VISHNU
UNIVERSAL LEARNING

BVRIT H

BVRIT HYDERABAD College of
Engineering for Women (UGC-Autonomous)

PPS LAB ACTIVITY

Department of CSE Certified that this is a Bonafide Record of
the word done by:

Adifa Saniya - 93

G.Hansini - 71

Gayathri – 10

Of Class CSE C of Year 1 of Semester 1 in PPS Laboratory

DATE: SIGNATURE:

// C Program to implement Snake and Ladder Game

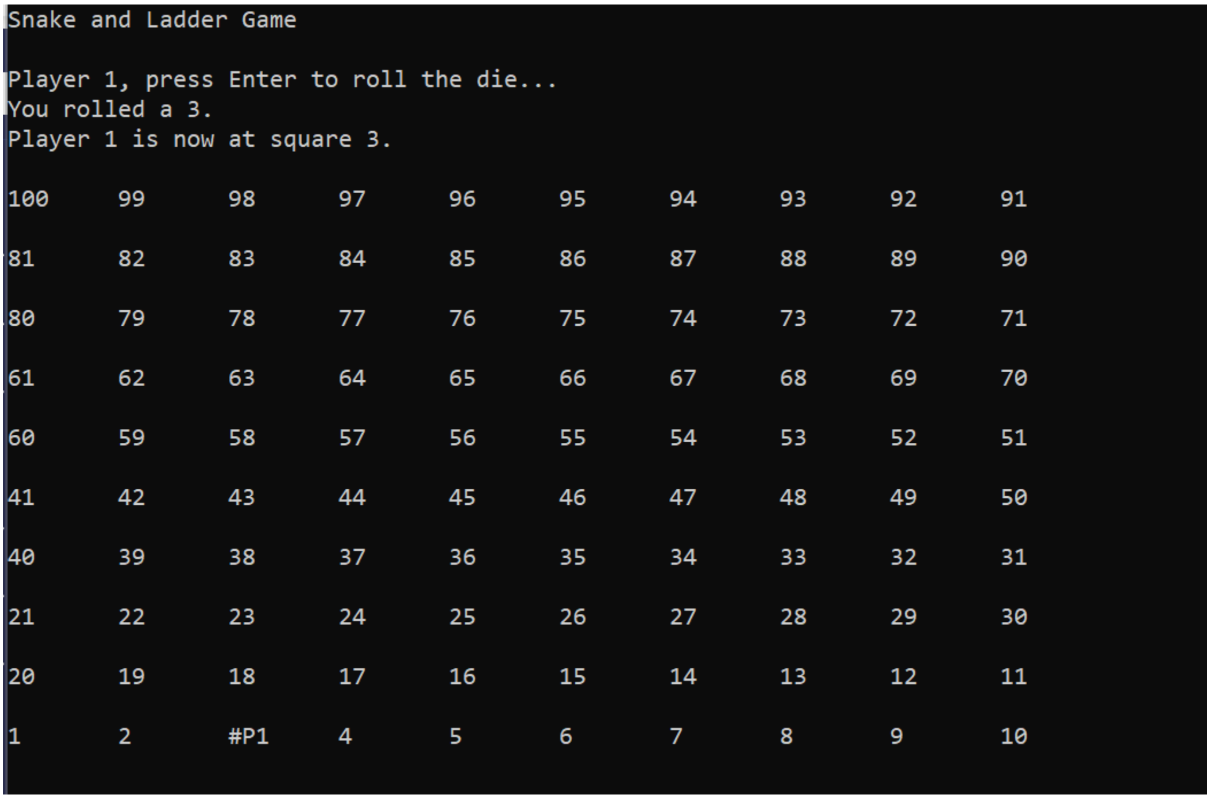
Implement snake and ladder game. It’s a 2 player game. It should start at square 1 and should finish at square 100. At regular interval it should have snakes and ladder. Snake should lead to decrease in current score/place of a player and ladder should lead to increase in score/place. Generate a random number from 1 to 6 considering it as a dice. Display what number is generated and accordingly update respective player score.(Traditional snake and ladder game). Once player reaches to square 100 display the winning message for that player.

```
#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]);
```

