A. Forgotten Episode

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

Polycarpus adores TV series. Right now he is ready to finish watching a season of a popular sitcom "Graph Theory". In total, the season has n episodes, numbered with integers from 1 to n.

Polycarpus watches episodes not one by one but in a random order. He has already watched all the episodes except for one. Which episode has Polycaprus forgotten to watch?

Input

The first line of the input contains integer n $(2 \le n \le 100000)$ — the number of episodes in a season. Assume that the episodes are numbered by integers from 1 to n.

The second line contains $n - 1$ integer $a_1, a_2, ..., a_n$ $(1 \le a_i \le n)$ — the numbers of episodes that Polycarpus has watched. All values of a_i are distinct.

Output

Print the number of the episode that Polycarpus hasn't watched.

Examples

input

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

output

```
4
```

B. Balancer

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya has k matches, placed in n matchboxes lying in a line from left to right. We know that k is divisible by n. Petya wants all boxes to have the same number of matches inside. For that, he can move a match from its box to the adjacent one in one move. How many such moves does he need to achieve the desired configuration?

Input

The first line contains integer n ($1 \le n \le 50000$). The second line contains n non-negative numbers that do not exceed 10^9, the i-th written number is the number of matches in the i-th matchbox. It is guaranteed that the total number of matches is divisible by n.

Output

Print the total minimum number of moves.

Examples

input

```
6
1 6 2 5 3 7
```

output

```
12
```

C. One-Based Arithmetic

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Prof. Vasechkin wants to represent positive integer n as a sum of addends, where each addends is an integer number containing only 1s. For example, he can represent 121 as 121=111+11+–1. Help him to find the least number of digits 1 in such sum.

Input

The first line of the input contains integer n ($1 \le n < 10^{15}$).

Output

Print expected minimal number of digits 1.

Examples

input

```
121
```

output

```
6
```

D. Berland Federalization

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

Recently, Berland faces federalization requests more and more often. The proponents propose to divide the country into separate states. Moreover, they demand that there is a state which includes exactly k towns.

Currently, Berland has n towns, some pairs of them are connected by bilateral roads. Berland has only $n - 1$ roads. You can reach any city from the capital, that is, the road network forms a *tree*.

The Ministry of Roads fears that after the reform those roads that will connect the towns of different states will bring a lot of trouble.

Your task is to come up with a plan to divide the country into states such that:

- each state is connected, i.e. for each state it is possible to get from any town to any other using its roads (that is, the roads that connect the state towns),
- there is a state that consisted of exactly k cities,
- the number of roads that connect different states is minimum.

Input

The first line contains integers n, k ($1 \le k \le n \le 400$). Then follow $n - 1$ lines, each of them describes a road in Berland. The roads are given as pairs of integers x_i, y_i ($1 \le x_i, y_i \le n$; $x_i \ne y_i$) — the numbers of towns connected by the road. Assume that the towns are numbered from 1 to n.

Output

The the first line print the required minimum number of "problem" roads t. Then print a sequence of t integers — their indices in the found division. The roads are numbered starting from 1 in the order they follow in the input. If there are multiple possible solutions, print any of them.

If the solution shows that there are no "problem" roads at all, print a single integer `0` and either leave the second line empty or do not print it at all.

Examples

input

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

output

```
1
2
```

input

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

output

```
2
3 4
```

input

```
1 1
```

output

```
0
```