

PROJECT TITLE: SNAKES AND LADDER GAME

Team Members:

Sl. No	Name	Roll No.
1	Samriddha Chattopadhyay	200020040
2	Naram Keshav	200020029
3	Samir Bhauraoji Giripunje	200030049
4	Bondugula Pranav	200020010

Problem Statement:

Write a C program to implement the snake and ladder game described below.

- Consider a grid of 10x10.
- The starting point is number '1' and winning point is number '100'.
- The players must throw a dice (in this case enter a number between 1-6).
Based on the number entered the respective player's pawn is either moved forward or backward based on whether a ladder is present or a snake is present.
- A file snakes.txt should hold information on the number of snakes present, length of snake and position of snake. For example - 4 small snakes with length 10 units and 2 large snakes with length 15 units can be considered. The

length indicates if the snake head is at number 26 then tail would be at 16 for a small length snake.

- e) A file ladder.txt should hold information on the number of ladders present, length of ladder and position of ladder. For example - 3 small ladders are present in the numbers 5, 33, 89. Small ladders are of the length 10 units. i.e if the base of the ladder is at 5 then its tip will be at 15. 2 large ladders are present in numbers 29 and 74. Large ladders are of the length 20 units. i.e if the base of the ladder is at 29 then its tip will be at 49.
- f) When the player reaches number 100, the winner must be declared.

Proposed Steps to solve the problem:

1. The main code will have definition and declaration of functions. Each function will implement the features mentioned in the problem statement. A welcome screen will be displayed at the beginning of the game, where the number of players and player names will be taken as input.
2. A structure will be used that will contain an integer length and number and an array position, which will be used to store the length, position and number of different types of snakes and ladders (i.e., one structure variable for small snake, one for long snake, one for small ladder and one for long ladder)
3. A structure for players to store their names and position as input (positions are initialised to 0).
4. Function 1 to take input from the text files and provide the appropriate data for snakes and ladders from the text files.
5. Function 2 to handle the usual game methods, i.e., moving forward when a number is inputted or a ladder is encountered, and moving back when a snake is encountered.
6. Function 3 for handling corner cases like skipping a chance when player seems to move more than 100 or when more than one six comes up or when three sixes come up, etc.

Work Plan:

1. Structures and i/o from files by 26th May
2. Function 2 and Function 3 by 28th May
3. Complete code by 2nd June

Role of team members:

Student Name	Role
Samriddha Chattopadhyay	Main function, Report
Naram Keshav	Structures and i/o functions
Samir Bhauraoji Giripunje	Function 2
Bondugula Pranav	Function 3