Snake And Ladder Game

Program:

```
                        /* Snake And Ladder Game */
/* Moghees Ahmad (20F-0244)          Anas Abdullah (20F-0267) */

#include"pch.h"
#include<iostream>
#include<iomanip>
#include<windows.h>                          //Pre processive directories
#include<string>
#include<time.h>
#include<stdlib.h>
#include<fstream>

using namespace std;

string name[4];
int score1, score2, score3, score4;

/*Prototypes Of Functions*/

void menu();

void playgame();
void names();
void Table();
void score();
void index_check(int& score1, int& score2, int& score3, int& score4);
int position(int& x);
int reward(int& score1, int& score2, int& score3, int& score4);
int snake(int& score);
int ladder(int& score);

void display();

void credits();

void rule();

void writeRecord(string name, int num);

/*Main Function*/

int main()
{
      menu();                                //Calling 'menu' function
      return 0;
}

/*This Function Will display menu then it will get choice from user and do
what user asks.*/

void menu()
{
      /* Play in Full Screen Mode (Font size 16 would be good) */

      display();                             //Calling 'Display' function
      int choice;                            //Asking user what to do
      cin >> choice;
      switch (choice)
      {
      case 1:
            playgame();                   //Calling 'playgame' function
            break;
      case 2:
            credits();                          //Calling 'Credits'
function
            break;
      case 3:
            rule();                             //Calling 'rule' function
            break;
      case 4:
      {
            system("cls");
            cout << "\n\n\n\n\n\n\t\t\tRECORDS\n";
            cout << "\n\t\t" << setfill('_') << setw(40) << " " << "\n\n";
            string  ch;
            ifstream in;
            in.open("RECORD.txt");
            while (in.eof() == 0)               //Displaying Records
            {
                  cout << "\t\t";
                  getline(in, ch);
                  cout << ch;
            }
            in.close();
            cout << endl;
            cout << "\t\t" << setfill('_') << setw(40) << " " << "\n\n";
            system("pause");
            system("cls");
            menu();
            break;
      }
      case 5:
            system("exit");                     //End Game
      }
}

/*This Function will display menu when it is called*/

void display()
{
      system("color 60");
      cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t\t" <<
setfill('_') << setw(35) << " \n\n" << "\t\t\t\t\t\t\t\t\t"
            << "WELCOME TO SNAKE AND LADDER GAME.\n" <<
"\t\t\t\t\t\t\t\t\t" << setfill('_') << setw(35) << " \n\n";
      Sleep(2000);
      cout << "\n\t\t\t\t\t\t\t\t\t\tLOADING ";
      Sleep(500); cout << ".";
      Sleep(500); cout << ".";
      Sleep(500); cout << ".";
      system("cls");
      cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t\t    MENU\n\n"
            << "\t\t\t\t\t\t\t\t\t1.PLAY GAME\n\n\t\t\t\t\t\t\t\t\t"
            << "2.CREDITS\n\n\t\t\t\t\t\t\t\t\t3.RULES
\n\n\t\t\t\t\t\t\t\t\t"
            << "4.RECORDS \n\n\t\t\t\t\t\t\t\t\t5.EXIT\n";
}

/*This Function Will Display Rules*/

void rule()
{
      system("cls");
      cout << "\n\t\t\t\t\t\t\tRULES\n\n\t\t1. As the game is based on
four players so after selecting the play game option first game\n"
```

```
            << "\t\twill get the names of all players by input the
player’s nameand then your game will start\n"
            << "\t\tfrom a toss, who won the toss has the first turn then
the second, thirdand then the fourth player.\n"
            << "\n\t\t2. Board contains at five ladders to climb up and
five snakes to bite down.Suppose if\n"
            << "\t\tthe player is on S(snake) index which is the 50th
index on board then its updates position\n"
            << "\t\tmay be down to the 10th index and if the player is on
L(ladder) index which is the 40th\n"
            << "\t\tindex on board than its updated position may be upon
70th index.\n"
            << "\n\t\t3. Any player can start playing by getting six on
dice.\n"
            << "\n\t\t4. If there’s six on dice the player gets another
turn to roll dice.\n"
            << "\n\t\t5. If two or more players are at the same point
index then all goes back to the initial state\n"
            << "\t\texcept the latest one.\n"
            << "\t\tThe first player to reach home will be the winner.\n"
            << "\n\t\t6. The first winner can give six moves forward to
any other player as a gift.\n\n\n";
      system("pause");
      system("cls");
      menu();
}


/*This Function Will Display Credits*/


void credits()
{
      system("cls");
      cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n" << setfill(' ') << setw(82) <<
"Credits\n\n" << setfill(' ') << setw(85) << "MOGHEES AHMAD\t20F-0244\n"
            << setfill(' ') << setw(87) << "ANUS ABDULLAH\t20F-
0267\n\n\n";
```

```
	system("pause");
	system("cls");
	menu();
}

/*This Function will save records*/

void writeRecord(string name, int num)
{
	ofstream out;
	out.open("RECORD.txt");
	out << name << " wins the game.After " << num << " turns." << endl
<< endl;
	out.close();
}

/*This Function will Run The Game*/

void playgame()
{
	names();
	system("pause");
	score();
}

/*This function will get names of players*/

void names()
{
	system("cls");
	cout << "\t\t\tWelcome To Sanke And Ladder Game\n\nPlease Enter Your
Names Here:\n\n";
	for (int a = 1, b = 0; a <= 4; a++, b++)
```

```
      {
            cout << "Player " << a << ": ";
            cin >> name[b];
            cout << endl;
      }
      cout << endl;
}



/*This Function Will Calculate and Display scores of Players*/


void score()
{
      int  turn[4];
                                                      //Array Declaration
      int dice, x, c1 = 0, c2 = 0, c3 = 0, c4 = 0, count = 0, moves = 0;
                         //Variables Declaration


      system("cls");
      cout << "\n\n\n\n\t   TURNS\n\n";
      for (int t = 1, player, c = 0; c < 4; c++, t++)
            //Toss
      {
            turn[c] = rand() % 4 + 1;
            player = turn[c] - 1;
            cout << "\tTurn " << t << " = " << name[player] << endl;
      }
      cout << endl;
      system("pause"); system("cls"); system("color 07");


      srand(time(NULL));
```

```
    while (c1 != 3 && c2 != 3 && c3 != 3 && c4 != 3)
    //Dice rolling and Scores Calculation
    {
        Table();
        if (count == 0)
            moves++;
        for (int e = 0; e < 4; e++)
        {
            dice = rand() % 6 + 1;
            x = rand() % 6 + 1;
    //In case player gets 6 this dice will roll
            switch (turn[e])
            {
                        /*Score Calculation For Player 1*/
            case 1:

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                cout << "Player 1 :\n ";
                if (score1 >= 100)
                {
                    if (c1 == 0)
                        cout << name[0] << " finishes the game
with " << position(c1) << " position.\n\n";
                    if (c1 == 1 && count == 0)
                    {
                        cout << name[0] << " Is Winner after "
<< moves << " moves\n\n";
                        writeRecord(name[0], moves);
                        reward(score1, score2, score3,
score4);
                        count++;
                    }
                }
                else
```

```
                        {
                              cout << "\nDice = " << dice;
                              if (dice == 6 || score1 >= 6)
                                    score1 += dice;
                              if (dice == 6)
                              {
                                    cout << " + " << x; score1 += x;
                              }
                              snake(score1);
                              ladder(score1);
                              if (score1 <= 100)
                                    cout << endl << name[0] << " = " <<
score1 << endl;
                        }

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        break;
                                    /*Score Calculation For Player 2*/
                  case 2:

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                        cout << "Player 2 :\n ";
                        if (score2 >= 100)
                        {
                              if (c2 == 0)
                                    cout << name[1] << " finishes the game
with " << position(c2) << " position.\n\n";
                              if (c2 == 1 && count == 0)
                              {
                                    cout << name[1] << " Is Winner after "
<< moves << " moves\n\n";
                                    writeRecord(name[1], moves);
                                    reward(score1, score2, score3,
score4);
```

```
                        count++;
                    }
                }
                else
                {
                    cout << "\nDice = " << dice;
                    if (dice == 6 || score2 >= 6)
                        score2 += dice;
                    if (dice == 6)
                    {
                        cout << " + " << x; score2 += x;
                    }
                    snake(score2);
                    ladder(score2);
                    if (score2 <= 100)
                        cout << endl << name[1] << " = " <<
score2 << endl;
                }

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                break;
                        /*Score Calculation For Player 3*/
            case 3:

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                cout << "Player 3 :\n ";
                if (score3 >= 100)
                {
                    if (c3 == 0)
                        cout << name[2] << " finishes the game
with " << position(c3) << " position.\n\n";
                    if (c3 == 1 && count == 0)
                    {
```

```
                                    cout << name[2] << " Is Winner after "
<< moves << " moves\n\n";
                                    writeRecord(name[2], moves);
                                    reward(score1, score2, score3,
score4);
                                    count++;
                              }
                        }
                        else
                        {
                              cout << "\nDice = " << dice;
                              if (dice == 6 || score3 >= 6)
                                    score3 += dice;
                              if (dice == 6)
                              {
                                    cout << " + " << x; score3 += x;
                              }
                              snake(score3);
                              ladder(score3);
                              if (score3 <= 100)
                                    cout << endl << name[2] << " = " <<
score3 << endl;
                        }

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        break;
                                    /*Score Calculation For Player 4*/
                  case 4:

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                        cout << "Player 4 :\n ";
                        if (score4 >= 100)
                        {
                              if (c4 == 0)
                        cout << name[3] << " finishes the game
with " << position(c4) << " position.\n\n";
                    if (c4 == 1 && count == 0)
                    {
                        cout << name[3] << " Is Winner after "
<< moves << " moves\n\n";
                        writeRecord(name[3], moves);
                        reward(score1, score2, score3,
score4);
                        count++;
                    }
                }
                else
                {
                    cout << "\nDice = " << dice;
                    if (dice == 6 || score4 >= 6)
                        score4 += dice;
                    if (dice == 6)
                    {
                        cout << " + " << x; score4 += x;
                    }
                    snake(score4);
                    ladder(score4);
                    if (score4 <= 100)
                        cout << endl << name[3] << " = " <<
score4 << endl;
                }

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                break;
            }
            system("pause");
            index_check(score1, score2, score3, score4);
        }
        system("cls");
    }
    menu();
}

/*This Function will display the table and the positions of players*/

void Table()
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
    int count = 100, x = 0;                   // variables decleration
    int Array[10][10];                               // Array decleration
    cout << "\t\t\t\tSNAKE AND LADDER GAME.\n\n";
    for (int i = 0; i < 10; ++i)                                 // loop
to store and print numbers
    {
        if (x == 0)
        {
            for (int j = 0; j < 10; j++)
            {
                if (count == 100)
                {

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
            //Changing colors of text
                    cout << "\tHOME\t";
                    count--;
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                }
                else if (count >= 91)
       // To print S for Snakes and L for laders
                {
                    if (count == 99) {
```

```
      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
                                    cout << "S\t";
                                    count--; j++;

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              else if (count == 97)
                              {
                                    if (count == score1 || count == score2
|| count == score3 || count == score4)
                                    {
                                          /*This Will Display Player's
Position on Table*/
                                          if (count == score1)
                                          {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                                                cout << "P1\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                          }
                                          if (count == score2)
                                          {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                                                cout << "P2\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                          }
                                          if (count == score3)
                                          {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                                                cout << "P3\t";
      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                          }
                                          if (count == score4)