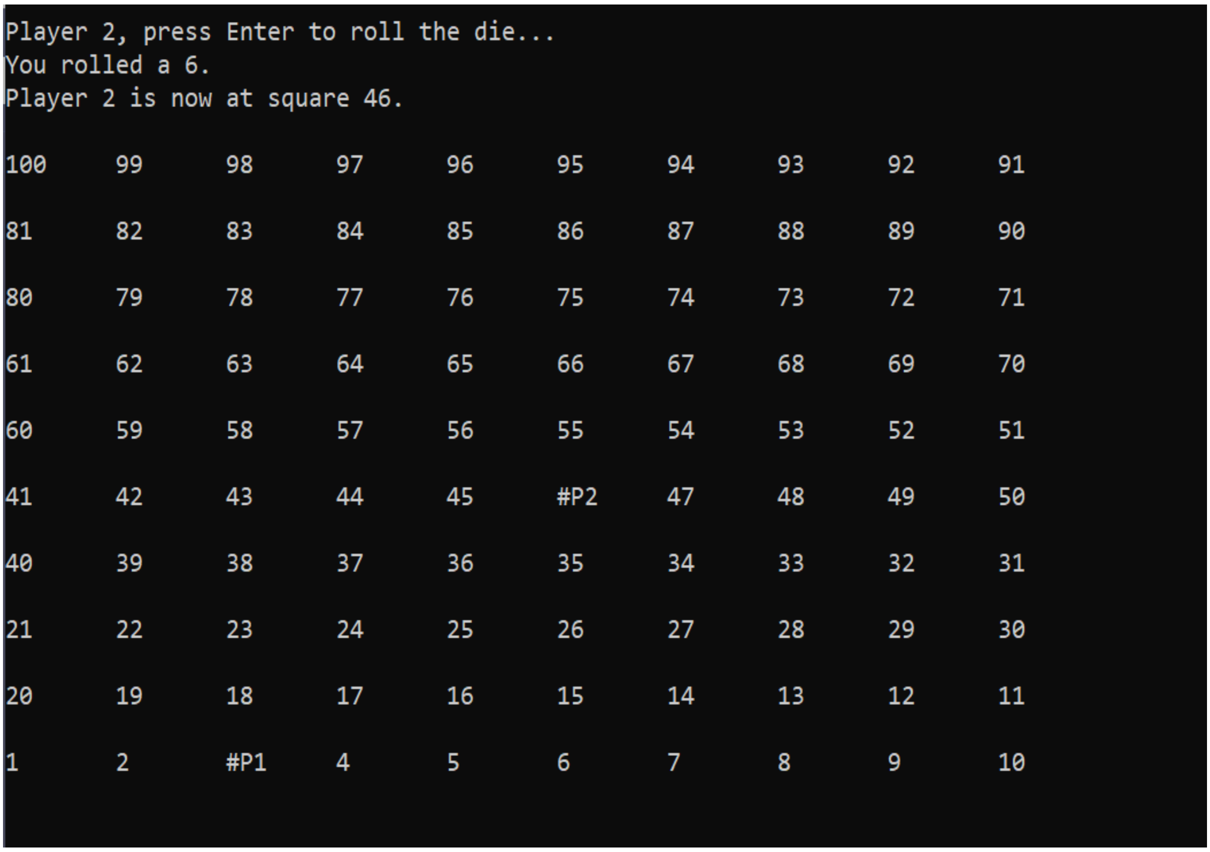
```
                        if (iterRL % 10 == 0) {
                                printf("\n\n");
                                alt = 0;
                                iterRL -= 30;
                        }
