Discrete Mathematics Assignment



# Session 2023 - 2027

**Submitted by:**

Abdul Hanan 2023-CS-105

Muhammad Owais 2023-CS-131

# Supervised by:

Waqas Ali

# Course:

CSC-101 Discrete Mathematics

Department of Computer Science

**University of Engineering and Technology Lahore Pakistan**

Table of Contents

[1. Question#1: Analysis of Games Using Graph Theory 5](#_Toc154843692)

[**1.1** **Investigate how certain games can be represented and analyzed using graph theory. (Give example of at least two games)?** 5](#_Toc154843693)

[**1.1.1 Chess:** 5](#_Toc154843694)

[**1.1.2 Snakes and ladder:** 8](#_Toc154843695)

[**1.2 Is Tic-Tac-Toe an example of a game that can be represented with graphs? Describe the graph and what you are trying to achieve or avoid. Consider studying Tic-Tac-Toe generalizations.** 10](#_Toc154843701)

[2. Question#2: Programming Tasks in Graph Theory 12](#_Toc154843702)

[**2.1** **Undirected Graph Degrees:** 12](#_Toc154843703)

[**2.2** **Directed Graph Degrees:** 13](#_Toc154843704)

[**2.3** **Bipartite Graph Check:** 13](#_Toc154843705)

[**2.4** **Adjacency Matrix:** 14](#_Toc154843706)

[**2.5** **Edge Listing from Adjacency Matrix:** 15](#_Toc154843707)

[**2.6** **Incidence Matrix Construction:** 15](#_Toc154843708)

[3 Question#3: Programming Tasks in Number Theory 16](#_Toc154843709)

[**3.1** **Prime Factorization:** 16](#_Toc154843710)

[**3.2** **Euclidean Algorithm for GCD:** 16](#_Toc154843711)

[**3.3** **LCM Calculation:** 17](#_Toc154843712)

[**3.4** **Bezout Coefficients:** 17](#_Toc154843713)

[**3.5** **Modular Inverse:** 17](#_Toc154843714)

[**3.6** **RSA Encryption:** 17](#_Toc154843715)

[**3.7** **RSA Decryption Key:** 18](#_Toc154843716)

Table of Figures

[**Figure 1: Chess Possible Moves** 4](#_Toc154842452)

[**Figure 2: Chess Moves with evaluation** 5](#_Toc154842453)

[**Figure 3: Knight's Movement** 6](#_Toc154842454)

[**Figure 4: Snakes and ladder** 8](#_Toc154842455)

[**Figure 5: Solved Version of Snakes and ladder** 9](#_Toc154842456)

[**Figure 6: Tic Tac Toe** 10](#_Toc154842457)

[**Figure 7: Undirected Graph Degrees Solution** 11](#_Toc154842458)

[**Figure 8: Directed Graphs Solution** 12](#_Toc154842459)

[**Figure 9: Bipartite Graph Solution** 13](#_Toc154842460)

[**Figure 10: Adjacency Matrix Solution** 13](#_Toc154842461)

[**Figure 11: Edge Listing from Adjacency Matrix Solution** 14](#_Toc154842462)

[**Figure 12: Incidence Matrix Construction Solution** 15](#_Toc154842463)

[**Figure 13: Prime Factorization Solution** 15](#_Toc154842464)

[**Figure 14: GCD Euclidean Algorithm Solution** 15](#_Toc154842465)

[**Figure 15: LCM Calculation Solution** 16](#_Toc154842466)

[**Figure 16: Bezout Coefficients Solution** 16](#_Toc154842467)

[**Figure 17: Modular Inverse Solution** 16](#_Toc154842468)

[**Figure 18: RSA Encryption Solution** 17](#_Toc154842469)

[**Figure 19: RSA Decryption Key Solution** 17](#_Toc154842470)

# Question#1: Analysis of Games Using Graph Theory

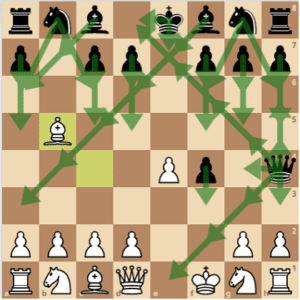
## **Investigate how certain games can be represented and analyzed using graph theory. (Give example of at least two games)?**

Graph theory can be applied to various games to model their structure and analyze their properties. Let's start by explaining how graph theory can be used to represent and analyze Chess and .