                                          {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                                                cout << "P4\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                          }
                                    }
                                    else
                                    {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 2);
                                          Array[i][j] = count;
                                                      // storing numbers in
array
                                          cout << Array[i][j] << "\t";
                                                //Displaying numbers

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                    }
                                    count--; j++;
                              }
                              Array[i][j] = count;
                              if (count == score1 || count == score2 ||
count == score3 || count == score4)
                              {
                                    if (count == score1)
                                    {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                                          cout << "P1\t";
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              if (count == score2)
                              {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                                    cout << "P2\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              if (count == score3)
                              {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                                    cout << "P3\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              if (count == score4)
                              {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                                    cout << "P4\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                        }
                        else
                              cout << Array[i][j] << "\t";
                        count--;
                  }
            }
            i++;
```

```
        cout << "\n\t\n\t";
    }
    if (x % 2 == 0)
    {
        count = count - 9;
        for (int k = 0; k < 10; k++)
        {
            if (count == 1)
            {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                cout << "START\t";
                count++; k++;

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
            }
            else if (count == 69)
            {
                Array[i][k] = count;

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
                cout << "S\t"; count++; k++;

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
            }
            else if (count == 30)
            {
                Array[i][k] = count++;

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
                cout << "S\t"; k++;

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7); break;
            }
```

```
                        else if (count == 66 || count == 43 || count == 6)
// Printing L for Ladders
                        {
                              Array[i][k] = count;

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 2);
                              cout << "L\t"; count++; k++;

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                        else if (count == 48 || count == 45 || count == 25
|| count == 8)
                        {
                              Array[i][k] = count;
                              if (count == score1 || count == score2 ||
count == score3 || count == score4)
                              {
                                    if (count == score1)
                                    {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                                          cout << "P1\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                    }
                                    if (count == score2)
                                    {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                                          cout << "P2\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                    }
                                    if (count == score3)
                                    {
```

```

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                                          cout << "P3\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                    }
                                    if (count == score4)
                                    {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                                          cout << "P4\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                    }
                              }
                              else
                              {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
                                    cout << Array[i][k] << "\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              count++; k++;
                        }
                        else if (count == 87 || count == 62 || count ==
27)
                        {
                              Array[i][k] = count;
                              if (count == score1 || count == score2 ||
count == score3 || count == score4)
                              {
                                    if (count == score1)
                                    {
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                                     cout << "P1\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                               }
                               if (count == score2)
                               {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                                     cout << "P2\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                               }
                               if (count == score3)
                               {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                                     cout << "P3\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                               }
                               if (count == score4)
                               {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                                     cout << "P4\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                               }
                         }
                         else
                         {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 2);
```

```
                                    cout << Array[i][k] << "\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              count++; k++;
                        }
                        Array[i][k] = count;
                        if (count == score1 || count == score2 || count ==
score3 || count == score4)
                        {
                              if (count == score1)
                              {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                                    cout << "P1\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              if (count == score2)
                              {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                                    cout << "P2\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              if (count == score3)
                              {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                                    cout << "P3\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              if (count == score4)
```

```
                    {

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                        cout << "P4\t";

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                    }
                }
                else
                    cout << Array[i][k] << "\t";
                count++;
            }
        }
        else if (x % 2 != 0)
        {
            count = count - 11;
            for (int y = 0; y < 10; y++)
            {
                if (count == 54)
                {
                    Array[i][y] = count;
                    if (count == score1 || count == score2 ||
count == score3 || count == score4)
                    {
                        if (count == score1)
                        {

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                            cout << "P1\t";

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                        if (count == score2)
                        {
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                                    cout << "P2\t";
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              if (count == score3)
                              {
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                                    cout << "P3\t";
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                              if (count == score4)
                              {
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                                    cout << "P4\t";
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                              }
                        }
                        else
                        {
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 2);
                              cout << Array[i][y] << "\t";
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                        count--; y++;
                  }
                  else if (count == 74 || count == 57)
                  {
```

```
                              Array[i][y] = count;

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
                              cout << "S\t"; count--; y++;

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                        else if (count == 76 || count == 35)
                        {
                              Array[i][y] = count;

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 2);
                              cout << "L\t"; count--; y++;

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                        else if (count == 78)
                        {
                              Array[i][y] = count;
                              if (count == score1 || count == score2 ||
count == score3 || count == score4)
                              {
                                    if (count == score1)
                                    {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                                          cout << "P1\t";

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                                    }
                                    if (count == score2)
                                    {

      SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                                          cout << "P2\t";
```

```
   SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                  }
                  if (count == score3)
                  {

   SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                     cout << "P3\t";

   SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                  }
                  if (count == score4)
                  {

   SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                     cout << "P4\t";

   SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                  }
               }
               else
               {

   SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
                  cout << Array[i][y] << "\t";

   SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
               }
               count--; y++;
            }
            Array[i][y] = count;
            if (count == score1 || count == score2 || count ==
score3 || count == score4)
            {
               if (count == score1)
```

```
                        {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 3);
                              cout << "P1\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                        if (count == score2)
                        {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 5);
                              cout << "P2\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                        if (count == score3)
                        {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
                              cout << "P3\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                        if (count == score4)
                        {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 1);
                              cout << "P4\t";

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
                        }
                  }
                  else
                        cout << Array[i][y] << "\t";
                  count--;
```

```
            }
        }
        ++x;
        cout << "\n\t\n\t";
    }
    cout << "\t" << setfill('_') << setw(60) << " " << "\n\n";

            /*Positions of Sankes And Ladders*/

    cout << "\n\t\tSNAKES\t\t\t\t LADDERS\n"
        << "\t\t-> 99 to 78\t\t\t-> 76 to 97\n" << "\t\t-> 74 to
45\t\t\t-> 66 to 87\n"
        << "\t\t-> 69 to 48\t\t\t-> 43 to 62\n" << "\t\t-> 57 to
25\t\t\t-> 35 to 54\n"
        << "\t\t-> 30 to 8 \t\t\t-> 6  to 27\n";
    cout << "\n\t\t\tSCORE BOARD\n\n";
}

/*This Function will send players to START if two or more players lie on
same index.Only one player will stay there.*/

void index_check(int& score1, int& score2, int& score3, int& score4)
{
    if (score1 == score2 && score1 < 100 && score2 < 100)
        score2 = 0;
    else if (score1 == score3 && score1 < 100 && score3 < 100)
        score3 = 0;
    else if (score1 == score4 && score1 < 100 && score4 < 100)
        score4 = 0;
    else if (score2 == score1 && score1 < 100 && score2 < 100)
        score1 = 0;
    else if (score2 == score3 && score2 < 100 && score3 < 100)
        score3 = 0;
```

```
      else if (score2 == score4 && score4 < 100 && score2 < 100)
            score4 = 0;
      else if (score3 == score1 && score1 < 100 && score3 < 100)
            score1 = 0;
      else if (score3 == score2 && score3 < 100 && score2 < 100)
            score2 = 0;
      else if (score3 == score4 && score3 < 100 && score4 < 100)
            score4 = 0;
      else if (score4 == score1 && score1 < 100 && score4 < 100)
            score1 = 0;
      else if (score4 == score2 && score2 < 100 && score4 < 100)
            score2 = 0;
      else if (score4 == score3 && score3 < 100 && score4 < 100)
            score3 = 0;
}

/*This Function will tell about the positions of players.*/

int position(int& x)
{
      static int y = 0;
      y++;
      x = y;
      return x;
}

      /*This function is to gift 6 moves*/

int reward(int& score1, int& score2, int& score3, int& score4)
{
      int reward, receiver, giftscores = 0;
                  //Variable Declaration
```

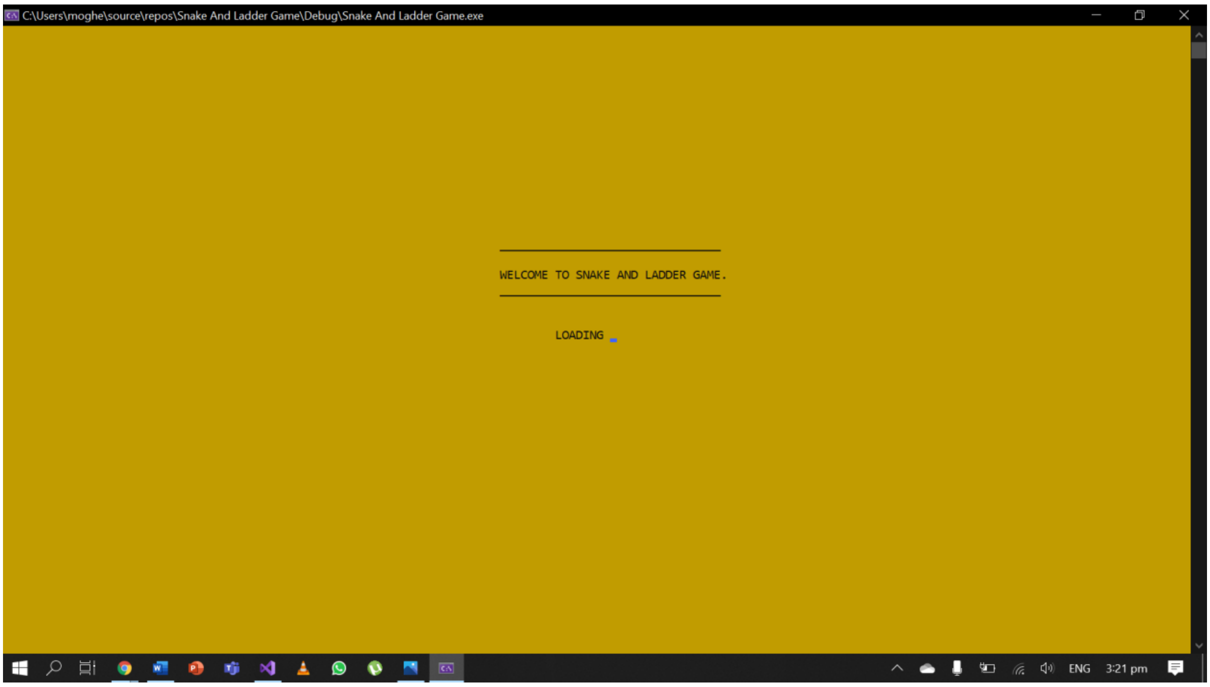
```
        cout << "\nWould You Like To Award 6 Moves To Any Other Player? (1 =
Yes/0 = No)\n";
        cin >> reward;
        if (reward == 0)
              cout << "\nNo Gift Given By The Winner!\n";
        if (reward == 1)
        {
              cout << "\nTo Which Player You Want To Gift 6 Moves?\n";
              cin >> receiver;
                          /*Adding score to the player winner chose*/
              switch (receiver)
              {
              case 1:
                    cout << " 6 Scores Have Been Awarded To Player 1 !\n";
                    score1 += 6;
                    return score1; break;
              case 2:
                    cout << " 6 Scores Have Been Awarded To Player 2 !\n";
                    score2 += 6;
                    return score2; break;
              case 3:
                    cout << " 6 Scores Have Been Awarded To Player 3 !\n";
                    score3 += 6;
                    return score3; break;
              case 4:
                    cout << " 6 Scores Have Been Awarded To Player 4 !\n";
                    score4 += 6;
                    return score4; break;
              }
        }
}
```

```
/*This function will check if the player gets ladder.*/

int ladder(int& score)
{
      if (score == 6 || score == 35 || score == 43 || score == 66 || score
== 76)
      {
            system("color 27");
            cout << "\nWOW!You Got A Ladder.\n";
            Sleep(1000); system("color 07");
      }
      (score == 6) ? score = 27 : (score == 35) ? score = 54 : (score ==
43) ? score = 62 :                  //Updating scores after getting Ladder
            (score == 66) ? score = 87 : (score == 76) ? score = 97 :
score = score;
      return score;
}

/*This function will check if the player gets snake.*/

int snake(int& score)
{
      if (score == 30 || score == 57 || score == 69 || score == 74 ||
score == 99)
      {
            system("color 47");
            cout << "\nOuch!You Got Bit By Snake.\n";
            Sleep(1000); system("color 07");
      }
      (score == 30) ? score = 8 : (score == 57) ? score = 25 : (score ==
69) ? score = 48 :                              //Updating scores after
getting Snake
            (score == 74) ? score = 45 : (score == 99) ? score = 78 :
score = score;
```

```
        return score;
}
```

Outputs:

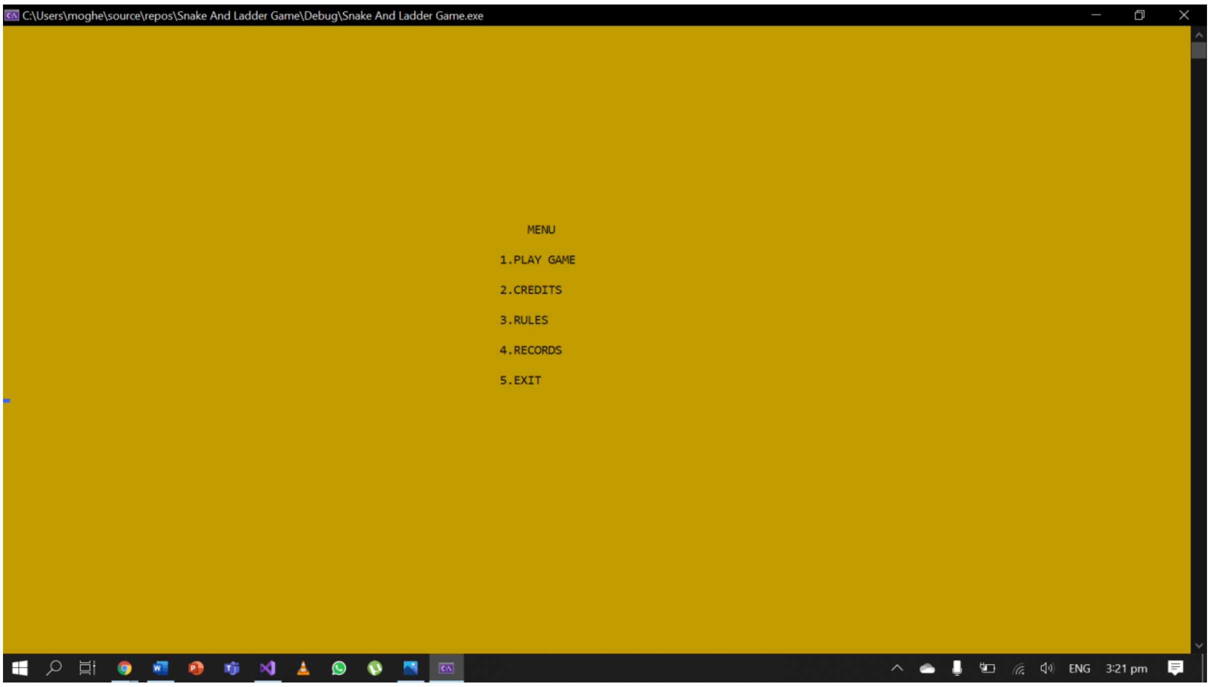
Welcome To Snake And Ladder Game !



C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe
ENG
3:21 pm

```
WELCOME TO SNAKE AND LADDER GAME.

        LOADING _
```

