CSE 4404-Algorithms Lab. Winter 2022
Date: February 1, 2023.
Target Group: B
Topic: Lab Test

Instructions:

- Task naming format: fullID_L02_T01_B.c/CPP
- **Solutions with less efficient approaches will be considered for partial marks.**

Task 1

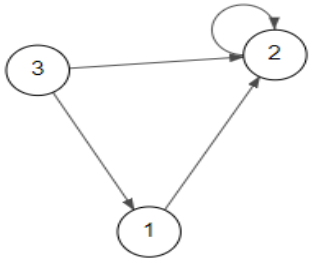
Create a program that identifies vertices in a directed graph that cannot be reached from a specified starting vertex. The graph is represented by n vertices (numbered 1 to n, where 1 ≤ n ≤ 100) and a set of directed edges connecting pairs of nodes (p -> q). A vertex r is considered reachable from a vertex p if there exists a direct edge from p to r, or if r can be reached from some other vertex q that is reachable from p. The goal is to identify all vertices (r) that are not reachable from the specified starting vertex (p).

Input

The first line of input consists of two integers, $N\ (\leq\ N\ \leq\ 100)$ and $M\ (1\ \leq\ M\ \leq\ N(N\ -\ 1)/2)$, where N represents the total number of houses and M represents the total number of edges. Following the graph definition, there will be one line containing a list of integers. The first integer on the line will specify how many integers follow. Each of the following integers represents a start vertex to be investigated by your program.

Output

For each start vertex to be investigated, your program should identify all the vertices which are inaccessible from the given start vertex. Each list should appear on one line, beginning with the count of inaccessible vertices and followed by the inaccessible vertex numbers.

Sample Input	**Sample Output**	**Explanation**
Testcase #1: 3 4 1 2 2 2 3 1 3 2 2 1 2	Output #1: 2 1 3 2 1 3	 If the starting vertex is 1, then from node 1, only node 2 is reachable. So unreachable vertices from node 2 are node 1 itself along with node 3. If the starting vertex is 2, then node 2 is reachable as there is a self-loop. But no other node is reachable from node 2. So unreachable vertices from node 2 are node 1 and node 3.

Task 2

Consider a number maze represented as a two-dimensional array of numbers comprehended between 0 and 9, as exemplified below. The maze can be traversed following any orthogonal direction (i.e., north, south, east, and west). Considering that each cell represents a cost, then finding the minimum cost to travel the maze from one entry point to an exit point may pose a reasonable challenge. Your task is to find the minimum cost value to go from the top-left corner to the bottom-right corner of a given number maze of size N × M where 1 ≤ N, M ≤ 999. Note that the solution for the given example is 24.

0	3	1	2	9
7	3	4	9	9
1	7	5	5	3
2	3	4	2	5

Input

The first input line contains two positive integers, N and M, defining the number of rows and columns. one per each row of the maze, containing the maze numbers separated by spaces. Then follows N lines with M integers each, defining the maze.

Output

For each maze, output one line with the required minimum value.

Sample Input	**Sample Output**
Testcase #1: 4 5 0 3 1 2 9 7 3 4 9 9 1 7 5 5 3 2 3 4 2 5 Testcase #2: 1 6 0 1 2 3 4 5	Output #1: 24. Output #2: 15