A. A Blend of Springtime

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

When the curtains are opened, a canvas unfolds outside. Kanno marvels at all the blonde colours along the riverside — not tangerines, but blossoms instead.
"What a pity it's already late spring," sighs Mino with regret, "one more drizzling night and they'd be gone."

"But these blends are at their best, aren't they?" Absorbed in the landscape, Kanno remains optimistic.

The landscape can be expressed as a row of consecutive cells, each of which either contains a flower of colour amber or buff or canary yellow, or is empty.

When a flower withers, it disappears from the cell that it originally belonged to, and it spreads petals of its colour in its two neighbouring cells (or outside the field if the cell is on the side of the landscape). In case petals fall outside the given cells, they simply become invisible.

You are to help Kanno determine whether it's possible that after some (possibly none or all) flowers shed their petals, at least one of the cells contains all three colours, considering both petals and flowers. Note that flowers can wither in arbitrary order.

Input
The first and only line of input contains a non-empty string $$$s$$$ consisting of uppercase English letters 'A', 'B', 'C' and characters '.' (dots) only ($$$\lvert s \rvert \leq 100$$$) — denoting cells containing an amber flower, a buff one, a canary yellow one, and no flowers, respectively.

Output
Output "Yes" if it's possible that all three colours appear in some cell, and "No" otherwise.

You can print each letter in any case (upper or lower).

Examples

input
```
.BAC.
```
output
```
Yes
```

input
```
AA..CB
```
output
```
No
```

Note
In the first example, the buff and canary yellow flowers can leave their petals in the central cell, blending all three colours in it.

In the second example, it's impossible to satisfy the requirement because there is no way that amber and buff meet in any cell.

B. A Tide of Riverscape

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Walking along a riverside, Mino silently takes a note of something.
"Time," Mino thinks aloud.

"What?"

"Time and tide wait for no man," explains Mino. "My name, taken from the river, always reminds me of this."

"And what are you recording?"

"You see it, tide. Everything has its own period, and I think I've figured out this one," says Mino with confidence.

Doubtfully, Kanno peeks at Mino's records.

The records are expressed as a string s of characters '`0`', '`1`' and '`.`', where '`0`' denotes a low tide, '`1`' denotes a high tide, and '`.`' denotes an unknown one (either high or low).

You are to help Mino determine whether it's possible that after replacing each '`.`' independently with '`0`' or '`1`', a given integer p is **not** a period of the resulting string. In case the answer is yes, please also show such a replacement to Mino.

In this problem, a positive integer p is considered a period of string s, if for all $1 \leq i \leq \lvert s \rvert - p$, the i-th and $(i + p)$-th characters of s are the same. Here $\lvert s \rvert$ is the length of s.

Input

The first line contains two space-separated integers n and p ($1 \leq p \leq n \leq 2000$) — the length of the given string and the supposed period, respectively.

The second line contains a string s of n characters — Mino's records. s only contains characters '`0`', '`1`' and '`.`', and contains at least one '`.`' character.

Output

Output one line — if it's possible that p is **not** a period of the resulting string, output any one of such strings; otherwise output "`No`" (without quotes, you can print letters in any case (upper or lower)).

Examples

input

```
10 7
1.0.1.0.1.
```

output

```
1000100010
```

input

```
10 6
1.0.1.1000
```

output

```
1001101000
```

input

```
10 9
1........1
```

output

```
No
```

Note

In the first example, 7 is not a period of the resulting string because the 1-st and 8-th characters of it are different.

In the second example, 6 is not a period of the resulting string because the 4-th and 10-th characters of it are different.

In the third example, 9 is always a period because the only constraint that the first and last characters are the same is already satisfied.

Note that there are multiple acceptable answers for the first two examples, you can print any of them.

C. A Mist of Florescence

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

As the boat drifts down the river, a wood full of blossoms shows up on the riverfront. "I've been here once," Mino exclaims with delight, "it's breathtakingly amazing."

"What is it like?"