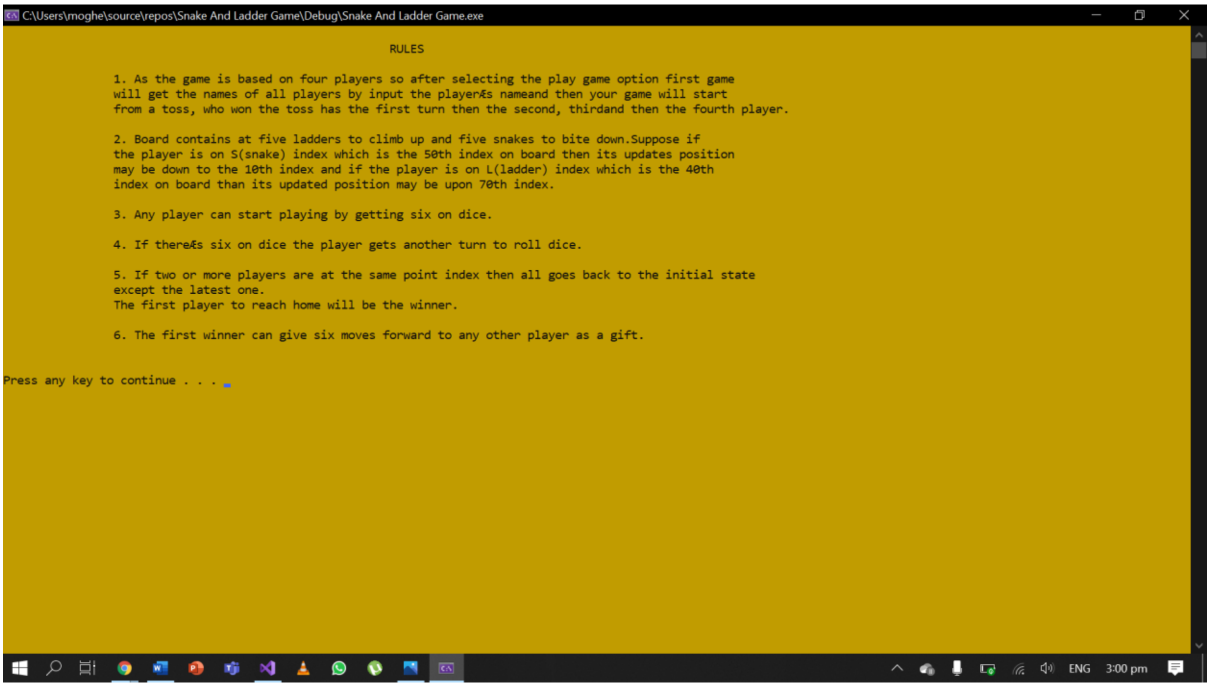
Choose Any Option



C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe
ENG
3:21 pm

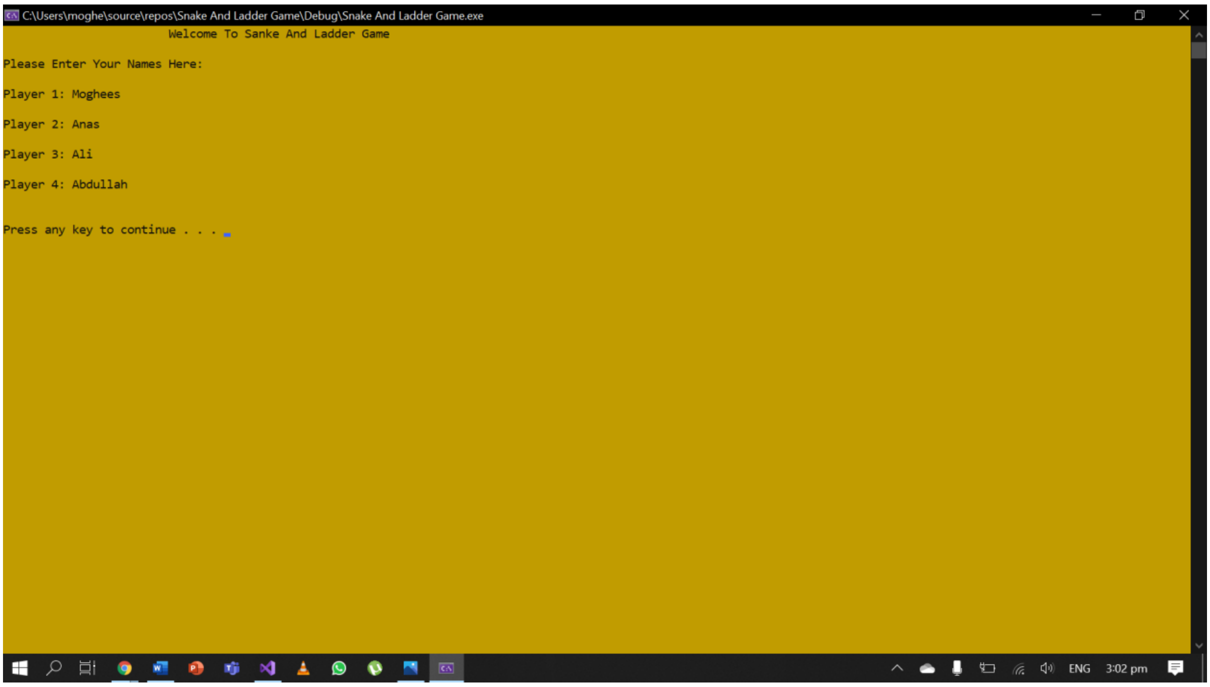
```
    MENU
1.PLAY GAME
2.CREDITS
3.RULES
4.RECORDS
5.EXIT
```

These Are The Rules Of Game



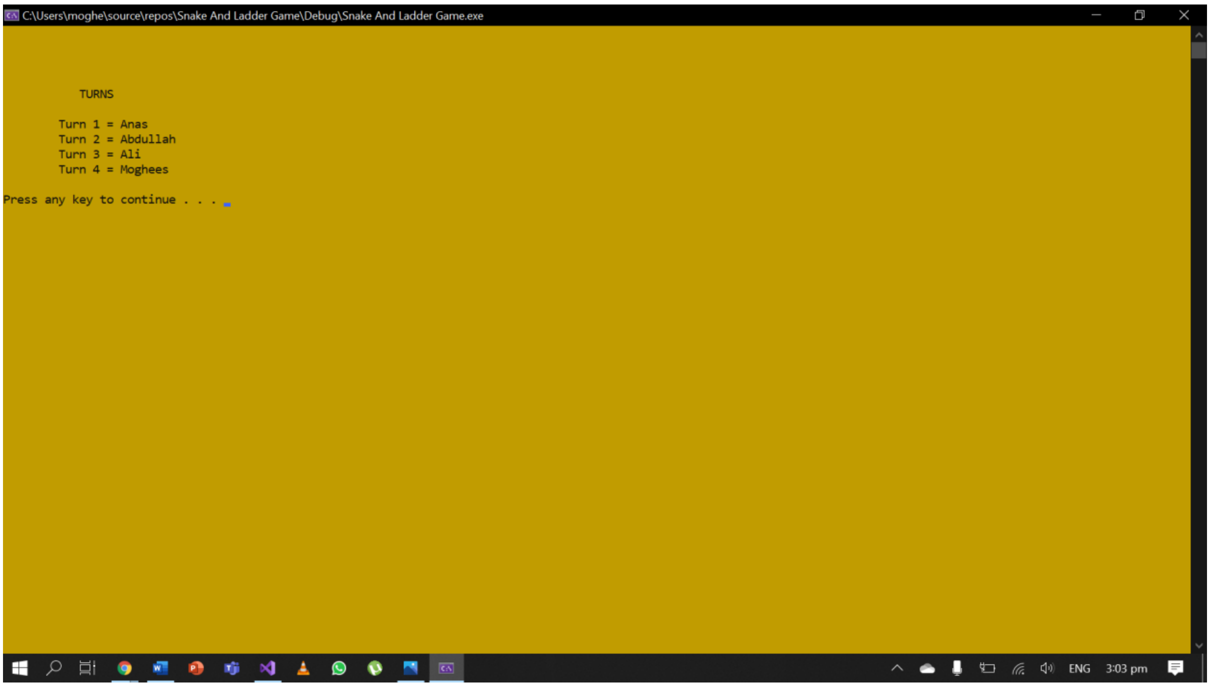
C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe

```
                                                       RULES

               1. As the game is based on four players so after selecting the play game option first game
               will get the names of all players by input the playerÆs nameand then your game will start
               from a toss, who won the toss has the first turn then the second, thirdand then the fourth player.

               2. Board contains at five ladders to climb up and five snakes to bite down.Suppose if
               the player is on S(snake) index which is the 50th index on board then its updates position
               may be down to the 10th index and if the player is on L(ladder) index which is the 40th
               index on board than its updated position may be upon 70th index.

               3. Any player can start playing by getting six on dice.

               4. If thereÆs six on dice the player gets another turn to roll dice.

               5. If two or more players are at the same point index then all goes back to the initial state
               except the latest one.
               The first player to reach home will be the winner.

               6. The first winner can give six moves forward to any other player as a gift.


Press any key to continue . . .
```



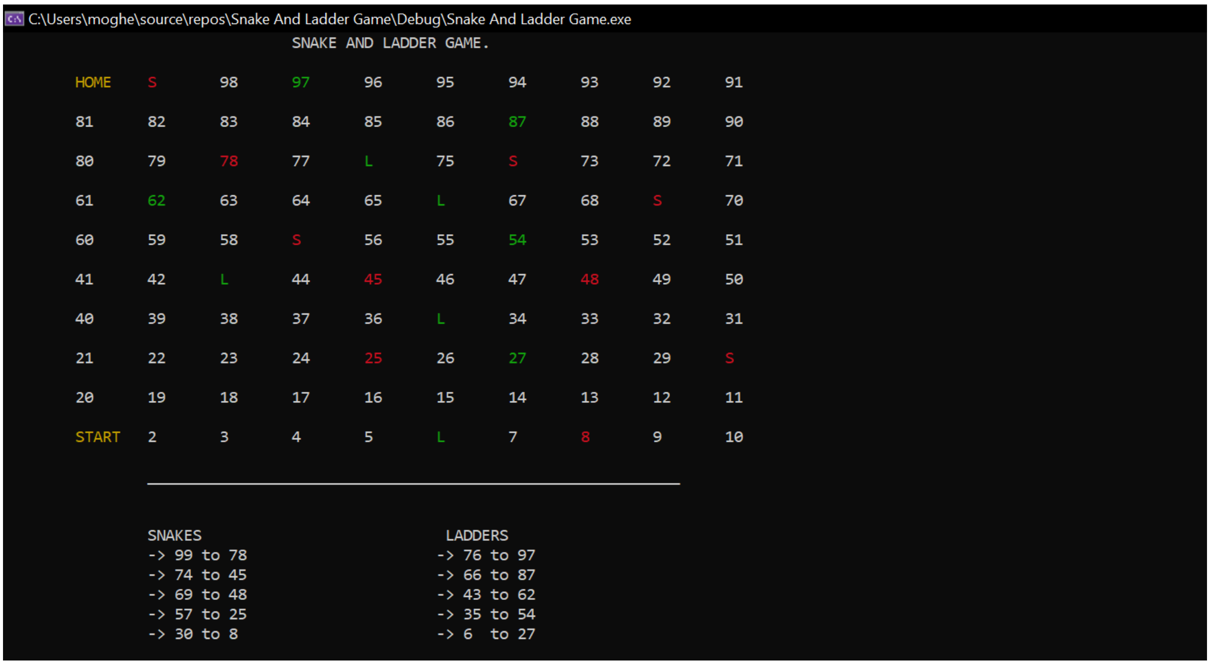
C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe

```
                        Welcome To Sanke And Ladder Game

Please Enter Your Names Here:

Player 1: Moghees

Player 2: Anas

Player 3: Ali

Player 4: Abdullah


Press any key to continue . . .
```


