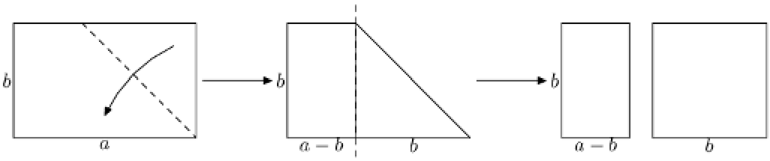
Codeforces Round #296 (Div. 2)

A. Playing with Paper

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

One day Vasya was sitting on a not so interesting Maths lesson and making an origami from a rectangular a mm $\times$ b mm sheet of paper ($a > b$). Usually the first step in making an origami is making a square piece of paper from the rectangular sheet by folding the sheet along the bisector of the right angle, and cutting the excess part.



After making a paper ship from the square piece, Vasya looked on the remaining $(a - b)$ mm $\times$ b mm strip of paper. He got the idea to use this strip of paper in the same way to make an origami, and then use the remainder (if it exists) and so on. At the moment when he is left with a square piece of paper, he will make the last ship from it and stop.

Can you determine how many ships Vasya will make during the lesson?

Input

The first line of the input contains two integers a, b ($1 \le b < a \le 10^{12}$) — the sizes of the original sheet of paper.

Output

Print a single integer — the number of ships that Vasya will make.

Sample test(s)

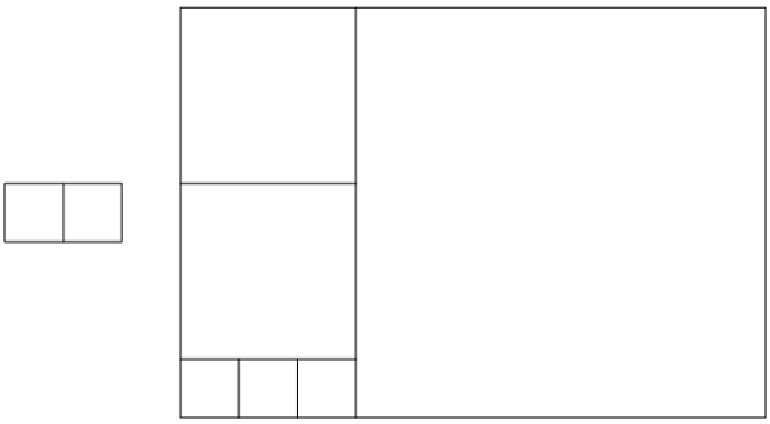
input
```
2 1
```
output
```
2
```

input
```
10 7
```
output
```
6
```

input
```
1000000000000 1
```
output
```
1000000000000
```

Note

Pictures to the first and second sample test.



B. Error Correct System

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

Ford Prefect got a job as a web developer for a small company that makes towels. His current work task is to create a search engine for the website of the company. During the development process, he needs to write a subroutine for comparing strings S and T of equal length to be "similar". After a brief search on the Internet, he learned about the *Hamming distance* between two strings S and T of the same length, which is defined as the number of positions in which S and T have different characters. For example, the Hamming distance between words "`permanent`" and "`pergament`" is two, as these words differ in the fourth and sixth letters.

Moreover, as he was searching for information, he also noticed that modern search engines have powerful mechanisms to correct errors in the request to improve the quality of search. Ford doesn't know much about human beings, so he assumed that the most common mistake in a request is swapping two arbitrary letters of the string (not necessarily adjacent). Now he wants to write a function that determines which two letters should be swapped in string S, so that the Hamming distance between a new string S and string T would be as small as possible, or otherwise, determine that such a replacement cannot reduce the distance between the strings.

Help him do this!

Input

The first line contains integer n $(1 \le n \le 200\,000)$ — the length of strings S and T.

The second line contains string S.

The third line contains string T.

Each of the lines only contains **lowercase** Latin letters.

Output

In the first line, print number x — the minimum possible Hamming distance between strings S and T if you swap at most one pair of letters in S.

In the second line, either print the indexes i and j $(1 \le i, j \le n,\ i \ne j)$, if reaching the minimum possible distance is possible by swapping letters on positions i and j, or print "`-1 -1`", if it is not necessary to swap characters.

