INTERACTIVE SNAKE GAME USING LINKED LIST

Project Report

Submitted for the course: Data Structures and Algorithms (CSE2003)

Submitted by:

Shubham Kumar (18BCE0356)

Niket Jain (18BCE0337)

Ayush Shinha (18BCE0299)

To the Professor:

Dr. Shalini N

Under Slot: G2

INDEX

ABSTRACT

Snakes and ladders is an ancient Indian board game regarded today as a worldwide classic. It is played between two or more players on a game board having numbered, gridded squares. A number of ladders and snakes are pictured on the board, each connecting two specific board squares. The objective of the game is to navigate one's game piece according to die rolls from start to the finish, helped or hindered by ladders or snakes respectively The game is a simple race contest based on sheer luck, probability and is really popular among young children. The following project illustrates the beautiful game of snakes and ladders in the form of an interactive computer program. This project has been made keeping in mind that the modern society has been strongly influenced by the presence of computers.

Thus, in order to take advantage of the capability of a computer, our goal is to make a two player game where both the players will be from the real world and no virtual player will be playing. Position of the ladders and snakes would be displayed according to eh level of difficulty selected by the user. Also, we have added some colours to the main program which will differentiate player 1 from player 2. And other colours are also added depending upon the situation this game creates. Hence this makes it really attractive and interactive game to be played.

Basically, this games work on a principle of linked list, a very effective data structure that uses pointers concept. All the boxes in the grid are actually connected to next box using a pointer and a die is rolled using the basic random integer function available in C++. This function creates a random integer number whose value is between zero to six. According to that given die number, the player moves ahead by the number obtained and thereafter the adventure begins. Now, any snake or ladder can be encountered by the head of the player and depending upon the case it is, suitable actions would be taken upon the player encountering the case. This continues until any player reaches the maximum number that is 100.

This way, we were able to successfully achieve this goal and thus successfully completed out project.

Details About the Related Work

To implement the game, we made use of the linked list and pointers as it helped us in locating positions and connecting ladders. Snakes have also been made using linked list by pointing the coordinates using a linked list. Also the position was easily detected with the help of the above mentioned data structure. The structure plays a crucial role in the program as the linked list would use the data types declared in the structure to carry out operations i.e. moving the pointer of the player to the next box.

We further mind stormed upon the different functions and the main function which was to be used in our programme. For example, there is a separate function used to generate random numbers which are within the range of 1 to 6, which acts like a dice. Finally, the main() function is where all the functions are called and executed. With further clarity of the functions and after weighing our abilities we distributed the work among the three group members. We further mind stormed upon the different functions and the main function which was to be used in our program. With further clarity of the functions and after weighing our abilities we distributed the work among the four group members.

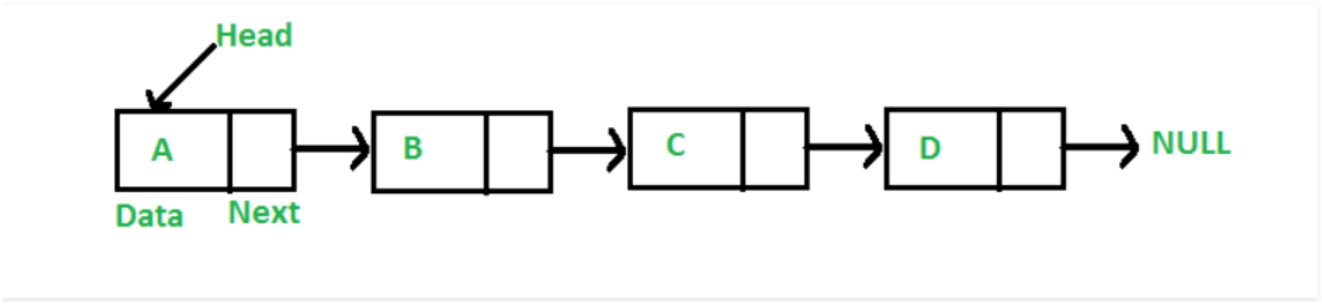
By review 3 we were done with all the codes and were further working on how to improve our codes.

The code was then developed and errors removed and further enhancements made. Finally, we ended up with this project of ours and made completed the final report.

METHODOLOGY

LINKED LIST:

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using
pointers as shown in the below image:



In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

Why Linked List?

Arrays can be used to store linear data of similar types, but arrays have following limitations.

1) The size of the arrays is fixed: So we must know the upper limit on the number of elements in advance. Also, generally, the allocated memory is equal to the upper limit irrespective of the usage.

2) Inserting a new element in an array of elements is expensive, because room has to be created for the new elements and to create room existing elements have to shifted.

For example, in a system if we maintain a sorted list of IDs in an array id[].

id[] = [1000, 1010, 1050, 2000, 2040].

Drawbacks:

1) Random access is not allowed. We have to access elements sequentially starting from the first node. So we cannot do binary search with linked lists efficiently with its default implementation.

2) Extra memory space for a pointer is required with each element of the list.

3) Not cache friendly. Since array elements are contiguous locations, there is locality of reference which is not there in case of linked lists.

Representation:

A linked list is represented by a pointer to the first node of the linked list. The first node is called head. If the linked list is empty, then value of head is NULL.
Each node in a list consists of at least two parts:
1) data

2) Pointer (Or Reference) to the next node
In C, we can represent a node using structures. Below is an example of a linked list node with an integer data.
In Java, LinkedList can be represented as a class and a Node as a separate class. The LinkedList class contains a reference of Node class type.

MOTIVATION OF THE PROJECT

We felt that this project would enlighten us with the many applications of data structures. We did not have any complicated ideas about the project. It was mutually felt that this project is not only about getting marks, it is also about gaining the knowledge required to implement data structures such as linked lists, with the help of pointers and structures.

We further mind stormed upon the different functions and the main function which was to be used in our programme. With further clarity of the functions and after weighing our abilities we distributed the work among the four group members. Hence, through the execution of a simple project, such as building the snake and ladder game, our team members have made the very best attempt of understanding the code presented in the document.

Although making a game is quite a common idea, the three of us felt that bringing a traditional game such as snakes and ladders to the computer could double the joy that we would acquire while playing this game. As mentioned earlier, we tried to make it interesting since the players won’t get to know what will have happened the next move and this would create a different level of excitement among them making this game a super hit.

