Mini-project set $n°2$: **Linked Lists**

1 Snake Game

Objective

Create a console-based Snake game using linked lists. The game will use a linked list to represent the snake's body and allow players to control the snake's movement, grow in length, and detect collisions with the walls or itself.

Background

The Snake game is a classic game where the player controls a snake that moves around the screen, consuming food to grow longer. The player must avoid collisions with walls or the snake's own body. The snake is represented by a sequence of connected body segments, and the game will implement this using a linked list where each node represents a segment of the snake.

Tasks

1. **Move Snake:** Implement a function to move the snake. The snake's head will move based on user input, and the body will follow the head.
2. **Grow Snake:** Implement a function to grow the snake when it eats food. This will involve adding a new node at the front of the snake's list.
3. **Check for Collisions:** Implement a function to check for collisions with walls or the snake's own body. The game should end if the snake collides with itself or the wall.
4. **Display the Game Board:** Implement a function to display the current state of the game board, showing the snake, food, and empty spaces.
5. **Generate Food:** Implement a function to randomly generate food on the game board.
6. **Game Loop:** Implement the main game loop that continuously moves the snake, updates the game board, and checks for user input.

2 Student Grading Management System

Objective

Design a system to manage student grades in various subjects using a list of lists. This system will store data for multiple students, track their grades across multiple subjects, and calculate averages. You will implement basic functionalities like adding students, updating grades, and generating reports.

Data Representation

The student data will be stored as a list of lists where each element represents one student and contains:

- The student's name (string),
- A list of grades for their subjects (list of floats).

Tasks

1. **Add Student:** Implement a function to add a new student along with their grades in various subjects.
2. **Update Student Grades:** Implement a function to update the grade for a particular subject of a student.
3. **Calculate Average Grade:** Implement a function that calculates the average grade for a student across all subjects they are enrolled in.
4. **Generate Student Report:** Implement a function that generates a report for a student, displaying their name, grades for each subject, and their overall average grade.
5. **Display All Students:** Implement a function to display a report for all students, showing their name, individual grades, and average grade.