INVESTIGATORY PROJECT

DEVELOPED BY: R.E.KEERTHANA

SUBJECT: COMPUTER SCIENCE

TEACHER: DEBJIT BISWAS

PYTHON PROJECT WITH INTERFACE OF MySQL ON

"TEXT BASED SNAKE AND LADDER"

CONTENTS

ACKNOWLEDGEMENT

I would like to express my gratitude to the Computer Science teacher Shri. Debjit Biswas for his guidance and support in completing my project. Though his involvement was not direct, he supported me by building my foundation in python and MySQL with endless efforts.

I would also like to extend my gratitude to my parents for providing me with all the facilities that was required.

CERTIFICATE OF ORIGINALITY

This is an original work done by me. I accept that thousands of students do this project, which would be available over the internet, but I have put my own efforts into this project and made it successful. Not a single part of this project violates copyright & plagiarism. I further agree that my name typed below is intended to have, and shall have, the same validity as my handwritten signature.

R.E. KEERTHANA

The Introduction

At some point of time, we all have played the game 'Snake and Ladder'. The original version consists of a board with 100 tiles numbered on it with some snakes and ladders in between, one coin for each player and a dice to roll.

It is a multi-player game, i.e. it can be played between more than two players.

The main objective of the players is to reach 100 first, for which the player is required to roll dice in their turn and move their coin accordingly, while doing so if the player lands on a snake or a ladder they must move down to the tail or climb to the top of the ladder respectively.

The player who reaches the 100^{th} tile first wins the game.

Now, I thought it would be a great idea to recreate the game with **my own ideas** and this is the result!!!

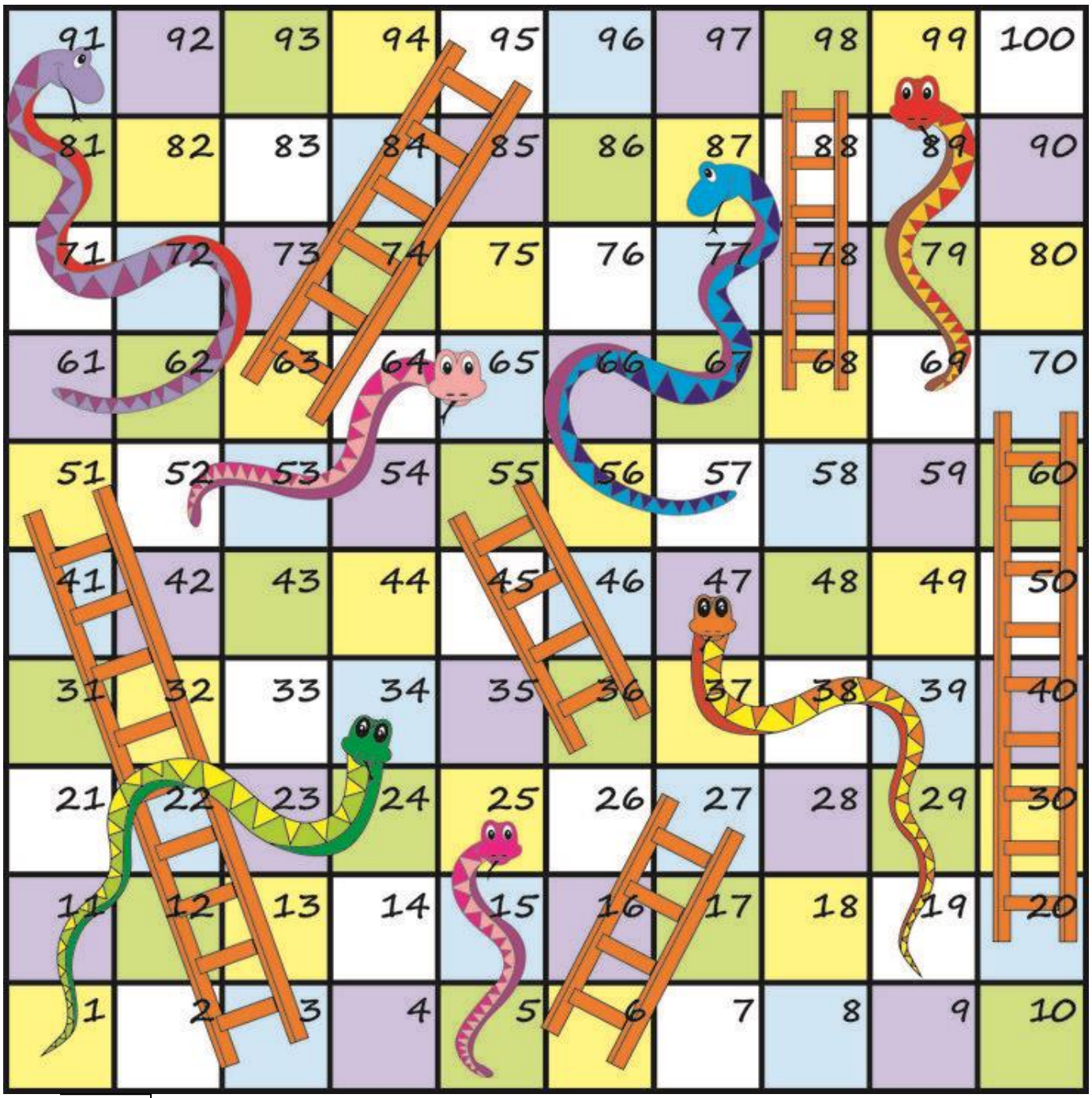
Although I could not get to make the visuals for the game but I made a text based version of the original game with dual player facility.

The program consists of many small functions that are been invoked in the main function.

I have also made the game with MySQL connectivity making sure to keep track on the previous matches played in that device (the game can be executed in any device having python and MySQL), which will be displayed at the end of each gameplay.

The entire code of this game is provided in the last part of the project along with a couple of walkthroughs.

The Basic Layout:



The game is based on this board.

Block wise Explanation:-

#BLOCK-1

```
import time
import random
import sys
import mysql.connector as mc
from tabulate import tabulate
```

- Module time is imported.
- Module random is imported.
- Module sys is imported.
- mysql.connector is imported with an alias name 'mc'
- from the module tabulate the function tabulate is imported.

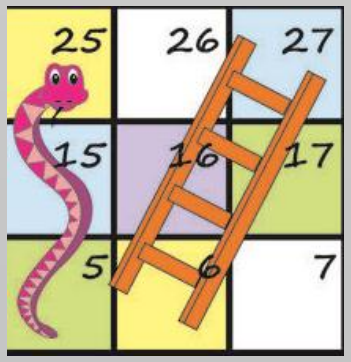
#BLOCK-2

```
MAX_VALUE = 100 # winning point

# snake takes you down from 'key' to 'value'
snakes_at = {
    25:5,
    34:1,
    47:19,
    65:52,
    87:57,
    91:61,
    99:69
}

# ladder takes you up from 'key' to 'value'
ladders_at = {
    3:51,
    6:27,
    20:70,
    36:55,
    63:95,
    68:98
}
# messages when turns
turns = [
    "",
    "Go.",
    "Your turn.",
    "Lets win this.",
    "Are you ready?",
    "Please proceed."
]
# messages when snake
snake_bite = [
    "dang",
    "bummer",
    "boohoo",
    "OHH NOOO",
    "snake bite"
]
# messages when ladder
ladder_climb = [
    "woww",
    "woohoo",
    "yaayyy"
    "nailed it",
    "HIP HIP HURRAY!"
]
```

- The final position of the game is 100, where the game ends and a player wins. The variable MAX_VALUE holds that.
- The dictionary snakes_at holds the positions where the snakes are found at, in the game as the keys and the place where their tails end as the values.
- Similarly, the dictionary ladders_at holds the position where the ladders are found at, in the game as the keys and the place where they end at as the values.
- FOR EXAMPLE:-



- The variable turns, snake_bite, ladder_climb holds the list of messages to motivate the players, emphasis the fact of getting a snake bite and ladder jump respectively to make the game more interactive.

