Start Program

function connectDatabase():

try:

- Open a connection to the 'students system.db' database
- Get a cursor for the database connection
- Execute queries to create 'students' and 'lessons' tables if they don't exist
- Join the two tables wuth each others
- Commit changes to the database
- Return the database connection and cursor

except DatabaseError as e:

- Print an error message
- Exit the program with an error code

function doesStudentExist(cursor, studentId):

- Execute a query to check if a student with the given ID exists in the database
- Return True if a student is found, else return False

function getStudentInput(cursor):

- Loop until valid student ID is entered:
  - Prompt the user to enter a student ID
  - Check if the student ID already exists
  - Break the loop if the ID is unique, otherwise prompt again

- Loop until valid name is entered:
  - Prompt the user to enter a name

    - Check if the name contains only letters without spaces
    - Break the loop if valid, otherwise prompt again

- Loop until valid last name is entered:
    - Prompt the user to enter a last name
    - Check if the last name contains only letters
    - Break the loop if valid, otherwise prompt again

- Loop until valid age is entered:
    - Prompt the user to enter an age
    - Break the loop if a valid numeric value is entered, otherwise prompt again

- Loop until valid school year is entered:
    - Prompt the user to enter a school year
    - Break the loop if a valid numeric value is entered, otherwise prompt again

- Loop until valid registration date is entered:
    - Prompt the user to enter a registration date in the format '1/1/2024'
    - Check if the input matches the date format
    - Break the loop if valid, otherwise prompt again

- Loop until valid lesson name is entered:
    - Prompt the user to enter lessons separated by commas
    - Check if the input contains only letters
    - Break the loop if valid, otherwise prompt again

- Return the gathered student information

function getStudentInputForUpdate():

- Similar to getStudentInput but without checking for unique student ID

function addStudent(cursor, connection):

- Print a message to add a student
- Call getStudentInput to get student information
- Check if the student ID is unique
- Insert the student and lessons data into the database
- Print a success message
- Commit changes to the database

function deleteStudent(cursor, connection):

- