

# Chutes and Ladders Board Game

Nellie Garcia

CIS 5

Spring 2019

42411

## **Introduction**

### Chutes and Ladders

Chutes and Ladders is a simple boardgame where the ultimate goal is to reach the number 100. In order to do such, the player must roll a die and move their character. If the player lands on a number with a ladder symbol on it, the player must go up to the top of the ladder. If the player lands on a number with a chute, the player must go down until it ends. For example, landing on the number 80 is a ladder all the way up to the number 100. The traditional game can have up to 4 players, but my version only has up to 2 players.

## **Summary**

Project Size: 600+ lines

Number of variables: 20

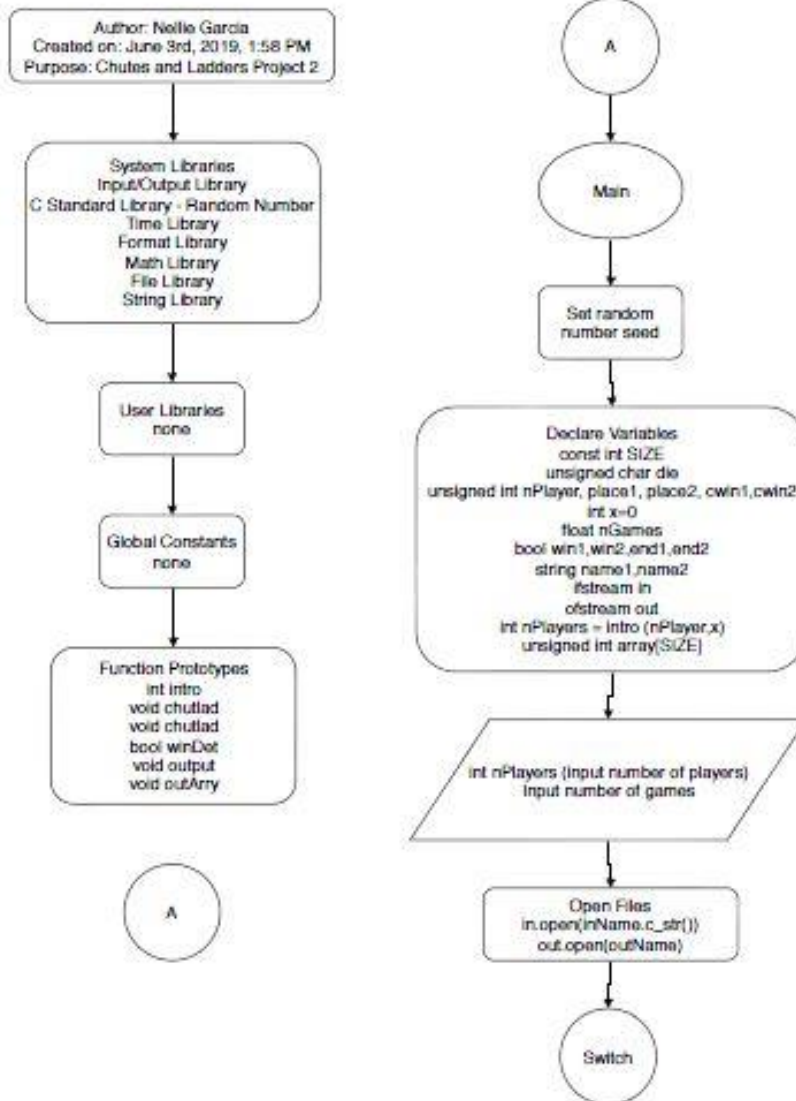
When I began on Project 2, I immediately went to starting my functions, deciding to worry about what I would do with arrays later. I created a function for the intro so that I could return an int value, then created two void functions, both used for testing chutes and ladders. These two functions have the same name, chutlad, but have different variables inside so that I could demonstrate using overloading.