#BLOCK-3

```
# introduction
def welcome_msg():
    msg = """
    Welcome to The Snake and Ladder Game.
    Version: 1.0.0
    Developed by R.E.Keerthana.
    **************************************************
    DISCLAIMER: This game is completely based on chance.
    **************************************************
    HOW TO PLAY:
        1. Decide who goes first.
        2. Press Enter instead of rolling dice... ;)
        3. Climb UP ladders(when you reach them!) to win fast.
        4. Slide DOWN snakes... T_T
        5. Land exactly on the final position (100) to win.
    ALL THE BEST!
    """
    print(msg)
```

The function welcome_msg prints an introduction message also telling the rules to play.

#BLOCK-4

```
# taking player names
def get_player_names():
    player1_name = None
    while not player1_name:
        player1_name = input("Please enter a valid name for first player: ").strip()

    player2_name = None
    while not player2_name:
        player2_name = input("Please enter a valid name for second player: ").strip()

    print("\nMatch will be played between '" + player1_name + "' and '" + player2_name + "'\n")
    return player1_name, player2_name
```

The function get_player_names() performs the task of getting the players' names without any trailing or leading spaces and stores the names in the variable player1_name and player2_name.

It also prints the statement saying the match will be played between so and so players.

#BLOCK-5

```
# virtual method for rolling dice
def get_dice_value():
    time.sleep(1)
    dice_value = random.randint(1,6)
    print("Its a " + str(dice_value))
    return dice_value
```

The function get_dice_value simulates a real life rolling dice by generating a random number from 1 to 6.

To make it look more realistic I have also used time.sleep(1) here to pause the program for 1 second.

#BLOCK-6

```
# function used when snake
def got_snake_bite(old_value, current_value, player_name):
    print("\n" + random.choice(snake_bite).upper() + "  ~~~~~~~~~~>")
    print("\n" + player_name + " got a snake bite. Down from " + str(old_value) + " to " + str(current_value))
```

- The function got_snake_bite takes 3 parameters namely the old value or the previous value of the player in which they landed after rolling the dice, the current value or the present value of the player in which they landed after the snake bite, and the respective player name.
- It also displays a random statement from the list snake_bite (REFER #BLOCK-2) along with a text depiction of snake.
- Lastly, it shows the player's stats.

#BLOCK-7

```
# function used when ladder
def got_ladder_climb(old_value, current_value, player_name):
    print("\n" + random.choice(ladder_climb).upper() + "    ##########")
    print("\n" + player_name + " climbed the ladder from " + str(old_value) + " to " + str(current_value))
```

- The function got_ladder_climb takes 3 parameters namely the old value or the previous value of the player in which they landed after rolling the dice, the current value or the present value of the player in which they landed after the snake bite, and the respective player name.
- It also displays a random statement from the list ladder_climb (REFER #BLOCK-2) along with a text depiction of ladder.
- Lastly, it shows the player's stats.

#BLOCK-8

```
# moving up or down acc. snake or ladder
def snake_ladder(player_name, current_value, dice_value):
    time.sleep(1)
    old_value = current_value
    current_value = current_value + dice_value

    if current_value > MAX_VALUE:
        print("You need " + str(MAX_VALUE - old_value) + " to win this game. Keep trying.")
        return old_value

    print("\n" + player_name + " moved from " + str(old_value) + " to " + str(current_value))
    if current_value in snakes_at:
        final_value = snakes_at.get(current_value)
        got_snake_bite(current_value, final_value, player_name)

    elif current_value in ladders_at:
        final_value = ladders_at.get(current_value)
        got_ladder_climb(current_value, final_value, player_name)

    else:
        final_value = current_value

    return final_value
```

