BFS 101

Problem Statement

In this problem, you will be given a **directed unweighted graph** G with N nodes and E edges. Your task is to find the **number of nodes directly connected** to a given starting node X, and compute the **minimum number of edges** required to reach each node from X using Breadth-First Search (BFS).

Input Format

- The first line contains two space-separated integers N and E:
  - N – the number of nodes in the graph.
  - E – the number of directed edges.
- The next E lines each contain two integers A and B, denoting a **directed edge** from node A to node B.
- The final line contains a single integer X ($0 \leq X < N$), the node from which the BFS traversal will begin.

Output Format

- On the first line, print a single integer — the number of nodes directly reachable from node X (i.e., the number of nodes to which there is an **outgoing edge** from X).
- On the second line, print N space-separated integers, where the i-th integer represents:
  - The **minimum number of edges** required to reach node i from node X.
  - If a node i is not reachable from X, print -1 for that node.

Constraints

- $1 \leq N \leq 1000$
- $1 \leq E \leq 5000$
- $0 \leq X < N$

Example

Input

```
5 7
1 2
1 4
```

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

Output

```
2
-1 0 1 2 1 -1
```

Explanation

- Node 1 has direct edges to nodes 2 and 4, so the first output is 2.
- Using BFS starting at node 1, the distances to other nodes are:
  - Node 1: distance 0 (starting node)
  - Node 2: 1 edge away
  - Node 3: 2 edges away ($1 \to 2 \to 3$)
  - Node 4: 1 edge away
  - Node 5: unreachable from 1 through the given edges, so -1