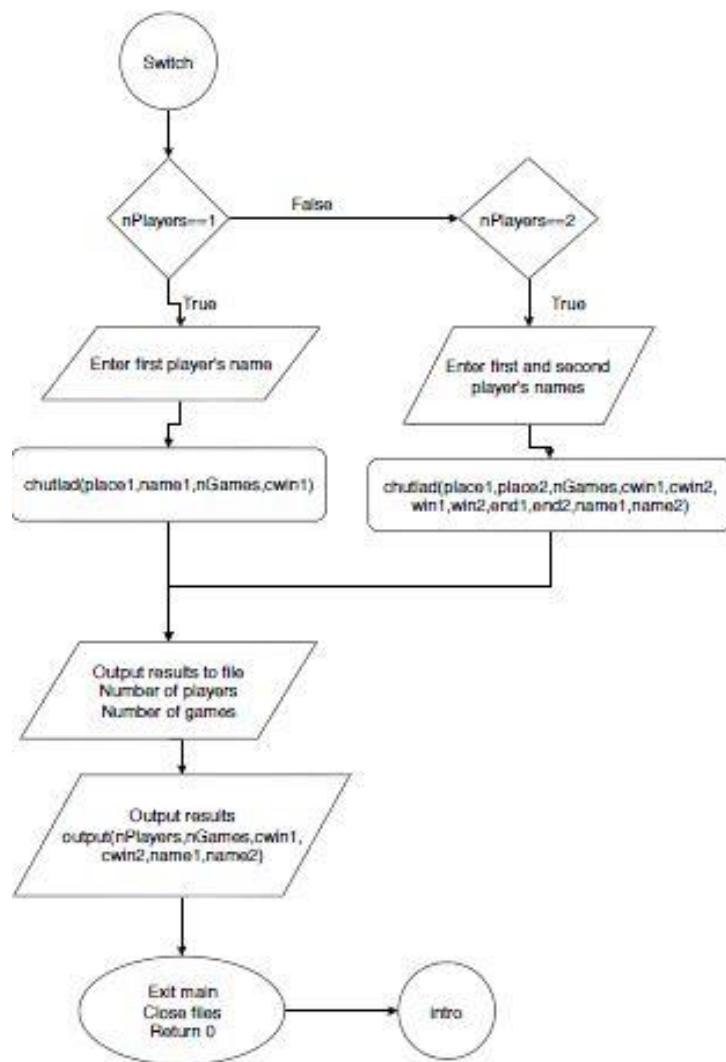
I also created a bool function called winDet. This function was used to determine if a win had occurred in the one-player version of the game. If the function returned true, then the counter for player one would go up, and it would move onto the next game, or end the games altogether.

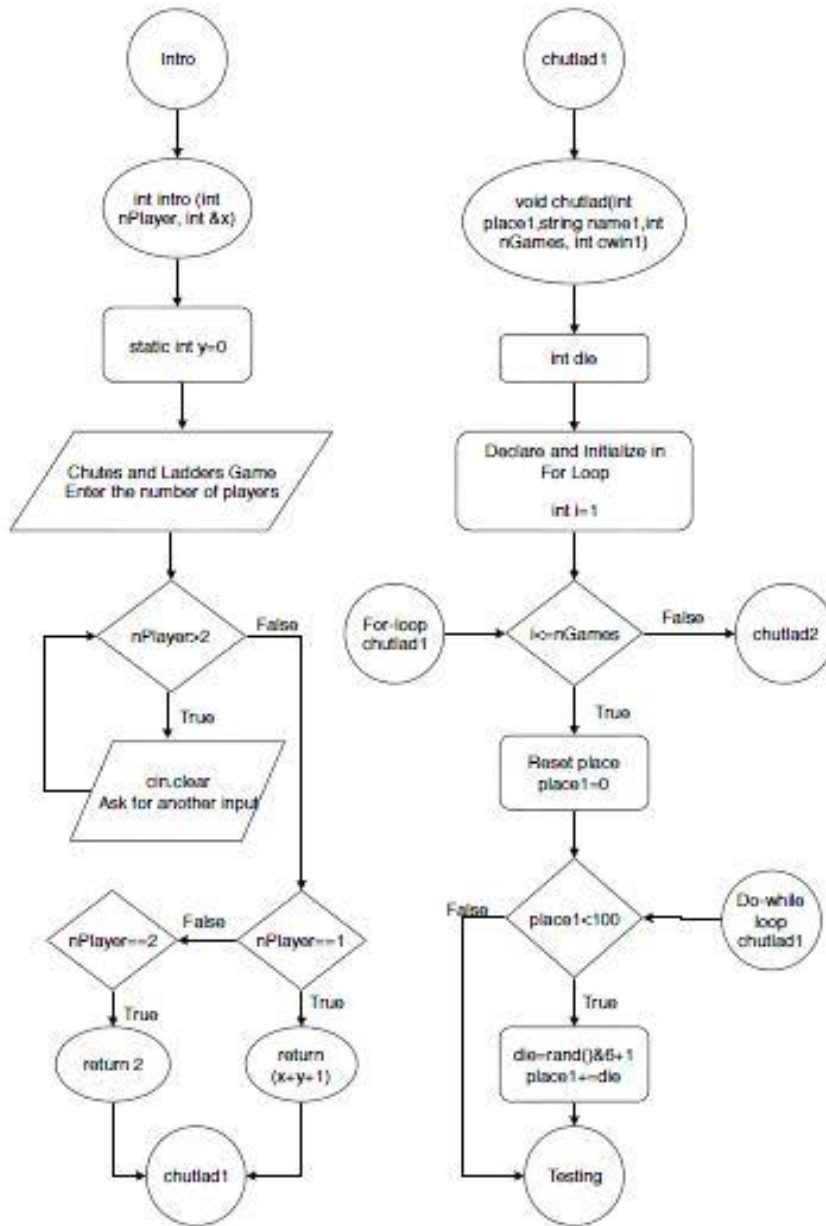
My second to last function is called output. This function is used solely for the process of outputting the results at the end of the program. In other words, it outputs the amount of games, the amount of players, and the number of wins each player accumulated.

