

PRACTICE ASSIGNMENT #5: BOARD GAME

Snakes and Ladders is a common board game Lucy has played with her family. She wants to develop a modern version of the game where moves are made automatically and the snakes and ladders are hidden from view.

The game involves players moving around a 6×6 grid from square 1 to square 36. Players move by throwing two six-sided dice and move that many spaces.

If a player lands on a square at the bottom of a ladder, the player will automatically move to the square at the top of the ladder. If a player lands on a square with the head of a snake, the player will automatically move to the square at the tail of the snake.



In order to win, a player must reach square 36, but does not need to land exactly on the square; for example, a player on square 34 who rolls a 4 will still be the winner.

Analyse the scenario requirements and develop a program that will do the following:

1. Allow two players to take turns to play the game.
2. Display the result of two six-sided dice rolls and move the players around the board automatically.
3. After each turn, the board must display where both player pieces are on the board.
4. The game must load the following messages from an external file when the game starts and display them when these conditions are met:
 - a. If a player lands on the foot of a ladder, the game must display a message that a ladder has moved the player to a new position.
 - b. If a player lands on the head of a snake, the game must display a message that a snake has moved the player to a new position.
 - c. When a player has won the game, the game must display a congratulations message and the winning score.
5. The locations for the snakes and ladders should be stored in an external file and read into the game when it starts.

Snakes: 12:2, 14:11, 17:4, 31:19, 35:22

Ladders: 3:16, 5:7, 15:25, 18:20, 21:32

In the final solution, you should demonstrate the use of:

- Data validation
- Random dice rolls
- File input to load the locations of snakes and ladders
- Structured code
- Error handling routines

Source files are provided for this exercise; download them from <http://zzed.uk/10583-PA5>