C++ CODE

```
#include<iostream>

#include<stdlib.h>

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<dos.h>

#include<windows.h>

#include<time.h>

using namespace std;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

struct node

{

int data;

node *next;

node *snake;

node *ladder;

}*p,*p1,*p2,*head,*tail;



////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void initialize()
```

```
{
head=NULL;
tail=NULL;
for(int i=1;i<=100;i++)
{
p=new node;
p->data=i;
if(head==NULL)
{
head=p;
tail=p;
tail->next=NULL;
p->snake=NULL;
p->ladder=NULL;
}
else
{
tail->next=p;
tail=p;
p->snake=NULL;
p->ladder=NULL;
}
}
}


//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void print_initialize()
{
```

```
system("cls");
cout<<"\t\t\tThe Layout of the Game Matrix is:\n\n\n\t";
p=head;
while(p!=NULL)
{
if(p->data%10==0)
{cout<<"\n";}
cout<<p->data<<"\t";
p=p->next;
}
}


/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void form_snake(int o,int t)
{
for(p1=head;p1->data!=o;p1=p1->next)
{}
for(p2=head;p2->data!=t;p2=p2->next)
{}
p1->snake=p2;
}


/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void form_ladder(int o,int t)
{
for(p1=head;p1->data!=o;p1=p1->next)
```

```
{}
for(p2=head;p2->data!=t;p2=p2->next)

{}
p1->ladder=p2;
}



/////////////////////////////////////////////////////////////////////////////////////////////////////////////

void check_snake()
{

cout<<"\n\nThe position of the snakes are:\n";
for(p1=head;p1!=NULL;p1=p1->next)
{
if(p1->snake!=NULL)
cout<<endl<<p1->data<<"->>>"<<p1->snake->data<<endl;
}
}



/////////////////////////////////////////////////////////////////////////////////////////////////////////////

void check_ladder()
{
cout<<"The position of the ladders are:\n";
for(p1=head;p1!=NULL;p1=p1->next)
{
if(p1->ladder!=NULL)
cout<<endl<<p1->data<<"->>>"<<p1->ladder->data<<endl;
}
}

int b1=0,s1=0,b2=0,s2=0;

int bite_count_p2()
{
    ++b2;return (b2);
}
int ladder_count_p2()
{
    ++s2;return (s2);
}



/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

int bite_count_p1()
{
    ++b1;return (b1);
}
int ladder_count_p1()
{
    ++s1;return (s1);
}




/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
void p1_chance()
{
int num;
l1:num=rand();
if(num>6 || num<=0)
goto l1;
cout<<"The outcome of the dice for PLAYER 1 is ->>"<<num<<endl;
while(num>0 && p1!=NULL && p1->data!=100)
{
num--;
p1=p1->next;
}
if(p1->snake!=NULL)
{
  p1=p1->snake;
  printf("\n\n---OOPS SNAKE BITE PLAYER 1---\n\n");
  system("color 43");
  bite_count_p1();
}

else if(p1->ladder!=NULL)
{
   p1=p1->ladder;
   printf("\n\n--- YAHOO!!STEPPING UP PLAYER 1 ---\n\n");
   system("color 34");
   ladder_count_p1();
}
else
{
    system("color 70");
}

}



/////////////////////////////////////////////////////////////////////////////////////////////////////////////


void p2_chance()
{
int num;
l1:num=rand();
if(num>6 || num==0)
goto l1;
cout<<"The outcome of the dice for PLAYER 2 is ->>"<<num<<endl;
while(num>0 && p2!=NULL && p2->data!=100)
{
num--;
p2=p2->next;
}

if(p2->snake!=NULL)
{
    p2=p2->snake;
    printf("\n\n--- OOPS SNAKE BITE PLAYER 2 ---\n\n");
    system("color 43");
    bite_count_p2();

}
```

```
else if(p2->ladder!=NULL)
{
   p2=p2->ladder;
   printf("\n\n--YAHOO!!STEPPING UP PLAYER 2--\n\n");
   system("color 34");
   ladder_count_p2();
}
else
{
   system("color 80");
}

}



//////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void homepage()
{

cout<<endl<<endl<<"\t\tWelcome";Sleep(500);cout<<" to";Sleep(500);cout<<" an";
Sleep(500);cout<<" Interactive";Sleep(500);cout<<" Snake";
Sleep(500);cout<<" Game";Sleep(500);cout<<" Using ";Sleep(500);
cout<<" Linked List";Sleep(500);
cout<<endl;
cout<<"\t\t    @_@"<<endl;
for(int i=0;i<5;i++)
{
for(int j=i;j<20;j++)
cout<<" ";
cout<<"^^^^"<<endl;
Sleep(100);
}
for(int i=15;i<20;i++)
{
for(int j=i;j>0;j--)
cout<<" ";
cout<<"^^^^"<<endl;
Sleep(100);
}
for(int i=0;i<5;i++)
{
for(int j=i;j<20;j++)
cout<<" ";
cout<<"^^^^"<<endl;
Sleep(100);
}
for(int i=0;i<10;i++)
{
cout<<"\t\t\t\t|--|"<<endl;
Sleep(100);
}
cout<<"Project By:-"<<endl<<endl;
Sleep(500);
cout<<"\tShubham Kumar"<<endl<<endl;
Sleep(500);
cout<<"\tNiket Jain"<<endl<<endl;
Sleep(500);
cout<<"\tAyush Sinha"<<endl<<endl;
```

```
Sleep(500);
cout<<"-------------------------------------------------------------------"<<endl<<endl;
cout<<"Project submitted to:-"<<endl<<endl;
Sleep(500);
cout<<"\tProf. Shalini L."<<endl<<endl;
Sleep(500);
}



////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void current_status()
{
cout<<endl;
for(p=head;p!=NULL;p=p->next)
{cout<<"\t";
if(p->data/10==0)
{
cout<<p->data;
}
else
{
cout<<p->data;
}
if(p1->data==p->data)
cout<<"[p1]";
if(p->data%10==int())
cout<<endl;
if(p2->data==p->data)
cout<<"[p2]";
}
}



//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void result_p1(const char* player1)
{
    system("color B0");
    cout<<"\n\n\t\t\t\t\t"<<player1<<" WINS!!\n\n";

}
void result_p2(const char* player2)
{
    system("color B0");
    cout<<"\n\n\t\t\t\t\t"<<player2<<" WINS!!\n\n";

}



//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void easy()
{
    form_ladder(6,33);
    form_ladder(16,24);
    form_ladder(21,44);
    form_ladder(35,66);
    form_snake(37,19);
    form_snake(89,54);
```

```
}


//////////////////////////////////////////////////////////////////////////////////////////////////////////

void medium()

{
    form_ladder(6,33);
    form_ladder(16,24);
    form_ladder(78,96);
    form_ladder(42,74);
    form_snake(63,25);
    form_snake(77,34);
    form_snake(89,54);
    form_snake(51,3);

}


//////////////////////////////////////////////////////////////////////////////////////////////////////////

void hard()

{

    form_ladder(35,66);
    form_ladder(78,96);
    form_ladder(3,45);
    form_snake(48,11);
    form_snake(63,25);
    form_snake(77,34);
    form_snake(89,54);
    form_snake(96,43);
    form_snake(55,21);
    form_snake(91,1);
    form_snake(11,4);
    form_snake(80,61);

}



//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


void chance(int i)
{
    if(i%2==0)
    {
    system("cls");
    p1_chance();
    current_status();
    getch();
    }
    else
    {
    system("cls");
    p2_chance();
    current_status();
    getch();
    }
```

```
}



//////////////////////////////////////////////////////////////////////////////////////////////////////////////


int main()
{
system("color F0");
char player1[50],player2[50];
homepage();
getch();
string another_game;
int p1_score=0,p2_score=0;
int num_game=0;

system("cls");
cout<<"\n\n\n\n\n\t\t\t\tEnter the name of player 1:-";
gets(player1);
cout<<endl<<endl;
cout<<"\t\t\t\tEnter the name player 2:-";
gets(player2);
do
{

initialize();
print_initialize();
getch();
system("cls");
int level;
cout<<"\n\n\n\t\t\t\tEnter the level of your game:-
\n\t\t\t\t1.Easy\n\t\t\t\t2.medium\n\t\t\t\t3.hard\n\n\n";
```

```
cout<<"LEVEL: ";
cin>>level;
switch(level)
{
   case 1:easy();
        break;
   case 2:medium();
        break;
   case 3:hard();
        break;
}
system("cls");

check_ladder();
check_snake();
getch();
system("cls");
cout<<"\t\t"<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl;
cout<<"\t\t\t Let the game begin";
getch();
int i;
int c1=0,c2=0;
for(i=0,p1=head,p2=head;
p1->next!=NULL,p2->next!=NULL,p1->data!=100,p2->data!=100;
i++)
{
   chance(i);

   if (p1->data==100 or p1->next==NULL)
   {
       result_p1(player1);
       p1_score++;
       break;
   }
   else if (p2->data==100 or p2->next==NULL)
   {
       result_p2(player2);
       p2_score++;
       break;
   }
   else
   {
       continue;
   }
}
int k1,j1,k2,j2;
k1=bite_count_p1()-1;
j1=ladder_count_p1()-1;
k2=bite_count_p2()-1;
j2=ladder_count_p2()-1;

getch();

system("cls");

cout<<"\n\n\n\n\n"<<endl;
cout<<"\t\t\t\tThe player1 was bitten by the snake "<<k1<<" time"<<endl;
cout<<"\t\t\t\tThe player1 used the ladder "<<j1<<" time"<<endl;
cout<<"\n\n\n\n"<<endl;
cout<<"\t\t\t\tThe player2 was bitten by the snake "<<k2<<" time"<<endl;
```

```
cout<<"\t\t\t\tThe player2 used the ladder "<<j2<<" time"<<endl;
cout<<"\n\n\n"<<endl;
getch();
system("cls");
cout<<"\n\n\n\n\n\t\t\t\t\tWANT ANOTHER GAME ??"<<endl;
cout<<"\n\n\t\t\t\t\tEnter Yes or No: ";
cin>>another_game;
++num_game;
}
while(another_game=="Yes");
system("cls");
if (p1_score>p2_score)
{
   cout<<"\n\n\n\n\n\t\t\t\t\tAfter round "<<num_game<<", "<<player1<<" is the winner
!!\n\n\n\n"<<endl;
}
else if (p2_score>p1_score)
{
   cout<<"\n\n\n\n\n\t\t\t\t\tAfter round "<<num_game<<", "<<player2<<" is the winner
!!\n\n\n\n"<<endl;
}
else
{
   cout<<"\n\n\n\n\n\t\t\t\t\tAfter round "<<num_game<<" the game is tie !!\n\n\n\n"<<endl;
}
return 0;
}
```

