VK Cup 2012 Finals

A. Privatization

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

There is a developed network of flights between Berland and Beerland. All of them belong to the Berland state company BerAvia. Each flight connects some Berland city with some Beerland city. For each flight airplanes fly in both directions.

Changes are coming to Berland — the state decided to privatize BerAvia, namely, to sell out all flights to t private companies. Each of these companies wants to get the maximal number of flights, so if the Berland flights are sold unevenly, Berland can be accused of partiality. Berland Government decided to sell the flights as evenly as possible between the t companies.

The *unevenness* of the distribution of flights between companies is calculated as follows. For each city i (both Berland and Beerland) we'll calculate the value of

$$w_i = \max_j a_{ij} - \min_j a_{ij},$$

where a_{ij} is the number of flights from city i, which belong to company j. The sum of w_i for all cities in both countries is called the unevenness of the distribution. The distribution with the minimal unevenness is the most even one.
Help the Berland government come up with the most even distribution plan of selling flights.

Input

The first input line contains four integers n, m, k and t ($1 \le n, m, t \le 200; 1 \le k \le 5000$), where n, m are the numbers of cities in Berland and Beerland, correspondingly, k is the number of flights between them, and t is the number of private companies. Next k lines describe the flights, one per line, as pairs of positive integers x_i, y_i ($1 \le x_i \le n; 1 \le y_i \le m$), where x_i and y_i are the indexes of cities in Berland and Beerland, correspondingly, connected by the i-th flight. There is at most one flight between any pair of cities, each flight connects cities of different countries. The cities in Berland are indexed from 1 to n, and in Beerland — from 1 to m.

Output

Print the unevenness of the sought plan on the first line. On the second line print a sequence of k integers $c_1, c_2, ..., c_k$ ($1 \le c_i \le t$), where c_i is the index of the company that should buy the i-th flight. Assume that the flights are indexed from 1 to k in the order they appear in the input. If there are multiple solutions, print any of them.

Sample test(s)

input

```
3 5 8 2
1 4
1 3
3 3
1 2
1 1
2 1
1 5
2 2
```

output

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

B. Polycarpus is Looking for Good Substrings

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

We'll call string $s[a, b] = s_a s_{a+1} \dots s_b$ $(1 \le a \le b \le |s|)$ a *substring* of string $s = s_1 s_2 \dots s_{|s|}$, where $|s|$ is the length of string s.

The *trace* of a non-empty string t is a set of characters that the string consists of. For example, the trace of string "`aab`" equals {'`a`', '`b`'}.

Let's consider an arbitrary string s and the set of its substrings with trace equal to C. We will denote the number of substrings from this set that are maximal by inclusion by $r(C, s)$. Substring $s[a, b]$ of length $n = b - a + 1$ belonging to some set is called maximal by inclusion, if there is no substring $s[x, y]$ in this set with length greater than n, such that $1 \le x \le a \le b \le y \le |s|$. Two substrings of string s are considered different even if they are equal but they are located at different positions of s.

Polycarpus got a challenging practical task on a stringology exam. He must do the following: given string s and non-empty sets of characters C_1, C_2, ..., C_m, find $r(C_i, s)$ for each set C_i. Help Polycarpus to solve the problem as he really doesn't want to be expelled from the university and go to the army!

Input

The first line contains a non-empty string s $(1 \le |s| \le 10^6)$.

The second line contains a single integer m $(1 \le m \le 10^4)$. Next m lines contain descriptions of sets C_i. The i-th line contains string c_i such that its trace equals C_i. It is guaranteed that all characters of each string c_i are different.

Note that C_i are not necessarily different. All given strings consist of lowercase English letters.

Output

Print m integers — the i-th integer must equal $r(C_i, s)$.

Sample test(s)

input

```
aaaaa
2
a
a
```

output

```
1
1
```

input

```
abacaba
3
ac
ba
a
```

output

```
1
2
4
```

C. Cowboys

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

A very tense moment: n cowboys stand in a circle and each one points his colt at a neighbor. Each cowboy can point the colt to the person who follows or precedes him in clockwise direction. Human life is worthless, just like in any real western.

The picture changes each second! Every second the cowboys analyse the situation and, if a pair of cowboys realize that they aim at each other, they turn around. In a second all such pairs of neighboring cowboys aiming at each other turn around. All actions happen instantaneously and simultaneously in a second.

We'll use character "A" to denote a cowboy who aims at his neighbour in the clockwise direction, and character " B" for a cowboy who aims at his neighbour in the counter clockwise direction. Then a string of letters "A" and "B" will denote the circle of cowboys, the record is made from the first of them in a clockwise direction.

For example, a circle that looks like "ABBBABBBA" after a second transforms into "BABBBABBA" and a circle that looks like "BABBA" transforms into "ABABB".

This picture illustrates how the circle "BABBA" transforms into "ABABB"

A second passed and now the cowboys' position is described by string s. Your task is to determine the number of possible states that lead to s in a second. Two states are considered distinct if there is a cowboy who aims at his clockwise neighbor in one state and at his counter clockwise neighbor in the other state.

Input

The input data consists of a single string s. Its length is from 3 to 100 characters, inclusive. Line s consists of letters "A" and "B".

