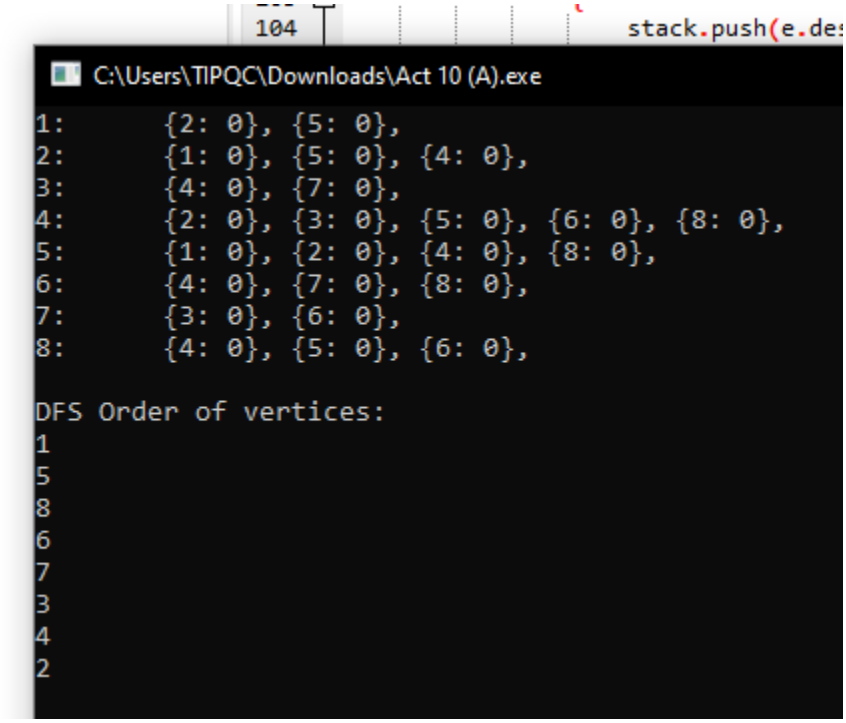
Activity No. 10	
Implementing DFS and Graph Applications	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed:** 11/27/2024
Section: CPE21S4	**Date Submitted:** 11/27/2024
Name(s): Mamaril, Justin Kenneth Reyes, Alexzander San Jose, Alexander San Juan ,Edson Ray Titong, Lee Ivan	**Instructor:** Engr. Maria Rizette Sayo

A. Output(s) and Observation(s)

Output:



Observation:
The code creates a graph, adds edges to it, and performs a depth-first search (DFS) starting from vertex 1. It then prints the graph structure and the order in which vertices are visited during the DFS traversal.

B. Answers to Supplementary Activity

Answer the following questions:
1. A person wants to visit different locations indicated on a map. He starts from one location (vertex) and wants to visit every vertex until it finishes from one vertex, backtracks, and then explore other vertex from same vertex. Discuss which algorithm would be most helpful to accomplish this task.

Answer: Depth-First Search (DFS) is the best algorithm for this task because it explores as far as possible from the starting vertex, backtracks, and continues exploring other paths from the same starting point.

2. Describe a situation where in the DFS of a graph would possibly be unique.

Answer: DFS can be unique when the graph is disconnected (it has separate sections) or when the order of neighbors affects the path it takes, leading to different traversal results.

3. Demonstrate the maximum number of times a vertex can be visited in the DFS. Prove your claim through code and demonstrated output.

Answer: In standard DFS, each vertex is visited once. However, if there are cycles in the graph, DFS might backtrack and revisit vertices, but typically each vertex is visited only once during the main traversal.

4. What are the possible applications of the DFS?

Answer: DFS is used for tasks like finding connected components, cycle detection, solving puzzles, pathfinding, and topological sorting in graphs.

5. Identify the equivalent of DFS in traversal strategies for trees. In order to efficiently answer this question, provide a graphical comparison, examine pseudocode and code implementation.

Answer: In trees, DFS is like preorder, inorder, and postorder traversals, where the algorithm explores deep into the tree before moving to other branches, with different orders of visiting nodes.

C. Conclusion & Lessons Learned

We as a group concluded that DFS is a powerful algorithm for exploring graphs by following one path fully before backtracking. It is useful for tasks like finding cycles, paths, and sorting. We learned how the graph's structure and neighbor order can affect the traversal. In trees, DFS works like preorder, inorder, or postorder traversal.

D. Assessment Rubric

E. External References