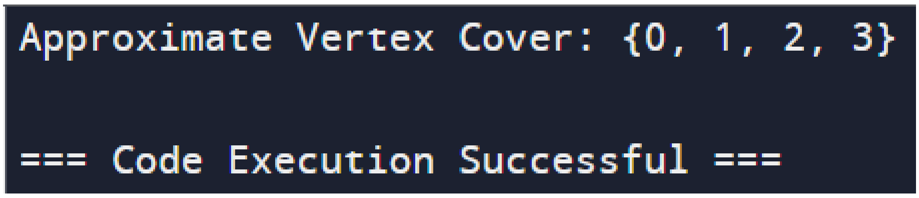
121. Approximation Algorithm Vertex Cover, Set Cover

PROGRAM:-

```
def vertex_cover_approximation(graph):
    """
    2-Approximation algorithm for the Vertex Cover problem.

    Parameters:
    graph (dict): Adjacency list of the graph.

    Returns:
    set: Approximate vertex cover.
    """
    # Initialize the vertex cover
    cover = set()
    edges = set((u, v) for u in graph for v in graph[u])

    # While there are edges in the graph
    while edges:
        # Pick an arbitrary edge (u, v)
        u, v = edges.pop()

        # Add both endpoints u and v to the vertex cover
        cover.add(u)
        cover.add(v)

        # Remove all edges incident to u and v
        edges = {e for e in edges if u not in e and v not in e}

    return cover

# Example usage:
graph = {
    0: [1, 2],
    1: [0, 2],
    2: [0, 1, 3],
    3: [2]
}

print("Approximate Vertex Cover:", vertex_cover_approximation(graph))
```

OUTPUT:-



```
Approximate Vertex Cover: {0, 1, 2, 3}


=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(V+E)