1

```python
import collections

def bfs(graph, root):
    visited, queue = set(), collections.deque([root])
    visited.add(root)
    while queue:
        vertex = queue.popleft()
        print(str(vertex) + " ", end="")
        for neighbour in graph[vertex]:
            if neighbour not in visited:
                visited.add(neighbour)
                queue.append(neighbour)
def dfs(graph, start, visited=None):
    if visited is None:
        visited = set()
    visited.add(start)
    print(start, end="")
    for next in graph[start]:
        print("->", end="")
        dfs(graph, next, visited)
    return visited

if __name__ == '__main__':
    graph = {0: [1, 2], 1: [], 2: [4, 5], 3: [5], 4: [], 5: []}
    print("Following is Breadth First Traversal: ")
    bfs(graph, 0)
    print("")
    print("Following is Depth First Traversal: ")
    dfs(graph, 0)
    print(" ")
```

```
ishadp@pop-os:~/Documents/Code/AI$ /bin/python3 /home/ishadp/Documents/Code/Practical-AI/1.py
Following is Breadth First Traversal:
0 1 2 4 5
Following is Depth First Traversal:
0->1->2->4->5
```