"Look, Kanno, you've got your paintbrush, and I've got my words. Have a try, shall we?"

There are four kinds of flowers in the wood, Amaranths, Begonias, Centaureas and Dianthuses.

The wood can be represented by a rectangular grid of n rows and m columns. In each cell of the grid, there is exactly one type of flowers.

According to Mino, the numbers of connected components formed by each kind of flowers are a, b, c and d respectively. Two cells are considered in the same connected component if and only if a path exists between them that moves between cells sharing common edges and passes only through cells containing the same flowers.

You are to help Kanno depict such a grid of flowers, with n and m arbitrarily chosen under the constraints given below. It can be shown that at least one solution exists under the constraints of this problem.

Note that you can choose arbitrary n and m under the constraints below, they are not given in the input.

Input

The first and only line of input contains four space-separated integers a, b, c and d ($1 \leq a, b, c, d \leq 100$) — the required number of connected components of Amaranths, Begonias, Centaureas and Dianthuses, respectively.

Output

In the first line, output two space-separated integers n and m ($1 \leq n, m \leq 50$) — the number of rows and the number of columns in the grid respectively.

Then output n lines each consisting of m consecutive English letters, representing one row of the grid. Each letter should be among '`A`', '`B`', '`C`' and '`D`', representing Amaranths, Begonias, Centaureas and Dianthuses, respectively.

In case there are multiple solutions, print any. You can output each letter in either case (upper or lower).

Examples

input

```
5 3 2 1
```

output

```
4 7
DDDDDDD
DABACAD
DBABACD
DDDDDDD
```

input

```
50 50 1 1
```

output

```
4 50
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ABABABABABABABABABABABABABABABABABABABABABABABABAB
BABABABABABABABABABABABABABABABABABABABABABABABABA
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
```

input

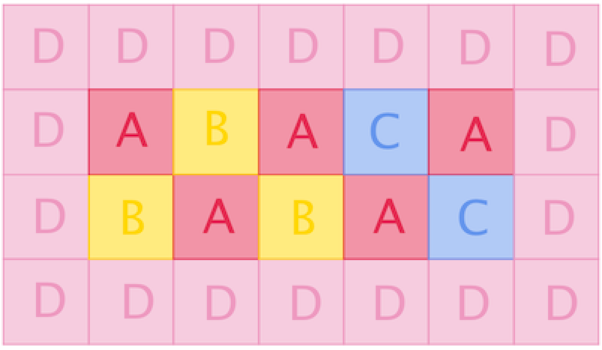
```
1 6 4 5
```

output

```
7 7
DDDDDDD
DDDBDBD
DDCDCDD
DBDADBD
DDCDCDD
DBDBDDD
DDDDDDD
```

Note

In the first example, each cell of Amaranths, Begonias and Centaureas forms a connected component, while all the Dianthuses form one.



D. A Shade of Moonlight

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

Gathering darkness shrouds the woods and the world. The moon sheds its light on the boat and the river.
"To curtain off the moonlight should be hardly possible; the shades present its mellow beauty and restful nature." Intonates Mino.

"See? The clouds are coming." Kanno gazes into the distance.

"That can't be better," Mino turns to Kanno.

The sky can be seen as a one-dimensional axis. The moon is at the origin whose coordinate is 0.

There are n clouds floating in the sky. Each cloud has the same length l. The i-th initially covers the range of $(x_i, x_i + l)$ (**endpoints excluded**). Initially, it moves at a velocity of v_i, which equals either 1 or -1.

Furthermore, no pair of clouds intersect initially, that is, for all $1 \leq i \lt j \leq n$, $\lvert x_i - x_j \rvert \geq l$.

With a wind velocity of w, the velocity of the i-th cloud becomes $v_i + w$. That is, its coordinate increases by $v_i + w$ during each unit of time. Note that the wind can be strong and clouds can change their direction.

