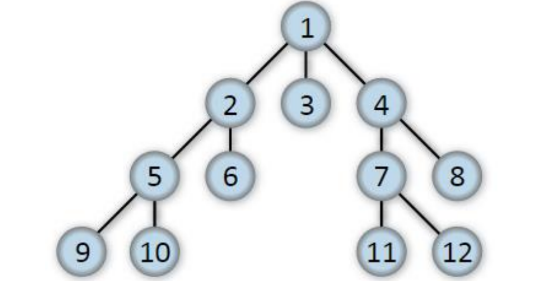
LAB 3

Task 1. Construct the given tree, write a breadth first search function def bfs(graph, start\_node, goal\_node) which should return goal node if the data is found in the tree else return None.



**Objectives:**

This task aims to implement a breadth-first search (BFS) function to traverse the tree and search for a specified goal node using Python

**Code:**

def bfs(graph, start\_node, goal\_node):

    visited = set()

    queue = []

    if start\_node == goal\_node:

        return start\_node

    visited.add(start\_node)

    queue.append(start\_node)

    while queue:

        path = queue.pop(0)

        # print '->' after each node except the last one

        print(path, end='->' if path != f'{goal\_node}' else '')

        if path == goal\_node:

        return path

        for neighbour in graph[path]:

            if neighbour not in visited:

                visited.add(neighbour)

                queue.append(neighbour)

    return None

graph = {

    '1': ['2', '3', '4'],

    '2': ['5', '6'],

    '3': [],

    '4': ['7', '8'],

    '5': ['9', '10'],

    '6': [],

    '7': ['11', '12'],

    '8': [],

    '9': [],

    '10': [],

    '11': [],

    '12': []

}

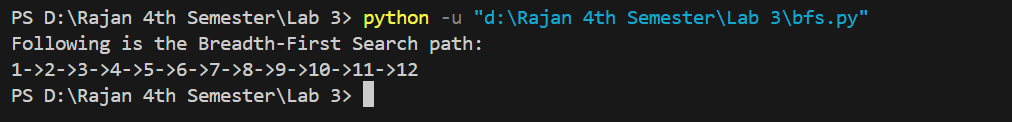
start\_node = '1'

goal\_node = '12'

print("Following is the Breadth-First Search path:")

bfs(graph, start\_node,goal\_node)

**OUTPUT:**



**Task 2.** Construct the tree as in the previous question, write a depth first search function def dfs(graph, start\_node, goal\_node) which should return goal node if the data is found in the tree else return None.

**Objective:**

This task aims to implement a depth-first search (BFS) function to travers the tree and search for a specified goal node using Python.

**Code:**

visited = set()

def dfs(visited, graph, node):

    if node not in visited:

        print(node, end="->" if node != node[-1] else " "

        visited.add(node)

        for neighbour in graph[node]:

            dfs(visited, graph, neighbour)

graph = {

  '5' : ['3','7'],

  '3' : ['2', '4'],

  '7' : ['8'],

  '2' : [],

  '4' : ['8'],

  '8' : []

}

print("Following is the Depth-First Search")

dfs(visited, graph, '5')

**Output:**

