

# Snake and Ladder Game – DSA Project

## ## Overview

This is a terminal-based Snake and Ladder game implemented in the C programming language as part of a **Data Structures and Algorithms (DSA)** project.

The project demonstrates multiple DSA concepts, especially **Doubly Linked Lists**, to simulate the game board and manage player movement efficiently.

---

## ## 🔑 Key Features

### - **Dynamic Game Board**

The 100-square board is implemented using a **Doubly Linked List**, where each node represents a square and allows two-way traversal.

### - **Multi-Player Support**

Up to 10 players supported. Player state is maintained via structured data types.

### - **Snakes and Ladders Logic**

Handled using pointer redirection to jump to destination squares.

### - **Randomized and Manual Dice Rolls**

Choose between `rand()`-based auto roll and manual user input. Includes logic for rolling a 6 to get an extra turn.

### - **Visual Game Display**

Clean console layout showing player positions and square numbers.

## ## □ DSA Concepts Implemented

1. **\*\*Doubly Linked List\*\*** – Used to form the game board with 100 connected nodes.
2. **\*\*Arrays\*\*** – For fast square access and managing player states.
3. **\*\*Structures\*\*** – `NODE` and `PLAYER` for organized data management.
4. **\*\*Pointers\*\*** – To handle player movement, jumps, and node traversal.
5. **\*\*Modular Programming\*\*** – Clear separation of logic via `.c` and `.h` files.

---

## ## ✂ Compilation Instructions

```
gcc -Wall -c main.c game_logic.c
gcc main.o game_logic.o -o Dsa
./Dsa
```

## ## File Summary -

File	Description
main.c	Handles game loop, dice, player turns
game_logic.c	Contains all board logic and movement rules
game_logic.h	Struct definitions and function prototypes

## Sample Output

\*\*\*\*\* SNAKE AND LADDER GAME \*\*\*\*\*

Snakes: 46 -> 9, 77 -> 23, 99 -> 1

Ladders: 14 -> 36, 43 -> 81, 71 -> 98

Player 1 rolled a 6

Player 1 got one more chance to roll die

Player 1 rolled a 4

Player 1 moved to 10

### ✓ Conclusion

This project is a great example of how **classical DSA concepts** can be applied to build a functional and interactive game. It showcases the use of **linked lists, structures, arrays, and pointers** in a real-world C application.