Problem – 3 Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not. (using DFS)

```python
from collections import defaultdict


class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = defaultdict(list)

    def add_edge(self, u, v):
        self.graph[u].append(v)
        self.graph[v].append(u)

    def is_cyclic_util(self, v, visited, parent):
        visited[v] = True

        for neighbor in self.graph[v]:
            if not visited[neighbor]:
                if self.is_cyclic_util(neighbor, visited, v):
                    return True
            elif parent != neighbor:
                return True

        return False

    def contains_cycle(self):
        visited = [False] * self.V

        for i in range(self.V):
            if not visited[i]:
                if self.is_cyclic_util(i, visited, -1):
```

```python
        return True

    return False


V = 5
E = 4
g = Graph(V)
edges = [(0, 1), (1, 2), (2, 1), (3, 4)]
for u, v in edges:
    g.add_edge(u, v)


if g.contains_cycle():
    print("The graph contains a cycle.")
else:
    print("The graph does not contain any cycle.")
```
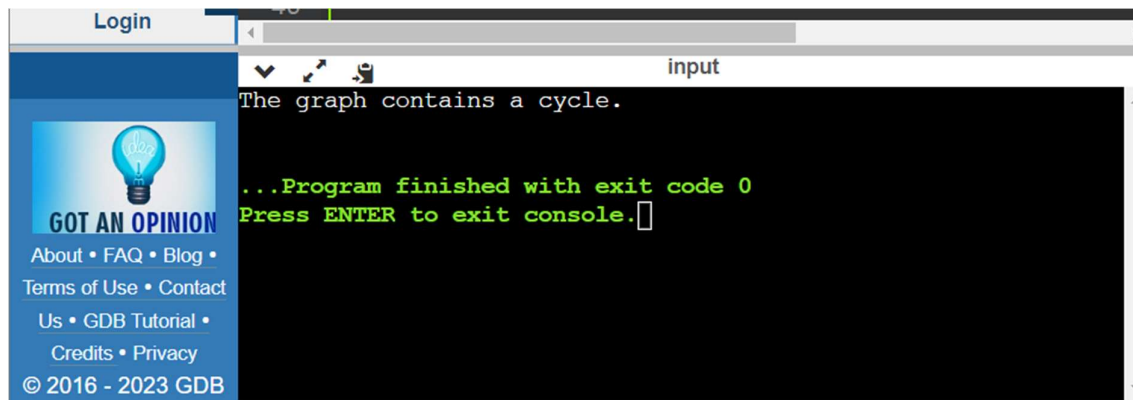
input

```
The graph contains a cycle.

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