# SNAKE AND LADDERS

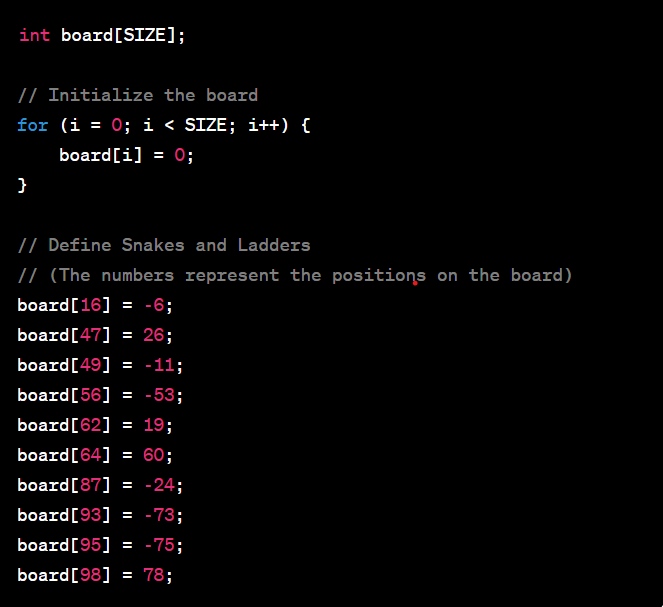
DONE BY: PAVANI M

AKSHARA M

**A Snake and Ladders game code is a program that simulates the classic board game known as "Snakes and Ladders" or "Chutes and Ladders." The game is played on a square board with numbered squares, and players take turns rolling a die and moving their game pieces based on the result. The board contains special positions, some of which are connected by snakes or ladders, influencing the players' progress.**

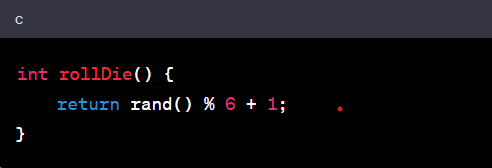
**1.Board Representation:**

* **The game board is represented using 1D array named ‘board’. Each element of the array corresponds to a position on the board. The array is initialised to all zeros initially.**
* **Certain positions on the board represent snakes and ladders, and their effects are defined in the ‘board’ array.**

****

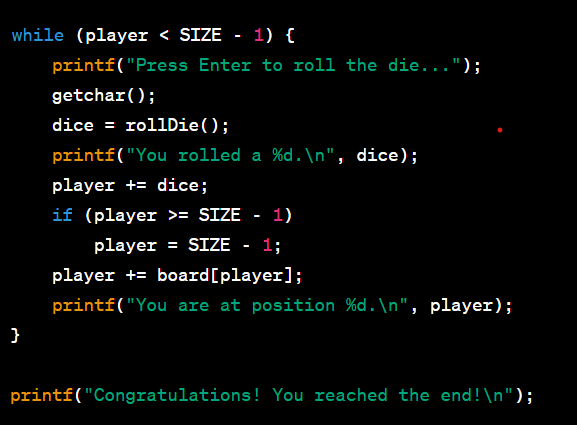
**2.Rolling the die:**

* **The ‘rolldie’ function generates a random number between 1 and 6, simulating the roll of a die.**
* **Rand() typically generates a random integer in the range [0,’RAND\_MAX’], where ‘RAND\_MAX’ is a constant defined in <stdlib.h>**

****

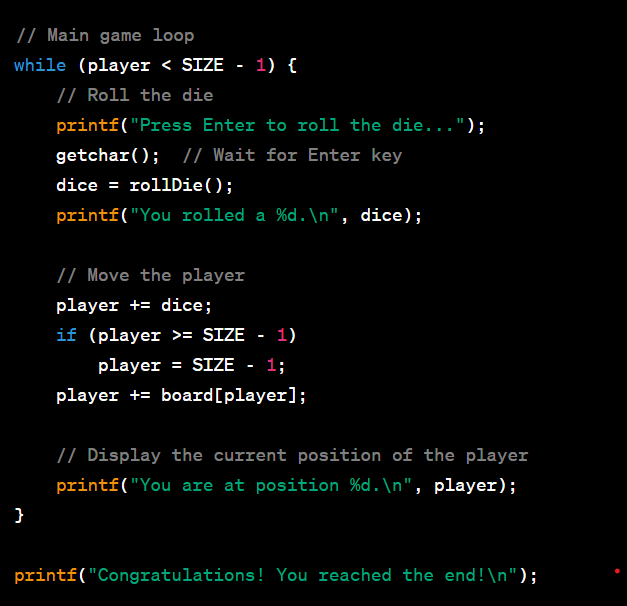
**3.Game loop:**

* **The main game loop continues until one of the players reaches or exceeds the last position on the board.**
* **Inside the loop, players take turns rolling the die, moving forward, and checking for snake and ladders.**