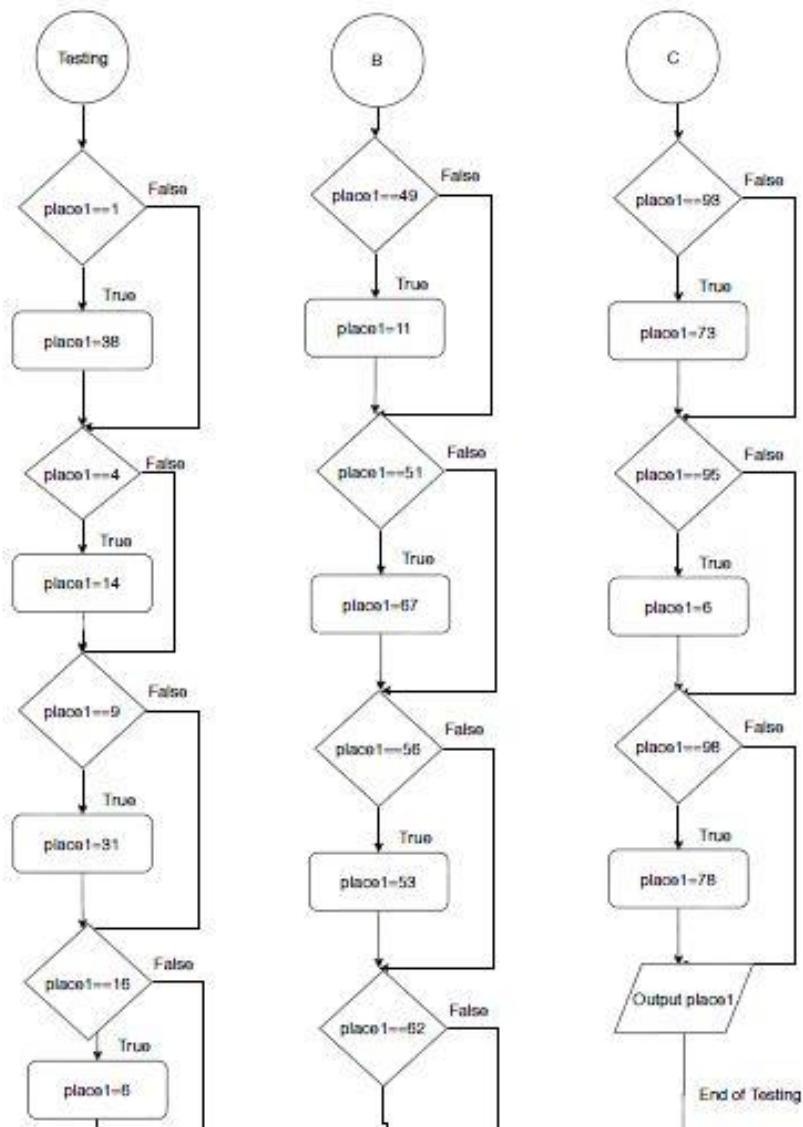
My next step was to create an array, so that I could then move onto demonstrating sorting. However, while I created the array, I had a lot of trouble with outputting it somehow. Therefore, while I have the array, I will not be demonstrating outputting it nor sorting it.

