National University of Computer and Emerging Sciences

School of Computing Fall 2019 Islamabad Campus

CS201/CS218: Data Structures (Fall 2019)

Assignment 2

(Deadline: 04th November, 2019 09:00 PM)

Submission: You are required to use Visual Studio for the assignment. Combine all your work (solution folder) in one .zip file after performing "Clean Solution". Name the .zip file as ROLL_NUM.zip (e.g. 18i-0001_A_01.zip). Submit zip file on slate within given deadline. Failure to submit according to above format would result in deduction of 10% marks.
Comments: Comment your code properly. Marks (minimum 10%) will be deducted for the undocumented code.
Deadline: Deadline to submit assignment is **04th November, 2019 09:00 PM**. No submission will be considered for grading outside slate or after the deadline. Correct and timely submission of assignment is responsibility of every student; hence no relaxation will be given to anyone.
Plagiarism: **-50% marks** in the assignment if any significant part of assignment is found plagiarized. A code is considered plagiarized if **more than 20%** code is not your own work.

SNAKE AND LADDER GAME

Snakes and ladders is an ancient south Asian board game. It consists of 10X10 grid board which contains some snakes and ladders at specific boxes/indexes. One hundred is the maximum and a must to win score for each player. First player reaching 100 gets to win the game and is immediately declared as first Winner.

Following are a few basic features of the game;

1. **Dice Board:** A board that display the boxes. Each box can either contain a number (1-100) or a player's pawn and a special field to represent presence of snake head or leader tail.

2. **Snakes:** Any player's pawn landing at a special box which have snake head will result in a severe loss in player score.

3. **Leader:** Any player's pawn landing at a special box which have leader tail will result in a significant increase in player score.

Instructions

In this assignment, you all are required to implement the ancient snake and ladder game with some new exciting rules and implementation guidelines.

Follow the following guidelines to the core to get the maximum marks;

1. **Board:** The game board will be implemented by using a 4D linked list (Next, Previous, Up & Down pointers).

    a. Each node next will points to the next box containing the next number.

    b. Each node previous will points to the previous box containing the previous number.

c. Each node will have up pointer set to null by default except the special nodes which are in path of a ladder. Such nodes up pointers will formulate the complete leader path. Hence any player landing at a special node with ladder tail flag will move up in the path with the help of up pointers until it reaches the top of ladder.

d. Each node will have down pointer set to null by default except the special nodes which are in path of a snake. Such nodes down pointers will formulate the complete snake. Hence any player landing at a special node with snake head flag will move down in the path with the help of down pointers until it reaches the tail of the snake.

2. **Player:** The game can be played between any numbers of players (>=2). The turns of players will be handled by using Queues (Implement using Arrays). The game will not end until we have winners.

   a. 2 Players: Game will end when one player reaches 100.

   b. 3 Players: Game will end when we have first and second winner.

   c. 4 or greater: Game will end when we have first, second and third winner.

3. **Dice:** The dice rolling will be a simple random number generator; which generate number from 1 to 6.

4. **Turn Rules**

   a. Each player will have to get a 6 or 1 on dice roll exactly in order to enter the board. The very first 6 of each player will be considered 1 which means each player will start the game from box 1.

   b. A player gets to roll the dice again; if he gets a 6. Each dice roll will be pushed in the diceRoll Stack. Assume a player rolls 6, 6 & 4. The player pawn will move in accordance with the stack popped dice roll. In our case, player pawn will first move 4 boxes then 6 boxes and then 6 boxes.

   c. If a player roll consecutive 6 exactly three times; he will lose his current turn.

   d. However, if a player roll consecutive 6 exactly four times; next player will lose his turn. Assume, Player B have turn after Player A. Player A rolled consecutive 6 exactly four times and then any number. Player B will end up losing his turn.

   e. If any player roll sum in a specific turn is divisible by three; the player turn queue will reverse itself. Assume, the player turn queue is as follows:

