**Easy Tasks (60-74 points)**

1. **Degree Calculator**
   * **Objective**: Write a program that calculates the degree of each vertex in a graph.
   * **Input Example**:
     + Graph: Vertices = {A, B, C}, Edges = {(A, B), (B, C), (C, A)}
   * **Output Example**:
     + Degree of A: 2
     + Degree of B: 2
     + Degree of C: 2
2. **Adjacency Matrix Builder**
   * **Objective**: Implement a program to create the adjacency matrix of a graph.
   * **Input Example**:
     + Graph: Vertices = {1, 2, 3}, Edges = {(1, 2), (2, 3)}
   * **Output Example**:
     + Adjacency Matrix:

Изображение выглядит как текст, снимок экрана, Шрифт, типография

Автоматически созданное описание

1. **Path Finder**
   * **Objective**: Create a program to find a path between two vertices in a graph.
   * **Input Example**:
     + Graph: Vertices = {1, 2, 3, 4}, Edges = {(1, 2), (2, 3), (3, 4)}
     + Start Vertex: 1
     + End Vertex: 4
   * **Output Example**:
     + Path: 1 -> 2 -> 3 -> 4

**Medium Tasks (75-89 points)**

1. **Subgraph Identifier**
   * **Objective**: Write a program that determines if a given graph is a subgraph of another.
   * **Input Example**:
     + Main Graph: Vertices = {A, B, C, D}, Edges = {(A, B), (B, C), (C, D)}
     + Subgraph: Vertices = {B, C}, Edges = {(B, C)}
   * **Output Example**:
     + Is Subgraph: True
2. **Sum of Degrees Checker**
   * **Objective**: Implement a program to verify the Sum of Degrees of Vertices Theorem in a graph.
   * **Input Example**:
     + Graph: Vertices = {1, 2, 3, 4}, Edges = {(1, 2), (2, 3), (3, 4), (4, 1)}
   * **Output Example**:
     + Sum of Degrees: 8 (which is twice the number of edges)
     + Theorem Verified: True
3. **Incidence Matrix Generator**
   * **Objective**: Create a program that generates the incidence matrix of a graph.
   * **Input Example**:
     + Graph: Vertices = {A, B, C}, Edges = {(A, B), (B, C)}
   * **Output Example**:
     + Incidence Matrix:

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание

**Hard Tasks (90-100 points)**

1. **Graph Isomorphism Tester**
   * **Objective**: Write a program that tests if two graphs are isomorphic.
   * **Input Example**:
     + Graph 1: Vertices = {1, 2, 3}, Edges = {(1, 2), (2, 3)}
     + Graph 2: Vertices = {A, B, C}, Edges = {(A, B), (B, C)}
   * **Output Example**:
     + Are Isomorphic: True
2. **Circuit Finder**
   * **Objective**: Implement a program to find a circuit in a graph.
   * **Input Example**:
     + Graph: Vertices = {A, B, C, D}, Edges = {(A, B), (B, C), (C, D), (D, A)}
   * **Output Example**:
     + Circuit: A -> B -> C -> D -> A
3. **Walks and Trails Analyzer**
   * **Objective**: Create a program that identifies all possible walks and trails between two vertices in a graph.
   * **Input Example**:
     + Graph: Vertices = {1, 2, 3, 4}, Edges = {(1, 2), (2, 3), (3, 1), (3, 4)}
     + Start Vertex: 1
     + End Vertex: 4
   * **Output Example**:
     + Walks: 1 -> 2 -> 3 -> 4, 1 -> 3 -> 2 -> 3