C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe
ENG
3:03 pm

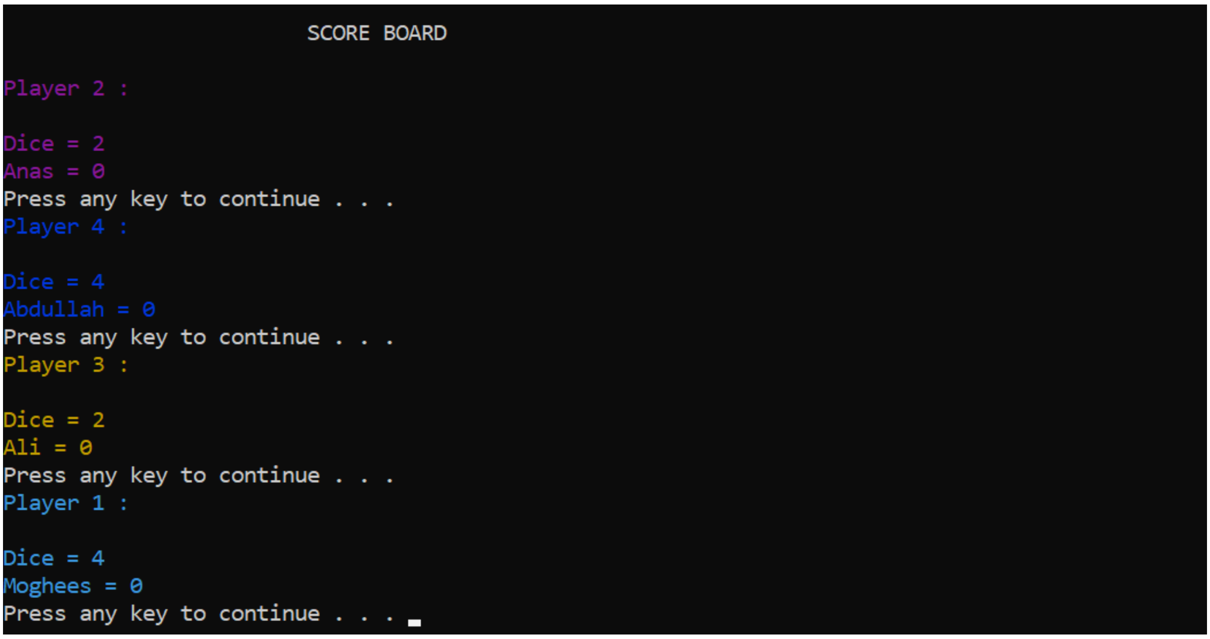
```
           TURNS

        Turn 1 = Anas
        Turn 2 = Abdullah
        Turn 3 = Ali
        Turn 4 = Moghees

Press any key to continue . . .
```

Let's Start The Game


C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe

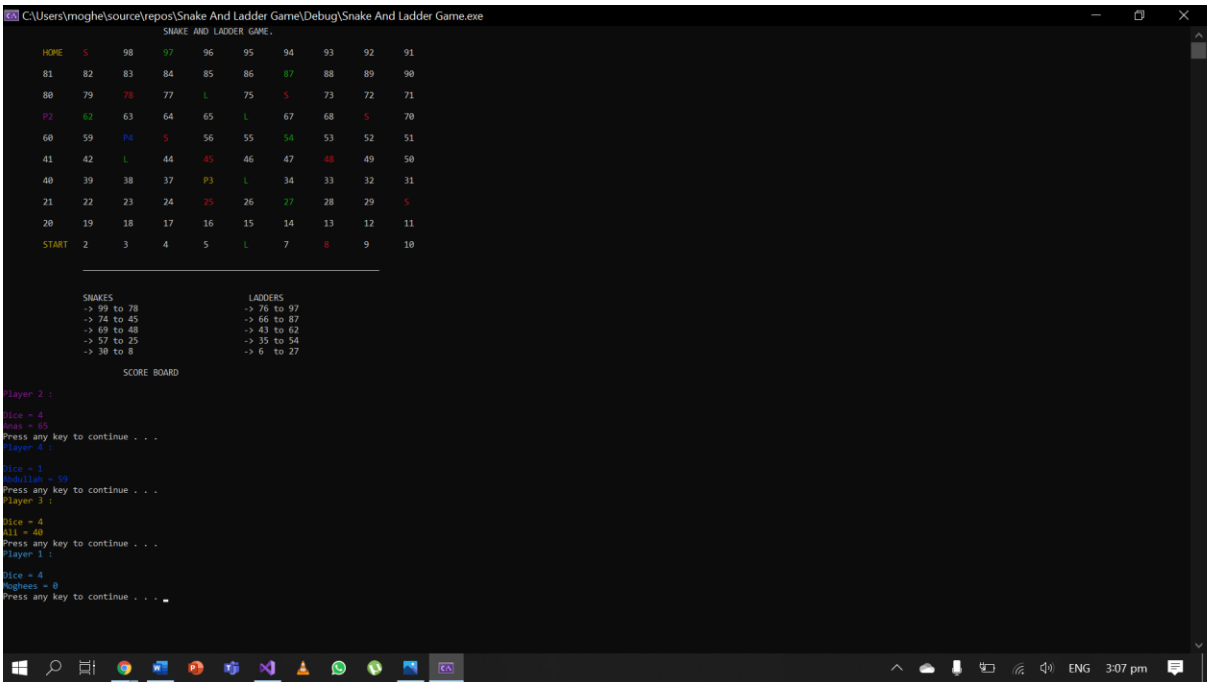
```
                                SNAKE AND LADDER GAME.

        HOME     S        98       97       96       95       94       93       92       91

        81       82       83       84       85       86       87       88       89       90

        80       79       78       77       L        75       S        73       72       71

        61       62       63       64       65       L        67       68       S        70

        60       59       58       S        56       55       54       53       52       51

        41       42       L        44       45       46       47       48       49       50

        40       39       38       37       36       L        34       33       32       31

        21       22       23       24       25       26       27       28       29       S

        20       19       18       17       16       15       14       13       12       11

        START    2        3        4        5        L        7        8        9        10

                ___________________________________________________________


                SNAKES                            LADDERS
                -> 99 to 78                      -> 76 to 97
                -> 74 to 45                      -> 66 to 87
                -> 69 to 48                      -> 43 to 62
                -> 57 to 25                      -> 35 to 54
                -> 30 to 8                       -> 6  to 27
```

Here Are Your Scores



```
                        SCORE  BOARD

Player 2 :

Dice = 2
Anas = 0
Press any key to continue . . .
Player 4 :

Dice = 4
Abdullah = 0
Press any key to continue . . .
Player 3 :

Dice = 2
Ali = 0
Press any key to continue . . .
Player 1 :

Dice = 4
Moghees = 0
Press any key to continue . . .
```

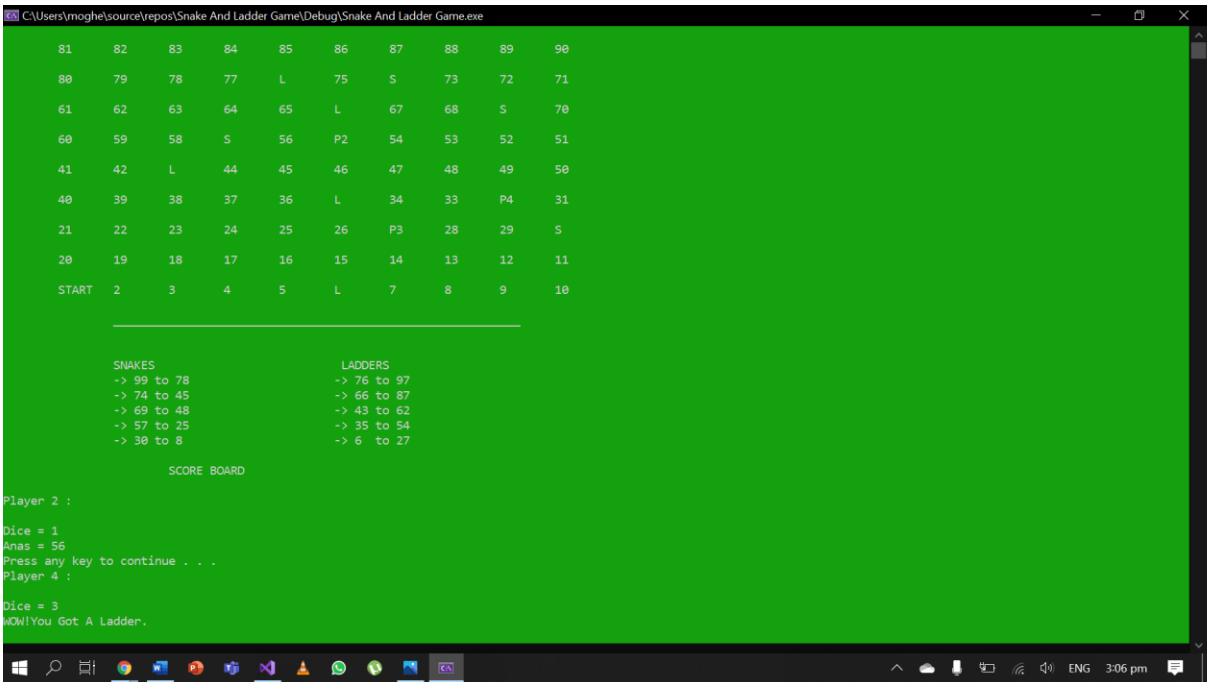
Your Scores Are Displayed On Table By P1,P2,P3 and P4



C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe

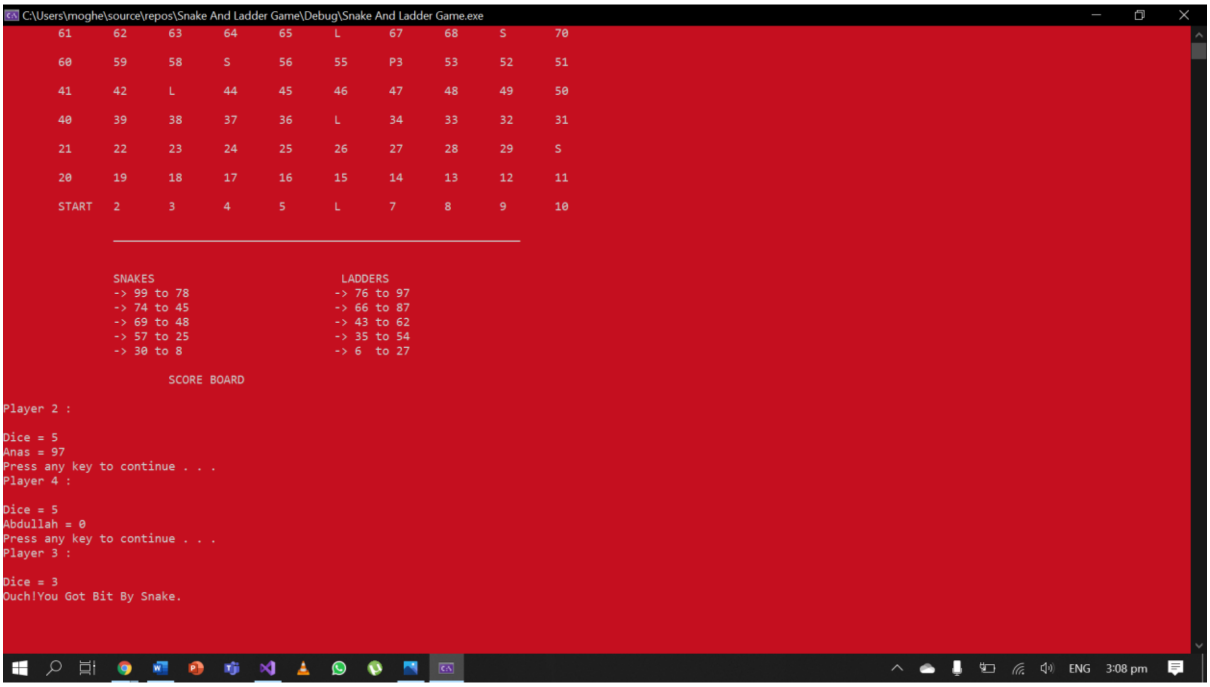
```
                         SNAKE  AND  LADDER  GAME.

      HOME    S       98      97      96      95      94      93      92      91

      81      82      83      84      85      86      87      88      89      90

      80      79      78      77      L       75      S       73      72      71

      P2      62      63      64      65      L       67      68      S       70

      60      59      P4      S       56      55      54      53      52      51

      41      42      L       44      45      46      47      48      49      50

      40      39      38      37      P3      L       34      33      32      31

      21      22      23      24      25      26      27      28      29      S

      20      19      18      17      16      15      14      13      12      11

      START   2       3       4       5       L       7       8       9       10

              ___________________________________________________


              SNAKES                          LADDERS
              -> 99 to 78                    -> 76 to 97
              -> 74 to 45                    -> 66 to 87
              -> 69 to 48                    -> 43 to 62
              -> 57 to 25                    -> 35 to 54
              -> 30 to 8                     -> 6  to 27

                    SCORE  BOARD

Player 2 :

Dice = 4
Anas = 65
Press any key to continue . . .
Player 4 :

Dice = 1
Abdullah = 59
Press any key to continue . . .
Player 3 :

Dice = 4
Ali = 40
Press any key to continue . . .
Player 1 :

Dice = 4
Moghees = 0
Press any key to continue . . .
```

Woooow!!! You Got A Ladder


C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe
ENG
3:06 pm

```
        81      82      83      84      85      86      87      88      89      90

        80      79      78      77      L       75      S       73      72      71

        61      62      63      64      65      L       67      68      S       70

        60      59      58      S       56      P2      54      53      52      51

        41      42      L       44      45      46      47      48      49      50

        40      39      38      37      36      L       34      33      P4      31

        21      22      23      24      25      26      P3      28      29      S

        20      19      18      17      16      15      14      13      12      11

        START   2       3       4       5       L       7       8       9       10

                ___________________________________________________________


                SNAKES                          LADDERS
                -> 99 to 78                    -> 76 to 97
                -> 74 to 45                    -> 66 to 87
                -> 69 to 48                    -> 43 to 62
                -> 57 to 25                    -> 35 to 54
                -> 30 to 8                     -> 6  to 27

                        SCORE BOARD

Player 2 :

Dice = 1
Anas = 56
Press any key to continue . . .
Player 4 :

Dice = 3
WOW!You Got A Ladder.
```