                }
                if (iterRL == 10)
                        break;
    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

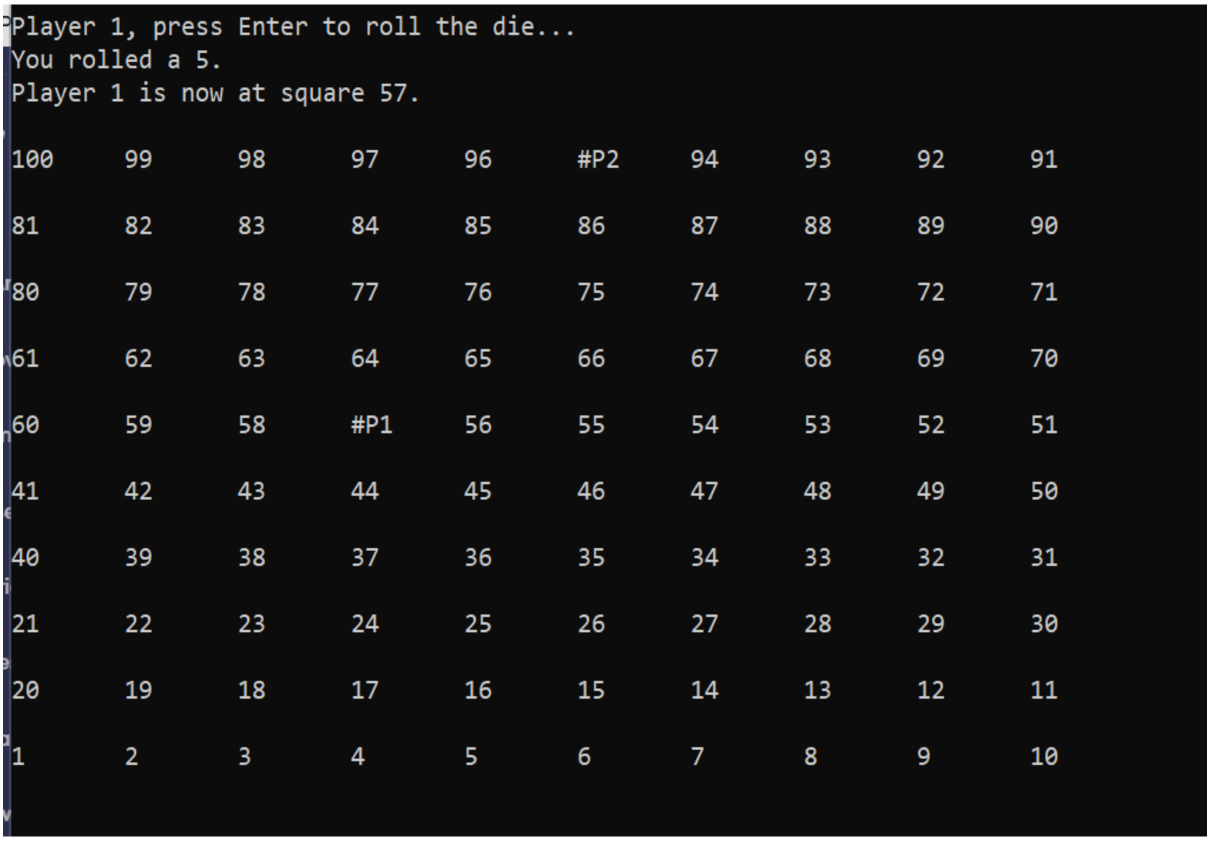
OUTPUT:



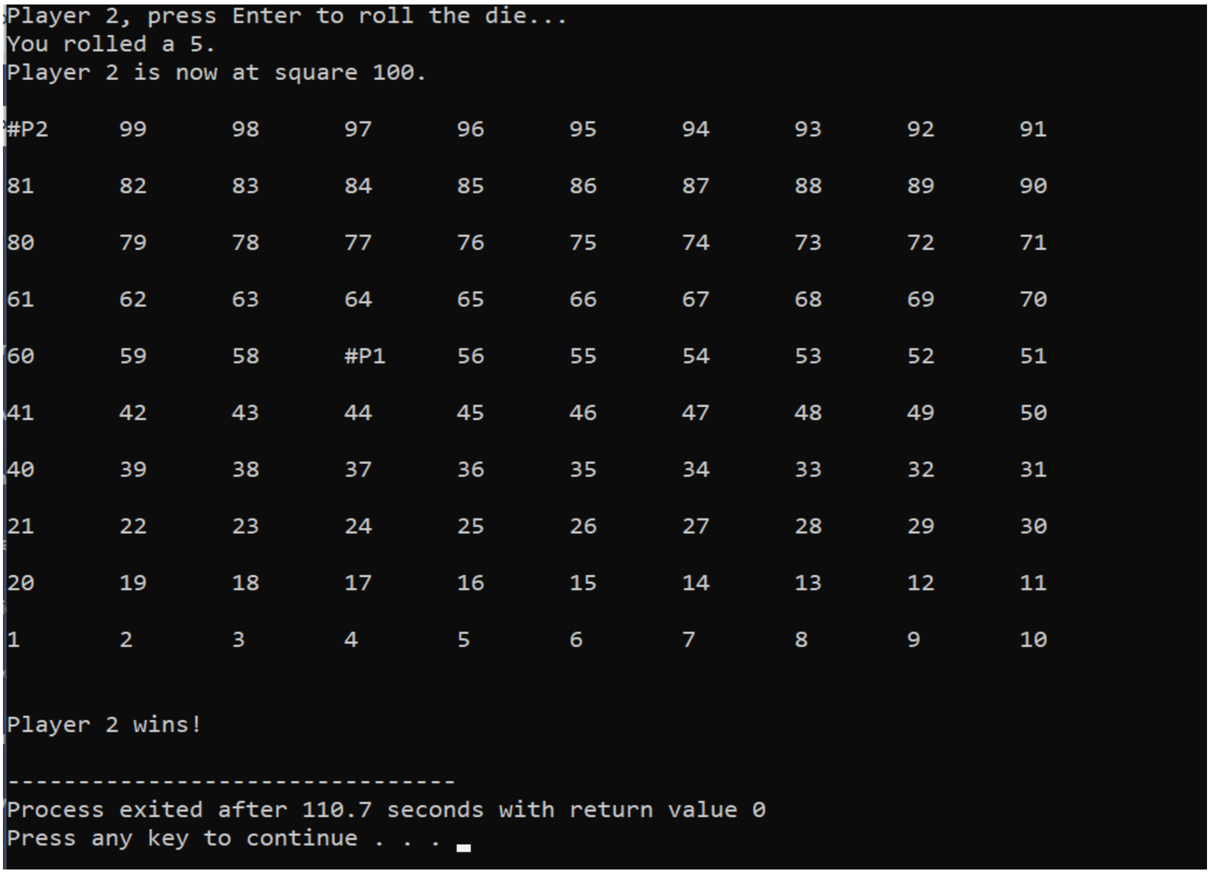
```
Snake and Ladder Game

