Codeforces Round #571 (Div. 2)

A. Vus the Cossack and a Contest

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

Vus the Cossack holds a programming competition, in which n people participate. He decided to award them all with pens and notebooks. It is known that Vus has exactly m pens and k notebooks.

Determine whether the Cossack can reward **all** participants, giving each of them at least one pen and at least one notebook.

Input

The first line contains three integers n, m, and k ($1 \leq n, m, k \leq 100$) — the number of participants, the number of pens, and the number of notebooks respectively.

Output

Print "Yes" if it possible to reward all the participants. Otherwise, print "No".

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

Examples

input
```
5 8 6
```
output
```
Yes
```

input
```
3 9 3
```
output
```
Yes
```

input
```
8 5 20
```
output
```
No
```

Note

In the first example, there are 5 participants. The Cossack has 8 pens and 6 notebooks. Therefore, he has enough pens and notebooks.

In the second example, there are 3 participants. The Cossack has 9 pens and 3 notebooks. He has more than enough pens but only the minimum needed number of notebooks.

In the third example, there are 8 participants but only 5 pens. Since the Cossack does not have enough pens, the answer is "No".

C. Vus the Cossack and Strings

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

Vus the Cossack has two binary strings, that is, strings that consist only of "0" and "1". We call these strings a and b. It is known that $|b| \leq |a|$, that is, the length of b is at most the length of a.

The Cossack considers every substring of length $|b|$ in string a. Let's call this substring c. He matches the corresponding characters in b and c, after which he counts the number of positions where the two strings are different. We call this function $f(b, c)$.

For example, let $b = 00110$, and $c = 11000$. In these strings, the first, second, third and fourth positions are different.

Vus the Cossack counts the number of such substrings c such that $f(b, c)$ is **even**.

For example, let $a = 01100010$ and $b = 00110$. a has four substrings of the length $|b|$: 01100, 11000, 10001, 00010.

- $f(00110, 01100) = 2$;
- $f(00110, 11000) = 4$;
- $f(00110, 10001) = 4$;
- $f(00110, 00010) = 1$.

Since in three substrings, $f(b, c)$ is even, the answer is 3.

Vus can not find the answer for big strings. That is why he is asking you to help him.

Input

The first line contains a binary string a ($1 \le |a| \le 10^6$) — the first string.

The second line contains a binary string b ($1 \le |b| \le |a|$) — the second string.

Output

Print one number — the answer.

Examples

input

```
01100010
00110
```

output

```
3
```

input

```
1010111110
0110
```

output

```
4
```

Note

The first example is explained in the legend.

In the second example, there are five substrings that satisfy us: 1010, 0101, 1111, 1111.

D. Vus the Cossack and Numbers

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

Vus the Cossack has n real numbers a_i. It is known that the sum of all numbers is equal to 0. He wants to choose a sequence b the size of which is n such that the sum of all numbers is 0 and each b_i is either $\lfloor a_i \rfloor$ or $\lceil a_i \rceil$. In other words, b_i equals a_i rounded up or down. It is not necessary to round to the nearest integer.

For example, if $a = [4.58413, 1.22491, -2.10517, -3.70387]$, then b can be equal, for example, to $[4, 2, -2, -4]$.

Note that if a_i is an integer, then there is no difference between $\lfloor a_i \rfloor$ and $\lceil a_i \rceil$, b_i will always be equal to a_i.

Help Vus the Cossack find such sequence!

Input

The first line contains one integer n ($1 \le n \le 10^5$) — the number of numbers.

Each of the next n lines contains one real number a_i ($|a_i| < 10^5$). It is guaranteed that each a_i has exactly 5 digits after the decimal point. It is guaranteed that the sum of all the numbers is equal to 0.

Output

In each of the next n lines, print one integer b_i. For each i, $|a_i - b_i| < 1$ must be met.

If there are multiple answers, print any.

