```
// C Program to implement Snake and Ladder Game
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
// Function to roll a six-sided die
int rollDie() { return rand() % 6 + 1; }

// global variables to store postions of player1 and player2
int player1 = 0, player2 = 0;

// Function to print the board
void printBoard()
{
   // logic to print a snake-ladder Game board
   // programmer can implement their own logic for the board,
    // this is one way to print a snake ladder board.
   int board[101];
   for (int i = 1; i <= 100; i++) {
      board[i] = i;
   }

   int alt = 0; // to switch between the alternate nature of the board
   int iterLR = 101; // iterator to print from left to right
   int iterRL = 80;  // iterator to print from right to left
   int val = 100;
   while (val--) {
```

```
if (alt == 0) {
    iterLR--;
    if (iterLR == player1) {
        printf("#P1    ");
    }
    else if (iterLR == player2) {
        printf("#P2    ");
    }
    else
        printf("%d    ", board[iterLR]);

    if (iterLR % 10 == 1) {
        printf("\n\n");
        alt = 1;
        iterLR -= 10;
    }
}
else {
    iterRL++;
    if (iterRL == player1) {
        printf("#P1    ");
    }
    else if (iterRL == player2) {
        printf("#P2    ");
    }
    else
```

```
            printf("%d    ", board[iterRL]);

        if (iterRL % 10 == 0) {
            printf("\n\n");
            alt = 0;
            iterRL -= 30;
        }
      }
      if (iterRL == 10)
        break;
   }
   printf("\n");
}

// Function to move the player
int movePlayer(int currentPlayer, int roll)
{
   int newPosition = currentPlayer + roll;
   // Define the positions of snakes and ladders on the
   // board
   int snakesAndLadders[101];

   for (int i = 0; i <= 100; i++) {
      snakesAndLadders[i] = 0;
   }
```

```
    // here positive weights represent a ladder
    // and negative weights represent a snake.
  snakesAndLadders[6] = 40;
  snakesAndLadders[23] = -10;
  snakesAndLadders[45] = -7;
  snakesAndLadders[61] = -18;
  snakesAndLadders[65] = -8;
  snakesAndLadders[77] = 5;
  snakesAndLadders[98] = -10;

  int newSquare
    = newPosition + snakesAndLadders[newPosition];

  if (newSquare > 100) {
    return currentPlayer; // Player cannot move beyond
                // square 100
  }

  return newSquare;
}

int main()
{
  srand(time(0)); // Initialize random seed
  int currentPlayer = 1;
  int won = 0;
```

```
printf("Snake and Ladder Game\n");

while (!won) {

   printf(
      "\nPlayer %d, press Enter to roll the die...",
      currentPlayer);
   getchar(); // Wait for the player to press Enter
   int roll = rollDie();
   printf("You rolled a %d.\n", roll);

   if (currentPlayer == 1) {
      player1 = movePlayer(player1, roll);
      printf("Player 1 is now at square %d.\n\n",
            player1);
      printBoard();
      if (player1 == 100) {
         printf("Player 1 wins!\n");
         won = 1;
      }
   }
   else {
      player2 = movePlayer(player2, roll);
      printf("Player 2 is now at square %d.\n\n",
            player2);
```

```
            printBoard();
            if (player2 == 100) {
                printf("Player 2 wins!\n");
                won = 1;
            }
        }

        // Switch to the other player
        currentPlayer = (currentPlayer == 1) ? 2 : 1;
    }

    return 0;
}
```