A. Guest From the Past

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

Kolya Gerasimov loves kefir very much. He lives in year 1984 and knows all the details of buying this delicious drink. One day, as you probably know, he found himself in year 2084, and buying kefir there is much more complicated.

Kolya is hungry, so he went to the nearest milk shop. In 2084 you may buy kefir in a plastic liter bottle, that costs a rubles, or in glass liter bottle, that costs b rubles. Also, you may return empty glass bottle and get c ($c < b$) rubles back, but you cannot return plastic bottles.

Kolya has n rubles and he is really hungry, so he wants to drink as much kefir as possible. There were no plastic bottles in his 1984, so Kolya doesn't know how to act optimally and asks for your help.

Input

First line of the input contains a single integer n ($1 \le n \le 10^{18}$) — the number of rubles Kolya has at the beginning.

Then follow three lines containing integers a, b and c ($1 \le a \le 10^{18}$, $1 \le c < b \le 10^{18}$) — the cost of one plastic liter bottle, the cost of one glass liter bottle and the money one can get back by returning an empty glass bottle, respectively.

Output

Print the only integer — maximum number of liters of kefir, that Kolya can drink.

Examples

input

```
10
11
9
8
```

output

```
2
```

input

```
10
5
6
1
```

output

```
2
```

Note

In the first sample, Kolya can buy one glass bottle, then return it and buy one more glass bottle. Thus he will drink 2 liters of kefir.

In the second sample, Kolya can buy two plastic bottle and get two liters of kefir, or he can buy one liter glass bottle, then return it and buy one plastic bottle. In both cases he will drink two liters of kefir.

B. War of the Corporations

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

A long time ago, in a galaxy far far away two giant IT-corporations Pineapple and Gogol continue their fierce competition. Crucial moment is just around the corner: Gogol is ready to release it's new tablet Lastus 3000.

This new device is equipped with specially designed artificial intelligence (AI). Employees of Pineapple did their best to postpone the release of Lastus 3000 as long as possible. Finally, they found out, that the name of the new artificial intelligence is similar to the name of the phone, that Pineapple released 200 years ago. As all rights on its name belong to Pineapple, they stand on changing the name of Gogol's artificial intelligence.

Pineapple insists, that the name of their phone occurs in the name of AI as a substring. Because the name of technology was already printed on all devices, the Gogol's director decided to replace some characters in AI name with "#". As this operation is pretty expensive, you should find the minimum number of characters to replace with "#", such that the name of AI doesn't contain the name of the phone as a substring.

Substring is a continuous subsequence of a string.

Input

The first line of the input contains the name of AI designed by Gogol, its length doesn't exceed $100\,000$ characters. Second line contains the name of the phone released by Pineapple 200 years ago, its length doesn't exceed 30. Both string are non-empty and consist of only small English letters.

Output

Print the minimum number of characters that must be replaced with "#" in order to obtain that the name of the phone doesn't occur in the name of AI as a substring.

Examples

input

```
intellect
tell
```

output

```
1
```

input

```
google
apple
```

output

```
0
```

input

```
sirisiri
sir
```

output

```
2
```

Note

In the first sample AI's name may be replaced with "`int#llect`".

In the second sample Gogol can just keep things as they are.

In the third sample one of the new possible names of AI may be "`s#ris#ri`".

C. K-special Tables

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

People do many crazy things to stand out in a crowd. Some of them dance, some learn by heart rules of Russian language, some try to become an outstanding competitive programmers, while others collect funny math objects.

Alis is among these collectors. Right now she wants to get one of k-special tables. In case you forget, the table $n \times n$ is called k-special if the following three conditions are satisfied:

- every integer from 1 to n^2 appears in the table exactly once;
- in each row numbers are situated in increasing order;
- the sum of numbers in the k-th column is maximum possible.

Your goal is to help Alice and find at least one k-special table of size $n \times n$. Both rows and columns are numbered from 1 to n, with rows numbered from top to bottom and columns numbered from left to right.

Input

The first line of the input contains two integers n and k ($1 \le n \le 500, 1 \le k \le n$) — the size of the table Alice is looking for and the column that should have maximum possible sum.

Output

First print the sum of the integers in the k-th column of the required table.

Next n lines should contain the description of the table itself: first line should contains n elements of the first row, second line should contain n elements of the second row and so on.

If there are multiple suitable table, you are allowed to print any.