- The function snake_ladder takes 3 parameters namely the player name, current value or the present value of the player before rolling the dice and the dice value which came after rolling the dice.
- Firstly, it updates the variable current_value with the sum of the present value and the dice value of the respective player that we have got as arguments.
- Then it checks whether the current_value exceeds the MAX_VALUE (REFER #BLOCK-2), if yes, then the statement saying you need a specific amount (difference of the old value and the maximum value) of value to win, is displayed and the functions terminates.
- If not so, then it simply prints a line of stats for the respective player.
- Next it checks whether the current value is in either snakes_at dictionary or ladders_at dictionary, if any of them turns to be True, then the corresponding value of that key is saved as final value and the function of got_snake_bite (REFER #BLOCK-6) and got_ladder_climb (REFER #BLOCK-7) is called respectively with required arguments.
- If none of the above mentioned conditions are true then the final value is returned.

#BLOCK-9

```
# checking for winner
def check_win(player_name, position,looser):
    time.sleep(1)
    if MAX_VALUE == position:
        global pwd
        con = mc.connect(host = 'localhost', user = 'root', passwd = pwd)
        print("\n\n\nThats it.\n\n" + player_name + " won the game.")
        print("Congratulations! " + player_name)
        print("\nThank you for playing the game.")
        winner = player_name
        cur = con.cursor() # myql connection
        cur.execute('create database if not exists Snake_And_Ladder')
        cur.execute('use Snake_And_Ladder')
        cur.execute('create table if not exists Score_Board(Winner_History varchar(30) not null,Looser_History varchar(30) not null)')
        names = (winner,looser)
        query = "insert into score_board values(%s,%s)" # all previous game history of
        cur.execute(query,names) # winners and loosers stored
        con.commit() # effecting change

        l = []
        query3 = "select * from score_board"
        cur.execute(query3)
        table = cur.fetchall()
        for row in table:
            row = list(row)
            l.append(row)
        print("\n")
        print("^"*39)
        print("|             SCORE_BORAD              |")
        print("-"*39)
        print(tabulate(l, headers=['Winner_History', 'Looser_History'],tablefmt='orgtbl'))
        print("-"*39)

        downloads = cur.rowcount # no. of record stored
        print('Total number of matches played as of now:',downloads)
        con.close()
        sys.exit(0) # exists
```

- The function check_win takes 3 parameters namely the player name for whose move the function is checking, the position of the player where they stand and the other player name, and checks whether the player wins the game with the move or not, this whole function will execute only if the position and maximum value is equal, i.e. when the player reaches the last tile.
- The function establishes connection with MySQL by accessing the password given at the starting of the game (#REFER BLOCK-10), and creates a database named “Snake_And_Ladder” (if it is not already created) and a table named “Score_Board” (if not already created) having two columns “Winner_History” and “Looser_History”, which is filled by the end of each game as we get to know the winner.
- Now we fetch all the records of the table and put them in a list, so that it can be accessed during tabulate function, which shows the entire table in a beautiful format.
- Finally, the total number of matches played in the particular device is shown, the MySQL connection is closed and the program terminates.

#BLOCK-10

```
def main_game():
    welcome_msg()
    time.sleep(1)
    player1_name, player2_name = get_player_names()
    time.sleep(1)
    global pwd
    pwd = input("Please enter password of mysql:")
    try:
        con = mc.connect(host = 'localhost', user = 'root', passwd = pwd)
    except:
        print("Authentication Error Occurred.")
        print("Incorrect Password!")
        sys.exit(0)
    player1_current_position = 0
    player2_current_position = 0

    while True:
        time.sleep(1)
        input_1 = input("\n" + player1_name + ": " + random.choice(turns) + " Hit the enter to roll dice: ")
        print("\nRolling dice...")
        dice_value = get_dice_value()
        time.sleep(1)
        print(player1_name + " moving....")
        player1_current_position = snake_ladder(player1_name, player1_current_position, dice_value)

        check_win(player1_name, player1_current_position,player2_name)

        input_2 = input("\n" + player2_name + ": " + random.choice(turns) + " Hit the enter to roll dice: ")
        print("\nRolling dice...")
        dice_value = get_dice_value()
        time.sleep(1)
        print(player2_name + " moving....")
        player2_current_position = snake_ladder(player2_name, player2_current_position, dice_value)

        check_win(player2_name, player2_current_position,player1_name)
```