Output

Print the sought number of states.

Sample test(s)

input
```
BABBBABBA
```
output
```
2
```

input
```
ABABB
```
output
```
2
```

input
```
ABABAB
```
output
```
4
```

Note

In the first sample the possible initial states are "ABBBABBAB" and "ABBBABBBA".

In the second sample the possible initial states are "AABBB" and "BABBA".

D. Cutting a Fence

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya the carpenter has an estate that is separated from the wood by a fence. The fence consists of n planks put in a line. The fence is not closed in a circle. The planks are numbered from left to right from 1 to n, the i-th plank is of height a_i. All planks have the same width, the lower edge of each plank is located at the ground level.

Recently a local newspaper "Malevich and Life" wrote that the most fashionable way to decorate a fence in the summer is to draw a fuchsia-colored rectangle on it, the lower side of the rectangle must be located at the lower edge of the fence.

Vasya is delighted with this idea! He immediately bought some fuchsia-colored paint and began to decide what kind of the rectangle he should paint. Vasya is sure that the rectangle should cover k consecutive planks. In other words, he will paint planks number x, $x+1$, ..., $x+k-1$ for some x $(1 \le x \le n-k+1)$. He wants to paint the rectangle of maximal area, so the rectangle height equals $min\ a_i$ for $x \le i \le x+k-1$, x is the number of the first colored plank.

Vasya has already made up his mind that the rectangle width can be equal to one of numbers of the sequence $k_1, k_2, ..., k_m$. For each k_i he wants to know the expected height of the painted rectangle, provided that he selects x for such fence uniformly among all $n-k_i+1$ possible values. Help him to find the expected heights.

Input

The first line contains a single integer n $(1 \le n \le 10^6)$ — the number of planks in the fence. The second line contains a sequence of integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ where a_i is the height of the i-th plank of the fence.

The third line contains an integer m $(1 \le m \le 10^6)$ and the next line contains m space-separated integers $k_1, k_2, ..., k_m$ $(1 \le k_i \le n)$ where k_i is the width of the desired fuchsia-colored rectangle in planks.

Output

Print m whitespace-separated real numbers, the i-th number equals the expected value of the rectangle height, if its width in planks equals k_i. The value will be considered correct if its absolute or relative error doesn't exceed 10^{-9}.

Sample test(s)

input

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

output

```
2.000000000000000
1.500000000000000
1.000000000000000
```

input

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

output

```
1.000000000000000
1.000000000000000
1.000000000000000
```

Note

Let's consider the first sample test.

- There are three possible positions of the fence for $k_1 = 1$. For the first position $(x = 1)$ the height is 3, for the second one $(x = 2)$ the height is 2, for the third one $(x = 3)$ the height is 1. As the fence position is chosen uniformly, the expected height of the fence equals $\frac{3+2+1}{3} = 2$;
- There are two possible positions of the fence for $k_2 = 2$. For the first position $(x = 1)$ the height is 2, for the second one $(x = 2)$ the height is 1. The expected height of the fence equals $\frac{2+1}{2} = 1.5$;
- There is the only possible position of the fence for $k_3 = 3$. The expected height of the fence equals 1.

E. IT Restaurants

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

City N. has a huge problem with roads, food and IT-infrastructure. In total the city has n junctions, some pairs of them are connected by bidirectional roads. The road network consists of n - 1 roads, you can get from any junction to any other one by these roads. Yes, you're right — the road network forms an undirected tree.

Recently, the Mayor came up with a way that eliminates the problems with the food and the IT-infrastructure at the same time! He decided to put at the city junctions restaurants of two well-known cafe networks for IT professionals: "iMac D0naldz" and "Burger Bing". Since the network owners are not friends, it is strictly prohibited to place two restaurants of different networks on neighboring junctions. There are other requirements. Here's the full list:

- each junction must have at most one restaurant;
- each restaurant belongs either to "iMac D0naldz", or to "Burger Bing";
- each network should build at least one restaurant;
- there is no pair of junctions that are connected by a road and contains restaurants of different networks.

The Mayor is going to take a large tax from each restaurant, so he is interested in making the total number of the restaurants as large as possible.

Help the Mayor to analyze the situation. Find all such pairs of (a, b) that a restaurants can belong to "iMac D0naldz", b restaurants can belong to "Burger Bing", and the sum of $a + b$ is as large as possible.

Input

The first input line contains integer n $(3 \le n \le 5000)$ — the number of junctions in the city. Next n - 1 lines list all roads one per line. Each road is given as a pair of integers x_i, y_i $(1 \le x_i, y_i \le n)$ — the indexes of connected junctions. Consider the junctions indexed from 1 to n.

It is guaranteed that the given road network is represented by an undirected tree with n vertexes.

Output

Print on the first line integer z — the number of sought pairs. Then print all sought pairs (a, b) in the order of increasing of the first component a.

Sample test(s)

input

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

output

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

input

```
10
1 2
2 3
3 4
5 6
6 7
7 4
8 9
9 10
10 4
```

output

```
6
1 8
2 7
3 6
6 3
7 2
8 1
```

Note

The figure below shows the answers to the first test case. The junctions with "iMac D0naldz" restaurants are marked red and "Burger Bing" restaurants are marked blue.