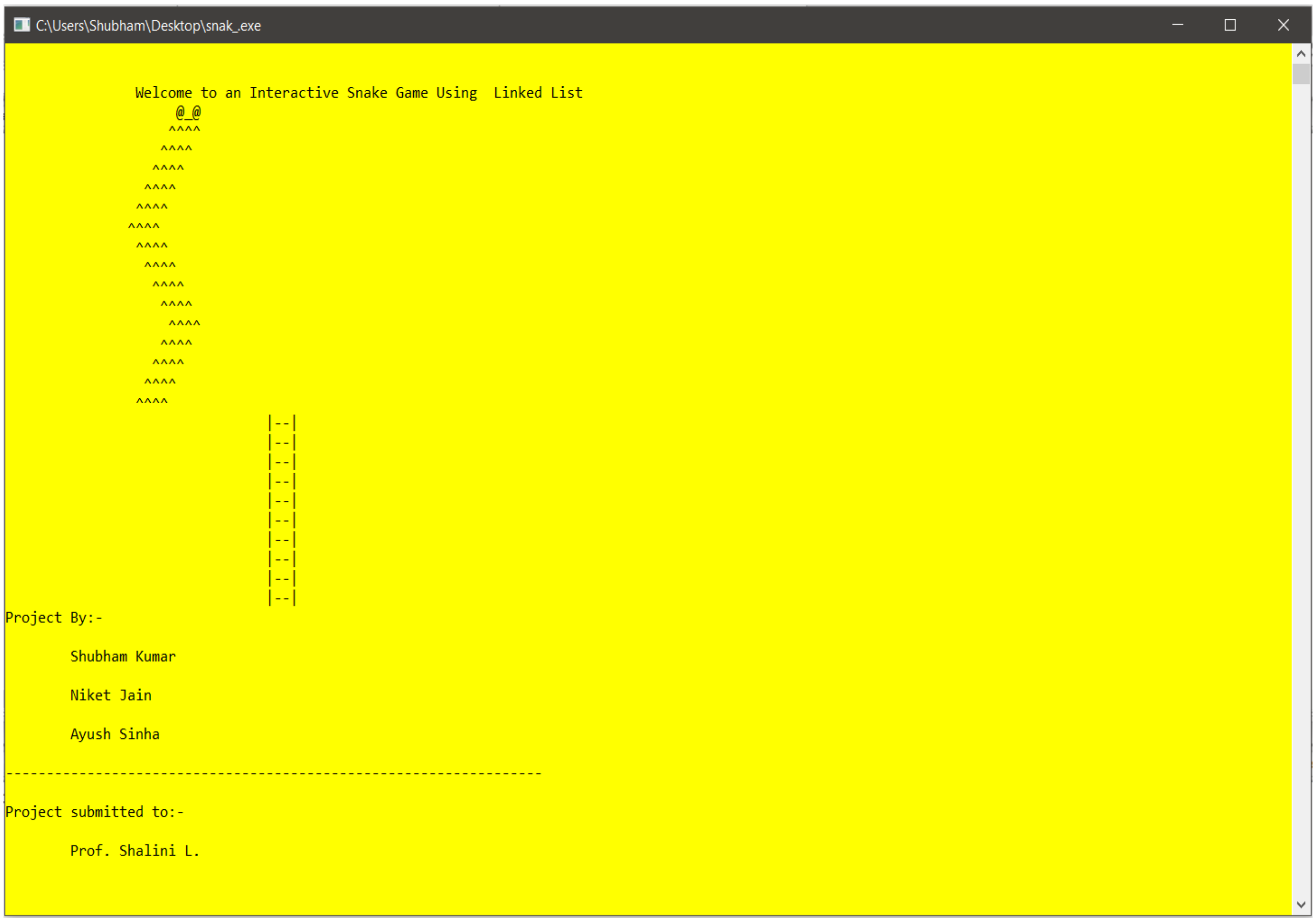
RESULT

We obtain the result we aimed for. We successfully created the old classic game which is "Snake's and Ladders". One of the most fun and entertaining game has been made digitally. We used various codes and algorithms to achieve the best outcome possible with the least number of lines and efficient time and space complexities. After many errors and bugs we have sufficed the desired output.

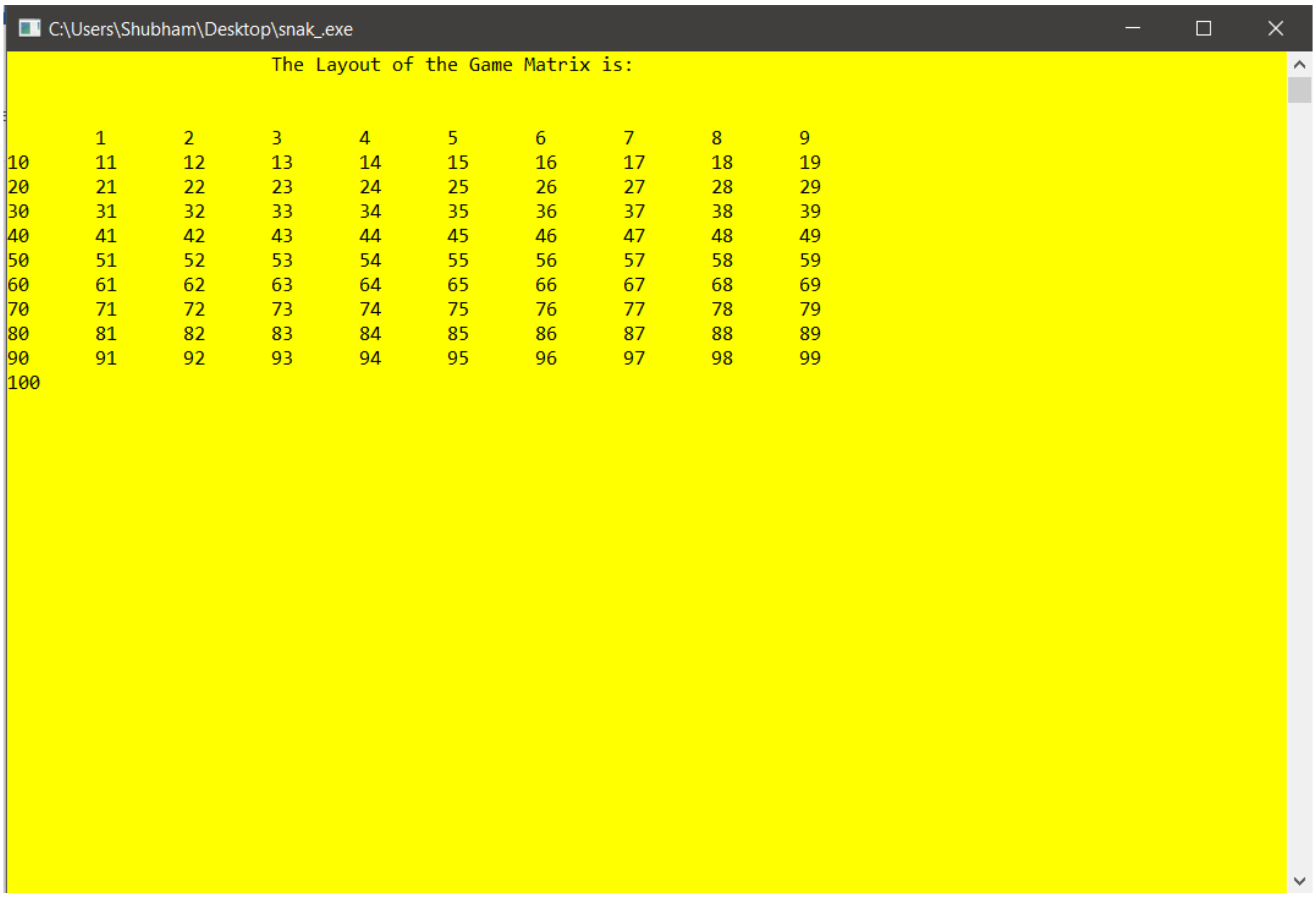
We have also introduced the number of series played by the players as in total number of games won by both the players and whoever wins the most of them, wins the series.
The program runs smoothly and swiftly without any problems; the main agenda of the game has been kept the same. We roll a dice and see where we land up. Many number of shoots have been provided and similarly many number of snakes as well. The first to reach 100 would win the game. We have also introduced a two-way setup for the game and were able to run it successfully.

We would like to thank our professor Dr. Shalini N for believing in us and pushing us to the boundaries for achieving this task. We required the motivation and support for doing it.

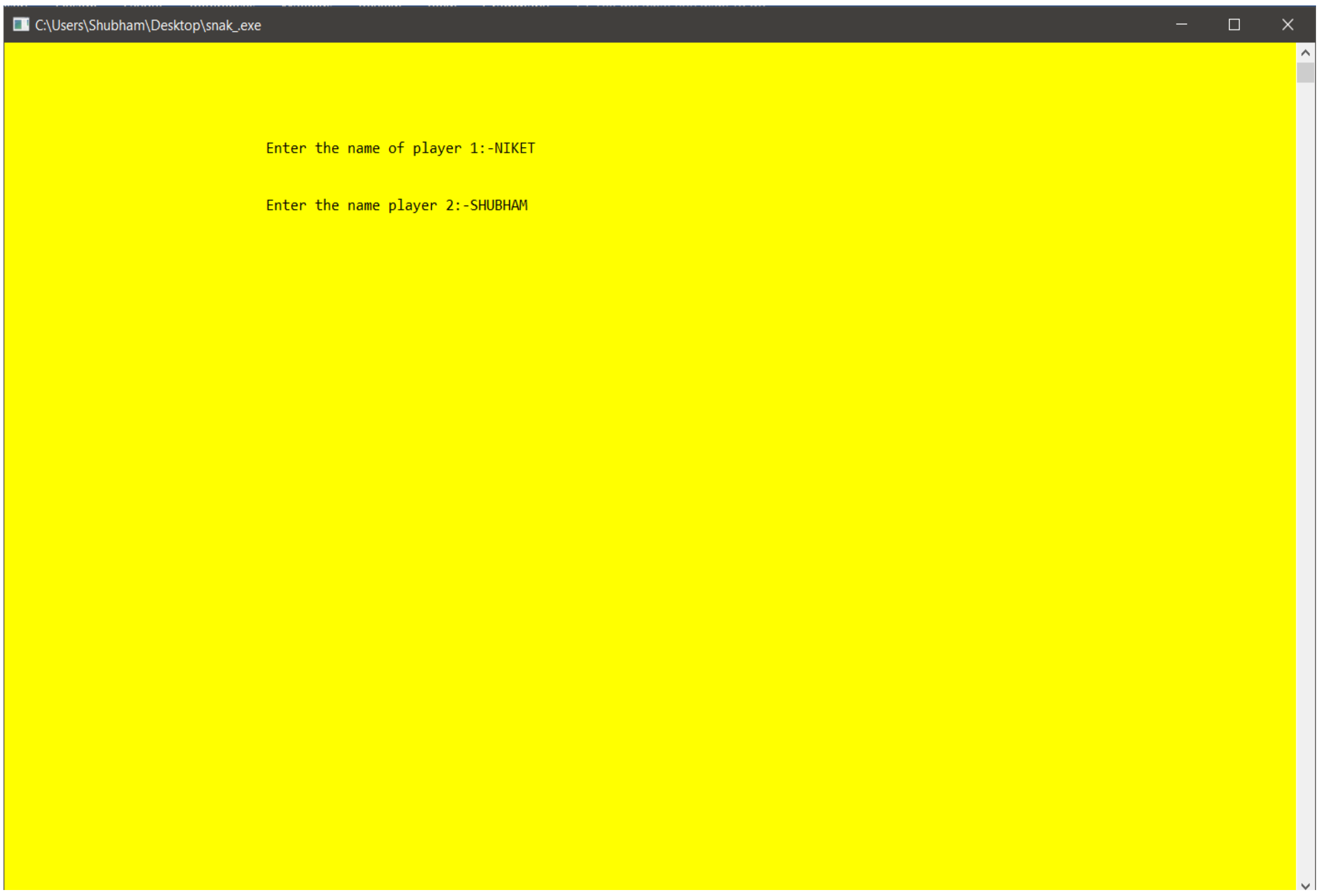
The various output screens are shown in the coming pages:-



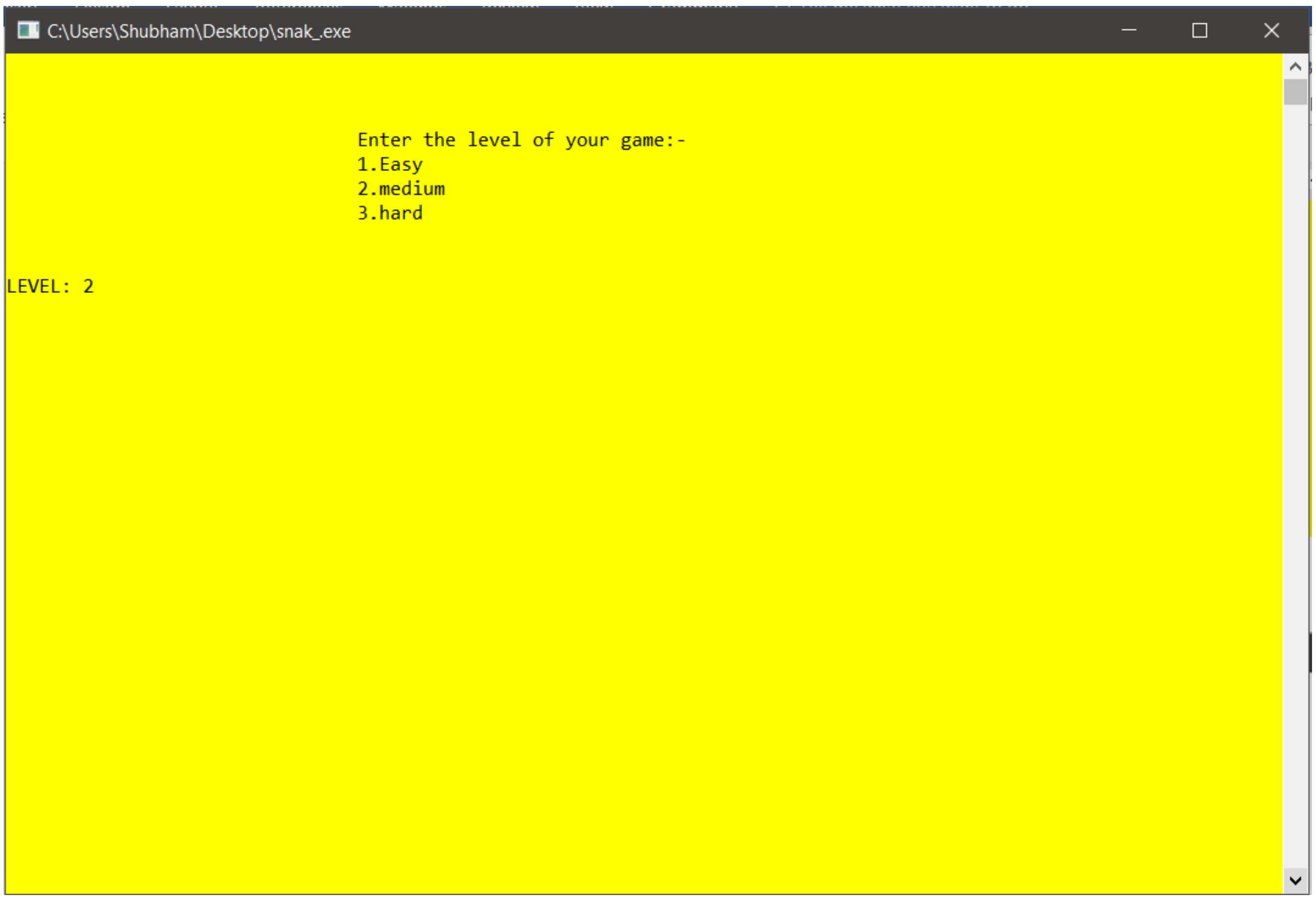
```
                Welcome to an Interactive Snake Game Using  Linked List
                     @_@
                    ^^^^
                   ^^^^
                  ^^^^
                 ^^^^
                ^^^^
               ^^^^
                ^^^^
                 ^^^^
                  ^^^^
                   ^^^^
                    ^^^^
                   ^^^^
                  ^^^^
                 ^^^^
                ^^^^
                                |--|
                                |--|
                                |--|
                                |--|
                                |--|
                                |--|
                                |--|
                                |--|
                                |--|
                                |--|
Project By:-

        Shubham Kumar

        Niket Jain

        Ayush Sinha

------------------------------------------------------------------

Project submitted to:-

        Prof. Shalini L.
```



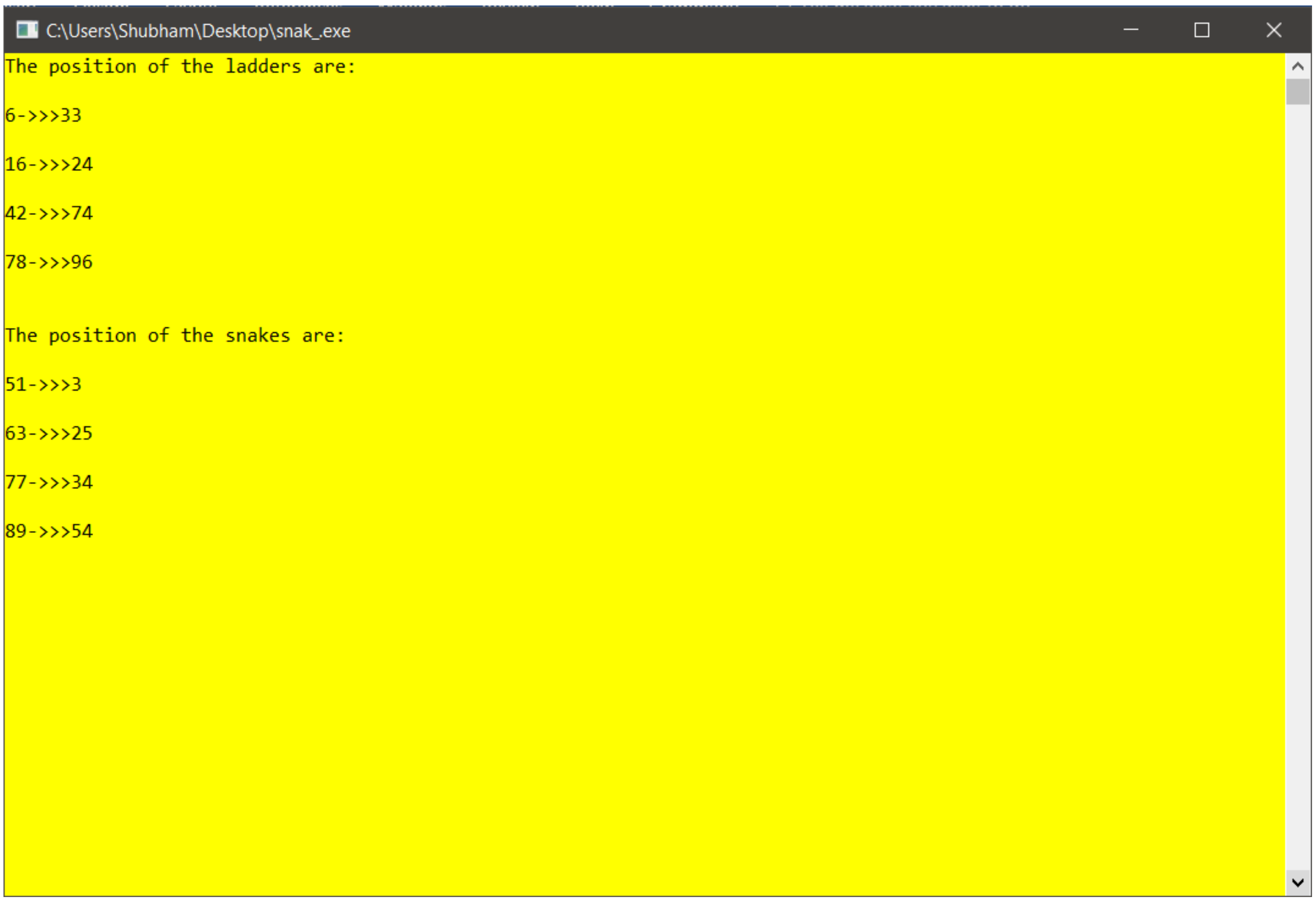
```
                        The Layout of the Game Matrix is:


        1       2       3       4       5       6       7       8       9
10      11      12      13      14      15      16      17      18      19
20      21      22      23      24      25      26      27      28      29
30      31      32      33      34      35      36      37      38      39
40      41      42      43      44      45      46      47      48      49
50      51      52      53      54      55      56      57      58      59
60      61      62      63      64      65      66      67      68      69
70      71      72      73      74      75      76      77      78      79
80      81      82      83      84      85      86      87      88      89
90      91      92      93      94      95      96      97      98      99
100
```



```
Enter the name of player 1:-NIKET


Enter the name player 2:-SHUBHAM
```

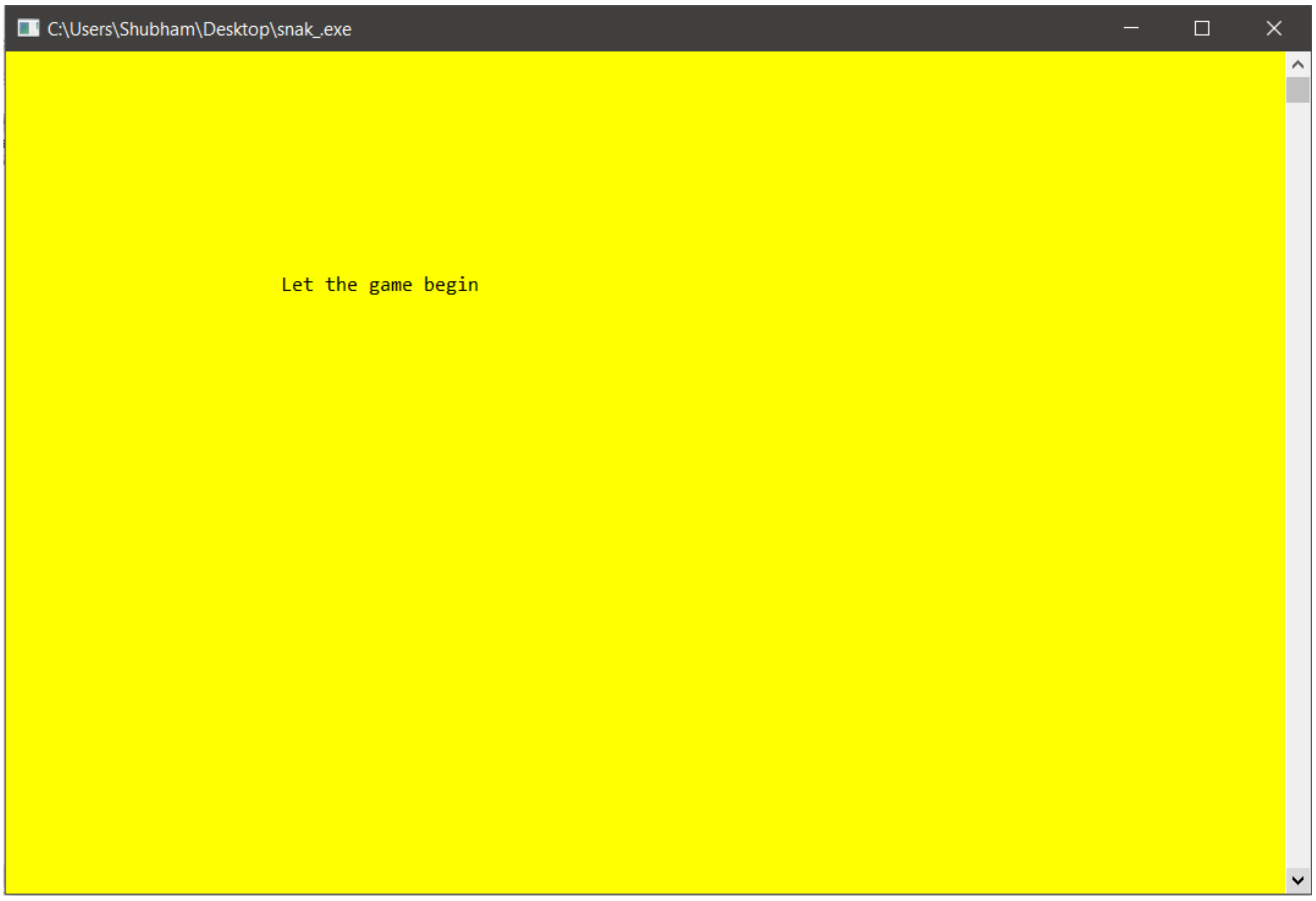


```
                                   Enter the level of your game:-
                                   1.Easy
                                   2.medium
                                   3.hard


LEVEL: 2
```



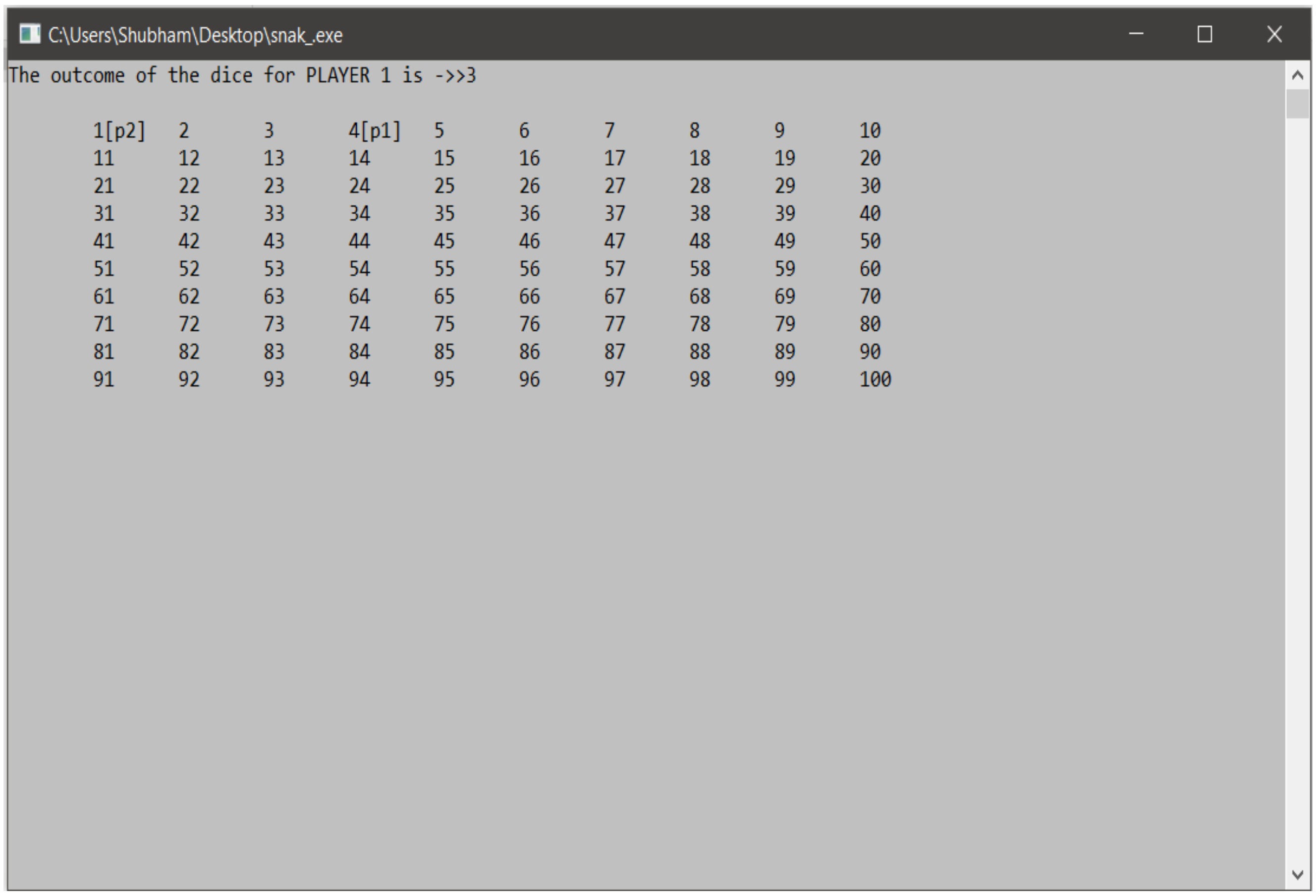
C:\Users\Shubham\Desktop\snak_.exe

```
The position of the ladders are:

6->>>33

16->>>24

42->>>74

78->>>96


The position of the snakes are:

51->>>3

63->>>25

77->>>34

89->>>54
```



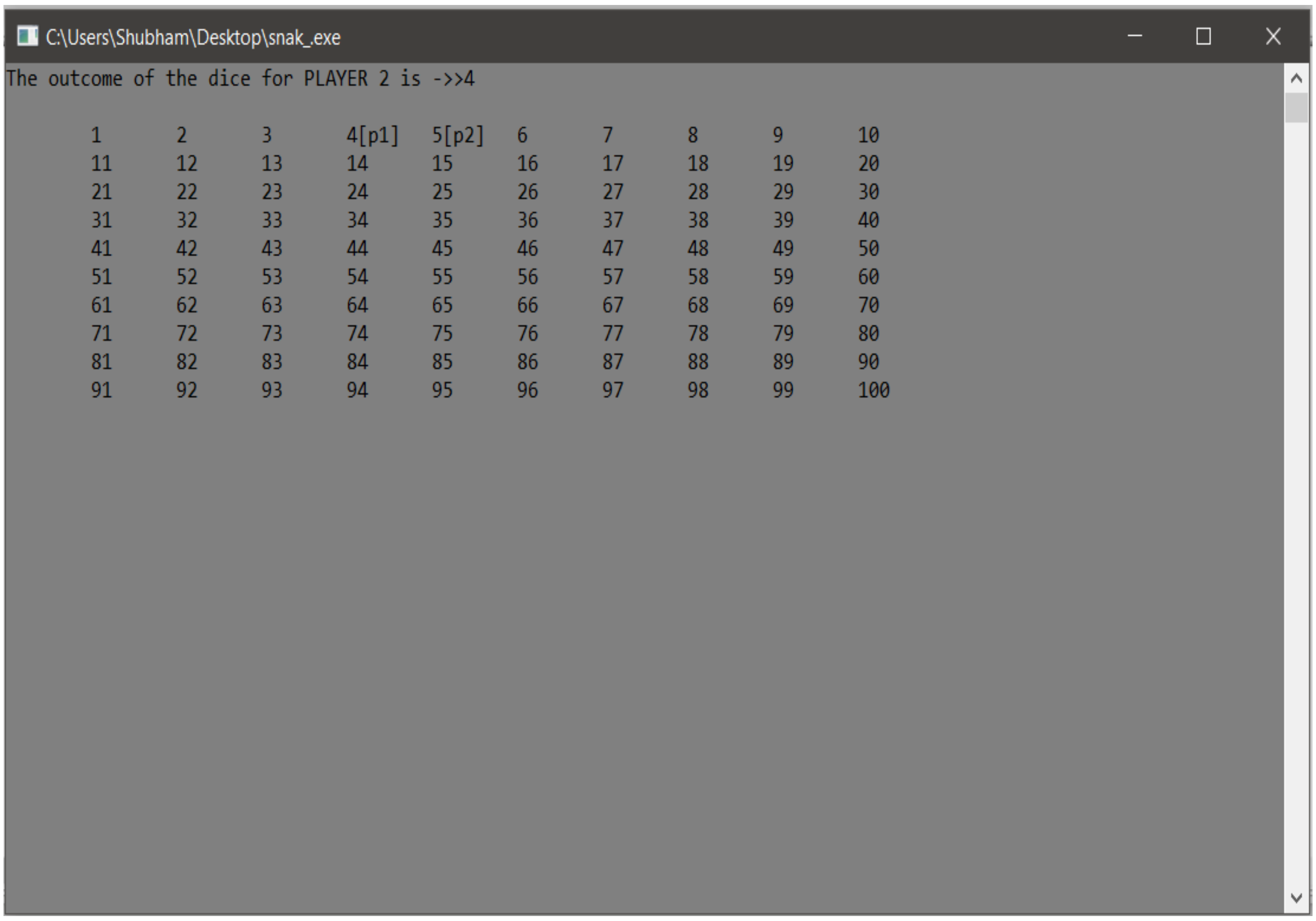
C:\Users\Shubham\Desktop\snak_.exe

```
                         Let the game begin
```



