Good Bye 2015

A. New Year and Days

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

Today is Wednesday, the third day of the week. What's more interesting is that tomorrow is the last day of the year 2015.

Limak is a little polar bear. He enjoyed this year a lot. Now, he is so eager to the coming year 2016.

Limak wants to prove how responsible a bear he is. He is going to regularly save candies for the entire year 2016! He considers various saving plans. He can save one candy either on some fixed day of the week or on some fixed day of the month.

Limak chose one particular plan. He isn't sure how many candies he will save in the 2016 with his plan. Please, calculate it and tell him.

Input

The only line of the input is in one of the following two formats:

- "x `of week`" where x $(1 \le x \le 7)$ denotes the day of the week. The 1-st day is Monday and the 7-th one is Sunday.
- "x `of month`" where x $(1 \le x \le 31)$ denotes the day of the month.

Output

Print one integer — the number of candies Limak will save in the year 2016.

Sample test(s)

input
```
4 of week
```
output
```
52
```

input
```
30 of month
```
output
```
11
```

Note

Polar bears use the Gregorian calendar. It is the most common calendar and you likely use it too. You can read about it on Wikipedia if you want to – https://en.wikipedia.org/wiki/Gregorian_calendar. The week starts with Monday.

In the first sample Limak wants to save one candy on each Thursday (the 4-th day of the week). There are 52 Thursdays in the 2016. Thus, he will save 52 candies in total.

In the second sample Limak wants to save one candy on the 30-th day of each month. There is the 30-th day in exactly 11 months in the 2016 — all months but February. It means that Limak will save 11 candies in total.

B. New Year and Old Property

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

The year 2015 is almost over.

Limak is a little polar bear. He has recently learnt about the binary system. He noticed that the passing year has exactly one zero in its representation in the binary system — $2015_{10} = 11111011111_2$. Note that he doesn't care about the number of zeros in the decimal representation.

Limak chose some interval of years. He is going to count all years from this interval that have exactly one zero in the binary representation. Can you do it faster?

Assume that all positive integers are always written without leading zeros.

Input

The only line of the input contains two integers a and b ($1 \le a \le b \le 10^{18}$) — the first year and the last year in Limak's interval respectively.

Output

Print one integer – the number of years Limak will count in his chosen interval.

Sample test(s)

input
```
5 10
```
output
```
2
```

input
```
2015 2015
```
output
```
1
```

input
```
100 105
```
output
```
0
```

input
```
72057594000000000 72057595000000000
```
output
```
26
```

Note

In the first sample Limak's interval contains numbers $5_{10} = 101_2$, $6_{10} = 110_2$, $7_{10} = 111_2$, $8_{10} = 1000_2$, $9_{10} = 1001_2$ and $10_{10} = 1010_2$. Two of them (101_2 and 110_2) have the described property.

C. New Year and Domino

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

They say "*years are like dominoes, tumbling one after the other*". But would a year fit into a grid? I don't think so.

Limak is a little polar bear who loves to play. He has recently got a rectangular grid with h rows and w columns. Each cell is a square, either empty (denoted by '`.`') or forbidden (denoted by '`#`'). Rows are numbered 1 through h from top to bottom. Columns are numbered 1 through w from left to right.

Also, Limak has a single domino. He wants to put it somewhere in a grid. A domino will occupy exactly two adjacent cells, located either in one row or in one column. Both adjacent cells must be empty and must be inside a grid.

Limak needs more fun and thus he is going to consider some queries. In each query he chooses some rectangle and wonders, how many way are there to put a single domino inside of the chosen rectangle?

Input

The first line of the input contains two integers h and w ($1 \le h, w \le 500$) – the number of rows and the number of columns, respectively.

The next h lines describe a grid. Each line contains a string of the length w. Each character is either '`.`' or '`#`' — denoting an empty or forbidden cell, respectively.

The next line contains a single integer q ($1 \le q \le 100\,000$) — the number of queries.

Each of the next q lines contains four integers $r1_i$, $c1_i$, $r2_i$, $c2_i$ ($1 \le r1_i \le r2_i \le h$, $1 \le c1_i \le c2_i \le w$) — the i-th query. Numbers $r1_i$ and $c1_i$ denote the row and the column (respectively) of the upper left cell of the rectangle. Numbers $r2_i$ and $c2_i$ denote the row and the column (respectively) of the bottom right cell of the rectangle.

Output

Print q integers, i-th should be equal to the number of ways to put a single domino inside the i-th rectangle.