You are to help Mino count the number of pairs (i, j) ($i < j$), such that with a proper choice of wind velocity w not exceeding w_max in absolute value (possibly negative and/or fractional), the i-th and j-th clouds both cover the moon at the same future moment. This w doesn't need to be the same across different pairs.

Input

The first line contains three space-separated integers n, l, and w_max ($1 \leq n \leq 10^5$, $1 \leq l, w_\mathrm{max} \leq 10^8$) — the number of clouds, the length of each cloud and the maximum wind speed, respectively.

The i-th of the following n lines contains two space-separated integers x_i and v_i ($-10^8 \leq x_i \leq 10^8$, $v_i \in \{-1, 1\}$) — the initial position and the velocity of the i-th cloud, respectively.

The input guarantees that for all $1 \leq i \lt j \leq n$, $\lvert x_i - x_j \rvert \geq l$.

Output

Output one integer — the number of unordered pairs of clouds such that it's possible that clouds from each pair cover the moon at the same future moment with a proper choice of wind velocity w.

Examples

input

```
5 1 2
-2 1
2 1
3 -1
5 -1
7 -1
```

output

```
4
```

input

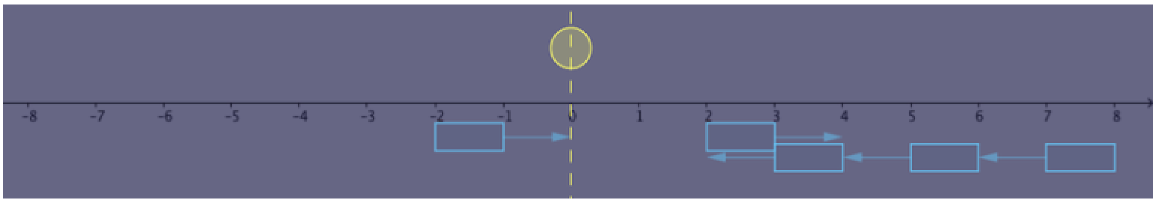
```
4 10 1
-20 1
-10 -1
0 1
10 -1
```

output

```
1
```

Note

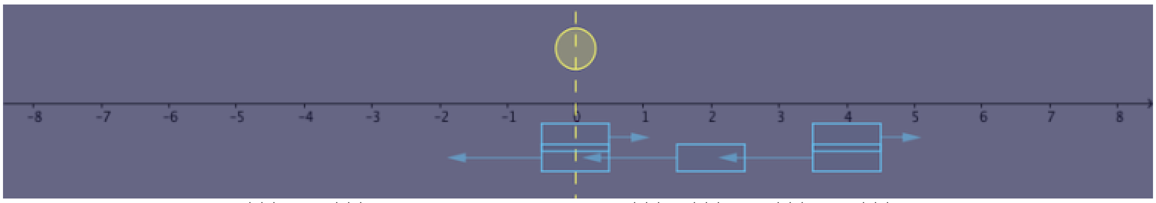
In the first example, the initial positions and velocities of clouds are illustrated below.



The pairs are:

- $(1, 3)$, covering the moon at time 2.5 with $w = -0.4$;
- $(1, 4)$, covering the moon at time 3.5 with $w = -0.6$;
- $(1, 5)$, covering the moon at time 4.5 with $w = -0.7$;
- $(2, 5)$, covering the moon at time 2.5 with $w = -2$.

Below is the positions of clouds at time 2.5 with $w = -0.4$. At this moment, the 1-st and 3-rd clouds both cover the moon.



In the second example, the only pair is $(1, 4)$, covering the moon at time 15 with $w = 0$.

Note that all the times and wind velocities given above are just examples among infinitely many choices.

E. A Trance of Nightfall

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

The cool breeze blows gently, the flowing water ripples steadily.
"Flowing and passing like this, the water isn't gone ultimately; Waxing and waning like that, the moon doesn't shrink or grow eventually."

"Everything is transient in a way and perennial in another."

Kanno doesn't seem to make much sense out of Mino's isolated words, but maybe it's time that they enjoy the gentle breeze and the night sky — the inexhaustible gifts from nature.