- The function ‘main_game’ is the most important function as it invokes all other functions in the program.
- As we can see that it calls the welcome_msg function, sleeps for a second, then calls the get_player_name function and again sleeps for a second.
- pwd variable is updated with a new value which is took from the user, it is the password for the MySQL which varies device to device. Then the program tries connecting, if the password was correct then the connection would be successful and the program will proceed, but if the password was wrong, then an exception block is used to handle the error so caused which displays statements mentioning the error and the program terminates then and there.
- At the beginning of the game, the position of both players are zero.
- A while loop is used in which the game for two players are defined, it starts with a second of pause.
- The players need to roll the dice by pressing the “Enter” button, after which get_dice_value function is invoked, then a second of pause after which a statement saying the player is moving is shown.
- Finally. The check_win function is called each time to check whether the game ends or not.

#BLOCK-11

```
#main program or calling main function which contains all the subfunctions.
pwd = None
main_game()
```

- The variable pwd is given the value of 'None' first due to the fact that it needs to be derived from the user and requires to be used in a sub function.
- This is the main portion of the game where the function main_game is called.

The Entire Code

```
import time
import random
import sys
import mysql.connector as mc
from tabulate import tabulate

MAX_VALUE = 100 # winning point

# snake takes you down from 'key' to 'value'
snakes_at = {
    25:5,
    34:1,
    47:19,
    65:52,
    87:57,
    91:61,
    99:69
}

# ladder takes you up from 'key' to 'value'
ladders_at = {
    3:51,
    6:27,
    20:70,
    36:55,
    63:95,
    68:98
}
# messages when turns
turns = [
    "",
    "Go.",
    "Your turn.",
    "Lets win this.",
    "Are you ready?",
    "Please proceed."
]
# messages when snake
snake_bite = [
    "dang",
    "bummer",
    "boohoo",
    "OHH NOOO",
    "snake bite"
]
# messages when ladder
ladder_climb = [
    "woww",
    "woohoo",
    "yaayyy"
    "nailed it",
    "HIP HIP HURRAY!"
]
```

```python
# introduction
def welcome_msg():
    msg = """
    Welcome to The Snake and Ladder Game.
    Version: 1.0.0
    Developed by R.E.Keerthana.
    ************************************************************
    DISCLAIMER: This game is completely based on chance.
    ************************************************************
    HOW TO PLAY:
        1. Decide who goes first.
        2. Press Enter instead of rolling dice... ;)
        3. Climb UP ladders(when you reach them!) to win fast.
        4. Slide DOWN snakes... T_T
        5. Land exactly on the final position (100) to win.
    ALL THE BEST!
    """
    print(msg)

# taking player names
def get_player_names():
    player1_name = None
    while not player1_name:
        player1_name = input("Please enter a valid name for first player: ").strip()

    player2_name = None
    while not player2_name:
        player2_name = input("Please enter a valid name for second player: ").strip()

    print("\nMatch will be played between '" + player1_name + "' and '" + player2_name + "'\n")
    return player1_name, player2_name

# virtual method for rolling dice
def get_dice_value():
    time.sleep(1)
    dice_value = random.randint(1,6)
    print("Its a " + str(dice_value))
    return dice_value

# function used when snake
def got_snake_bite(old_value, current_value, player_name):
    print("\n" + random.choice(snake_bite).upper() + "  ~~~~~~~~~~>")
    print("\n" + player_name + " got a snake bite. Down from " + str(old_value) + " to " + str(current_value))

# function used when ladder
def got_ladder_climb(old_value, current_value, player_name):
    print("\n" + random.choice(ladder_climb).upper() + "    ##########")
    print("\n" + player_name + " climbed the ladder from " + str(old_value) + " to " + str(current_value))