Player 1, press Enter to roll the die...
You rolled a 3.
Player 1 is now at square 3.

100     99      98      97      96      95      94      93      92      91

81      82      83      84      85      86      87      88      89      90

80      79      78      77      76      75      74      73      72      71

61      62      63      64      65      66      67      68      69      70

60      59      58      57      56      55      54      53      52      51

41      42      43      44      45      46      47      48      49      50

40      39      38      37      36      35      34      33      32      31

21      22      23      24      25      26      27      28      29      30

20      19      18      17      16      15      14      13      12      11

1       2       #P1     4       5       6       7       8       9       10
```



```
Player 2, press Enter to roll the die...
You rolled a 6.
Player 2 is now at square 46.

100     99      98      97      96      95      94      93      92      91

81      82      83      84      85      86      87      88      89      90

80      79      78      77      76      75      74      73      72      71

61      62      63      64      65      66      67      68      69      70

60      59      58      57      56      55      54      53      52      51

41      42      43      44      45      #P2     47      48      49      50

40      39      38      37      36      35      34      33      32      31

21      22      23      24      25      26      27      28      29      30

20      19      18      17      16      15      14      13      12      11

1       2       #P1     4       5       6       7       8       9       10
```



```
Player 1, press Enter to roll the die...
You rolled a 5.
Player 1 is now at square 57.

100     99      98      97      96      #P2     94      93      92      91

81      82      83      84      85      86      87      88      89      90

80      79      78      77      76      75      74      73      72      71

61      62      63      64      65      66      67      68      69      70

60      59      58      #P1     56      55      54      53      52      51

41      42      43      44      45      46      47      48      49      50

40      39      38      37      36      35      34      33      32      31

21      22      23      24      25      26      27      28      29      30

20      19      18      17      16      15      14      13      12      11

1       2       3       4       5       6       7       8       9       10
```



```
Player 2, press Enter to roll the die...
You rolled a 5.
Player 2 is now at square 100.

#P2     99      98      97      96      95      94      93      92      91

81      82      83      84      85      86      87      88      89      90

80      79      78      77      76      75      74      73      72      71

61      62      63      64      65      66      67      68      69      70

60      59      58      #P1     56      55      54      53      52      51

41      42      43      44      45      46      47      48      49      50

40      39      38      37      36      35      34      33      32      31

21      22      23      24      25      26      27      28      29      30

20      19      18      17      16      15      14      13      12      11

1       2       3       4       5       6       7       8       9       10


Player 2 wins!

--------------------------------
Process exited after 110.7 seconds with return value 0
Press any key to continue . . .
```