Sample test(s)

input

```
5 8
....#..#
.#......
##.#....
##..#.##
........
4
1 1 2 3
4 1 4 1
1 2 4 5
2 5 5 8
```

output

```
4
0
10
15
```

input

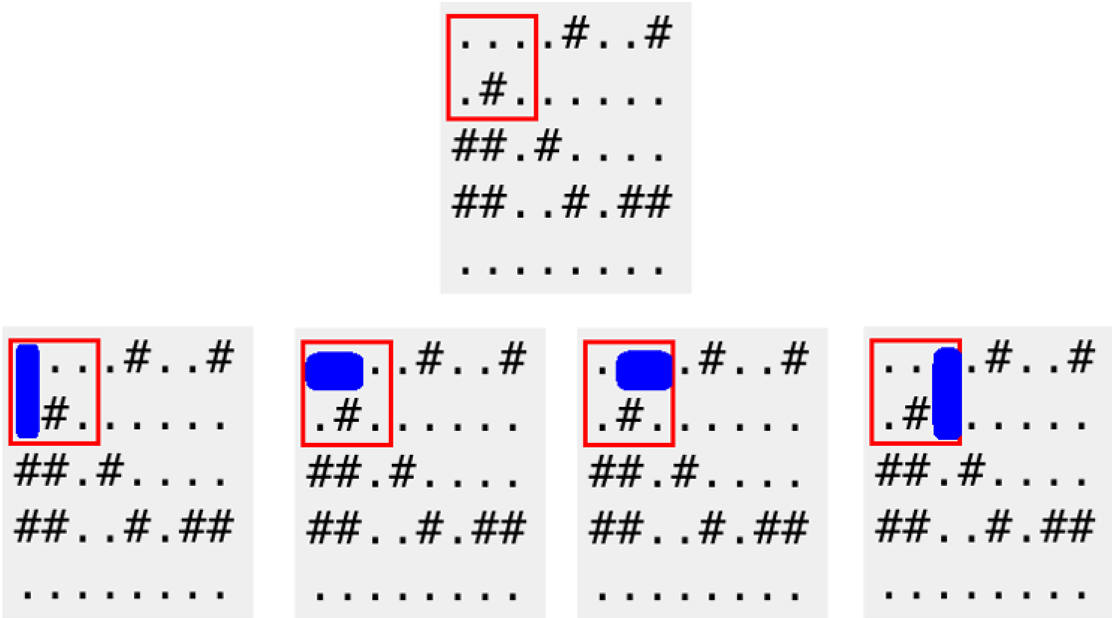
```
7 39
.......................................
.###..###..#..###.....###..###..#..###.
...#..#.#..#..#.........#..#.#..#..#...
.###..#.#..#..###.....###..#.#..#..###.
.#....#.#..#....#.....#....#.#..#..#.#.
.###..###..#..###.....###..###..#..###.
.......................................
6
1 1 3 20
2 10 6 30
2 10 7 30
2 2 7 7
1 7 7 7
1 8 7 8
```

output

```
53
89
120
23
0
2
```

Note

A red frame below corresponds to the first query of the first sample. A domino can be placed in 4 possible ways.



D. New Year and Ancient Prophecy

time limit per test: 2.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Limak is a little polar bear. In the snow he found a scroll with the ancient prophecy. Limak doesn't know any ancient languages and thus is unable to understand the prophecy. But he knows digits!

One fragment of the prophecy is a sequence of n digits. The first digit isn't zero. Limak thinks that it's a list of some special years. It's hard to see any commas or spaces, so maybe ancient people didn't use them. Now Limak wonders what years are listed there.

Limak assumes three things:

- Years are listed in the **strictly** increasing order;
- Every year is a positive integer number;
- There are no leading zeros.

Limak is going to consider all possible ways to split a sequence into numbers (years), satisfying the conditions above. He will do it without any help. However, he asked you to tell him the number of ways to do so. Since this number may be very large, you are only asked to calculate it modulo $10^9 + 7$.

Input

The first line of the input contains a single integer n $(1 \le n \le 5000)$ — the number of digits.

The second line contains a string of digits and has length equal to n. It's guaranteed that the first digit is not '0'.

Output

Print the number of ways to correctly split the given sequence modulo $10^9 + 7$.

Sample test(s)

input

```
6
123434
```

output

```
8
```

input

```
8
20152016
```

output

```
4
```

Note

In the first sample there are 8 ways to split the sequence:

- "123434" = "123434" (maybe the given sequence is just one big number)
- "123434" = "1" + "23434"
- "123434" = "12" + "3434"
- "123434" = "123" + "434"
- "123434" = "1" + "23" + "434"
- "123434" = "1" + "2" + "3434"
- "123434" = "1" + "2" + "3" + "434"
- "123434" = "1" + "2" + "3" + "4" + "34"

Note that we don't count a split "123434" = "12" + "34" + "34" because numbers have to be strictly increasing.