**1.1.1 Chess:** In Chess, graph theory can be employed to model the movement of pieces on the board.

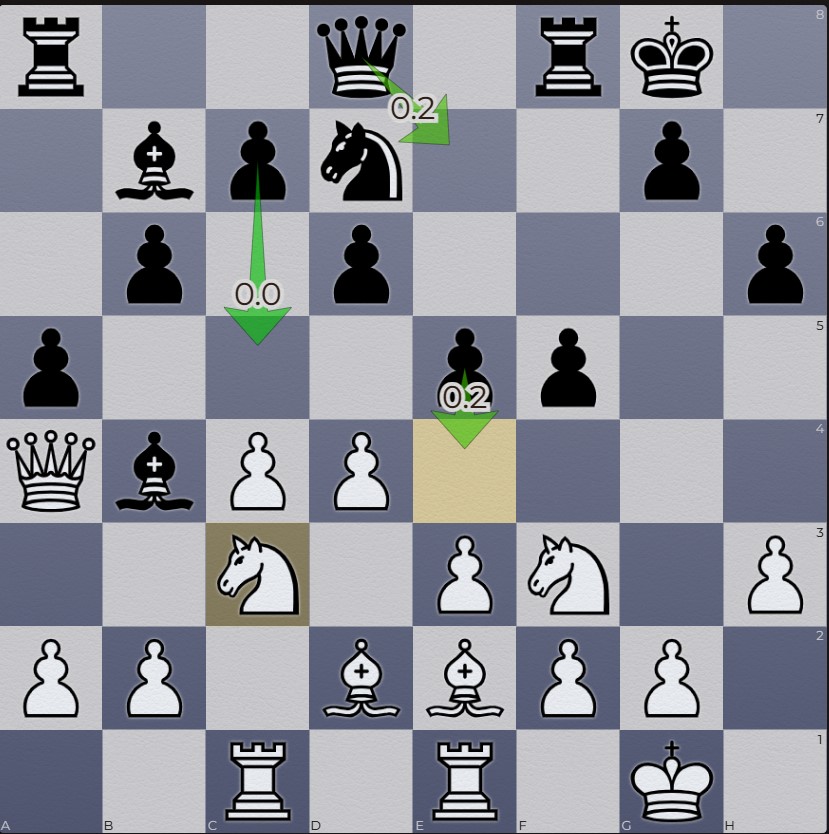
**Nodes (Vertices):** Each node represents a square on the Chessboard. There are 64 squares on a standard 8x8 Chessboard, so we have 64 nodes in our graph.**Edges:** Edges are drawn between nodes based on the legal moves of the pieces.

Here's a basic graphical representation:



**Figure 1: Chess Possible Moves**

Here is another example of possible moves in chess but this time with evaluations (possible advantage or disadvantage after a move). Here Black Queen to E7 and pawn to e4 are the best moves for black as they give 0.2 advantage to black.



**Figure 2: Chess Moves with evaluation**

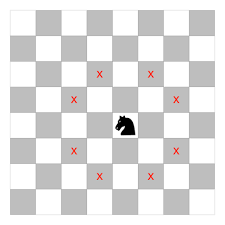
If you’ve ever played chess, you’ve probably noted how the knight’s peculiar L-shaped move takes it around the board.

You’re not alone; its idiosyncrasy been examined by poets, philosophers and mathematicians (including Euler) for a thousand years.

Among the most recent is New College student Michael Dwyer, who studied a classic math problem related to the knight for Professor of Mathematics Eirini Poimenidou’s “Graph Theory” course. He explained it at the class’s semester-end poster presentation.

“The general idea of the Knight’s Tour problem is that you’re trying – using only legal chess moves – to take a knight to every square of the chessboard exactly one time,” Dwyer said. “This relates to graph theory because we can translate the chess board and the possible moves the knight can make at each square into a graph.”

In the graph, he said, each square on the chessboard represents one vertex, and the edges between the vertices are represented by the legal moves the knight can make. From there he goes on to discuss the concept of the Hamiltonian cycle or path, of which the knight’s tour is a specialized case; how the problem changes in cases where you alter the size of the chessboard, where m and n represent the number of squares on a side; “a-b” moves by the knight, and more esoterica.



**Figure 3: Knight's Movement**

So by using graph theory, you can analyze various aspects of Chess:**Connectivity:** Check whether all squares are reachable from all other squares by following the edges (legal moves of the pieces).**Pathfinding:** Finding the shortest path between two squares for a specific piece.**Optimal strategies:** Analyzing optimal move sequences or understanding board positions in terms of graph theory properties like connected components, cycles, etc.

### 

### **1.1.2** **Snakes and ladder:**

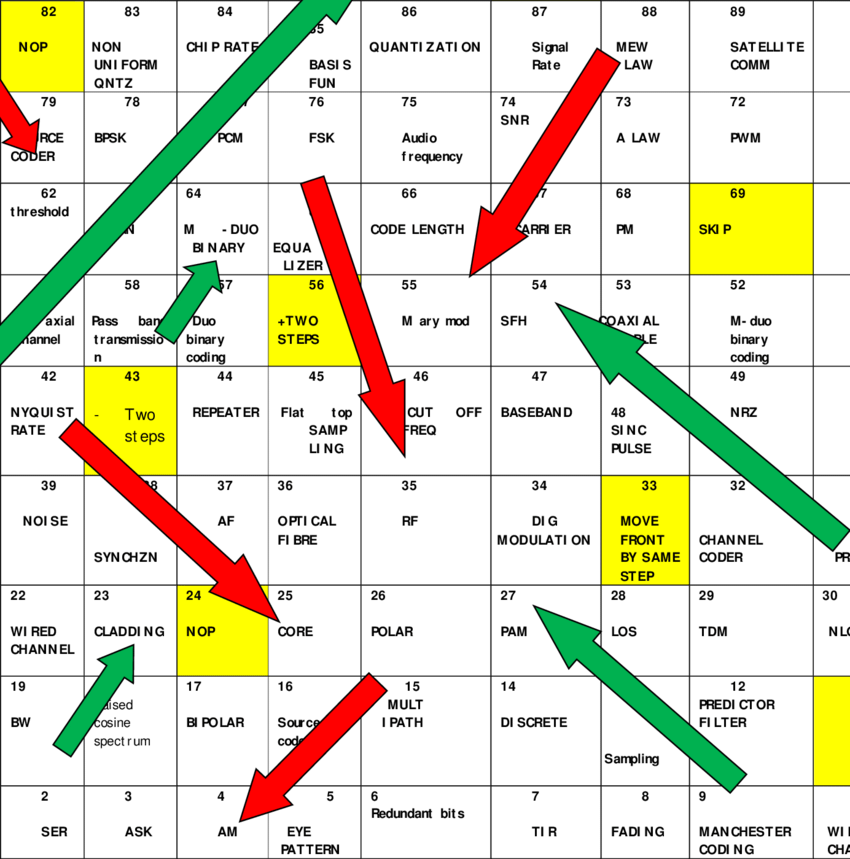
Another game that can be represented using graph theory is the game of Snake and Ladders (or Chutes and Ladders). This classic board game involves players moving across a board with squares numbered from 1 to a certain value, often 100.

**Nodes:** Each square on the board represents a node in the graph.

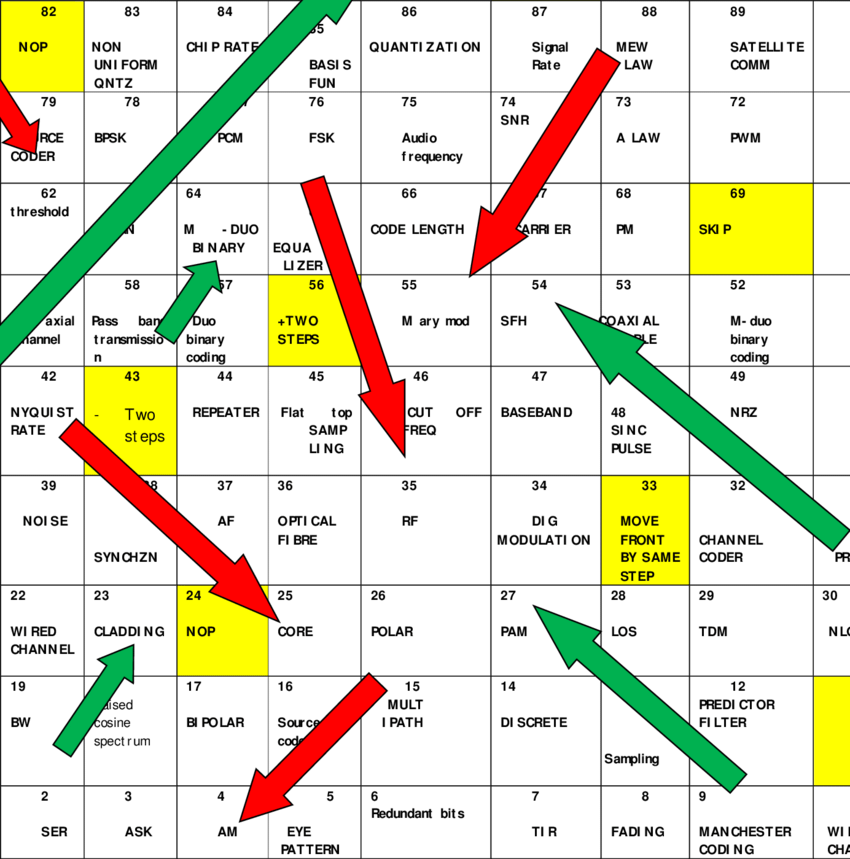
**Edges:** The connections between nodes are the ladder climbs or snake descents. If a player lands on a square with a ladder, they move up to the destination square. If they land on a square with a snake, they move down to the destination square.

By representing Snake and Ladders as a graph, one can analyze the probabilities of reaching different squares, the average number of moves required to win, or even design strategies for optimal play.

Here is an example of a simple puzzle.

  **Figure 4: Snakes and ladder**

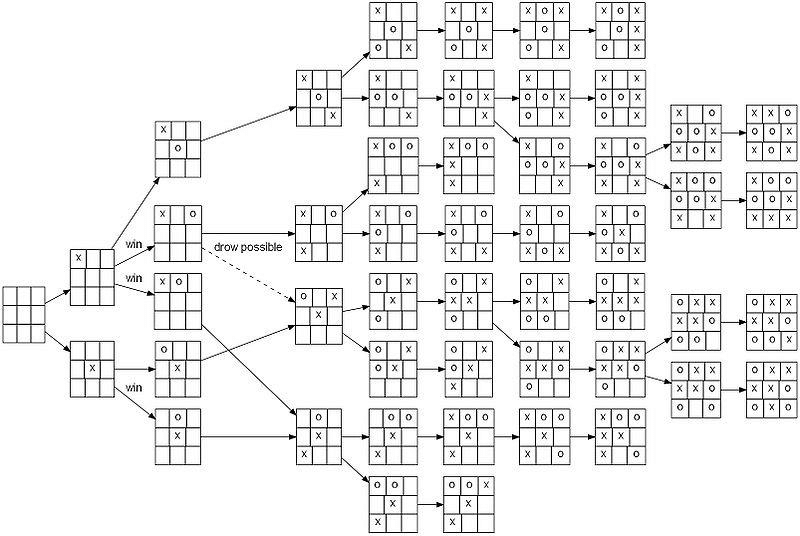
Here is the solved version of this puzzle.



**Figure 5: Solved Version of Snakes and ladder**

## **1.2 Is Tic-Tac-Toe an example of a game that can be represented with graphs? Describe the graph and what you are trying to achieve or avoid. Consider studying Tic-Tac-Toe generalizations.**

Yes, Tic-Tac-Toe can indeed be represented using graphs. To represent Tic-Tac-Toe as a graph, you can create a game tree that illustrates all possible moves and game states.Here's a basic explanation of how Tic-Tac-Toe can be represented as a graph:**Nodes (Vertices):** Each node in the graph represents a possible game state, including the positions of X’s and O’s on the Tic-Tac-Toe grid.**Edges:** Edges between nodes represent legal moves that a player can make. For example, if one node represents a game state with X in the top-left corner, an edge from that node would lead to another node with O in the top-left corner (as a response), and so on for all legal subsequent moves.

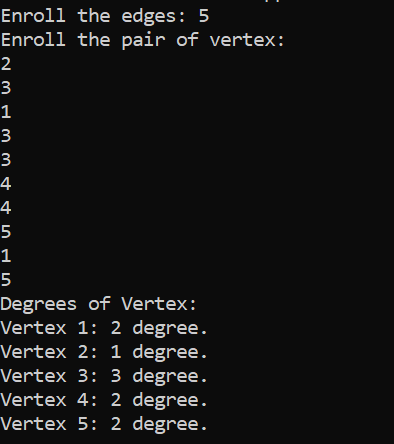
  **Figure 6: Tic Tac Toe**

The graph starts with the initial empty grid as the root node. From there, it branches out into all possible moves that players can make. As the game progresses, the graph expands to include all potential future moves until a terminal state is reached (win, loss, or draw).What you aim to achieve or avoid in studying Tic-Tac-Toe using this graph representation:**Winning Strategies:** By traversing the graph, you can identify winning strategies for either player by analyzing paths that lead to a win (three X’s or O’s in a row, column, or diagonal).**Avoiding Losing Moves**: Understanding the graph helps in identifying moves that lead to inevitable losses. This involves recognizing patterns that result in an opponent's victory if certain moves are made.**Optimal Play:** Graph theory allows for the determination of optimal plays or strategies by evaluating the graph and finding paths that lead to the best possible outcomes, whether it's a win or a draw.**For generalizations of Tic-Tac-Toe:** N x N Tic-Tac-Toe (larger boards): The graph representation expands significantly for larger grids, creating a more extensive game tree with more nodes and edges. Strategies and winning patterns become more complex.Tic-Tac-Toe on a Torus (wrap-around board): Generalizing the board to a torus (where the edges wrap around, creating a continuous grid) introduces additional complexities and alters winning conditions.Overall, by representing Tic-Tac-Toe as a graph, you can analyze different game states, strategies, and outcomes systematically, enabling a deeper understanding of the game and its variations.