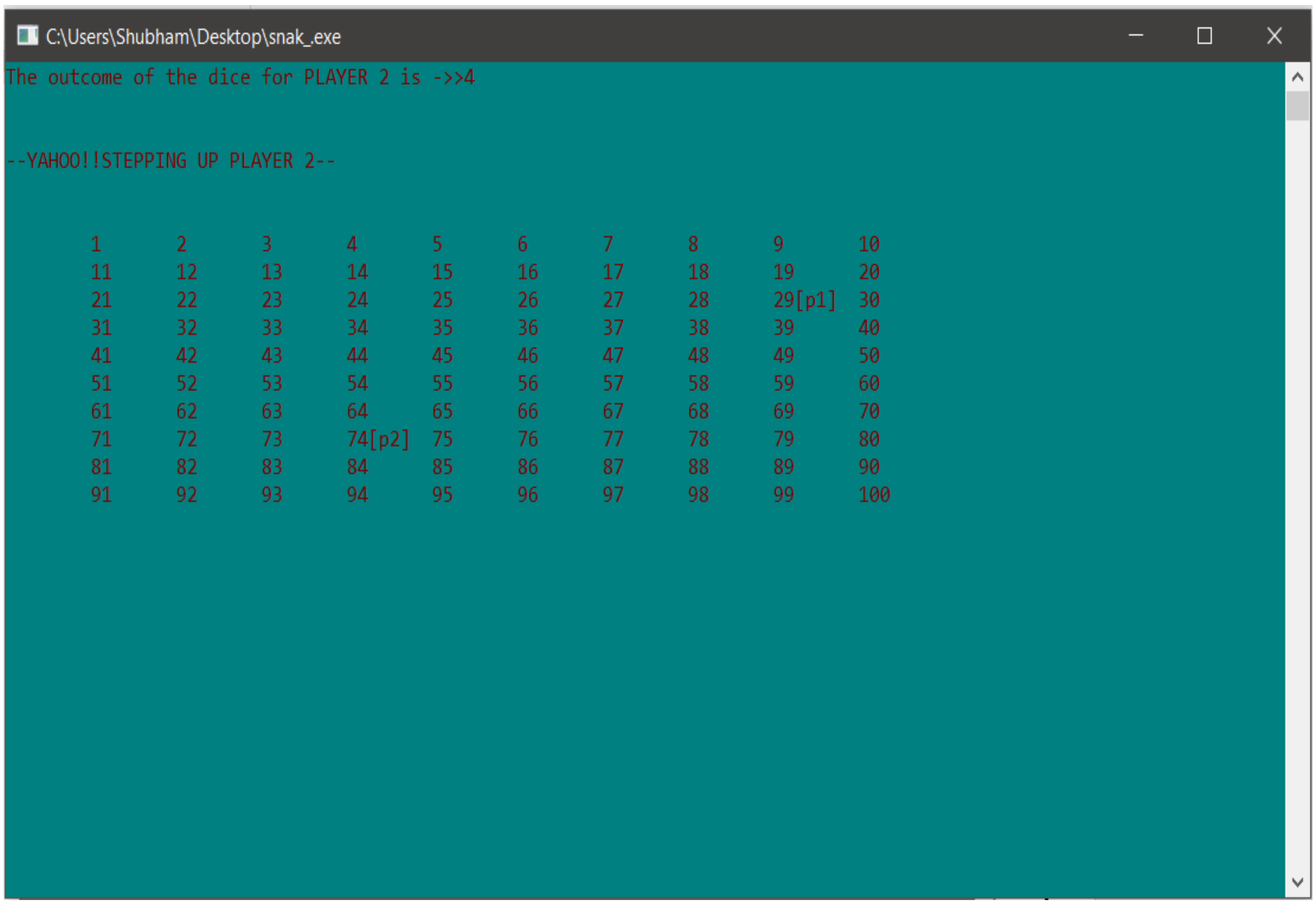
C:\Users\Shubham\Desktop\snak_.exe

```
The outcome of the dice for PLAYER 1 is ->>3

        1[p2]   2       3       4[p1]   5       6       7       8       9       10
        11      12      13      14      15      16      17      18      19      20
        21      22      23      24      25      26      27      28      29      30
        31      32      33      34      35      36      37      38      39      40
        41      42      43      44      45      46      47      48      49      50
        51      52      53      54      55      56      57      58      59      60
        61      62      63      64      65      66      67      68      69      70
        71      72      73      74      75      76      77      78      79      80
        81      82      83      84      85      86      87      88      89      90
        91      92      93      94      95      96      97      98      99      100
```



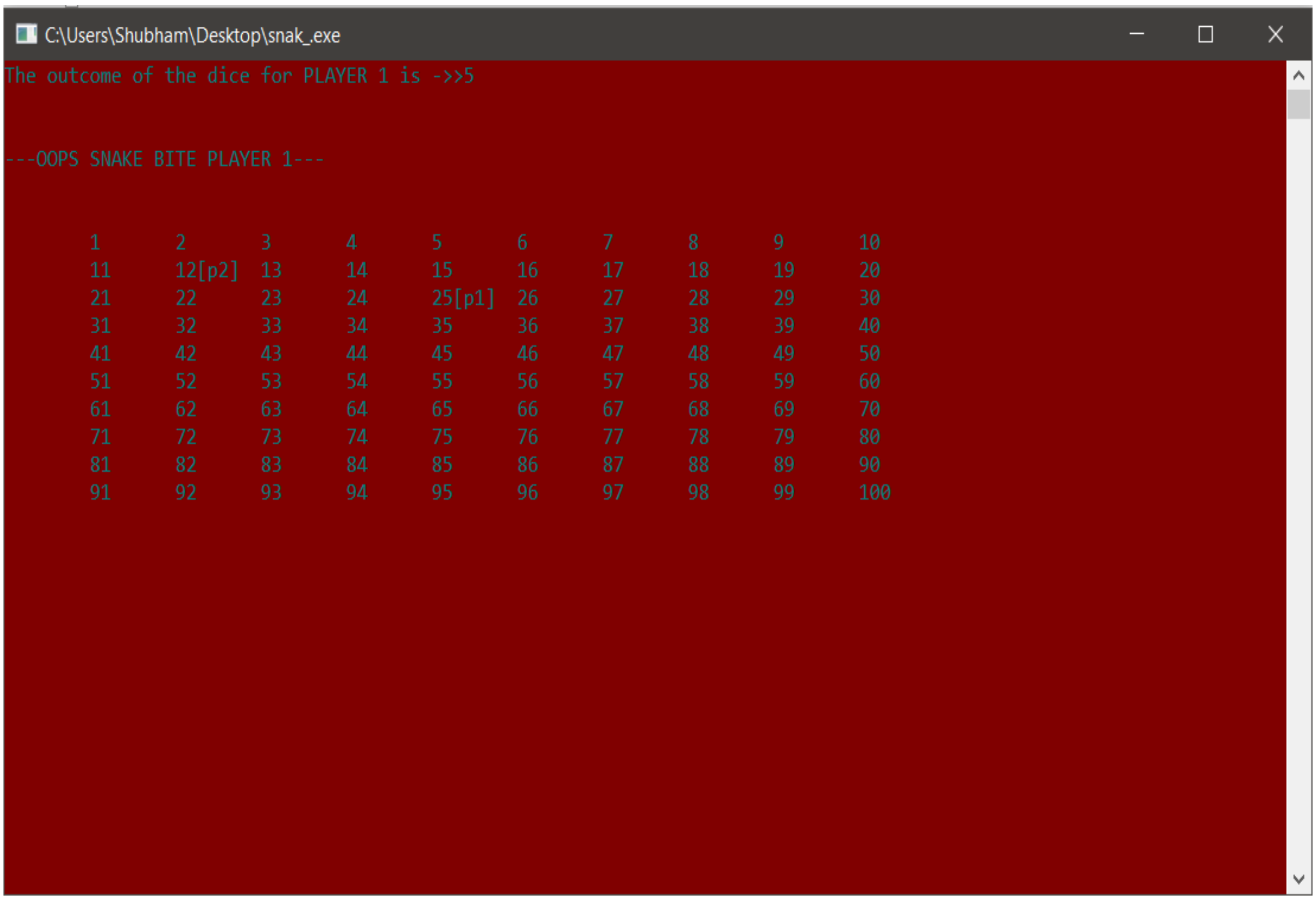
C:\Users\Shubham\Desktop\snak_.exe

```
The outcome of the dice for PLAYER 2 is ->>4

        1       2       3       4[p1]   5[p2]   6       7       8       9       10
        11      12      13      14      15      16      17      18      19      20
        21      22      23      24      25      26      27      28      29      30
        31      32      33      34      35      36      37      38      39      40
        41      42      43      44      45      46      47      48      49      50
        51      52      53      54      55      56      57      58      59      60
        61      62      63      64      65      66      67      68      69      70
        71      72      73      74      75      76      77      78      79      80
        81      82      83      84      85      86      87      88      89      90
        91      92      93      94      95      96      97      98      99      100
```



