Educational Codeforces Round 2

A. Extract Numbers

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

You are given string s. Let's call `word` any largest sequence of consecutive symbols without symbols `','` (comma) and `';'` (semicolon). For example, there are four `words` in string `"aba,123;1a;0"`: `"aba"`, `"123"`, `"1a"`, `"0"`. A word can be empty: for example, the string s=`";;"` contains three empty words separated by `';'`.

You should find all `words` in the given string that are nonnegative INTEGER numbers without leading zeroes and build by them new string a. String a should contain all `words` that are numbers separating them by `','` (the order of numbers should remain the same as in the string s). By all other `words` you should build string b in the same way (the order of numbers should remain the same as in the string s).

Here strings `"101"`, `"0"` are INTEGER numbers, but `"01"` and `"1.0"` are not.

For example, for the string `aba,123;1a;0` the string a would be equal to `"123,0"` and string b would be equal to `"aba,1a"`.

Input

The only line of input contains the string s ($1 \le |s| \le 10^5$). The string contains only symbols `'.'` (ASCII 46), `','` (ASCII 44), `';'` (ASCII 59), digits, lowercase and uppercase latin letters.

Output

Print the string a to the first line and string b to the second line. Each string should be surrounded by quotes (ASCII 34).

If there are no `words` that are numbers print dash (ASCII 45) on the first line. If all `words` are numbers print dash on the second line.

Examples

input
aba,123;1a;0
output
"123,0" "aba,1a"

input
1;;01,a0,
output
"1" ",01,a0,"

input
1
output
"1" -

input
a
output
- "a"

Note

In the second example the string s contains five words: `"1"`, `""`, `"01"`, `"a0"`, `""`.

B. Queries about less or equal elements

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

You are given two arrays of integers a and b. For each element of the second array b_j you should find the number of elements in array a that are less than or equal to the value b_j.

Input

The first line contains two integers n, m ($1 \le n, m \le 2 \cdot 10^5$) — the sizes of arrays a and b.

The second line contains n integers — the elements of array a ($-10^9 \le a_i \le 10^9$).

The third line contains m integers — the elements of array b ($-10^9 \le b_j \le 10^9$).

Output

Print m integers, separated by spaces: the j-th of which is equal to the number of such elements in array a that are less than or equal to the value b_j.

Examples

input
5 4 1 3 5 7 9 6 4 2 8
output
3 2 1 4

input
5 5 1 2 1 2 5 3 1 4 1 5
output
4 2 4 2 5

C. Make Palindrome

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

A string is called palindrome if it reads the same from left to right and from right to left. For example "kazak", "oo", "r" and "mikhailrubinchikkihcniburliahkim" are palindroms, but strings "abb" and "ij" are not.

You are given string s consisting of lowercase Latin letters. At once you can choose any position in the string and change letter in that position to any other lowercase letter. So after each changing the length of the string doesn't change. At first you can change some letters in s. Then you can permute the order of letters as you want. Permutation doesn't count as changes.

You should obtain palindrome with the minimal number of changes. If there are several ways to do that you should get the lexicographically (alphabetically) smallest palindrome. So firstly you should minimize the number of changes and then minimize the palindrome lexicographically.

Input

The only line contains string s ($1 \le |s| \le 2 \cdot 10^5$) consisting of only lowercase Latin letters.

Output

Print the lexicographically smallest palindrome that can be obtained with the minimal number of changes.

Examples

input
aabc
output
abba

input
aabcd
output
abcba

D. Area of Two Circles' Intersection

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

You are given two circles. Find the area of their intersection.

Input

The first line contains three integers x_1, y_1, r_1 ($-10^9 \le x_1, y_1 \le 10^9, 1 \le r_1 \le 10^9$) — the position of the center and the radius of the first circle.

The second line contains three integers x_2, y_2, r_2 ($-10^9 \le x_2, y_2 \le 10^9, 1 \le r_2 \le 10^9$) — the position of the center and the radius of the second circle.

Output

Print the area of the intersection of the circles. The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.

Examples

input
0 0 4 6 0 4
output
7.25298806364175601379

input
0 0 5 11 0 5
output
0.00000000000000000000

E. Lomsat gelral

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

You are given a rooted tree with root in vertex 1. Each vertex is coloured in some colour.

Let's call colour c dominating in the subtree of vertex v if there are no other colours that appear in the subtree of vertex v more times than colour c. So it's possible that two or more colours will be dominating in the subtree of some vertex.

The subtree of vertex v is the vertex v and all other vertices that contains vertex v in each path to the root.

For each vertex v find the sum of all dominating colours in the subtree of vertex v.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the number of vertices in the tree.

The second line contains n integers c_i ($1 \le c_i \le n$), c_i — the colour of the i-th vertex.

Each of the next $n - 1$ lines contains two integers x_j, y_j ($1 \le x_j, y_j \le n$) — the edge of the tree. The first vertex is the root of the tree.

Output

Print n integers — the sums of dominating colours for each vertex.

Examples

input
4 1 2 3 4 1 2 2 3 2 4
output
10 9 3 4

input
15 1 2 3 1 2 3 3 1 1 3 2 2 1 2 3 1 2 1 3 1 4 1 14 1 15 2 5 2 6 2 7 3 8 3 9 3 10 4 11 4 12 4 13
output
6 5 4 3 2 3 3 1 1 3 2 2 1 2 3

F. Edge coloring of bipartite graph

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

You are given an undirected bipartite graph without multiple edges. You should paint the edges of graph to minimal number of colours, so that no two adjacent edges have the same colour.

Input

The first line contains three integers a, b, m ($1 \le a, b \le 1000$, $0 \le m \le 10^5$), a is the size of the first part, b is the size of the second part, m is the number of edges in the graph.

Each of the next m lines contains two integers x, y ($1 \le x \le a$, $1 \le y \le b$), where x is the number of the vertex in the first part and y is the number of the vertex in the second part. It is guaranteed that there are no multiple edges.

Output

In the first line print integer c — the minimal number of colours. The second line should contain m integers from 1 to c — the colours of the edges (in the order they appear in the input).

If there are several solutions, you can print any one of them.

Examples

input
4 3 5 1 2 2 2 3 2 4 1 4 3
output
3 1 2 3 1 2