Codeforces Round #339 (Div. 1)

A. Peter and Snow Blower

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

Peter got a new snow blower as a New Year present. Of course, Peter decided to try it immediately. After reading the instructions he realized that it does not work like regular snow blowing machines. In order to make it work, you need to tie it to some point that it does not cover, and then switch it on. As a result it will go along a circle around this point and will remove all the snow from its path.

Formally, we assume that Peter's machine is a polygon on a plane. Then, after the machine is switched on, it will make a circle around the point to which Peter tied it (this point lies strictly outside the polygon). That is, each of the points lying within or on the border of the polygon will move along the circular trajectory, with the center of the circle at the point to which Peter tied his machine.

Peter decided to tie his car to point P and now he is wondering what is the area of the region that will be cleared from snow. Help him.

Input

The first line of the input contains three integers — the number of vertices of the polygon n ($3 \le n \le 100\,000$), and coordinates of point P.

Each of the next n lines contains two integers — coordinates of the vertices of the polygon in the clockwise or counterclockwise order. It is guaranteed that no three consecutive vertices lie on a common straight line.

All the numbers in the input are integers that do not exceed $1\,000\,000$ in their absolute value.

Output

Print a single real value number — the area of the region that will be cleared. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1,b)} \le 10^{-6}$.

Examples

input
3 0 0 0 1 -1 2 1 2
output
12.566370614359172464

input
4 1 -1 0 0 1 2 2 0 1 1
output
21.991148575128551812

Note

In the first sample snow will be removed from that area:

B. Skills

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

Lesha plays the recently published new version of the legendary game hacknet. In this version character skill mechanism was introduced. Now, each player character has exactly n skills. Each skill is represented by a non-negative integer a_i — the current skill level. All skills have the same maximum level A.

Along with the skills, global ranking of all players was added. Players are ranked according to the so-called Force. The *Force* of a player is the sum of the following values:

- The number of skills that a character has perfected (i.e., such that $a_i = A$), multiplied by coefficient c_f.
- The minimum skill level among all skills ($min\ a_i$), multiplied by coefficient c_m.

Now Lesha has m hacknetian currency units, which he is willing to spend. Each currency unit can increase the current level of any skill by 1 (if it's not equal to A yet). Help him spend his money in order to achieve the maximum possible value of the Force.

Input

The first line of the input contains five space-separated integers n, A, c_f, c_m and m ($1 \le n \le 100\,000$, $1 \le A \le 10^9$, $0 \le c_f, c_m \le 1000$, $0 \le m \le 10^{15}$).

The second line contains exactly n integers a_i ($0 \le a_i \le A$), separated by spaces, — the current levels of skills.

Output

On the first line print the maximum value of the Force that the character can achieve using no more than m currency units.

On the second line print n integers a'_i ($a_i \le a'_i \le A$), skill levels which one must achieve in order to reach the specified value of the Force, while using no more than m currency units. Numbers should be separated by spaces.

Examples

input
3 5 10 1 5 1 3 1
output
12 2 5 2

input
3 5 10 1 339 1 3 1
output
35 5 5 5

Note

In the first test the optimal strategy is to increase the second skill to its maximum, and increase the two others by **1**.

In the second test one should increase all skills to maximum.

C. Necklace

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

Ivan wants to make a necklace as a present to his beloved girl. A *necklace* is a cyclic sequence of beads of different colors. Ivan says that necklace is *beautiful* relative to the cut point between two adjacent beads, if the chain of beads remaining after this cut is a palindrome (reads the same forward and backward).

Ivan has beads of n colors. He wants to make a necklace, such that it's beautiful relative to as many cuts as possible. He certainly wants to use all the beads. Help him to make the most beautiful necklace.

Input

The first line of the input contains a single number n ($1 \le n \le 26$) — the number of colors of beads. The second line contains after n positive integers a_i — the quantity of beads of i-th color. It is guaranteed that the sum of a_i is at least 2 and does not exceed $100\,000$.

Output

In the first line print a single number — the maximum number of beautiful cuts that a necklace composed from given beads may have. In the second line print any example of such necklace.

Each color of the beads should be represented by the corresponding lowercase English letter (starting with a). As the necklace is cyclic, print it starting from any point.