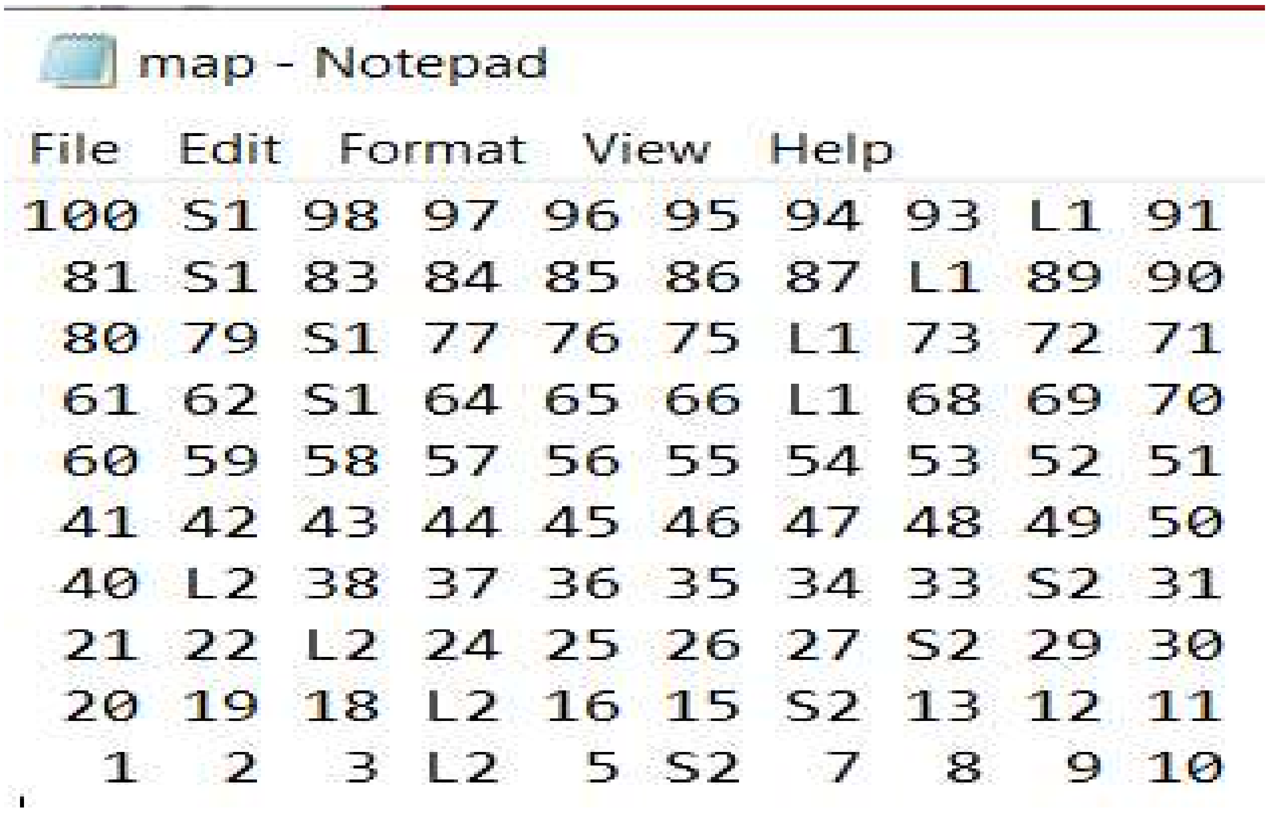
P1	**P2**	**P3**	**P4**

   P1 rolls the dice and ends up getting the roll sum which is divisible by three. The player pawn will move according to the rule b, c & d. Once P1 has completed his turn the turn Queue will reverse itself.

P4	**P3**	**P2**	**P1**

Now the next turn will be of P4 and not P2.

Following will be the map of the game: (Read from map.txt)



```
100 S1 98 97 96 95 94 93 L1 91
 81 S1 83 84 85 86 87 L1 89 90
 80 79 S1 77 76 75 L1 73 72 71
 61 62 S1 64 65 66 L1 68 69 70
 60 59 58 57 56 55 54 53 52 51
 41 42 43 44 45 46 47 48 49 50
 40 L2 38 37 36 35 34 33 S2 31
 21 22 L2 24 25 26 27 S2 29 30
 20 19 18 L2 16 15 S2 13 12 11
  1  2  3 L2  5 S2  7  8  9 10
```

What to submit

Submit your code according to the guidelines given above. Before submitting your code ensure successful execution of all unit tests (uploaded along with the assignment description). Moreover, you are supposed to submit two pages documentation summarizing the data structures and algorithms. The documentation should also provide the justification for the selection of data structures and algorithms.

The assignment will be graded according to the following policy.

- Successful execution of unit tests (90 marks). The marks for each unit test is indicated in the SnakeTest.cpp file.
- Documentation (10 marks).

Good Luck!