Examples

input

```
4 1
```

output

```
28
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

input

```
5 3
```

output

```
85
5 6 17 18 19
9 10 23 24 25
7 8 20 21 22
3 4 14 15 16
1 2 11 12 13
```

D. Finals in arithmetic

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

Vitya is studying in the third grade. During the last math lesson all the pupils wrote on arithmetic quiz. Vitya is a clever boy, so he managed to finish all the tasks pretty fast and Oksana Fillipovna gave him a new one, that is much harder.

Let's denote a *flip operation* of an integer as follows: number is considered in decimal notation and then reverted. If there are any leading zeroes afterwards, they are thrown away. For example, if we flip 123 the result is the integer 321, but flipping 130 we obtain 31, and by flipping 31 we come to 13.

Oksana Fillipovna picked some number a without leading zeroes, and flipped it to get number a_r. Then she summed a and a_r, and told Vitya the resulting value n. His goal is to find any valid a.

As Oksana Fillipovna picked some small integers as a and a_r, Vitya managed to find the answer pretty fast and became interested in finding some general algorithm to deal with this problem. Now, he wants you to write the program that for given n finds any a without leading zeroes, such that $a + a_r = n$ or determine that such a doesn't exist.

Input

The first line of the input contains a single integer n ($1 \le n \le 10^{100\,000}$).

Output

If there is no such positive integer a without leading zeroes that $a + a_r = n$ then print 0. Otherwise, print any valid a. If there are many possible answers, you are allowed to pick any.

Examples

input

```
4
```

output

```
2
```

input

```
11
```

output

```
10
```

input

```
5
```

output

```
0
```

input

```
33
```

output

```
21
```

Note

In the first sample $4 = 2 + 2$, $a = 2$ is the only possibility.

In the second sample $11 = 10 + 1$, $a = 10$ — the only valid solution. Note, that $a = 01$ is incorrect, because a can't have leading zeroes.

It's easy to check that there is no suitable a in the third sample.

In the fourth sample $33 = 30 + 3 = 12 + 21$, so there are three possibilities for a: $a = 30$, $a = 12$, $a = 21$. Any of these is considered to be correct answer.

E. Frog Fights

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

Ostap Bender recently visited frog farm and was inspired to create his own frog game.

Number of frogs are places on a cyclic gameboard, divided into m cells. Cells are numbered from 1 to m, but the board is cyclic, so cell number 1 goes right after the cell number m in the direction of movement. i-th frog during its turn can jump for a_i cells.

Frogs move in turns, game starts with a move by frog 1. On its turn i-th frog moves a_i cells forward, knocking out all the frogs on its way. If there is a frog in the last cell of the path of the i-th frog, that frog is also knocked out. After this the value a_i is decreased by the number of frogs that were knocked out during this turn. If a_i is zero or goes negative, then i-th frog doesn't make moves anymore.

After frog number 1 finishes its turn, frog number 2 starts to move, then frog number 3 and so on. After the frog number n makes its move, frog 1 starts to move again, then frog 2 and so on this process goes forever. If some frog was already knocked out from the board, we consider that it skips all its moves.

Help Ostap to identify, what frogs will stay on the board at the end of a game?

Input

First line of the input contains two integers n and m ($1 \le n \le 100000, 1 \le m \le 10^9, n \le m$) — number of frogs and gameboard size, respectively.

Following n lines contains frogs descriptions — two integers p_i and a_i ($1 \le p_i, a_i \le m$) — the number of cell occupied by i-th frog initially and initial jump length. All p_i are guaranteed to be distinct.

Output

In the first line output number of frogs on the final gameboard. In the second line output their numbers in any order.

Examples

input

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

output

```
1
3
```

input

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

output

```
2
1 4
```

Note

In the first sample first frog jumps 1 cell and finishes in cell number 3. Second frog jumps for 3 cells and finishes on cell number 3, knocking out frog number 1. Current jump length for frog number 2 is now 2. Third frog jumps to cell 2, then second frog jumps to cell 5. Third frog in turn finishes in cell 5 and removes frog 2 from the gameboard. Now, it's the only remaining frog in the game.

In the second sample first frog jumps 2 cells and knocks out frogs in cells 2 and 3. Its value a_i is now 0. Then fourth frog jumps and knocks out fifth frog and its a_i is now 0 too. These two frogs will remains on the gameboard forever.