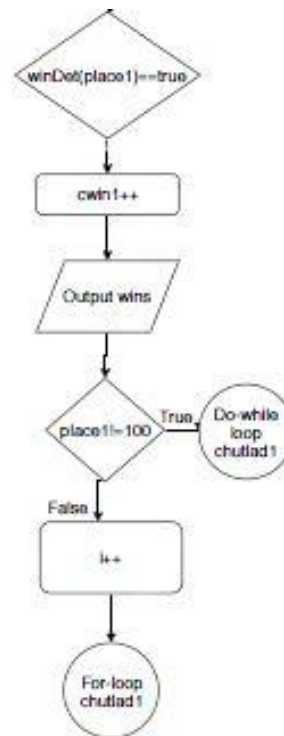
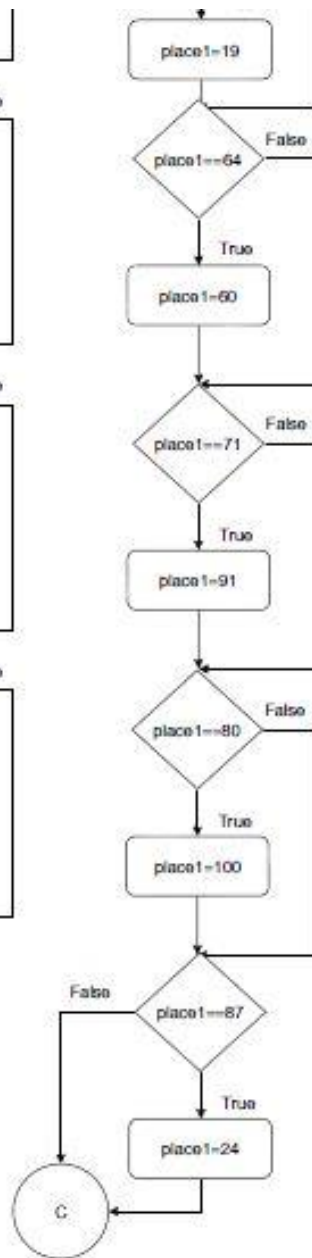
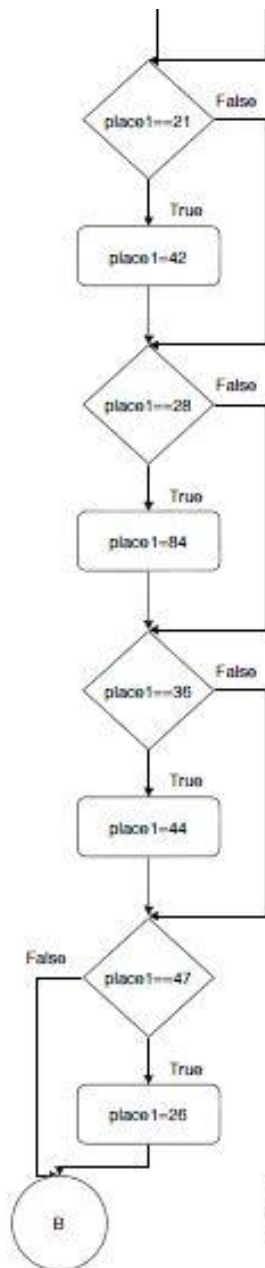
## **Flowcharts**