In the second sample there are 4 ways:

- "20152016" = "20152016"
- "20152016" = "20" + "152016"
- "20152016" = "201" + "52016"
- "20152016" = "2015" + "2016"

E. New Year and Three Musketeers

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

Do you know the story about the three musketeers? Anyway, you must help them now.

Richelimakieu is a cardinal in the city of Bearis. He found three brave warriors and called them the three musketeers. Athos has strength a, Borthos strength b, and Caramis has strength c.

The year 2015 is almost over and there are still n criminals to be defeated. The i-th criminal has strength t_i. It's hard to defeat strong criminals — maybe musketeers will have to fight together to achieve it.

Richelimakieu will coordinate musketeers' actions. In each hour each musketeer can either do nothing or be assigned to one criminal. Two or three musketeers can be assigned to the same criminal and then their strengths are summed up. A criminal can be defeated in exactly one hour (also if two or three musketeers fight him). Richelimakieu can't allow the situation where a criminal has strength bigger than the sum of strengths of musketeers fighting him — a criminal would win then!

In other words, there are three ways to defeat a criminal.

- A musketeer of the strength x in one hour can defeat a criminal of the strength not greater than x. So, for example Athos in one hour can defeat criminal i only if $t_i \leq a$.
- Two musketeers can fight together and in one hour defeat a criminal of the strength not greater than the sum of strengths of these two musketeers. So, for example Athos and Caramis in one hour can defeat criminal i only if $t_i \leq a + c$. Note that the third remaining musketeer can either do nothing or fight some other criminal.
- Similarly, all three musketeers can fight together and in one hour defeat a criminal of the strength not greater than the sum of musketeers' strengths, i.e. $t_i \leq a + b + c$.

Richelimakieu doesn't want musketeers to fight during the New Year's Eve. Thus, he must coordinate their actions in order to minimize the number of hours till all criminals will be defeated.

Find the minimum number of hours to defeat all criminals. If musketeers can't defeat them all then print "`-1`" (without the quotes) instead.

Input

The first line of the input contains a single integer n $(1 \leq n \leq 200\,000)$ — the number of criminals.

The second line contains three integers a, b and c $(1 \leq a, b, c \leq 10^8)$ — strengths of musketeers.

The third line contains n integers $t_1, t_2, ..., t_n$ $(1 \leq t_i \leq 10^8)$ — strengths of criminals.

Output

Print one line with the answer.

If it's impossible to defeat all criminals, print "`-1`" (without the quotes). Otherwise, print the minimum number of hours the three musketeers will spend on defeating all criminals.

Sample test(s)

input

```
5
10 20 30
1 1 1 1 50
```

output

```
2
```

input

```
5
10 20 30
1 1 1 1 51
```

output

```
3
```

input

```
7
30 20 10
34 19 50 33 88 15 20
```

output

```
-1
```

input

```
6
```

```
10 5 10
10 9 5 25 20 5
```

output

```
3
```

Note

In the first sample Athos has strength 10, Borthos 20, and Caramis 30. They can defeat all criminals in two hours:

- Borthos and Caramis should together fight a criminal with strength 50. In the same hour Athos can fight one of four criminals with strength 1.
- There are three criminals left, each with strength 1. Each musketeer can fight one criminal in the second hour.

In the second sample all three musketeers must together fight a criminal with strength 51. It takes one hour. In the second hour they can fight separately, each with one criminal. In the third hour one criminal is left and any of musketeers can fight him.

F. New Year and Cleaning

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

Limak is a little polar bear. His parents told him to clean a house before the New Year's Eve. Their house is a rectangular grid with h rows and w columns. Each cell is an empty square.

He is a little bear and thus he can't clean a house by himself. Instead, he is going to use a cleaning robot.

A cleaning robot has a built-in pattern of n moves, defined by a string of the length n. A single move (character) moves a robot to one of four adjacent cells. Each character is one of the following four: '`U`' (up), '`D`' (down), '`L`' (left), '`R`' (right). One move takes one minute.

A cleaning robot must be placed and started in some cell. Then it repeats its pattern of moves till it hits a wall (one of four borders of a house). After hitting a wall it can be placed and used again.

Limak isn't sure if placing a cleaning robot in one cell will be enough. Thus, he is going to start it $w \cdot h$ times, one time in each cell. Maybe some cells will be cleaned more than once but who cares?

Limak asks you one question. How much time will it take to clean a house? Find and print the number of minutes modulo $10^9 + 7$. It's also possible that a cleaning robot will never stop — then print "`-1`" (without the quotes) instead.

Placing and starting a robot takes no time, however, you must count a move when robot hits a wall. Take a look into samples for further clarification.

Input

The first line contains three integers n, h and w ($1 \le n, h, w \le 500\,000$) — the length of the pattern, the number of rows and the number of columns, respectively.