If there are multiple possible answers, print any of them.

Sample test(s)

input
```
9
pergament
permanent
```
output
```
1
4 6
```

input
```
6
wookie
cookie
```
output
```
1
-1 -1
```

input
```
4
petr
egor
```
output
```
2
1 2
```

input
```
6
double
bundle
```
output
```
2
4 1
```

Note

In the second test it is acceptable to print $i = 2, j = 3$.

C. Glass Carving

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

Leonid wants to become a glass carver (the person who creates beautiful artworks by cutting the glass). He already has a rectangular w mm $\times$ h mm sheet of glass, a diamond glass cutter and lots of enthusiasm. What he lacks is understanding of what to carve and how.

In order not to waste time, he decided to practice the technique of carving. To do this, he makes vertical and horizontal cuts through the entire sheet. This process results in making smaller rectangular fragments of glass. Leonid does not move the newly made glass fragments. In particular, a cut divides each fragment of glass that it goes through into smaller fragments.

After each cut Leonid tries to determine what area the largest of the currently available glass fragments has. Since there appear more and more fragments, this question takes him more and more time and distracts him from the fascinating process.

Leonid offers to divide the labor — he will cut glass, and you will calculate the area of the maximum fragment after each cut. Do you agree?

Input

The first line contains three integers w, h, n $(2 \le w, h \le 200\,000,\ 1 \le n \le 200\,000)$.

Next n lines contain the descriptions of the cuts. Each description has the form $H\ y$ or $V\ x$. In the first case Leonid makes the horizontal cut at the distance y millimeters $(1 \le y \le h - 1)$ from the lower edge of the original sheet of glass. In the second case Leonid makes a vertical cut at distance x $(1 \le x \le w - 1)$ millimeters from the left edge of the original sheet of glass. It is guaranteed that Leonid won't make two identical cuts.

Output

After each cut print on a single line the area of the maximum available glass fragment in mm ^{2}.

Sample test(s)

input

```
4 3 4
H 2
V 2
V 3
V 1
```

output

```
8
4
4
2
```

input

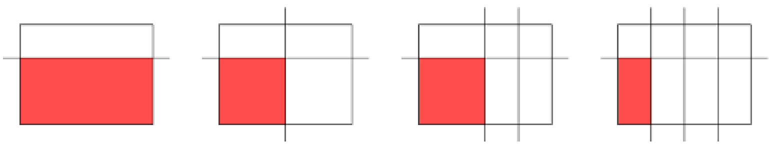
```
7 6 5
H 4
V 3
V 5
H 2
V 1
```

output

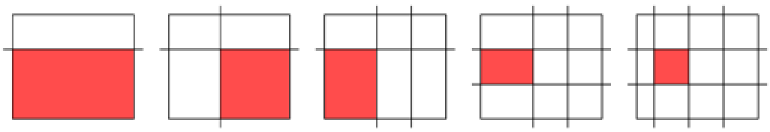
```
28
16
12
6
4
```

Note

Picture for the first sample test:



Picture for the second sample test:



D. Clique Problem

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

The clique problem is one of the most well-known NP-complete problems. Under some simplification it can be formulated as follows. Consider an undirected graph G. It is required to find a subset of vertices C of the maximum size such that any two of them are connected by an edge in graph G. Sounds simple, doesn't it? Nobody yet knows an algorithm that finds a solution to this problem in polynomial time of the size of the graph. However, as with many other NP-complete problems, the clique problem is easier if you consider a specific type of a graph.

Consider n distinct points on a line. Let the i-th point have the coordinate x_i and weight w_i. Let's form graph G, whose vertices are these points and edges connect exactly the pairs of points (i, j), such that the distance between them is not less than the sum of their weights, or more formally: $|x_i - x_j| \ge w_i + w_j$.

Find the size of the maximum clique in such graph.

Input

The first line contains the integer n $(1 \le n \le 200\ 000)$ — the number of points.

Each of the next n lines contains two numbers x_i, w_i $(0 \le x_i \le 10^9, 1 \le w_i \le 10^9)$ — the coordinate and the weight of a point. All x_i are different.

Output

Print a single number — the number of vertexes in the maximum clique of the given graph.

