Codeforces Round #135 (Div. 2)

A. k-String

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

A string is called a k-string if it can be represented as k concatenated copies of some string. For example, the string "aabaabaabaab" is at the same time a 1-string, a 2-string and a 4-string, but it is not a 3-string, a 5-string, or a 6-string and so on. Obviously any string is a 1-string.

You are given a string s, consisting of lowercase English letters and a positive integer k. Your task is to reorder the letters in the string s in such a way that the resulting string is a k-string.

Input

The first input line contains integer k ($1 \le k \le 1000$). The second line contains s, all characters in s are lowercase English letters. The string length s satisfies the inequality $1 \le |s| \le 1000$, where $|s|$ is the length of string s.

Output

Rearrange the letters in string s in such a way that the result is a k-string. Print the result on a single output line. If there are multiple solutions, print any of them.

If the solution doesn't exist, print "-1" (without quotes).

Sample test(s)

input
2 aazz
output
azaz

input
3 abcabcabz
output
-1

B. Special Offer! Super Price 999 Bourles!

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

Polycarpus is an amateur businessman. Recently he was surprised to find out that the market for paper scissors is completely free! Without further ado, Polycarpus decided to start producing and selling such scissors.

Polycaprus calculated that the optimal celling price for such scissors would be p bourles. However, he read somewhere that customers are attracted by prices that say something like "Special Offer! Super price 999 bourles!". So Polycarpus decided to lower the price a little if it leads to the desired effect.

Polycarpus agrees to lower the price by no more than d bourles so that the number of nines at the end of the resulting price is maximum. If there are several ways to do it, he chooses the maximum possible price.

Note, Polycarpus counts only the *trailing* nines in a price.

Input

The first line contains two integers p and d ($1 \le p \le 10^{18}$; $0 \le d < p$) — the initial price of scissors and the maximum possible price reduction.

Please, do not use the %lld specifier to read or write 64-bit integers in C++. It is preferred to use cin, cout streams or the %I64d specifier.

Output

Print the required price — the maximum price that ends with the largest number of nines and that is less than p by no more than d.

The required number shouldn't have leading zeroes.

Sample test(s)

input
1029 102
output
999

input
27191 17
output
27189

C. Color Stripe

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

A colored stripe is represented by a horizontal row of n square cells, each cell is pained one of k colors. Your task is to repaint the minimum number of cells so that no two neighbouring cells are of the same color. You can use any color from 1 to k to repaint the cells.

Input

The first input line contains two integers n and k ($1 \le n \le 5 \cdot 10^5$; $2 \le k \le 26$). The second line contains n uppercase English letters. Letter "A" stands for the first color, letter "B" stands for the second color and so on. The first k English letters may be used. Each letter represents the color of the corresponding cell of the stripe.

Output

Print a single integer — the required minimum number of repaintings. In the second line print any possible variant of the repainted stripe.

Sample test(s)

input
6 3 ABBACC
output
2 ABCACA

input
3 2 BBB
output
1 BAB

D. Choosing Capital for Treeland

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The country Treeland consists of n cities, some pairs of them are connected with *unidirectional* roads. Overall there are $n - 1$ roads in the country. We know that if we don't take the direction of the roads into consideration, we can get from any city to any other one.

The council of the elders has recently decided to choose the capital of Treeland. Of course it should be a city of this country. The council is supposed to meet in the capital and regularly move from the capital to other cities (at this stage nobody is thinking about getting back to the capital from these cities). For that reason if city a is chosen a capital, then all roads must be oriented so that if we move along them, we can get from city a to any other city. For that some roads may have to be inversed.

Help the elders to choose the capital so that they have to inverse the minimum number of roads in the country.

Input

The first input line contains integer n ($2 \le n \le 2 \cdot 10^5$) — the number of cities in Treeland. Next $n - 1$ lines contain the descriptions of the roads, one road per line. A road is described by a pair of integers s_i, t_i ($1 \le s_i, t_i \le n$; $s_i \ne t_i$) — the numbers of cities, connected by that road. The i-th road is oriented from city s_i to city t_i. You can consider cities in Treeland indexed from 1 to n.

Output

In the first line print the minimum number of roads to be inversed if the capital is chosen optimally. In the second line print all possible ways to choose the capital — a sequence of indexes of cities in the increasing order.

Sample test(s)

input
3 2 1 2 3
output
0 2

input
4 1 4 2 4 3 4
output
2 1 2 3

E. Parking Lot

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

A parking lot in the City consists of n parking spaces, standing in a line. The parking spaces are numbered from 1 to n from left to right.

When a car arrives at the lot, the operator determines an empty parking space for it. For the safety's sake the chosen place should be located as far from the already occupied places as possible. That is, the closest occupied parking space must be as far away as possible. If there are several such places, then the operator chooses the place with the minimum index from them. If all parking lot places are empty, then the car gets place number 1.

We consider the distance between the i-th and the j-th parking spaces equal to $4 \cdot |i - j|$ meters.

You are given the parking lot records of arriving and departing cars in the chronological order. For each record of an arriving car print the number of the parking lot that was given to this car.

Input

The first line contains two space-separated integers n and m $(1 \le n, m \le 2 \cdot 10^5)$ — the number of parking places and the number of records correspondingly.

Next m lines contain the descriptions of the records, one per line. The i-th line contains numbers t_i, id_i $(1 \le t_i \le 2;\ 1 \le id_i \le 10^6)$. If t_i equals 1, then the corresponding record says that the car number id_i arrived at the parking lot. If t_i equals 2, then the corresponding record says that the car number id_i departed from the parking lot.

Records about arriving to the parking lot and departing from the parking lot are given chronologically. All events occurred consecutively, no two events occurred simultaneously.

It is guaranteed that all entries are correct:

- each car arrived at the parking lot at most once and departed from the parking lot at most once,
- there is no record of a departing car if it didn't arrive at the parking lot earlier,
- there are no more than n cars on the parking lot at any moment.

You can consider the cars arbitrarily numbered from 1 to 10^6, all numbers are distinct. Initially all places in the parking lot are empty.

Output

For each entry of an arriving car print the number of its parking space. Print the numbers of the spaces in the order, in which the cars arrive to the parking lot.

Sample test(s)

input
7 11 1 15 1 123123 1 3 1 5 2 123123 2 15 1 21 2 3 1 6 1 7 1 8
output
1 7 4 2 7 4 1 3