Technocup 2017 - Elimination Round 3

A. Santa Claus and a Place in a Class

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

Santa Claus is the first who came to the Christmas Olympiad, and he is going to be the first to take his place at a desk! In the classroom there are n lanes of m desks each, and there are two working places at each of the desks. The lanes are numbered from 1 to n from the left to the right, the desks in a lane are numbered from 1 to m starting from the blackboard. Note that the lanes go perpendicularly to the blackboard, not along it (see picture).

The organizers numbered all the working places from 1 to $2nm$. The places are numbered by lanes (i. e. all the places of the first lane go first, then all the places of the second lane, and so on), in a lane the places are numbered starting from the nearest to the blackboard (i. e. from the first desk in the lane), at each desk, the place on the left is numbered before the place on the right.

The picture illustrates the first and the second samples.

Santa Clause knows that his place has number k. Help him to determine at which lane at which desk he should sit, and whether his place is on the left or on the right!

Input

The only line contains three integers n, m and k ($1 \le n, m \le 10\,000$, $1 \le k \le 2nm$) — the number of lanes, the number of desks in each lane and the number of Santa Claus' place.

Output

Print two integers: the number of lane r, the number of desk d, and a character s, which stands for the side of the desk Santa Claus. The character s should be "`L`", if Santa Clause should sit on the left, and "`R`" if his place is on the right.

Examples

input
```
4 3 9
```
output
```
2 2 L
```

input
```
4 3 24
```
output
```
4 3 R
```

input
```
2 4 4
```
output
```
1 2 R
```

Note

The first and the second samples are shown on the picture. The green place corresponds to Santa Claus' place in the first example, the blue place corresponds to Santa Claus' place in the second example.

In the third sample there are two lanes with four desks in each, and Santa Claus has the fourth place. Thus, his place is in the first lane at the second desk on the right.

B. Santa Claus and Keyboard Check

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Santa Claus decided to disassemble his keyboard to clean it. After he returned all the keys back, he suddenly realized that some pairs of keys took each other's place! That is, Santa suspects that each key is either on its place, or on the place of another key, which is located exactly where the first key should be.

In order to make sure that he's right and restore the correct order of keys, Santa typed his favorite patter looking only to his keyboard.

You are given the Santa's favorite patter and the string he actually typed. Determine which pairs of keys could be mixed. Each key must occur in pairs **at most once**.

Input

The input consists of only two strings s and t denoting the favorite Santa's patter and the resulting string. s and t are not empty and have the same length, which is at most 1000. Both strings consist only of lowercase English letters.

Output

If Santa is wrong, and there is no way to divide some of keys into pairs and swap keys in each pair so that the keyboard will be fixed, print «-1» (without quotes).

Otherwise, the first line of output should contain the only integer k ($k \geq 0$) — the number of pairs of keys that should be swapped. The following k lines should contain two space-separated letters each, denoting the keys which should be swapped. All printed letters must be distinct.

If there are several possible answers, print any of them. You are free to choose the order of the pairs and the order of keys in a pair.

Each letter must occur at most once. Santa considers the keyboard to be fixed if he can print his favorite patter without mistakes.

Examples

input

```
helloworld
ehoolwlroz
```

output

```
3
h e
l o
d z
```

input

```
hastalavistababy
hastalavistababy
```

output

```
0
```

input

```
merrychristmas
christmasmerry
```

output

```
-1
```

C. Santa Claus and Robot

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Santa Claus has Robot which lives on the infinite grid and can move **along its lines**. He can also, having a sequence of m points $p_1, p_2, ..., p_m$ with integer coordinates, do the following: denote its initial location by p_0. First, the robot will move from p_0 to p_1 along one of the shortest paths between them (please notice that since the robot moves only along the grid lines, there can be several shortest paths). Then, after it reaches p_1, it'll move to p_2, again, choosing one of the shortest ways, then to p_3, and so on, until he has visited all points in the given order. Some of the points in the sequence may coincide, in that case Robot will visit that point several times according to the sequence order.

While Santa was away, someone gave a sequence of points to Robot. This sequence is now lost, but Robot saved the protocol of its unit movements. Please, find the minimum possible length of the sequence.

Input

The first line of input contains the only positive integer n ($1 \le n \le 2 \cdot 10^5$) which equals the number of unit segments the robot traveled. The second line contains the movements protocol, which consists of n letters, each being equal either `L`, or `R`, or `U`, or `D`. k-th letter stands for the direction which Robot traveled the k-th unit segment in: `L` means that it moved to the left, `R` — to the right, `U` — to the top and `D` — to the bottom. Have a look at the illustrations for better explanation.

Output

The only line of input should contain the minimum possible length of the sequence.

Examples

input

```
4
RURD
```

output

```
2
```

input

```
6
RRULDD
```

output

```
2
```

input

```
26
RRRULURURUULULLLDLDDRDRDLD
```

output

```
7
```

input

```
3
RLL
```

output

```
2
```

input

```
4
LRLR
```

output

```
4
```

Note

The illustrations to the first three tests are given below.

The last example illustrates that each point in the sequence should be counted as many times as it is presented in the sequence.

D. Santa Claus and a Palindrome

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

Santa Claus likes palindromes very much. There was his birthday recently. k of his friends came to him to congratulate him, and each of them presented to him a string s_i having the same length n. We denote the beauty of the i-th string by a_i. It can happen that a_i is negative — that means that Santa doesn't find this string beautiful at all.

Santa Claus is crazy about palindromes. He is thinking about the following question: what is the maximum possible total beauty of a palindrome which can be obtained by concatenating some (possibly all) of the strings he has? Each present can be used at most once. Note that all strings have **the same length** n.