Examples

input
3 4 2 1
output
1 abacaba

input
1 4
output
4 aaaa

input
2 1 1
output
0 ab

Note

In the first sample a necklace can have at most one beautiful cut. The example of such a necklace is shown on the picture.

In the second sample there is only one way to compose a necklace.

D. Kingdom and its Cities

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

Meanwhile, the kingdom of K is getting ready for the marriage of the King's daughter. However, in order not to lose face in front of the relatives, the King should first finish reforms in his kingdom. As the King can not wait for his daughter's marriage, reforms must be finished as soon as possible.

The kingdom currently consists of n cities. Cities are connected by $n - 1$ bidirectional road, such that one can get from any city to any other city. As the King had to save a lot, there is only one path between any two cities.

What is the point of the reform? The key ministries of the state should be relocated to distinct cities (we call such cities *important*). However, due to the fact that there is a high risk of an attack by barbarians it must be done carefully. The King has made several plans, each of which is described by a set of important cities, and now wonders what is the best plan.

Barbarians can capture some of the cities that are not important (the important ones will have enough protection for sure), after that the captured city becomes impassable. In particular, an interesting feature of the plan is the minimum number of cities that the barbarians need to capture in order to make all the important cities isolated, that is, from all important cities it would be impossible to reach any other important city.

Help the King to calculate this characteristic for each of his plan.

Input

The first line of the input contains integer n ($1 \le n \le 100\,000$) — the number of cities in the kingdom.

Each of the next $n - 1$ lines contains two distinct integers u_i, v_i ($1 \le u_i, v_i \le n$) — the indices of the cities connected by the i-th road. It is guaranteed that you can get from any city to any other one moving only along the existing roads.

The next line contains a single integer q ($1 \le q \le 100\,000$) — the number of King's plans.

Each of the next q lines looks as follows: first goes number k_i — the number of important cities in the King's plan, ($1 \le k_i \le n$), then follow exactly k_i space-separated pairwise distinct numbers from 1 to n — the numbers of important cities in this plan.

The sum of all k_i's does't exceed $100\,000$.

Output

For each plan print a single integer — the minimum number of cities that the barbarians need to capture, or print -1 if all the barbarians' attempts to isolate important cities will not be effective.

Examples

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

input
7 1 2 2 3 3 4 1 5 5 6 5 7 1 4 2 4 6 7
output
2

Note

In the first sample, in the first and the third King's plan barbarians can capture the city **3**, and that will be enough. In the second and

the fourth plans all their attempts will not be effective.

In the second sample the cities to capture are **3** and **5**.

E. Puzzle Lover

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

Oleg Petrov loves crossword puzzles and every Thursday he buys his favorite magazine with crosswords and other word puzzles. In the last magazine Oleg found a curious puzzle, and the magazine promised a valuable prize for it's solution. We give a formal description of the problem below.

The puzzle field consists of two rows, each row contains n cells. Each cell contains exactly one small English letter. You also are given a word w, which consists of k small English letters. A *solution* of the puzzle is a sequence of field cells $c_1, \ldots, c_k$, such that:

- For all i from 1 to k the letter written in the cell c_i matches the letter w_i;
- All the cells in the sequence are pairwise distinct;
- For all i from 1 to $k - 1$ cells c_i and c_{i+1} have a common side.

Oleg Petrov quickly found a solution for the puzzle. Now he wonders, how many distinct solutions are there for this puzzle. Oleg Petrov doesn't like too large numbers, so calculate the answer modulo $10^9 + 7$.

Two solutions c_i and c'_i are considered distinct if the sequences of cells do not match in at least one position, that is there is such j in range from 1 to k, such that $c_j \neq c'_j$.

Input

The first two lines contain the state of the field for the puzzle. Each of these non-empty lines contains exactly n small English letters.

The next line is left empty.

The next line is non-empty and contains word w, consisting of small English letters.

The length of each line doesn't exceed $2\,000$.

Output

Print a single integer — the number of distinct solutions for the puzzle modulo $10^9 + 7$.

Examples

input
code edoc code
output
4

input
aaa aaa aa
output
14