

Snake and Ladder Assignment

Tayyab Nasir (218-BSCS-2019) (E2)

Documentation of code is already explained in .cpp as well as .txt file in form of remarks aside the code.

Front View:

The front view of this game is explained below. Where Player 1 represents the (\$) symbol and the second player, Player 2 represents the (#) symbol. In the structure as well. This will input names of two players specifically as explained below.

```
*****
Welcome to the Snake and ladder Game..... Player 1 = ($)
                                              Player 2 = (#)
*****
Enter the name of First Player = ASIM
```

After entering the name of first player, the second player name will be entered as shown below. Let suppose that the name of second player is AZHAR.

```
*****
Welcome to the Snake and ladder Game..... Player 1 = ($)
                                           Player 2 = (#)
*****

Enter the name of First Player = ASIM
Enter the name of Second Player = AZHAR
```

First View of puzzle:

The first view of puzzle is shown below, where the L represents the Ladder and S represents the snake head. The following order shows the snake and ladder pattern on specific numbers in puzzle display.

- 6 = Ladder
- 27 = Ladder
- 65 = Snake
- 72 = Ladder
- 97 = Snake

```
*****
Welcome to the Snake and ladder Game..... Player 1 = ($)
Player 2 = (#)
*****

Enter the name of First Player = ASIM
Enter the name of Second Player = AZHAR

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

Its the turn of ASIM
Enter any key for rolling the dice =
```

This will continuously ask both player to roll the dice until anyone of them wins.
This is done by taking a **goto** statements between the two functions, one for each player scenario specifically.
Now below a few effects have been shown.

Ladder Effect;

The given figure shows the ladder effect of one of the players, who come on the numbers having a ladder 'L'.

```

Player ( ASIM ) Move

Your dice is rolling . . . . .
The Number of dice is = 6

It was a LADDER !!

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

NOW Its the turn of AZHAR

Enter any key for rolling the dice =
```

Snake Effect:

The following picture shows the snake effect. The number will be decreased to specific value. After going having declination, the picture is as followed.

```
Player ( AZHAR ) Move

Your dice is rolling . . . . .
The Number of dice is = 4

It was a SNAKE !!

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

Its the turn of ASIM

Enter any key for rolling the dice =
```

Final Result for Winners:

After the loop started with the help of goto statement. The game will continuously run until one of the players wins the game (**making the condition false $P \leq 100$ and $Q \leq 100$**). Then if any player wins the game then it will show the result as explained in picture.

```
Player ( AZHAR ) Move

Your dice is rolling . . . . .
The Number of dice is = 4

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

*****

CONGRATULATIONS ASIM you WON !!

BETTER LUCK. TRY AGAIN AZHAR you LOST !!

*****
```

Conclusion

I have learnt so many things from this assignment. And the credit goes to **Allah Almighty** who helped me in making this assignment on my own, without any external assistance. Secondly the credit goes to Sir **Farhat Ullah** who encouraged me to let this happen. I got an exuberance of knowledge from this assignments, some are as followed:

- Loops
- 2D array Dynamic
- 2D array simple
- 1D array
- Calling arrays through functions for both Dynamic and Static 2D arrays
- Functions
- Syntaxes
- Pointers to 2D arrays
- Conditions
- Random Numbers at specific range

•
•
•
•

And Many More ALHAMDULILLAH