Gazing into the sky of stars, Kanno indulges in a night's tranquil dreams.

There is a set S of n points on a coordinate plane.

Kanno starts from a point P that can be chosen on the plane. P is not added to S if it doesn't belong to S. Then the following sequence of operations (altogether called a *move*) is repeated several times, in the given order:

1. Choose a line l such that it passes through at least two elements in S and passes through Kanno's current position. If there are multiple such lines, one is chosen equiprobably.
2. Move to one of the points that belong to S and lie on l. The destination is chosen equiprobably among all possible ones, including Kanno's current position (if it does belong to S).

There are q queries each consisting of two integers (t_i, m_i). For each query, you're to help Kanno maximize the probability of the stopping position being the t_i-th element in S after m_i moves with a proper selection of P, and output this maximum probability. Note that according to rule 1, P should belong to at least one line that passes through at least two points from S.

Input

The first line contains a positive integer n ($2 \leq n \leq 200$) — the number of points in S.

The i-th of the following n lines contains two space-separated integers x_i and y_i ($-10^4 \leq x_i, y_i \leq 10^4$) — the coordinates of the i-th point in S. The input guarantees that for all $1 \leq i \lt j \leq n$, $(x_i, y_i) \neq (x_j, y_j)$ holds.

The next line contains a positive integer q ($1 \leq q \leq 200$) — the number of queries.

Each of the following $$$q$$$ lines contains two space-separated integers $$$t$$$ and $$$m$$$ ($$$1 \leq t_i \leq n$$$, $$$1 \leq m_i \leq 10^4$$$) — the index of the target point and the number of moves, respectively.

Output

Output $$$q$$$ lines each containing a decimal number — the $$$i$$$-th among them denotes the maximum probability of staying on the $$$t_i$$$-th point after $$$m_i$$$ steps, with a proper choice of starting position $$$P$$$.

Your answer will be considered correct if each number in your output differs from the corresponding one in jury's answer by at most $$$10^{-6}$$$.

Formally, let your answer be $$$a$$$, and the jury's answer be $$$b$$$. Your answer is considered correct if $$$|a - b| \le 10^{-6}$$$.

Example

input

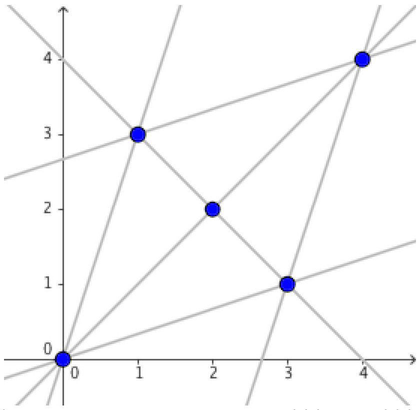
```
5
0 0
1 3
2 2
3 1
4 4
10
1 1
2 1
3 1
4 1
5 1
3 2
3 3
3 4
3 5
3 6
```

output

```
0.50000000000000000000
0.50000000000000000000
0.33333333333333331483
0.50000000000000000000
0.50000000000000000000
0.18518518518518517491
0.15226337448559670862
0.14494741655235482414
0.14332164812274550414
0.14296036624949901017
```

Note

The points in $$$S$$$ and possible candidates for line $$$l$$$ are depicted in the following figure.



For the first query, when $$$P = (-1, -3)$$$, $$$l$$$ is uniquely determined to be $$$3x = y$$$, and thus Kanno will move to $$$(0, 0)$$$ with a probability of $$$\frac 1 2$$$.

For the third query, when $$$P = (2, 2)$$$, $$$l$$$ is chosen equiprobably between $$$x + y = 4$$$ and $$$x = y$$$. Kanno will then move to the other four points with a probability of $$$\frac 1 2 \cdot \frac 1 3 = \frac 1 6$$$ each, or stay at $$$(2, 2)$$$ with a probability of $$$\frac 1 3$$$.