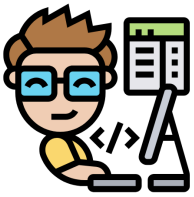

DAILY PROGRAMMING
CHALLENGE

WISDOM SPROUTS
IT Training Hub

---

# Detect a Cycle in an Undirected Graph

You are given an undirected graph represented by an adjacency list. Your task is to determine if the graph contains any cycle. A cycle is formed if you can traverse through a sequence of edges that starts and ends at the same vertex.

**Input:**

- An integer V representing the number of vertices.
- An integer E representing the number of edges.
- A list of edges, where each edge connects two vertices of the graph.

**Output:**

- Return true if the graph contains a cycle, otherwise return false.

**Examples:**

- Example 1
  Input: V = 5, E = 5
  Edges = [[0, 1], [1, 2], [2, 3], [3, 4], [4, 0]]
  Output: true
  Explanation:
  - The tree is as follows


  - This graph contains a cycle (0 -> 1 -> 2 -> 3 -> 4 -> 0).

**Constraints:**

- $1 \leq V \leq 10^4$
- $0 \leq E \leq 10^4$
- The graph can be disconnected, so check all components.
- The graph has no self-loops and no multiple edges between the same pair of nodes.

**Test Cases:**

1. Input: V = 5, E = 5
   Edges = [[0, 1], [1, 2], [2, 3], [3, 4], [4, 0]]
   Output: true
2. Input: V = 3, E = 2
   Edges = [[0, 1], [1, 2]]
   Output: false
3. Input: V = 1, E = 0
   Edges = []
   Output: false
4. Input: V = 4, E = 3
   Edges = [[0, 1], [1, 2], [2, 3]]
   Output: false

**Edge Cases:**

1. A graph with one node and no edges cannot contain a cycle
2. If the graph contains multiple components, check for cycles in all components.
3. If the graph forms a straight line (like a chain), no cycle is present.