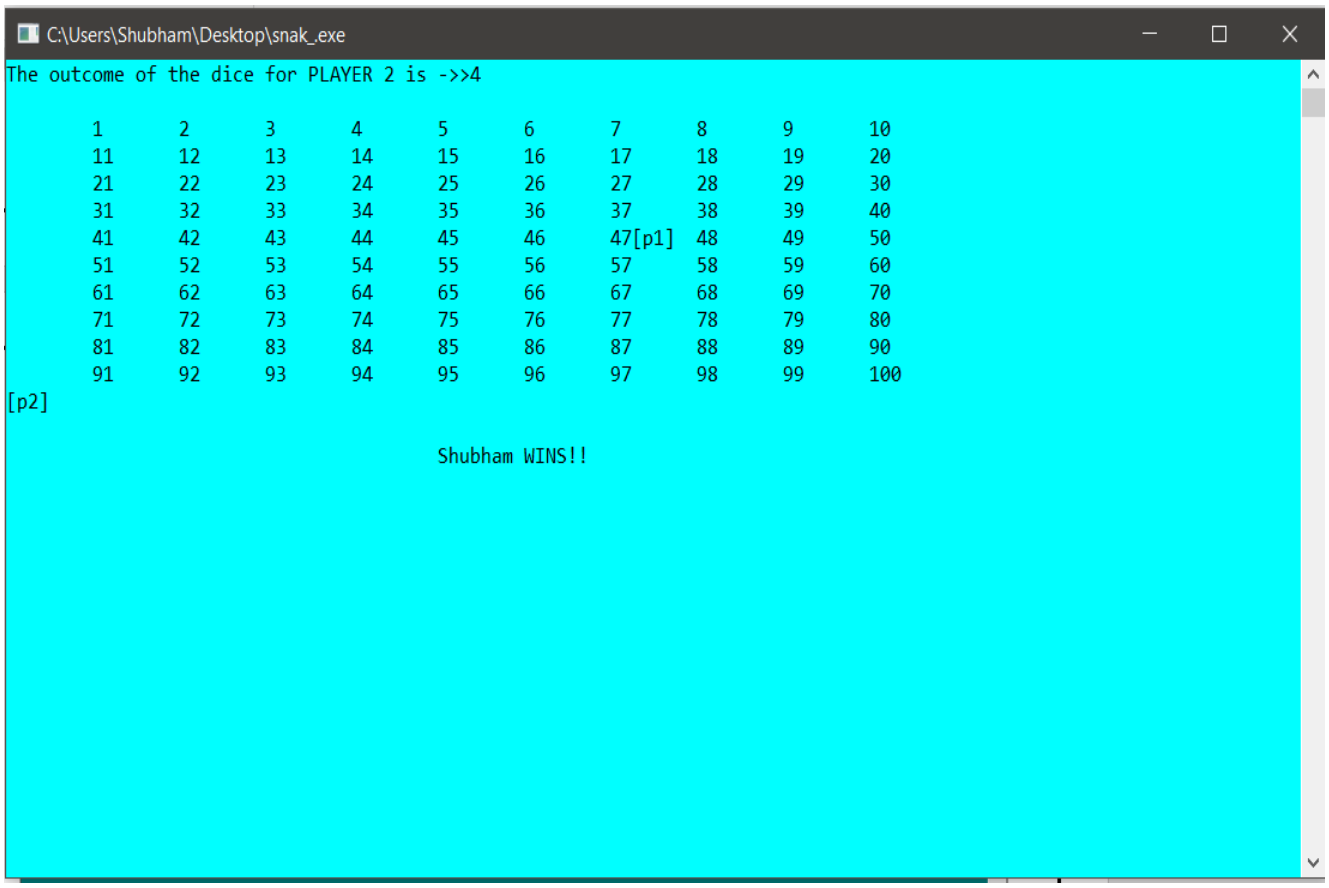
C:\Users\Shubham\Desktop\snak_.exe

```
The outcome of the dice for PLAYER 2 is ->>4


--YAHOO!!STEPPING UP PLAYER 2--



        1       2       3       4       5       6       7       8       9       10
        11      12      13      14      15      16      17      18      19      20
        21      22      23      24      25      26      27      28      29[p1]  30
        31      32      33      34      35      36      37      38      39      40
        41      42      43      44      45      46      47      48      49      50
        51      52      53      54      55      56      57      58      59      60
        61      62      63      64      65      66      67      68      69      70
        71      72      73      74[p2]  75      76      77      78      79      80
        81      82      83      84      85      86      87      88      89      90
        91      92      93      94      95      96      97      98      99      100
```