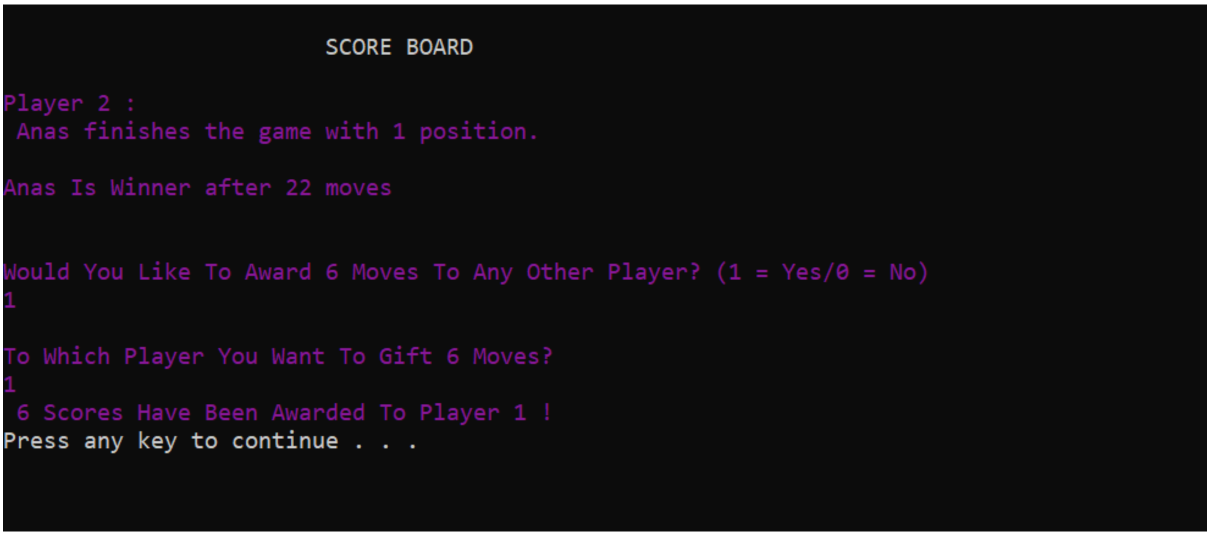
Ooohhh!!! You Are Bitten By Snake


C:\Users\moghe\source\repos\Snake And Ladder Game\Debug\Snake And Ladder Game.exe
ENG
3:08 pm

```
        61      62      63      64      65      L       67      68      S       70

        60      59      58      S       56      55      P3      53      52      51

        41      42      L       44      45      46      47      48      49      50

        40      39      38      37      36      L       34      33      32      31

        21      22      23      24      25      26      27      28      29      S

        20      19      18      17      16      15      14      13      12      11

        START   2       3       4       5       L       7       8       9       10

                ___________________________________________________________


                SNAKES                          LADDERS
                -> 99 to 78                    -> 76 to 97
                -> 74 to 45                    -> 66 to 87
                -> 69 to 48                    -> 43 to 62
                -> 57 to 25                    -> 35 to 54
                -> 30 to 8                     -> 6  to 27

                        SCORE BOARD

Player 2 :

Dice = 5
Anas = 97
Press any key to continue . . .
Player 4 :

Dice = 5
Abdullah = 0
Press any key to continue . . .
Player 3 :

Dice = 3
Ouch!You Got Bit By Snake.
```

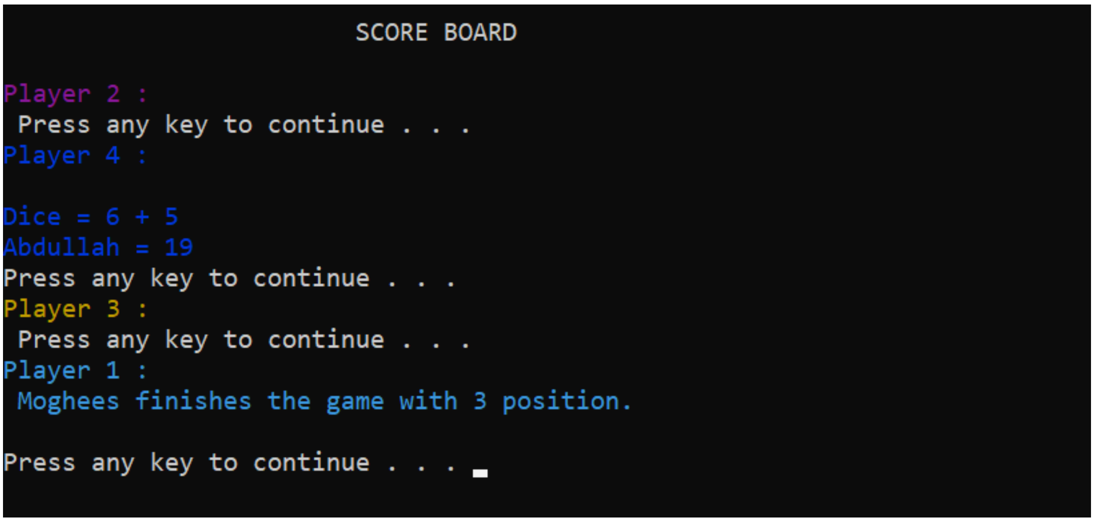
And Here Is Your Winner !

So, Mr. Winner! Do You Want To Gift Moves ???



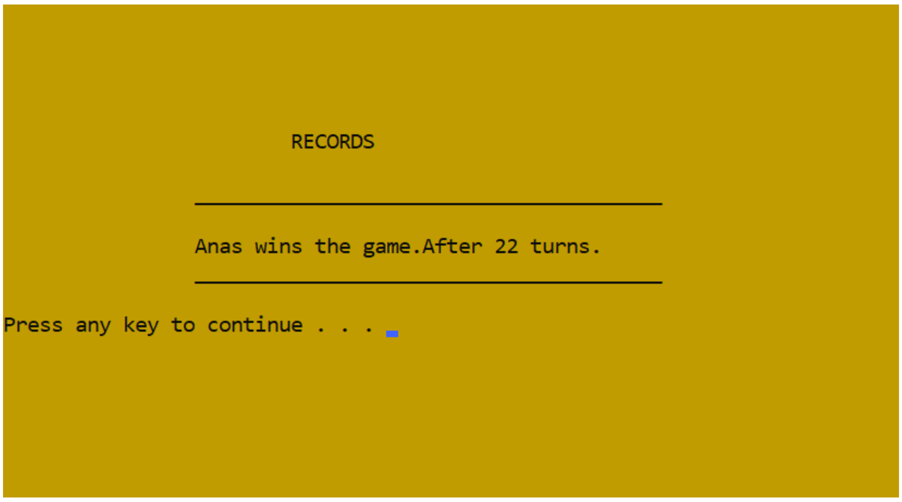
```
                         SCORE BOARD

Player 2 :
 Anas finishes the game with 1 position.

Anas Is Winner after 22 moves


Would You Like To Award 6 Moves To Any Other Player? (1 = Yes/0 = No)
1

To Which Player You Want To Gift 6 Moves?
1
 6 Scores Have Been Awarded To Player 1 !
Press any key to continue . . .
```

Here Is Our Third Winner



```
                        SCORE BOARD

Player 2 :
 Press any key to continue . . .
Player 4 :

Dice = 6 + 5
Abdullah = 19
Press any key to continue . . .
Player 3 :
 Press any key to continue . . .
Player 1 :
 Moghees finishes the game with 3 position.

Press any key to continue . . .
```

Mr. Winner ! Your Name Is Written In Our History!!



```
                    RECORDS

             ___________________________________

             Anas wins the game.After 22 turns.
             ___________________________________

Press any key to continue . . .
```