```

```python
# moving up or down acc. snake or ladder
def snake_ladder(player_name, current_value, dice_value):
    time.sleep(1)
    old_value = current_value
    current_value = current_value + dice_value

    if current_value > MAX_VALUE:
        print("You need " + str(MAX_VALUE - old_value) + " to win this game. Keep trying.")
        return old_value

    print("\n" + player_name + " moved from " + str(old_value) + " to " + str(current_value))
    if current_value in snakes_at:
        final_value = snakes_at.get(current_value)
        got_snake_bite(current_value, final_value, player_name)

    elif current_value in ladders_at:
        final_value = ladders_at.get(current_value)
        got_ladder_climb(current_value, final_value, player_name)

    else:
        final_value = current_value

    return final_value
```

```
# checking for winner
def check_win(player_name, position,looser):
    time.sleep(1)
    if MAX_VALUE == position:
        global pwd
        con = mc.connect(host = 'localhost', user = 'root', passwd = pwd)
        print("\n\n\nThats it.\n\n" + player_name + " won the game.")
        print("Congratulations! " + player_name)
        print("\nThank you for playing the game.")
        winner = player_name
        cur = con.cursor() # myql connection
        cur.execute('create database if not exists Snake_And_Ladder')
        cur.execute('use Snake_And_Ladder')
        cur.execute('create table if not exists Score_Board(Winner_History varchar(30) not null,Looser_History varchar(30) not null)')
        names = (winner,looser)
        query = "insert into score_board values(%s,%s)" # all previous game history of
        cur.execute(query,names) # winners and loosers stored
        con.commit() # effecting change

        l = []
        query3 = "select * from score_board"
        cur.execute(query3)
        table = cur.fetchall()
        for row in table:
            row = list(row)
            l.append(row)
        print("\n")
        print("^"*39)
        print("|             SCORE_BORAD              |")
        print("-"*39)
        print(tabulate(l, headers=['Winner_History', 'Looser_History'],tablefmt='orgtbl'))
        print("-"*39)

        downloads = cur.rowcount # no. of record stored
        print('Total number of matches played as of now:',downloads)
        con.close()
        sys.exit(0) # exists
```

```
def main_game():
    welcome_msg()
    time.sleep(1)
    player1_name, player2_name = get_player_names()
    time.sleep(1)
    global pwd
    pwd = input("Please enter password of mysql:")
    try:
        con = mc.connect(host = 'localhost', user = 'root', passwd = pwd)
    except:
        print("Authentication Error Occurred.")
        print("Incorrect Password!")
        sys.exit(0)
    player1_current_position = 0
    player2_current_position = 0

    while True:
        time.sleep(1)
        input_1 = input("\n" + player1_name + ": " + random.choice(turns) + " Hit the enter to roll dice: ")
        print("\nRolling dice...")
        dice_value = get_dice_value()
        time.sleep(1)
        print(player1_name + " moving....")
        player1_current_position = snake_ladder(player1_name, player1_current_position, dice_value)

        check_win(player1_name, player1_current_position,player2_name)

        input_2 = input("\n" + player2_name + ": " + random.choice(turns) + " Hit the enter to roll dice: ")
        print("\nRolling dice...")
        dice_value = get_dice_value()
        time.sleep(1)
        print(player2_name + " moving....")
        player2_current_position = snake_ladder(player2_name, player2_current_position, dice_value)

        check_win(player2_name, player2_current_position,player1_name)


#main program or calling main function which contains all the subfunctions.
pwd = None
main_game()
```

The Gameplay

Therefore, we had a glimpse of the entire code, but now we may look at the actual gameplay of the game, how interesting the gameplay will be and other stuff that words cannot explain.

#CASE-1 (Deals with the proper gameplay)

```
    Welcome to The Snake and Ladder Game.
    Version: 1.0.0
    Developed by R.E.Keerthana.
    **********************************************************
    DISCLAIMER: This game is completely based on chance.
    **********************************************************
    HOW TO PLAY:
        1. Decide who goes first.
        2. Press Enter instead of rolling dice... ;)
        3. Climb UP ladders(when you reach them!) to win fast.
        4. Slide DOWN snakes... T_T
        5. Land exactly on the final position (100) to win.
    ALL THE BEST!

Please enter a valid name for first player: Keerthana
Please enter a valid name for second player: Ganesh

Match will be played between 'Keerthana' and 'Ganesh'

Please enter password of mysql:keetu

Keerthana:  Hit the enter to roll dice:

Rolling dice...
Its a 3
Keerthana moving....

Keerthana moved from 0 to 3

HIP HIP HURRAY!    ##########

Keerthana climbed the ladder from 3 to 51

Ganesh: Your turn. Hit the enter to roll dice:

Rolling dice...
Its a 1
Ganesh moving....

Ganesh moved from 0 to 1

Keerthana: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 3
Keerthana moving....

Keerthana moved from 51 to 54

Ganesh: Go. Hit the enter to roll dice:

Rolling dice...
Its a 3
Ganesh moving....
```

```
Ganesh moved from 1 to 4

Keerthana: Your turn. Hit the enter to roll dice:

Rolling dice...
Its a 2
Keerthana moving....

Keerthana moved from 54 to 56

Ganesh: Are you ready? Hit the enter to roll dice:

Rolling dice...
Its a 3
Ganesh moving....

Ganesh moved from 4 to 7

Keerthana: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 4
Keerthana moving....

Keerthana moved from 56 to 60

Ganesh: Go. Hit the enter to roll dice:

Rolling dice...
Its a 1
Ganesh moving....

Ganesh moved from 7 to 8

Keerthana:  Hit the enter to roll dice:

Rolling dice...
Its a 1
Keerthana moving....

Keerthana moved from 60 to 61

Ganesh: Your turn. Hit the enter to roll dice:

Rolling dice...
Its a 5
Ganesh moving....

Ganesh moved from 8 to 13

Keerthana: Go. Hit the enter to roll dice:

Rolling dice...
Its a 5
Keerthana moving....
```

```
Keerthana moved from 61 to 66

Ganesh: Go. Hit the enter to roll dice:

Rolling dice...
Its a 2
Ganesh moving....

Ganesh moved from 13 to 15

Keerthana: Go. Hit the enter to roll dice:

Rolling dice...
Its a 6
Keerthana moving....

Keerthana moved from 66 to 72

Ganesh:  Hit the enter to roll dice:

Rolling dice...
Its a 3
Ganesh moving....

Ganesh moved from 15 to 18

Keerthana: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 1
Keerthana moving....

Keerthana moved from 72 to 73

Ganesh: Go. Hit the enter to roll dice:

Rolling dice...
Its a 4
Ganesh moving....

Ganesh moved from 18 to 22

Keerthana: Your turn. Hit the enter to roll dice:

Rolling dice...
Its a 2
Keerthana moving....

Keerthana moved from 73 to 75

Ganesh: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 3
Ganesh moving....
```

```
Ganesh moved from 22 to 25

DANG  ~~~~~~~~~~>

Ganesh got a snake bite. Down from 25 to 5

Keerthana: Are you ready? Hit the enter to roll dice:

Rolling dice...
Its a 3
Keerthana moving....

Keerthana moved from 75 to 78

Ganesh: Your turn. Hit the enter to roll dice:

Rolling dice...
Its a 1
Ganesh moving....

Ganesh moved from 5 to 6

YAAYYYNAILED IT    ##########

Ganesh climbed the ladder from 6 to 27

Keerthana: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 5
Keerthana moving....

Keerthana moved from 78 to 83

Ganesh:  Hit the enter to roll dice:

Rolling dice...
Its a 1
Ganesh moving....

Ganesh moved from 27 to 28

Keerthana: Are you ready? Hit the enter to roll dice:

Rolling dice...
Its a 1
Keerthana moving....

Keerthana moved from 83 to 84

Ganesh:  Hit the enter to roll dice:

Rolling dice...
Its a 2
Ganesh moving....
```

```