Examples

input

```
4
4.58413
1.22491
-2.10517
-3.70387
```

output

```
4
2
-2
```

```
-4
```

input

```
5
-6.32509
3.30066
-0.93878
2.00000
1.96321
```

output

```
-6
3
-1
2
2
```

Note

The first example is explained in the legend.

In the second example, we can round the first and fifth numbers up, and the second and third numbers down. We can round the fourth number neither up, nor down.

E. Vus the Cossack and a Field

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

Vus the Cossack has a field with dimensions $n \times m$, which consists of "0" and "1". He is building an infinite field from this field. He is doing this in this way:

1. He takes the current field and finds a new inverted field. In other words, the new field will contain "1" only there, where "0" was in the current field, and "0" there, where "1" was.
2. To the current field, he adds the inverted field to the right.
3. To the current field, he adds the inverted field to the bottom.
4. To the current field, he adds the current field to the bottom right.
5. He repeats it.

For example, if the initial field was:

$$\begin{matrix} 1 & 0 \\ 1 & 1 \end{matrix}$$

After the first iteration, the field will be like this:

$$\begin{matrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{matrix}$$

After the second iteration, the field will be like this:

$$\begin{matrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{matrix}$$

And so on...

Let's numerate lines from top to bottom from 1 to infinity, and columns from left to right from 1 to infinity. We call the submatrix (x_1, y_1, x_2, y_2) all numbers that have coordinates (x, y) such that $x_1 \le x \le x_2$ and $y_1 \le y \le y_2$.

The Cossack needs sometimes to find the sum of all the numbers in submatrices. Since he is pretty busy right now, he is asking you to find the answers!

Input

The first line contains three integers n, m, q ($1 \le n, m \le 1\,000$, $1 \le q \le 10^5$) — the dimensions of the initial matrix and the number of queries.

Each of the next n lines contains m characters c_{ij} ($0 \le c_{ij} \le 1$) — the characters in the matrix.

Each of the next q lines contains four integers x_1, y_1, x_2, y_2 ($1 \le x_1 \le x_2 \le 10^9$, $1 \le y_1 \le y_2 \le 10^9$) — the coordinates of the upper left cell and bottom right cell, between which you need to find the sum of all numbers.

Output

For each query, print the answer.

Examples

input

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

output

```
32
5
25
14
4
```

input

```
2 3 7
100
101
4 12 5 17
5 4 9 4
1 4 13 18
12 1 14 9
3 10 7 18
3 15 12 17
8 6 8 12
```

output

```
6
3
98
13
22
15
3
```

Note

The first example is explained in the legend.

F. Vus the Cossack and a Graph

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

Vus the Cossack has a simple graph with n vertices and m edges. Let d_i be a degree of the i-th vertex. Recall that a degree of the i-th vertex is the number of conected edges to the i-th vertex.

He needs to remain not more than $\lceil \frac{n+m}{2} \rceil$ edges. Let f_i be the degree of the i-th vertex after removing. He needs to delete them in such way so that $\lceil \frac{d_i}{2} \rceil \le f_i$ for each i. In other words, the degree of each vertex should not be reduced more than twice.

Help Vus to remain the needed edges!

Input

The first line contains two integers n and m ($1 \le n \le 10^6$, $0 \le m \le 10^6$) — the number of vertices and edges respectively.

Each of the next m lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) — vertices between which there is an edge.

It is guaranteed that the graph does not have loops and multiple edges.

It is possible to show that the answer always exists.

Output

In the first line, print one integer k ($0 \le k \le \lceil \frac{n+m}{2} \rceil$) — the number of edges which you need to **remain**.

In each of the next k lines, print two integers u_i and v_i ($1 \le u_i, v_i \le n$) — the vertices, the edge between which, you need to remain. You can not print the same edge more than once.

Examples

input

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

output

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

input

```
10 20
4 3
6 5
4 5
10 8
4 8
5 8
10 4
9 5
5 1
3 8
1 2
4 7
1 4
10 7
1 7
6 1
9 6
3 9
7 9
6 2
```

output

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