Sample test(s)

input

```
4
2 3
3 1
6 1
0 2
```

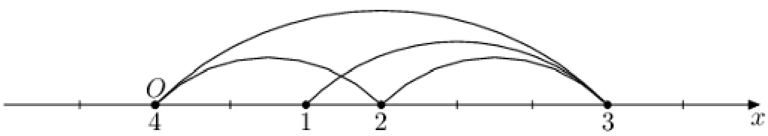
output

```
3
```

Note

If you happen to know how to solve this problem without using the specific properties of the graph formulated in the problem statement, then you are able to get a prize of one million dollars!

The picture for the sample test.



E. Data Center Drama

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

The project of a data center of a Big Software Company consists of n computers connected by m cables. Simply speaking, each computer can be considered as a box with multiple cables going out of the box. Very Important Information is transmitted along each cable in one of the two directions. As the data center plan is not yet approved, it wasn't determined yet in which direction information will go along each cable. The cables are put so that each computer is connected with each one, perhaps through some other computers.

The person in charge of the cleaning the data center will be Claudia Ivanova, the janitor. She loves to tie cables into bundles using cable ties. For some reasons, she groups the cables sticking out of a computer into groups of two, and if it isn't possible, then she gets furious and attacks the computer with the water from the bucket.

It should also be noted that due to the specific physical characteristics of the Very Important Information, it is strictly forbidden to connect in one bundle two cables where information flows in different directions.

The management of the data center wants to determine how to send information along each cable so that Claudia Ivanova is able to group all the cables coming out of each computer into groups of two, observing the condition above. Since it may not be possible with the existing connections plan, you are allowed to add the minimum possible number of cables to the scheme, and then you need to determine the direction of the information flow for each cable (yes, sometimes data centers are designed based on the janitors' convenience...)

Input

The first line contains two numbers, n and m ($1 \le n \le 100\,000$, $1 \le m \le 200\,000$) — the number of computers and the number of the already present cables, respectively.

Each of the next lines contains two numbers a_i, b_i ($1 \le a_i, b_i \le n$) — the indices of the computers connected by the i-th cable. The data centers often have a very complex structure, so a pair of computers may have more than one pair of cables between them and some cables may connect a computer with itself.

Output

In the first line print a single number p ($p \ge m$) — the minimum number of cables in the final scheme.

In each of the next p lines print a pair of numbers c_i, d_i ($1 \le c_i, d_i \le n$), describing another cable. Such entry means that information will go along a certain cable in direction from c_i to d_i.

Among the cables you printed there should be all the cables presented in the original plan in some of two possible directions. It is guaranteed that there is a solution where p doesn't exceed $500\,000$.

If there are several posible solutions with minimum possible value of p, print any of them.

Sample test(s)

input

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

output

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

input

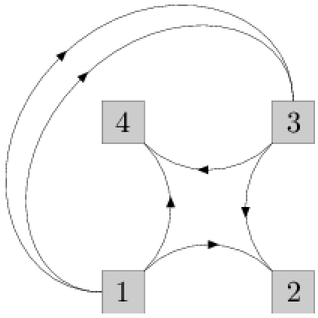
```
3 4
1 2
2 3
1 1
3 3
```

output

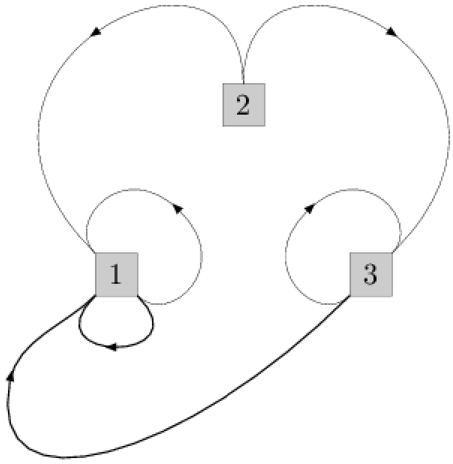
```
6
2 1
2 3
1 1
3 3
3 1
1 1
```

Note

Picture for the first sample test. The tied pairs of cables are shown going out from the same point.



Picture for the second test from the statement. The added cables are drawin in bold.



Alternative answer for the second sample test: