

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015

Certified

Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A Course End Project Report towards Advanced Data Structures Laboratory titled

**SNAKE AND LADDER PROBLEM**

Submitted in the partial fulfilment of the requirements for

the course end project of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted

By

Roll Number: 23885A0502

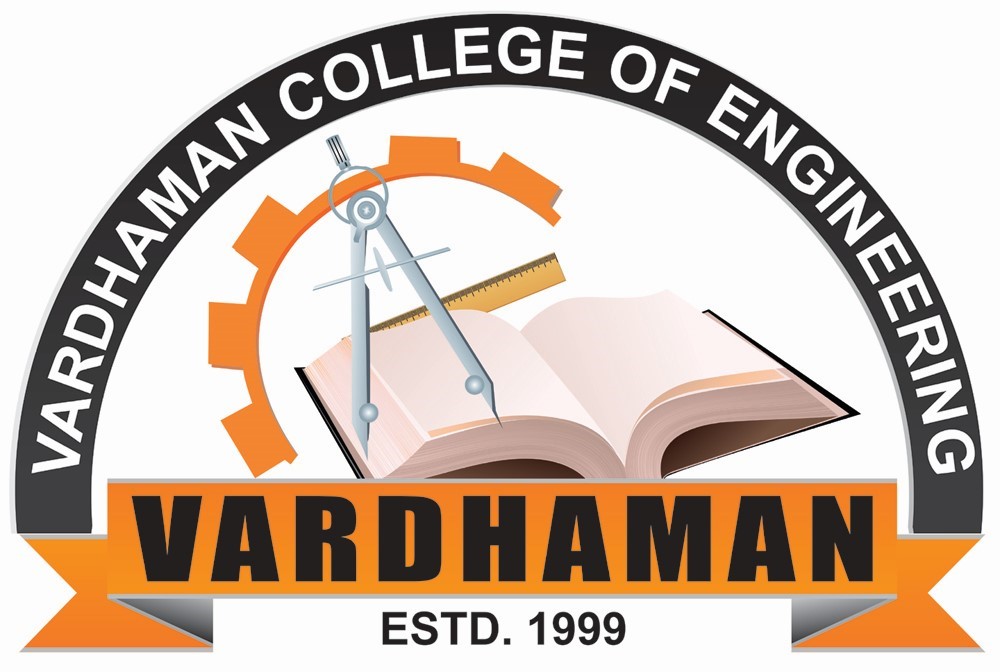
Name of the Student: G.PARINITHA

To

**Mr. M. NARESH GOUD**

**ASSOCIATE PROFESSOR**





**VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD**

An autonomous institute affiliated to JNTUH

**Department of Computer Science and Engineering**

**CERTIFICATE**

This is to certify that the Course End Project titled **“SNAKE AND LADDER GAME” is** carried out by Mr./Ms.**GAJING PARINITHA**, Roll Number **23885A0502** towards **A8513 – Advanced Data Structures Laboratory** course and submitted to **Department of Computer Science and Engineering**, in partial fulfilment of the requirements for the award of degree of **Bachelor of Technology** in **Department of Computer Science and Engineering** during the Academic year 2023-24.

**Name & Signature of the Instructor Name & Signature of the HOD**

**M.Naresh Goud Dr. Ramesh Karnati**

**Assistant Proffesor,**



**TABLE OF CONTENTS**

**CHAPTER NO TITILE**

1. ABSTRACT
2. PROBLEM STATEMENT
3. OBJECTIVES
4. DATA STRUCTURES
5. ALOGORITHM
6. CODE
7. OUTPUT
8. SCREENSHOTS
9. CONCLUSION



**1.ABSTRACT:**

Title: Snakes and Ladders - Abstract Representation

"Snakes and Ladders" is a traditional board game that combines luck and strategy. The game board consists of a grid with numbered squares, and players move their tokens based on the roll of a die. The objective is to reach the final square, typically numbered 100, before opponents.

The game incorporates a unique set of features: snakes and ladders. Certain squares have illustrations of snakes, which act as obstacles, forcing players to slide down to a lower-numbered square. Conversely, ladders connect specific squares, allowing players to climb higher on the board.

The game dynamics involve chance, as players rely on the randomness of dice rolls to advance. Strategic decisions arise as players weigh the risks of landing on snake squares against the rewards of reaching ladders. The presence of these elements adds unpredictability and excitement to the gameplay.

The abstract nature of "Snakes and Ladders" makes it a popular choice for all ages. It serves as an entertaining tool for teaching children about counting and probability while offering a nostalgic experience for adults. The game's simplicity, combined with its elements of chance and strategy, contributes to its enduring appeal across cultures and generations.

Top of Form

**2.PROBLEM STATEMENT**

You are given a standard Snake and Ladder game board of size N x N. The board has cells numbered from 1 to N\*N, where N is a positive integer. Some cells contain either a snake or a ladder.



**3.OBJECTIVES**

1. **Entertainment and Fun:**
   * Engage players in a fun and enjoyable gaming experience.
   * Create a visually appealing game board with attractive graphics.
2. **Educational:**
   * Reinforce basic counting skills for young players.
   * Incorporate educational elements such as math problems or trivia questions on certain squares.
3. **Social Interaction:**
   * Facilitate social interaction among players.
   * Encourage friendly competition and sportsmanship.
4. **Strategy Development:**
   * Include elements that require strategic thinking, such as decision-making points on the board.
   * Develop a balance between luck (rolling the dice) and strategy (making choices based on the current game situation).
5. **Learning Objectives:**
   * Introduce game mechanics and rules that promote learning.
   * Teach players the concept of cause and effect as they move through the board.

**4.DATA STRUCTURE USED:**

1. **Game Board Representation:**
   * **2D Array/Matrix:** The game board can be represented as a 2D array where each cell corresponds to a specific position on the board. The snakes and ladders are placed on specific cells, and players move through the array based on dice rolls.
2. **Player Positions:**
   * **Array or List:** To keep track of the current position of each player on the game board, you can use an array or list. Each element in the array represents the current position of a player.
3. **Snakes and Ladders:**
   * **HashMap or Dictionary:** To represent the mapping of snake heads to tails and ladder bottoms to tops, you can use a data structure like a HashMap (in Java, Python) or a Dictionary (in Python). Each key-value pair in the map corresponds to the head and tail (or bottom and top) of a snake or ladder.

**5.ALGORITHM :**

Here's an algorithm for the Snake and Ladder problem:

1. **Create the Board Graph:**
   * Create a graph representation where each cell on the board is a node.
   * For each cell, connect it to the next six cells (if they exist) since a die roll can result in a number from 1 to 6.
2. **Include Snake and Ladder Edges:**
   * For each snake, add an edge from the snake's head to its tail.
   * For each ladder, add an edge from the ladder's bottom to its top.
3. **BFS to Find Shortest Path:**
   * Initialize a queue for BFS and enqueue the starting node (position 1).
   * Create an array to store the minimum number of moves required to reach each cell. Initialize it with infinity, except for the starting cell (set its value to 0).
   * Perform BFS, updating the minimum moves array as you explore nodes.
   * Continue BFS until you reach the destination (position 100).
4. **Trace Back to Get Path:**
   * Once you reach the destination, you can trace back the shortest path using the information stored in the minimum moves array.

**6.CODE**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

// Function to roll a six-sided die

int rollDie() {

return rand() % 6 + 1;

}

// Function to play Snake and Ladder

void playGame() {

int playerPosition = 1;

 int dice, newPosition;

printf("Welcome to Snake and Ladder!\n");

while (playerPosition < 100) {

// Roll the die

dice = rollDie();

printf("You rolled a %d.\n", dice);

// Calculate the new position

newPosition = playerPosition + dice;

// Check for snakes and ladders

switch (newPosition) {

case 98: newPosition = 28; break;

case 95: newPosition = 24; break;

case 92: newPosition = 51; break;

case 83: newPosition = 19; break;

case 73: newPosition = 1; break;

case 48: newPosition = 9; break;

case 46: newPosition = 5; break;

case 16: newPosition = 6; break;

case 49: newPosition = 11; break;

case 64: newPosition = 60; break;

case 62: newPosition = 19; break;

case 87: newPosition = 24; break;

 case 89: newPosition = 26; break;

case 68: newPosition = 34; break;

case 65: newPosition = 51; break;

case 36: newPosition = 44; break;

case 43: newPosition = 77; break;

case 98: newPosition = 28; break;

}

// Update player position

if (newPosition <= 100) {

playerPosition = newPosition;

printf("You are now at position %d.\n", playerPosition);

} else {

printf("You need a %d to win.\n", 100 - playerPosition);

}

}

printf("Congratulations! You won!\n");

}

int main() {

// Seed the random number generator

srand(time(NULL));