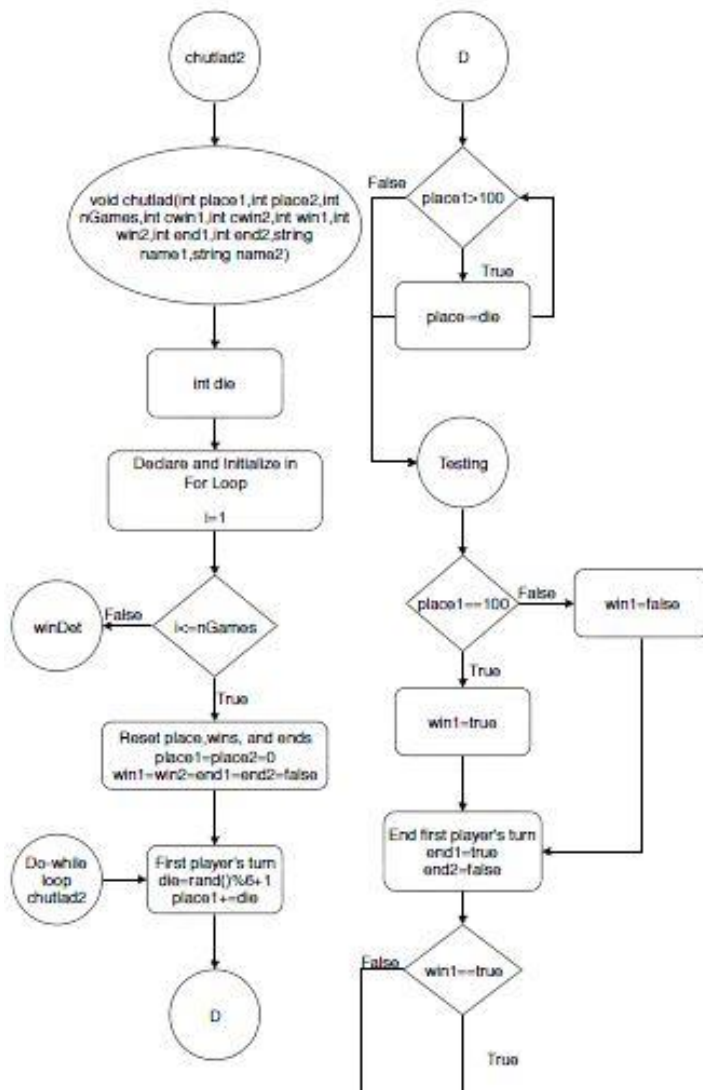




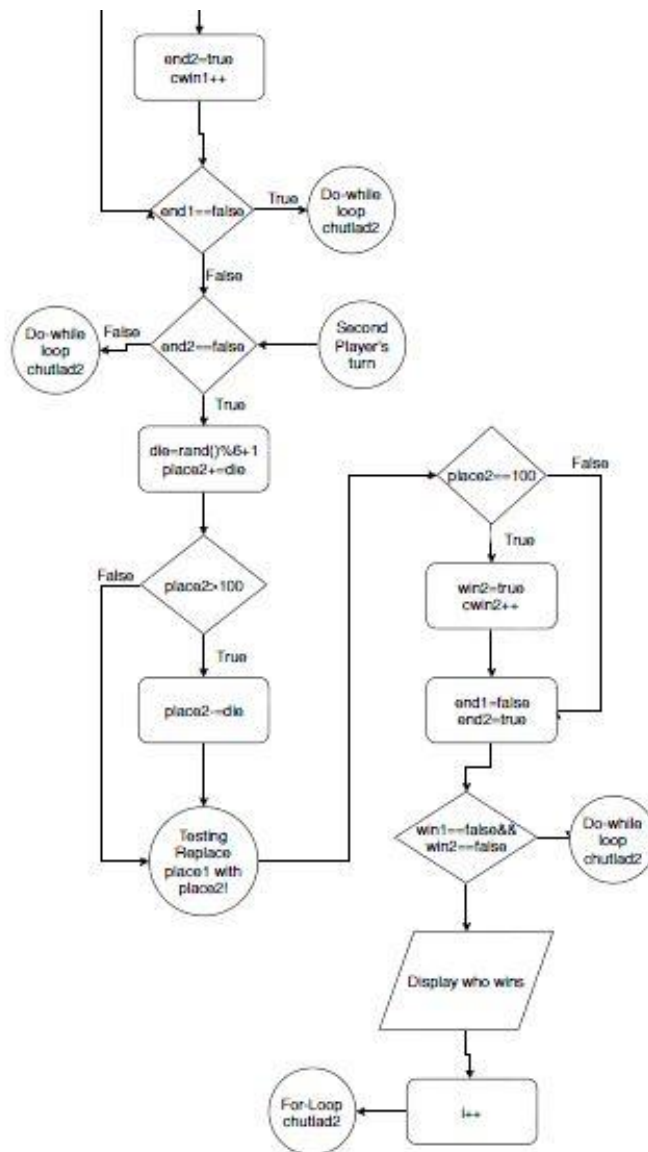


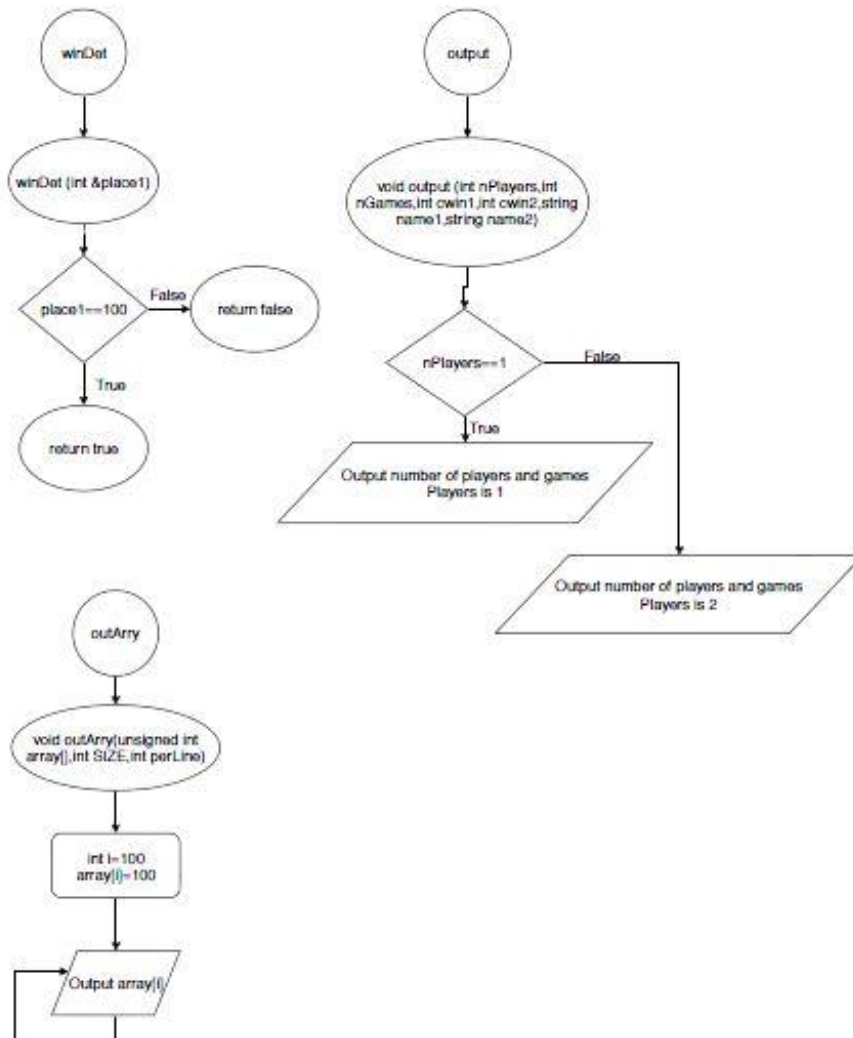


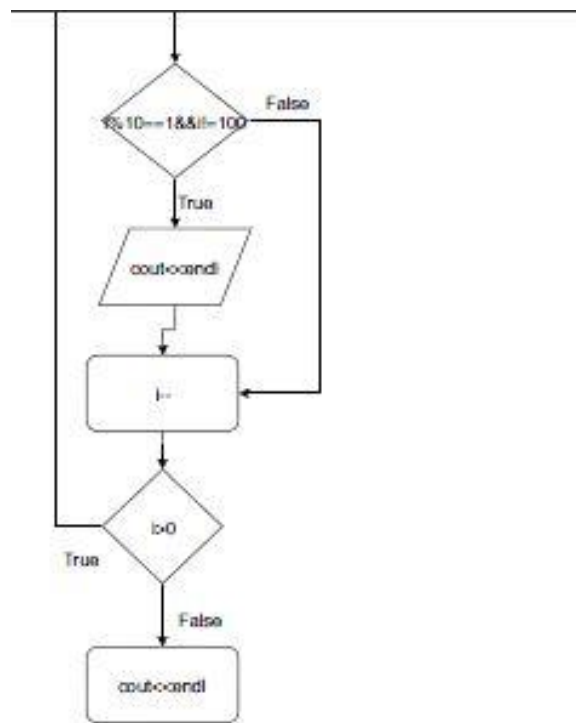












## Screenshots

```

A's number: 81
A's number: 83
A's number: 84
A's number: 86
A's number: 88
A's number: 94
A's number: 96
A's number: 98
A's moved down a chute!
A's number: 78
A's number: 82
A's number: 87
A's moved down a chute!
A's number: 24
A's number: 28
A's moved up a ladder!
A's number: 84
A's number: 88
A's number: 91
A's number: 96
A's number: 101
You cannot land over 100. Go back!
A's number: 96
A's number: 99
A's number: 103
You cannot land over 100. Go back!
A's number: 99
A's number: 100