Recall that a palindrome is a string that doesn't change after one reverses it.

Since the empty string is a palindrome too, the answer can't be negative. Even if all a_i's are negative, Santa can obtain the empty string.

Input

The first line contains two positive integers k and n divided by space and denoting the number of Santa friends and the length of every string they've presented, respectively ($1 \le k, n \le 100\,000$; $n \cdot k \le 100\,000$).

k lines follow. The i-th of them contains the string s_i and its beauty a_i ($-10\,000 \le a_i \le 10\,000$). The string consists of n lowercase English letters, and its beauty is integer. Some of strings may coincide. Also, equal strings can have different beauties.

Output

In the only line print the required maximum possible beauty.

Examples

input

```
7 3
abb 2
aaa -3
bba -1
zyz -4
abb 5
aaa 7
xyx 4
```

output

```
12
```

input

```
3 1
a 1
a 2
a 3
```

output

```
6
```

input

```
2 5
abcde 10000
abcde 10000
```

output

```
0
```

Note

In the first example Santa can obtain abbaaaxyxaaabba by concatenating strings 5, 2, 7, 6 and 3 (in this order).

E. Santa Claus and Tangerines

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

Santa Claus has n tangerines, and the i-th of them consists of exactly a_i slices. Santa Claus came to a school which has k pupils. Santa decided to treat them with tangerines.

However, there can be too few tangerines to present at least one tangerine to each pupil. So Santa decided to divide tangerines into parts so that no one will be offended. In order to do this, he can divide a tangerine or any existing part into two smaller equal parts. If the number of slices in the part he wants to split is odd, then one of the resulting parts will have one slice more than the other. It's forbidden to divide a part consisting of only one slice.

Santa Claus wants to present to everyone either a whole tangerine or exactly one part of it (that also means that everyone must get a positive number of slices). One or several tangerines or their parts may stay with Santa.

Let b_i be the number of slices the i-th pupil has in the end. Let Santa's *joy* be the minimum among all b_i's.

Your task is to find the maximum possible *joy* Santa can have after he treats everyone with tangerines (or their parts).

Input

The first line contains two positive integers n and k ($1 \le n \le 10^6$, $1 \le k \le 2 \cdot 10^9$) denoting the number of tangerines and the number of pupils, respectively.

The second line consists of n positive integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^7$), where a_i stands for the number of slices the i-th tangerine consists of.

Output

If there's no way to present a tangerine or a part of tangerine to everyone, print `-1`. Otherwise, print the maximum possible *joy* that Santa can have.

Examples

input

```
3 2
5 9 3
```

output

```
5
```

input

```
2 4
12 14
```

output

```
6
```

input

```
2 3
1 1
```

output

```
-1
```

Note

In the first example Santa should divide the second tangerine into two parts with 5 and 4 slices. After that he can present the part with 5 slices to the first pupil and the whole first tangerine (with 5 slices, too) to the second pupil.

In the second example Santa should divide both tangerines, so that he'll be able to present two parts with 6 slices and two parts with 7 slices.

In the third example Santa Claus can't present 2 slices to 3 pupils in such a way that everyone will have anything.

F. Santa Clauses and a Soccer Championship

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

The country Treeland consists of n cities connected with $n - 1$ bidirectional roads in such a way that it's possible to reach every city starting from any other city using these roads. There will be a soccer championship next year, and all participants are Santa Clauses. There are exactly $2k$ teams from $2k$ different cities.

During the first stage all teams are divided into k pairs. Teams of each pair play two games against each other: one in the hometown of the first team, and the other in the hometown of the other team. Thus, each of the $2k$ cities holds exactly one soccer game. However, it's not decided yet how to divide teams into pairs.

It's also necessary to choose several cities to settle players in. Organizers tend to use **as few cities as possible** to settle the teams.

Nobody wants to travel too much during the championship, so if a team plays in cities u and v, it wants to live in one of the cities on the shortest path between u and v (maybe, in u or in v). There is another constraint also: the teams from one pair must live in the same city.

Summarizing, the organizers want to divide $2k$ teams into pairs and settle them in the minimum possible number of cities m in such a way that teams from each pair live in the same city which lies between their hometowns.

Input

The first line of input contains two integers n and k ($2 \le n \le 2 \cdot 10^5$, $2 \le 2k \le n$) — the number of cities in Treeland and the number of pairs of teams, respectively.

The following $n - 1$ lines describe roads in Treeland: each of these lines contains two integers a and b ($1 \le a, b \le n$, $a \ne b$) which mean that there is a road between cities a and b. It's guaranteed that there is a path between any two cities.

The last line contains $2k$ distinct integers $c_1, c_2, ..., c_{2k}$ ($1 \le c_i \le n$), where c_i is the hometown of the i-th team. All these numbers are distinct.

Output

The first line of output must contain the only positive integer m which should be equal to the minimum possible number of cities the teams can be settled in.

The second line should contain m distinct numbers $d_1, d_2, ..., d_m$ ($1 \le d_i \le n$) denoting the indices of the cities where the teams should be settled.

The k lines should follow, the j-th of them should contain 3 integers u_j, v_j and x_j, where u_j and v_j are the hometowns of the j-th pair's teams, and x_j is the city they should live in during the tournament. Each of the numbers $c_1, c_2, ..., c_{2k}$ should occur in all u_j's and v_j's exactly once. Each of the numbers x_j should belong to $\{d_1, d_2, ..., d_m\}$.

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

Example

input

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

output

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

Note

In the first test the orginizers can settle all the teams in the city number 2. The way to divide all teams into pairs is not important, since all requirements are satisfied anyway, because the city 2 lies on the shortest path between every two cities from $\{2, 4, 5, 6\}$.