// Start the game

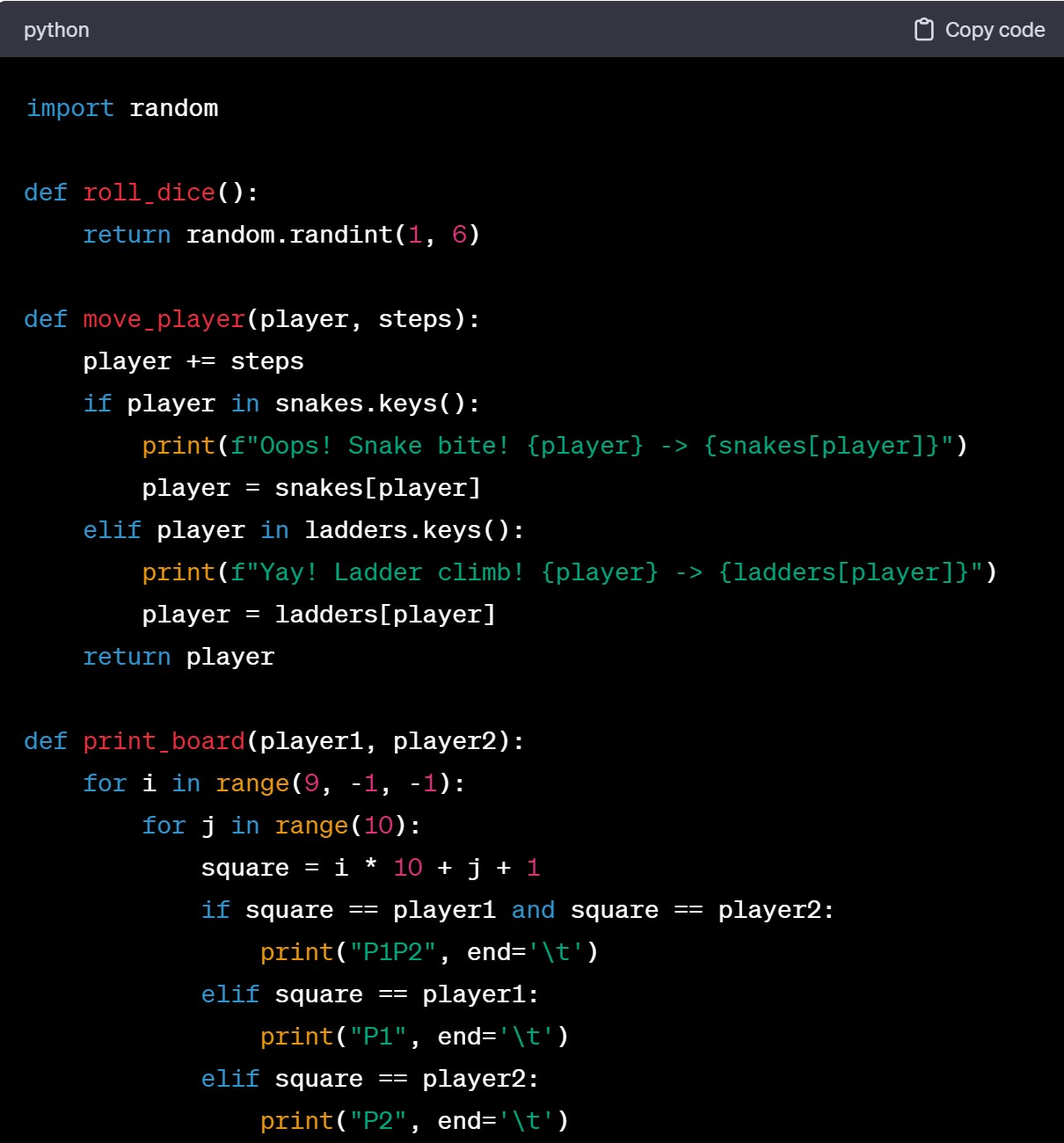
playGame();



return 0;