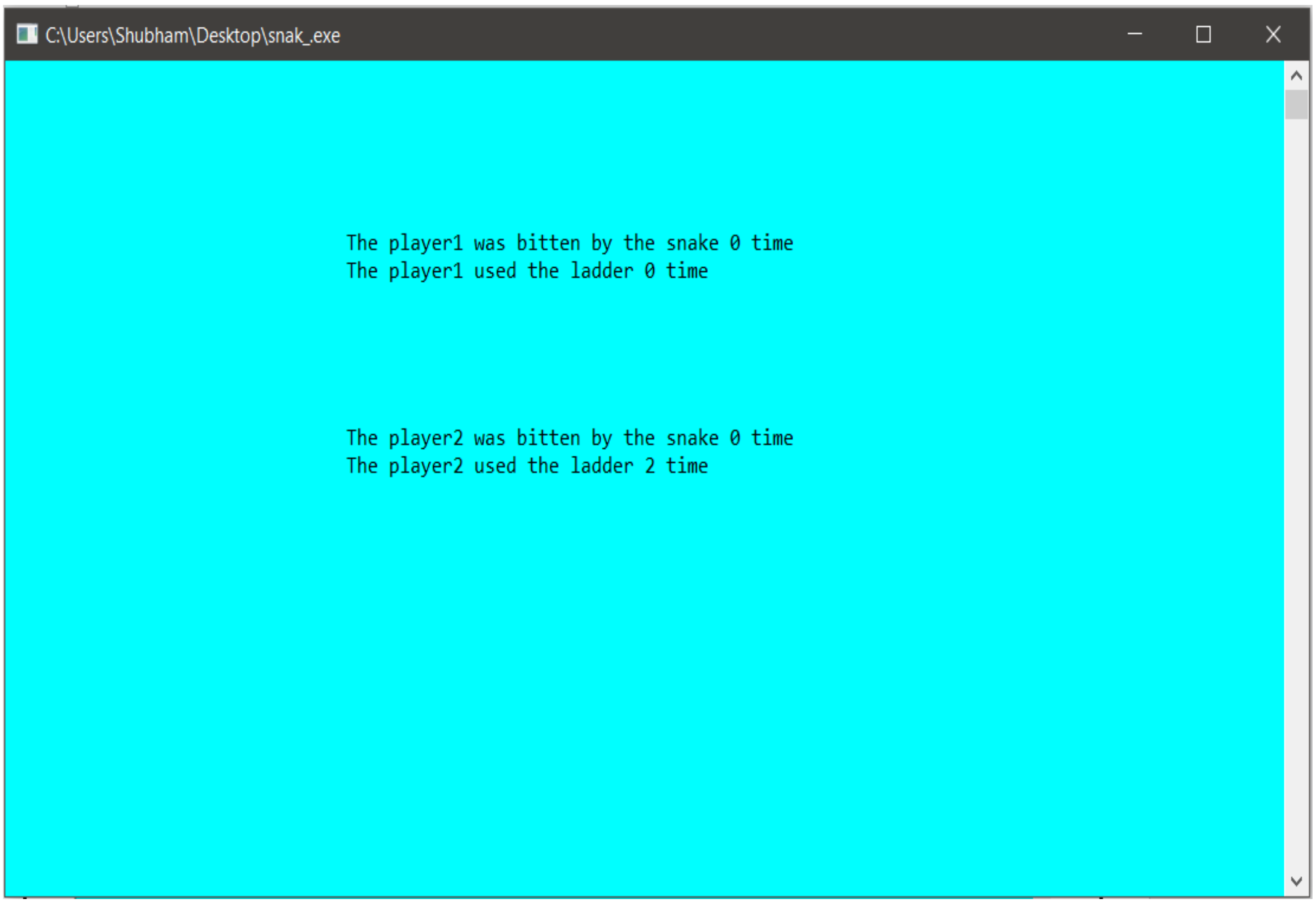


C:\Users\Shubham\Desktop\snak_.exe

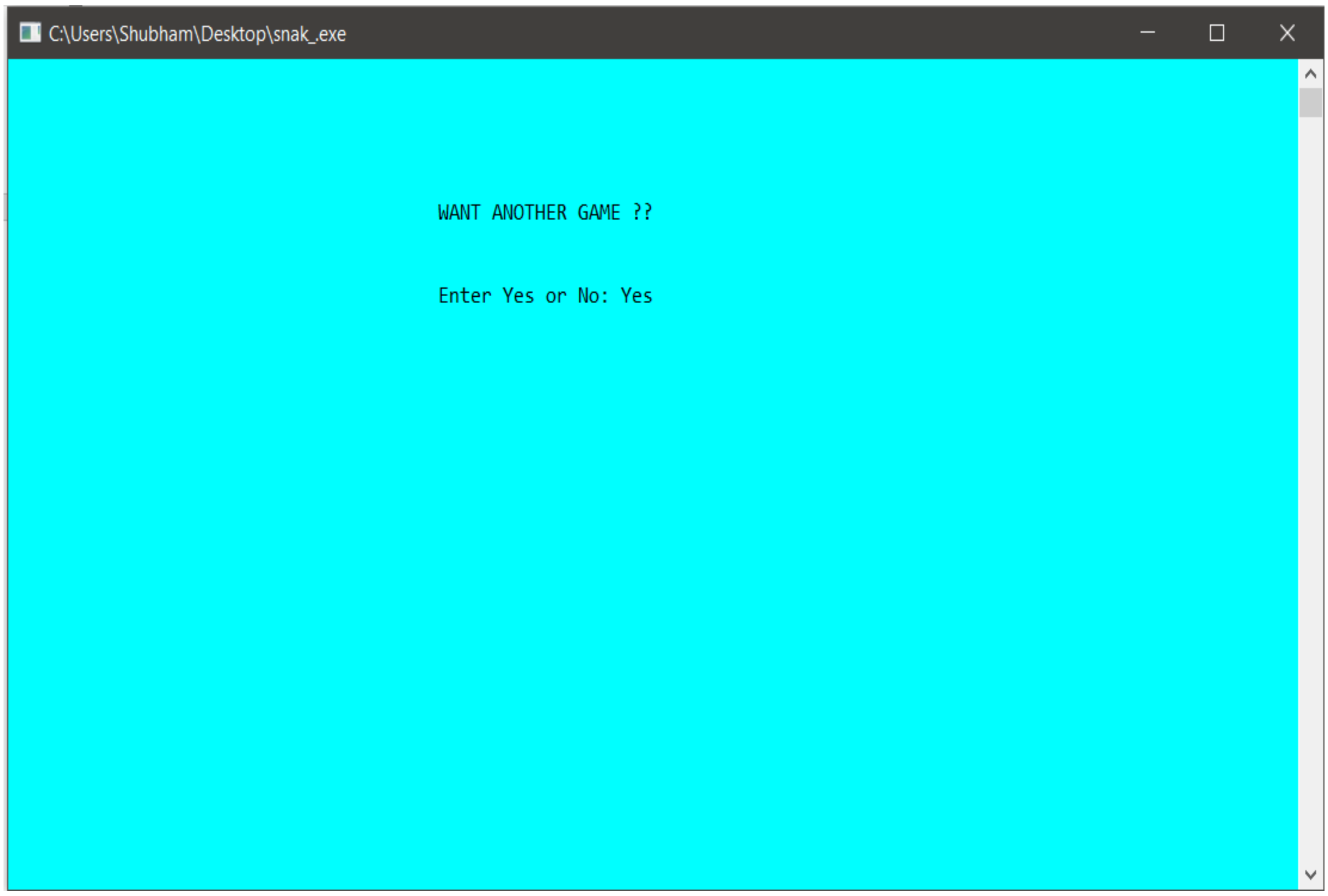
```
The outcome of the dice for PLAYER 1 is ->>5


---OOPS SNAKE BITE PLAYER 1---



        1       2       3       4       5       6       7       8       9       10
        11      12[p2]  13      14      15      16      17      18      19      20
        21      22      23      24      25[p1]  26      27      28      29      30
        31      32      33      34      35      36      37      38      39      40
        41      42      43      44      45      46      47      48      49      50
        51      52      53      54      55      56      57      58      59      60
        61      62      63      64      65      66      67      68      69      70
        71      72      73      74      75      76      77      78      79      80
        81      82      83      84      85      86      87      88      89      90
        91      92      93      94      95      96      97      98      99      100
```



```
The outcome of the dice for PLAYER 2 is ->>4

        1       2       3       4       5       6       7       8       9       10
        11      12      13      14      15      16      17      18      19      20
        21      22      23      24      25      26      27      28      29      30
        31      32      33      34      35      36      37      38      39      40
        41      42      43      44      45      46      47[p1]  48      49      50
        51      52      53      54      55      56      57      58      59      60
        61      62      63      64      65      66      67      68      69      70
        71      72      73      74      75      76      77      78      79      80
        81      82      83      84      85      86      87      88      89      90
        91      92      93      94      95      96      97      98      99      100
[p2]

                                        Shubham WINS!!
```

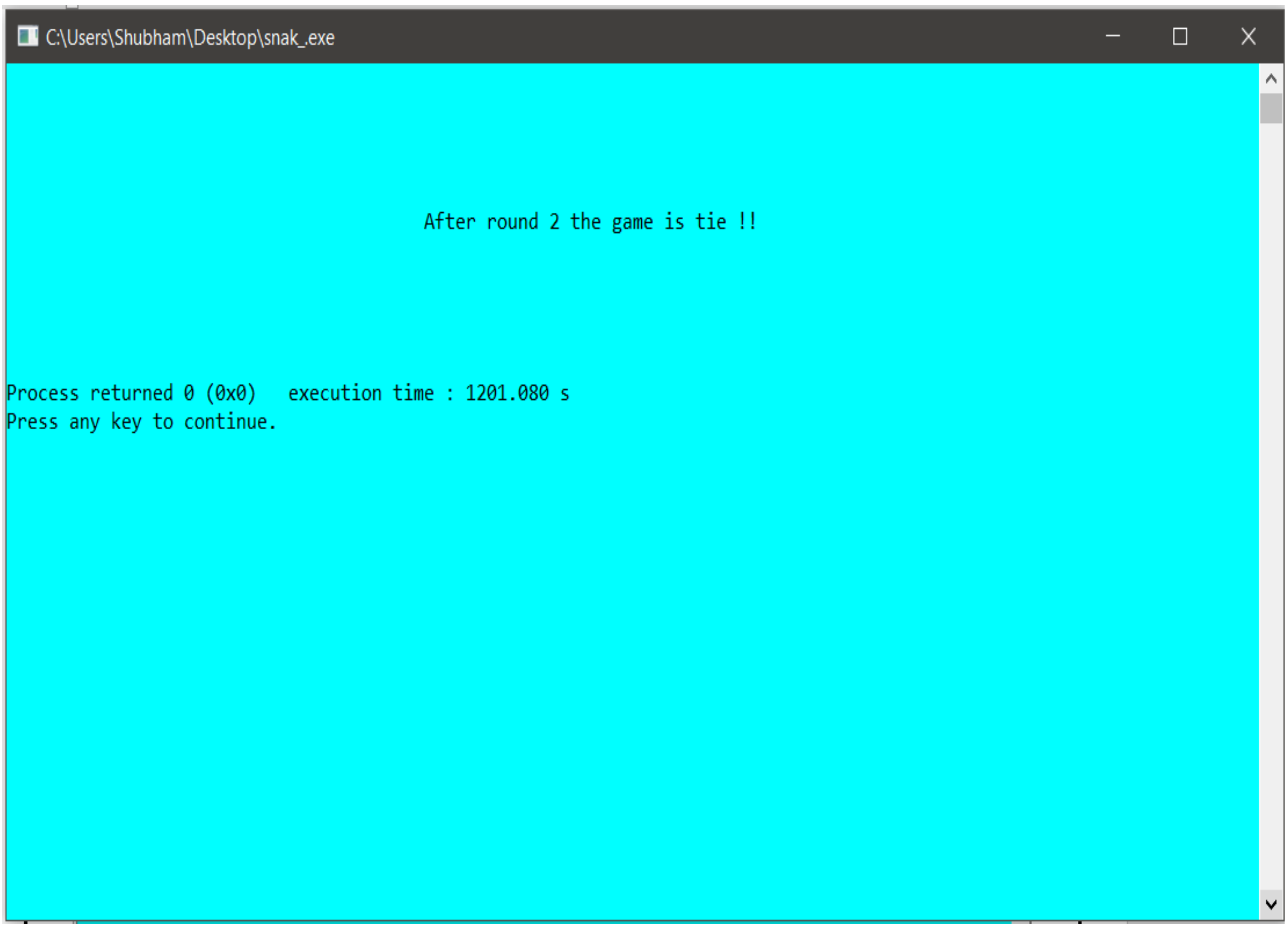


```
                                The player1 was bitten by the snake 0 time
                                The player1 used the ladder 0 time



                                The player2 was bitten by the snake 0 time
                                The player2 used the ladder 2 time
```



C:\Users\Shubham\Desktop\snak_.exe

```
                                        WANT ANOTHER GAME ??


                                        Enter Yes or No: Yes
```



C:\Users\Shubham\Desktop\snak_.exe

```
                                        After round 2 the game is tie !!




Process returned 0 (0x0)   execution time : 1201.080 s
Press any key to continue.
```

REFERENCES

1) www.geeksforgeeks.com

2) www.hackerrank.com

3) www.codespoint.com

4) www.wikipedia.org

5) www.github.com

6) www.stackoverflow.com