****

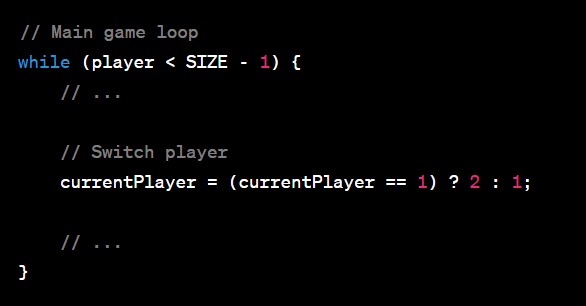
**5.Player Movement:**

* **Players(Player 1 and Player 2) have their positions on the board (‘player 1’ and ‘player2’ variables). They move forward by the number rolled on the die.**
* **The effects of snakes and ladders are applied by adding the corresponding value from the ‘board’ array to the players position.**

****

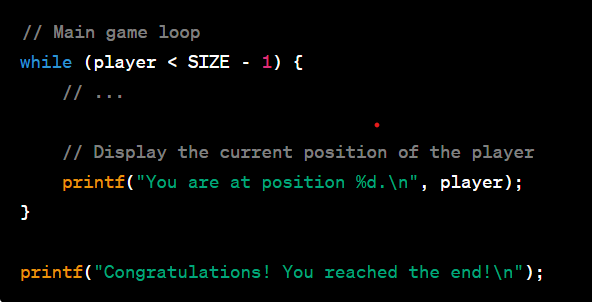
**5. Switching Players:**

* **The variable ‘current player’ is used to keep a track of the current player(1 or 2). It switches after each turn.**

****

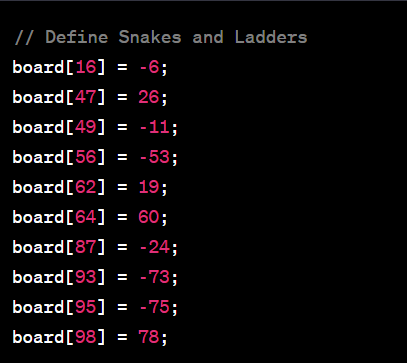
**6.Winning Condition:**

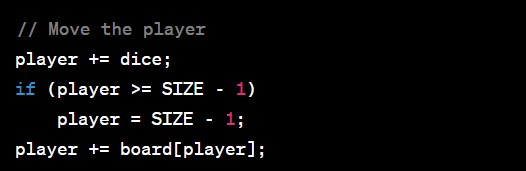
* **The game declares a winner when one of the players reaches or exceeds the last position on the board.**

****

**7.Snakes and ladders:**

* **Snakes and ladders are implemented as non-zero values in the ‘board’ array. When a player lands on a position with the snake or ladder, the player’s position is adjusted accordingly.**

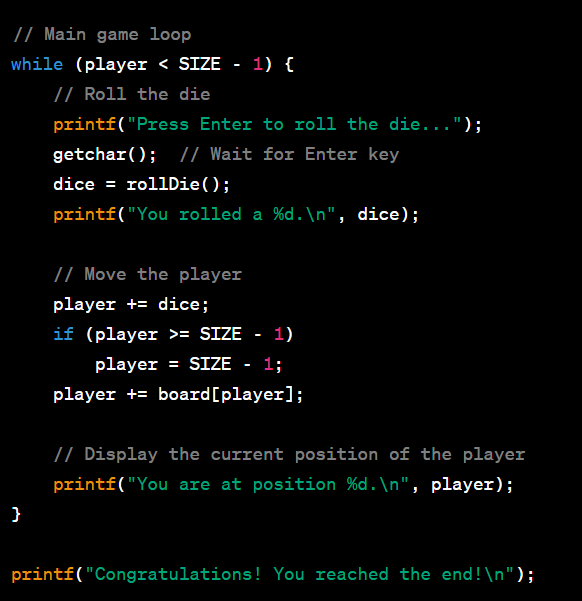
****



**So, the implementation of snakes and ladders is integrated into the movement logic within the game loop. The effects of landing on a aquare with a snake or ladder are applied during the players’s turn.**

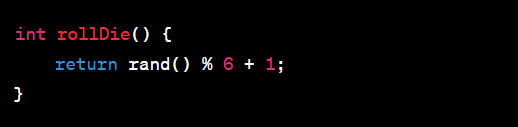
**8.Printing Information:**

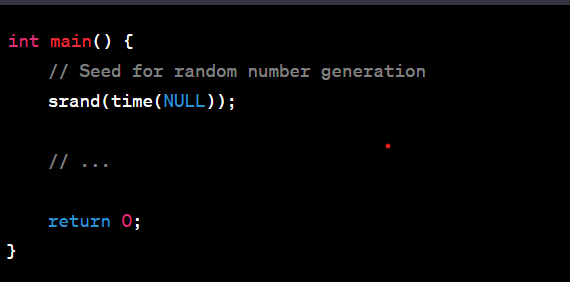
* **The code prints information about the game , such as the current player, the result of the die roll, and positions of both players.**