# Question#2: Programming Tasks in Graph Theory

## **Undirected Graph Degrees:**

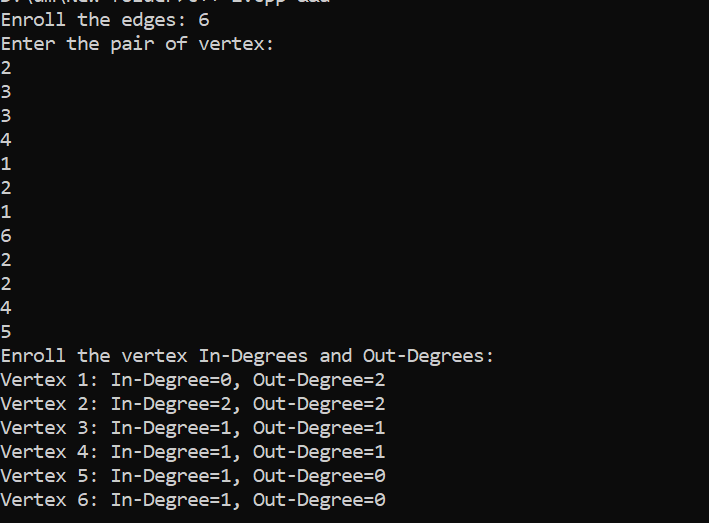




**Figure 7: Undirected Graph Degrees Solution**

## **Directed Graph Degrees:**

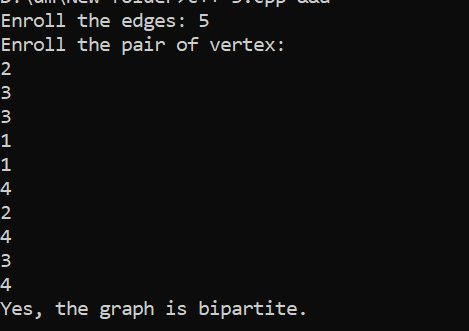




**Figure 8: Directed Graphs Solution**

## **Bipartite Graph Check:**

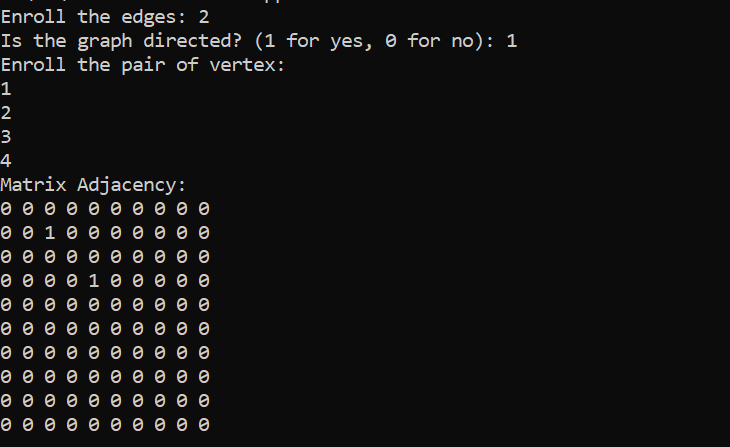




**Figure 9: Bipartite Graph Solution**

## **Adjacency Matrix:**

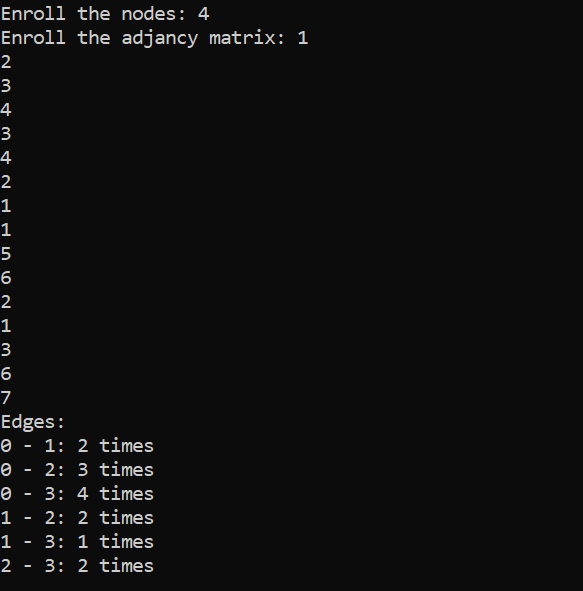




**Figure 10: Adjacency Matrix Solution**

## **Edge Listing from Adjacency Matrix:**

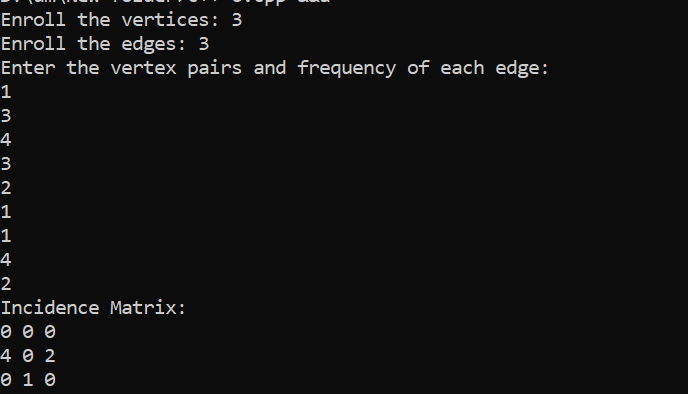




**Figure 11: Edge Listing from Adjacency Matrix Solution**

## **Incidence Matrix Construction:**





**Figure 12: Incidence Matrix Construction Solution**

# Question#3: Programming Tasks in Number Theory

### **Prime Factorization:**

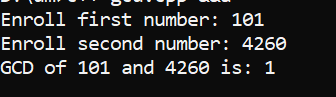




**Figure 13: Prime Factorization Solution**

### **Euclidean Algorithm for GCD:**

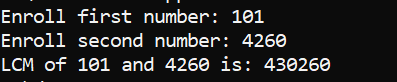




**Figure 14: GCD Euclidean Algorithm Solution**

### **LCM Calculation:**

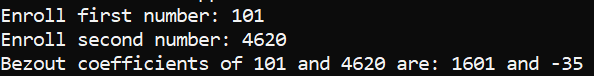
****



**Figure 15: LCM Calculation Solution**

### **Bezout Coefficients:**

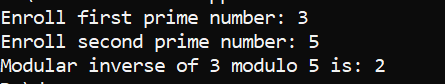




**Figure 16: Bezout Coefficients Solution**

### **Modular Inverse:**

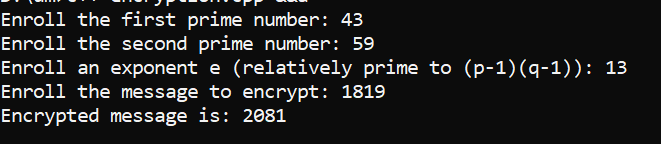




**Figure 17: Modular Inverse Solution**

### **RSA Encryption:**

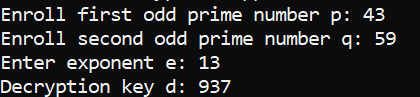




**Figure 18: RSA Encryption Solution**

### **RSA Decryption Key:**





**Figure 19: RSA Decryption Key Solution**