```

A wins!

```

A          wins = 3
Number of players = 1
Number of games  = 3

```

```

RUN SUCCESSFUL (total time: 20s)
|

```

```

A's number: 81
B's number: 52
A's number: 90
B's number: 56
B's moved down a chute!
B's number: 53
A's number: 96
B's number: 54
A's number: 100

```

A wins!

```

A          wins = 4
B          wins = 1
Number of players = 2
Number of games  = 5

```

```

RUN SUCCESSFUL (total time: 5s)

```

```
Chutes and Ladders Game
The goal is to reach number 100
You can have up to two players
Enter the number of players: 3
You cannot have more than two players
Enter the number of players: 1
Enter the number of games you want to play: 1
Enter the First Player's name: A
```

```
A's number: 6
A's number: 12
A's number: 17
A's number: 22
A's number: 23
A's number: 28
A's moved up a ladder!
A's number: 84
A's number: 90
A's number: 95
```

---

```
Chutes and Ladders Game
The goal is to reach number 100
You can have up to two players
Enter the number of players: 2
Enter the number of games you want to play: 5
Enter the First Player's name: A
Enter the Second Player's name: B
```

```
A's number: 2
B's number: 2
A's number: 5
B's number: 5
A's number: 7
B's number: 10
A's number: 8
B's number: 11
A's number: 9
A's moved up a ladder!
A's number: 31
B's number: 12
A's number: 34
```

## **Pseudocode**

```
/*
```

```
* File: main.cpp
```

```
* Author: Nellie Garcia
```

```
* Created on June 3rd, 2019, 1:58 PM
```

```
* Purpose: Chutes and Ladders Project 2 Version 1
```

```
*      Arrays
```

```
*/
```

```
//System Libraries
```

```
//Input/Output Library
```

```
//C Standard Library - Random Number
```

```
//Time Library
```

```
//Format Library
```

```
//Math Library
```

```
//File Library
```

```
//String Library
```

```
//Namespace Standard
```

```
//User Libraries
```

```
//Global Constants, no Global Variables are allowed
```

```
//Math/Physics/Conversions/Higher Dimensions - i.e. PI, e, etc...
```

```
//Function Prototypes
```

```
//Overloading
```

```
//Output results function
```

```
//Execution Begins Here!
```

```
    //Set the random number seed
```

```
    //Declare Variables
```

```
    //Size of array
```

```
    //Die input
```

```
    //Number of players
```

```
    //Place for player 1
```

```
    //Place for player 2
```

```
    //Counting first player's wins
```

```
    //Counting second player's wins
```

```
//Used for defaulted argument
//Number of games
//Win/Lose for player 1
//Win/Lose for player 2
//End first player's turn
//End second player's turn
//Name of first player
//Name of second player
//Input file
//Output file

//Initialize or input i.e. set variable values
//Function for determining nPlayer
//Array for board
//Set number of games
//Round to integer
//Open input file
//Open output file

//Map inputs -> outputs
    //One player

    //Two players

//Output results to file

//Output results

//Exit stage right or left!

//Start of game - Inputting nPlayer
```

```
//Static variable inside function

//Set number of players

//Validating user input

//Used for defaulted argument


//One player Chutes and Ladders Game

//For loop for number of games


//Reset place and wins


//Begin game
do {
    //Dice roll
    //Place before chutes or ladders
}

//Place cannot go over 100
//Backtrack place1


//Testing chutes and ladders


//Output place1


//Win Determination and Display Win


//Two-player Chutes and Ladders Game

//For loop for number of games


//Reset place, wins, and ends


//Loop turns in order
//First player's turn
```



//Dice roll

//Place before chutes or ladders

//Place cannot go over 100

//Backtrack place1

//Testing chutes and ladders

//Output place1

//Win Determination - Ternary Operator

//End first player's turn

//End of game?

//Second player's turn

//Dice roll

//Place before chutes or ladders

//Place cannot go over 100

//Testing chutes and ladders

//Output place2

//Win Determination

//End second player's turn

//Display who wins

```
//Boolean function to determine win
```

```
//Pass by reference
```

```
//Output results
```

```
//Exit
```

```
//Output an array - Similar to a board
```