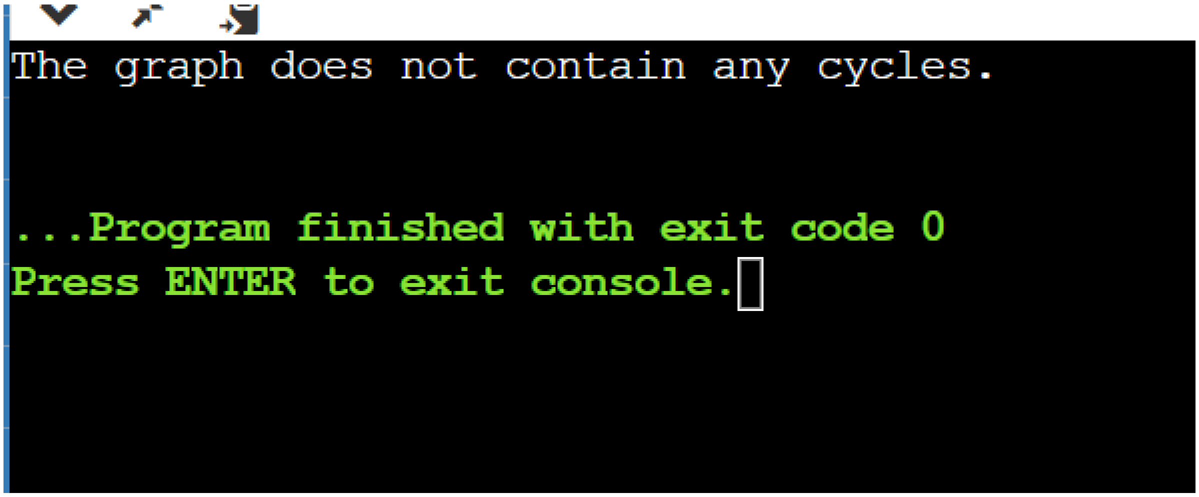
DFS DIRECTED GRAPH to DETECT A CYCLE:

Code:

```
def has_cycle(graph, node, visited, recursion_stack):
    visited[node] = True
    recursion_stack[node] = True

    for neighbor in graph[node]:
        if not visited[neighbor]:
            if has_cycle(graph, neighbor, visited, recursion_stack):
                return True
        elif recursion_stack[neighbor]:
            return True

    recursion_stack[node] = False
    return False

def contains_cycle(graph):
    visited = {node: False for node in graph}
    recursion_stack = {node: False for node in graph}

    for node in graph:
        if not visited[node]:
            if has_cycle(graph, node, visited, recursion_stack):
                return True

    return False

# Example directed graph represented as an adjacency list
graph = {
    1: [2],
    2: [3, 4],
    3: [4],
    4: []
}
```

```
if contains_cycle(graph):
    print("The graph contains at least one cycle.")
else:
    print("The graph does not contain any cycles.")
```

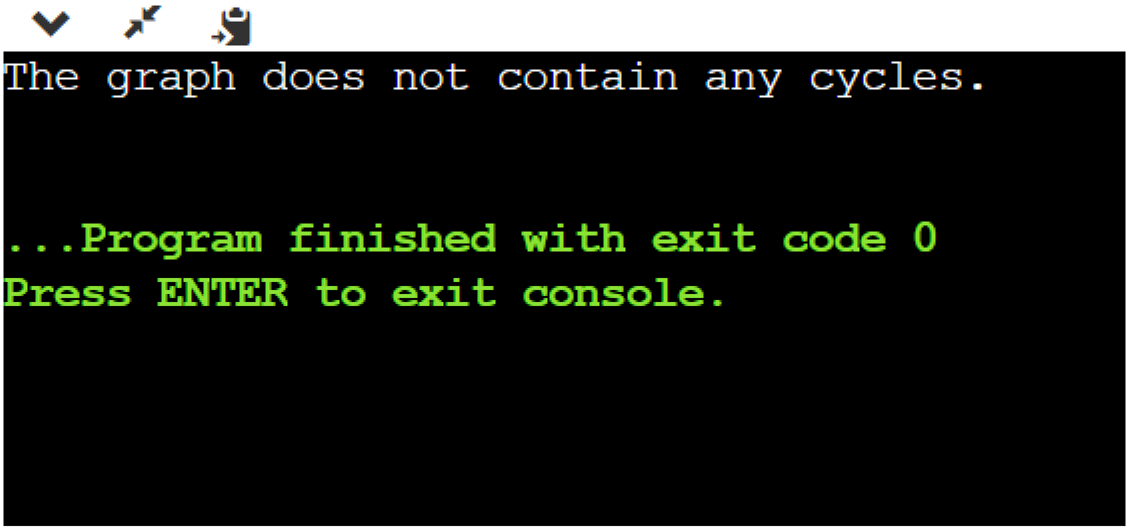


```
The graph does not contain any cycles.


...Program finished with exit code 0
Press ENTER to exit console.
```

DFS UNDIRECTED GRAPH TO DETECT A CYCLE:

Code:

```
def has_cycle(graph, node, parent, visited):
    visited[node] = True

    for neighbor in graph[node]:
        if not visited[neighbor]:
            if has_cycle(graph, neighbor, node, visited):
                return True
        elif neighbor != parent:
            return True

    return False

def contains_cycle(graph):
    visited = {node: False for node in graph}
    for node in graph:
        if not visited[node]:
            if has_cycle(graph, node, None, visited):
                return True

    return False

# Example undirected graph represented as an adjacency list
graph = {
    1: [2, 3],
    2: [1, 4],
    3: [1, 5],
    4: [2],
    5: [3]
}

if contains_cycle(graph):
    print("The graph contains at least one cycle.")
else:
    print("The graph does not contain any cycles.")
```



```
The graph does not contain any cycles.


...Program finished with exit code 0
Press ENTER to exit console.
```