Graph Weighting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

There is a connected undirected graph with N vertices numbered $1, 2 \dots, N$ and M edges. The i-th edge connects the vertex u_i and the vertex v_i. The graph may contain multiple edges between the same pair of vertices, but it does not contain self-loops.

For each $W = 0, 1, \dots, K$, solve the following problem:

Determine if there exists a way to assign a weight $w_i \in \{0, 1, \dots, L\}$ to the i-th edge for each i with $1 \leq i \leq M$, such that the weight of any spanning tree of the graph is exactly W. The weight of a spanning tree is defined as the sum of the weights of all the edges included in the spanning tree. If such an assignment exists, find the minimum value of $(w_1)^2 + (w_2)^2 + \dots + (w_M)^2$ over all such assignments.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N\ M\ K\ L \\
u_1\ v_1 \\
\vdots \\
u_M\ v_M
\end{array}
$$

- All values in the input are integers.
- $2 \leq N \leq 10^5$
- $N - 1 \leq M \leq 2 \times 10^5$
- $1 \leq L, K \leq 10^5$
- $1 \leq u_i, v_i \leq N$
- $u_i \neq v_i$
- A given undirected graph is connected.

Output

For each $W = 0, 1, \dots, K$, output the answer to the problem in this order, separated by spaces. Specifically, if no assignment meets the conditions, output `-1`. If an assignment exists, output the minimum value of $(w_1)^2 + (w_2)^2 + \dots + (w_M)^2$ over all such assignments.

Examples

standard input	standard output
4 4 3 2 1 2 2 3 2 4 3 4	0 1 3 4
2 3 2 1 1 2 2 1 1 2	0 3 -1
6 7 9 2 1 2 2 3 2 4 4 5 4 6 1 4 3 4	0 1 2 5 6 7 10 13 22 25

Note

In example 1, for instance, when $W = 2$, if we set $(w1, w2, w3, w4) = (0, 1, 1, 1)$, the weight of any spanning tree of the graph will be 2.

In example 2, it is not possible to make the weight of any spanning tree of the graph equal to 2.