of pairs of vertices v, u $(v < u)$, such that vertices v and u are close.

Input

The first line contains three integers n, l and w $(1 \le n \le 10^5, 1 \le l \le n, 0 \le w \le 10^9)$. The next $n - 1$ lines contain the descriptions of the tree edges. The i-th line contains two integers p_i, w_i $(1 \le p_i < (i + 1), 0 \le w_i \le 10^4)$, that mean that the i-th edge connects vertex $(i + 1)$ and p_i and has weight w_i.

Consider the tree vertices indexed from 1 to n in some way.

Output

Print a single integer — the number of close pairs.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Sample test(s)

input

```
4 4 6
1 3
1 4
1 3
```

output

```
4
```

input

```
6 2 17
1 3
2 5
2 13
1 6
5 9
```

output

```
9
```