The second line contains a string of length n — the pattern of n moves. Each character is one of uppercase letters '`U`', '`D`', '`L`' or '`R`'.

Output

Print one line with the answer.

If a cleaning robot will never stop, print "`-1`" (without the quotes). Otherwise, print the number of minutes it will take to clean a house modulo $10^9 + 7$.

Sample test(s)

input
```
1 10 2
R
```
output
```
30
```

input
```
3 4 6
RUL
```
output
```
134
```

input
```
4 1 500000
RLRL
```
output
```
-1
```

Note

In the first sample house is a grid with 10 rows and 2 columns. Starting a robot anywhere in the second column will result in only one move (thus, one minute of cleaning) in which robot will hit a wall — he tried to go right but there is no third column. Starting a robot anywhere in the first column will result in two moves. The total number of minutes is $10 \cdot 1 + 10 \cdot 2 = 30$.

In the second sample a started robot will try to move "`RULRULRULR...`" For example, for the leftmost cell in the second row robot will make 5 moves before it stops because of hitting an upper wall.

G. New Year and Cake

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

Limak is a little polar bear. According to some old traditions, his bear family prepared a New Year cake. And Limak likes cakes.

As you may know, a New Year cake is a strictly convex polygon with n vertices.

Parents won't allow Limak to eat more than half of a cake because he would get sick. After some thinking they decided to cut a cake along one of $n \cdot (n - 3) / 2$ diagonals. Then Limak will get a non-greater piece.

Limak understands rules but he won't be happy if the second piece happens to be much bigger. Limak's disappointment will be equal to the difference between pieces' areas, multiplied by two. It can be proved that it will be integer for the given constraints.

There are $n \cdot (n - 3) / 2$ possible scenarios. Consider them all and find the sum of values of Limak's disappointment, modulo $10^9 + 7$.

Input

The first line of the input contains a single integer n ($4 \le n \le 500\,000$) — the number of vertices in the polygon denoting the cake.

Each of the next n lines contains two integers x_i and y_i ($|x_i|, |y_i| \le 10^9$) — coordinates of the i-th point.

It's guaranteed that all points are distinct, polygon is strictly convex and points are given in the clockwise order.

Output

Print the sum of values of Limak's disappointment over all possible scenarios modulo $10^9 + 7$.

Sample test(s)

input

```
5
2 4
2 7
5 7
5 4
3 -2
```

output

```
90
```

input

```
4
-1000000000 -5000000
0 1234567
1 1
-5 -100000000
```

output

```
525185196
```

input

```
8
-10 0
-6 6
0 10
6 6
10 0
6 -6
0 -10
-6 -6
```

output

```
5216
```

Note

In the first sample possible values of Limak's disappointment are $0, 18, 18, 24, 30$.

H. New Year and Forgotten Tree

time limit per test: 7 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A *tree* is a connected undirected graph with $n - 1$ edges, where n denotes the number of vertices. Vertices are numbered 1 through n.

Limak is a little polar bear. His bear family prepares a New Year tree every year. One year ago their tree was more awesome than usually. Thus, they decided to prepare the same tree in the next year. Limak was responsible for remembering that tree.

It would be hard to remember a whole tree. Limak decided to describe it in his notebook instead. He took a pen and wrote $n - 1$ lines, each with two integers — indices of two vertices connected by an edge.

Now, the New Year is just around the corner and Limak is asked to reconstruct that tree. Of course, there is a problem. He was a very little bear a year ago, and he didn't know digits and the alphabet, so he just replaced each digit with a question mark — the only character he knew. That means, for any vertex index in his notes he knows only the number of digits in it. At least he knows there were no leading zeroes.

Limak doesn't want to disappoint everyone. Please, take his notes and reconstruct a New Year tree. Find any tree matching Limak's records and print its edges in any order. It's also possible that Limak made a mistake and there is no suitable tree – in this case print "`-1`" (without the quotes).

Input

The first line contains a single integer n $(2 \le n \le 200\,000)$ — the number of vertices.

Each of the next $n - 1$ lines contains two space-separated non-empty strings, both consisting of questions marks only. No string has more characters than the number of digits in n.

Output

If there is no tree matching Limak's records, print the only line with "`-1`" (without the quotes).

Otherwise, describe any tree matching Limak's notes. Print $n - 1$ lines, each with two space-separated integers – indices of vertices connected by an edge. You can print edges in any order.

Sample test(s)

input

```
12
? ?
? ?
? ?
? ??
?? ?
?? ??
? ??
? ?
? ?
? ?
? ?
```

output

```
3 1
1 6
9 1
2 10
1 7
8 1
1 4
1 10
5 1
10 11
12 1
```

input

```
12
?? ??
? ?
? ?
? ??
?? ?
?? ??
? ??
? ?
? ?
?? ??
? ?
```

output

```
-1
```