Ganesh moved from 28 to 30

Keerthana: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 1
Keerthana moving....

Keerthana moved from 84 to 85

Ganesh: Go. Hit the enter to roll dice:

Rolling dice...
Its a 5
Ganesh moving....

Ganesh moved from 30 to 35

Keerthana: Go. Hit the enter to roll dice:

Rolling dice...
Its a 5
Keerthana moving....

Keerthana moved from 85 to 90

Ganesh: Your turn. Hit the enter to roll dice:

Rolling dice...
Its a 2
Ganesh moving....

Ganesh moved from 35 to 37

Keerthana: Your turn. Hit the enter to roll dice:

Rolling dice...
Its a 3
Keerthana moving....

Keerthana moved from 90 to 93

Ganesh: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 6
Ganesh moving....

Ganesh moved from 37 to 43

Keerthana:  Hit the enter to roll dice:

Rolling dice...
Its a 2
Keerthana moving....
```

```
Keerthana moved from 93 to 95

Ganesh: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 3
Ganesh moving....

Ganesh moved from 43 to 46

Keerthana:  Hit the enter to roll dice:

Rolling dice...
Its a 6
Keerthana moving....
You need 5 to win this game. Keep trying.

Ganesh:  Hit the enter to roll dice:

Rolling dice...
Its a 1
Ganesh moving....

Ganesh moved from 46 to 47

SNAKE BITE  ~~~~~~~~~~>

Ganesh got a snake bite. Down from 47 to 19
```

```
Keerthana: Lets win this. Hit the enter to roll dice:

Rolling dice...
Its a 5
Keerthana moving....

Keerthana moved from 95 to 100



Thats it.

Keerthana won the game.
Congratulations! Keerthana

Thank you for playing the game.


****************************************
|            SCORE_BORAD               |
----------------------------------------
| Winner_History   | Looser_History   |
|------------------+------------------|
| Rosy             | Julie            |
| Anna             | Julie            |
| Anna             | Rosy             |
| Julie            | Ella             |
| Anna             | Ella             |
| Ella             | Elsa             |
| Elsa             | Anna             |
| Snake            | Mungoose         |
| Mungoose         | Snake            |
| Jackie           | Julie            |
| Boots            | Dora             |
| Dora             | Julie            |
| Boots            | Jcakie           |
| Neemo            | Dory             |
| John             | Ross             |
| Boots            | Benny            |
| Keerthana        | Abirami          |
| Jackie Chan      | Julie Chan       |
| Julie            | Dora             |
| Keerthana        | Tharun           |
| Tharun           | Elango           |
| Suganya          | Suresh           |
| Keerthana        | Ganesh           |
----------------------------------------
Total number of matches played as of now: 23
>>>
```

#CASE-2 (Deals with the gameplay having incorrect password input)

```
    Welcome to The Snake and Ladder Game.
    Version: 1.0.0
    Developed by R.E.Keerthana.
    **************************************************
    DISCLAIMER: This game is completely based on chance.
    **************************************************
    HOW TO PLAY:
        1. Decide who goes first.
        2. Press Enter instead of rolling dice... ;)
        3. Climb UP ladders(when you reach them!) to win fast.
        4. Slide DOWN snakes... T_T
        5. Land exactly on the final position (100) to win.
    ALL THE BEST!

Please enter a valid name for first player: Ganesh
Please enter a valid name for second player: Keerthana

Match will be played between 'Ganesh' and 'Keerthana'

Please enter password of mysql:hello
Authentication Error Occurred.
Incorrect Password!
>>>
```

BIBLIOGRAPHY

- Computer Science with Python Textbook for class XII.
  - Author : Sumita Arora
  - Published By : Dhanpat Rai & Co.
- https://stackoverflow.com/questions/9535954/printing-lists-as-tabular-data
- https://pypi.org/project/tabulate/

THANK YOU!