}

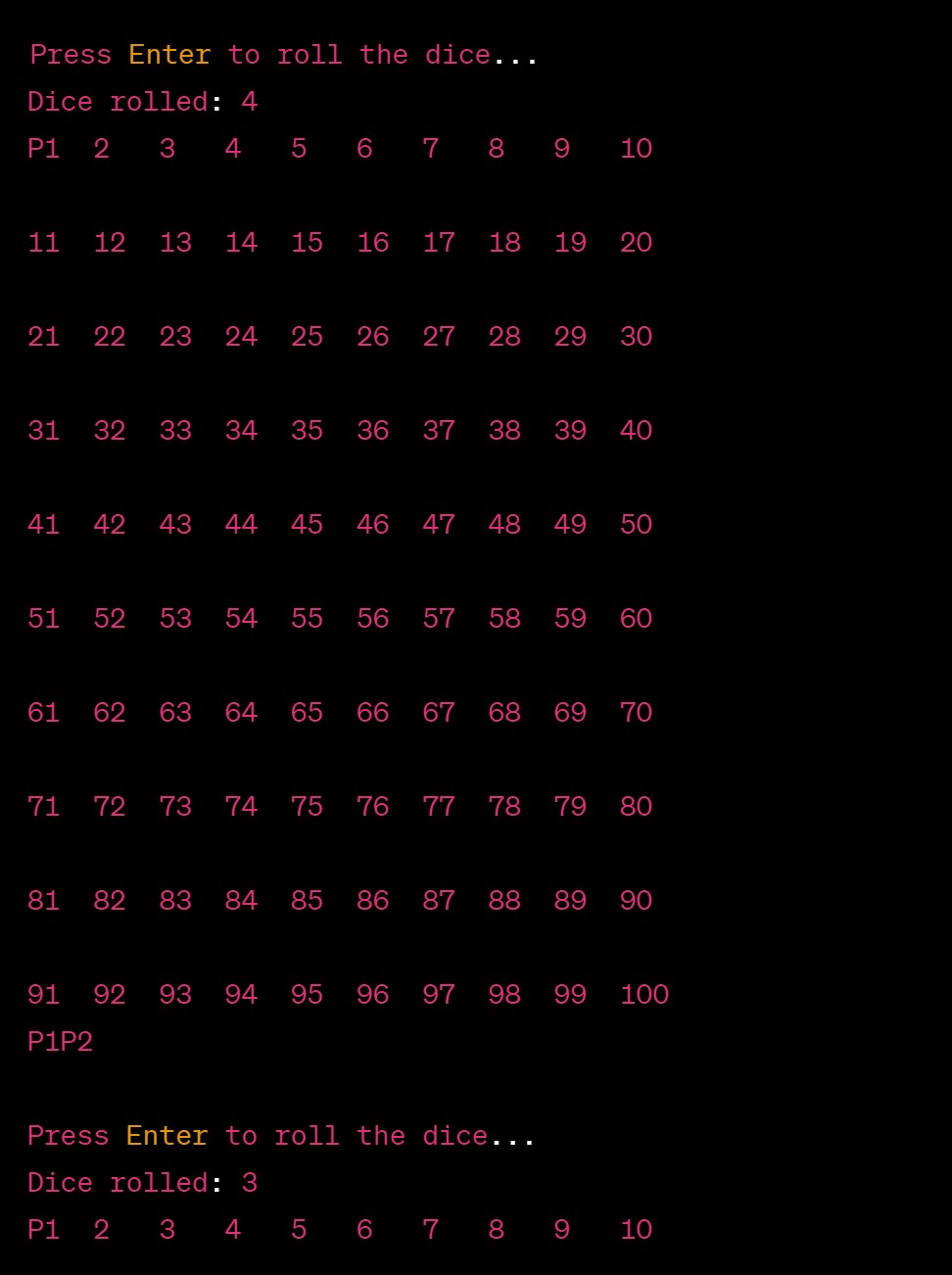
**8.SCREENSHOTS:**

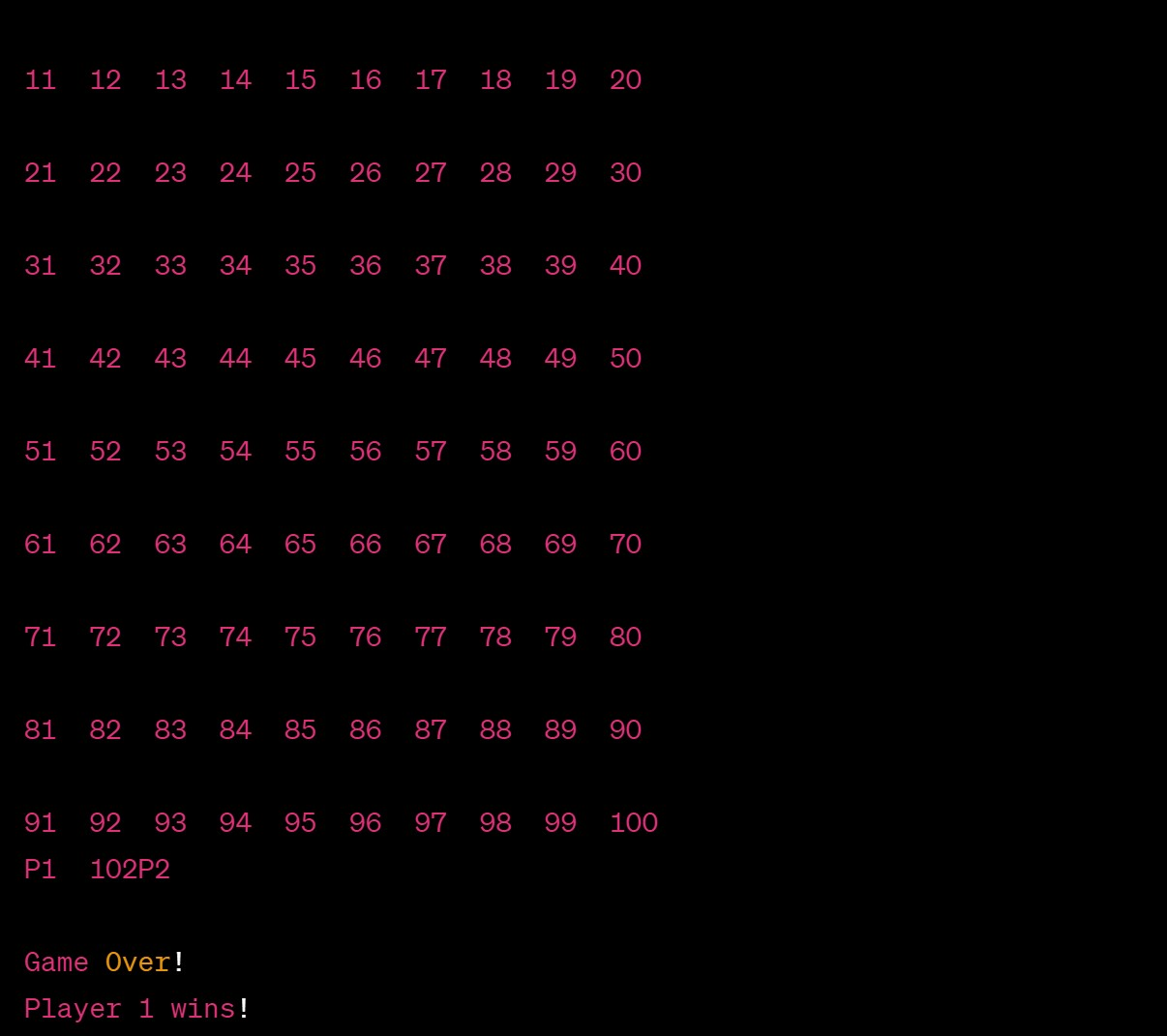
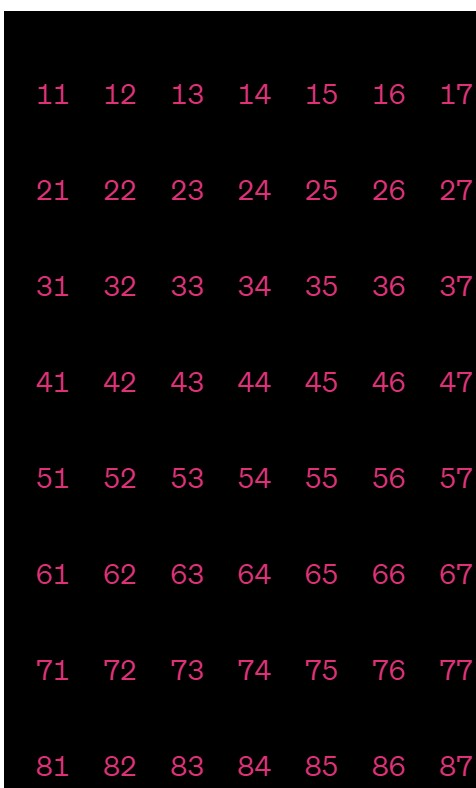






**Output:**





**9.CONCLUSION**

In conclusion, the Snake and Ladder game is a classic and simple board game that combines luck and strategy. The provided Python code offers a basic implementation of the game in a console-based format. Players take turns rolling a dice, move their positions on the board, and may encounter snakes or ladders, adding an element of surprise and strategy to the game.

The code can be a starting point for those looking to understand the fundamentals of programming or game development. It demonstrates concepts such as random number generation, conditional statements, and loops in Python. Additionally, it showcases how to incorporate elements like snakes and ladders to enhance the gameplay experience.

For further development, one could consider adding more features such as a graphical user interface (GUI), sound effects, or multiplayer support. Adapting the game to different themes or creating more complex rules could also provide additional challenges and entertainment. The flexibility of the code allows for customization and expansion based on individual preferences and learning goals.

Overall, the Snake and Ladder game serves as a fun and educational project, suitable for beginners in programming or those looking to explore game development concepts