****

**9. Random number generation:**

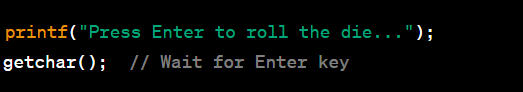
* **The ‘srand’ function is used to seed the random number generator based on the current time. This ensures different sequences of random numbers in each run.**

****

****

**10. Input from players:**

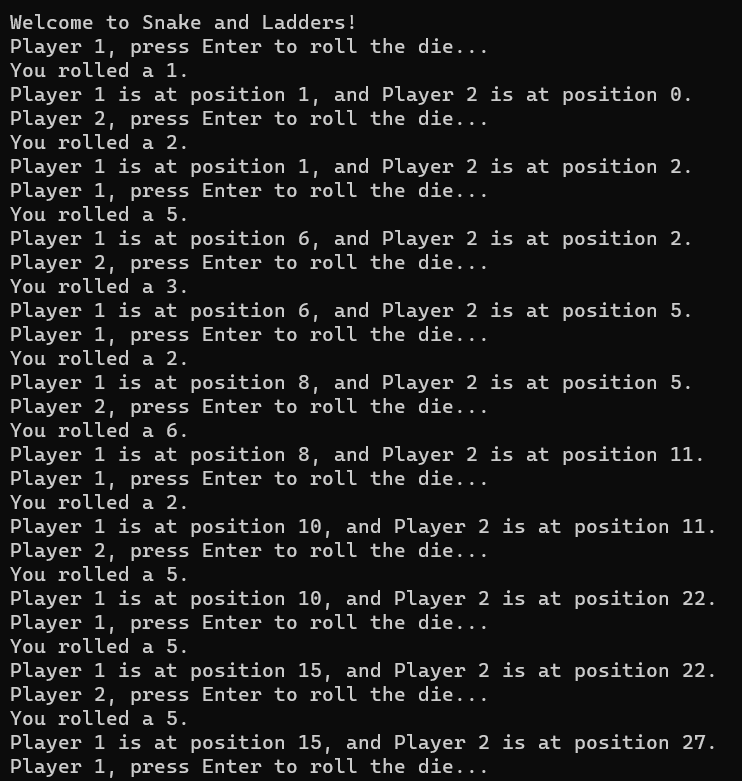
* **Players are prompted to press enter to roll the die. This provides a simple way to simulate player turns.**

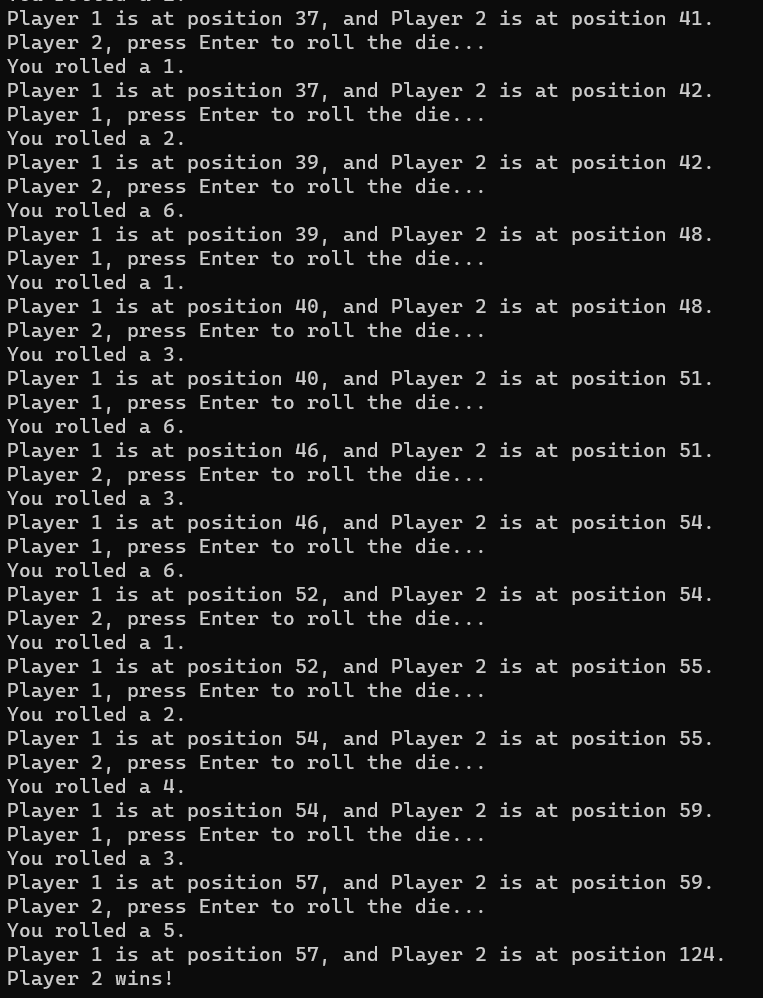
****

**11. Modularity:**

* **Functions like ‘rolldie’ and ‘playgame’ contribute to a modular structure, enhancing readability and maintainability.**

**OUTPUT OF THE ABOVE CODE:**

****

****