EXPERIMENT 6

Objective: Implementing algorithms to find Hamiltonian cycles in the graph.

Brief Theory:

A Hamiltonian cycle in a graph is a closed path that visits each vertex exactly once and returns to the starting vertex. Unlike Euler tours, Hamiltonian cycles focus on visiting vertices rather than edges. A graph that contains a Hamiltonian cycle is called a Hamiltonian graph.

To determine if a Hamiltonian cycle exists, there are no simple necessary and sufficient conditions like in Eulerian graphs. However, certain properties and algorithms can help find such cycles.

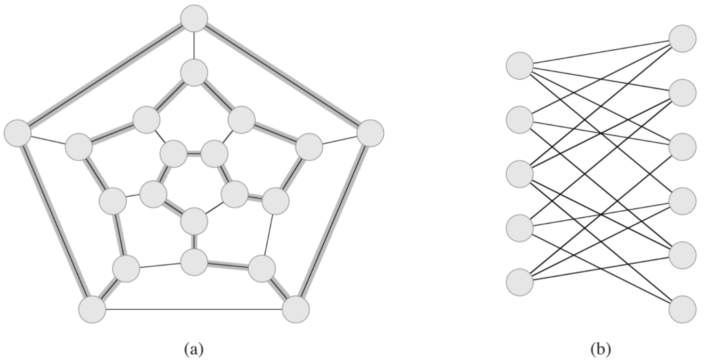
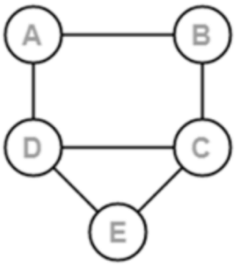
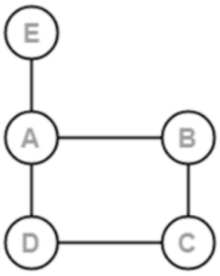


Figure 34.2 **(a)** A graph representing the vertices, edges, and faces of a dodecahedron, with a hamiltonian cycle shown by edges highlighted in blue. **(b)** A bipartite graph with an odd number of vertices. Any such graph is nonhamiltonian.

Task:

1) Write a program using backtracking to find Hamiltonian cycles in the given graphs.





2) Create a program that allows users to input a graph and checks if it satisfies basic conditions for Hamiltonian cycles, such as vertex degree and connectivity. (Use Linked representation)
3) Extend your implementation to visualize the graph and highlight the Hamiltonian cycle if it exists.

Apparatus and components required: Computer with C or C++ Compiler and Linux/Windows platform.

Experimental/numerical procedure: Coding, compilation, editing, run and debugging.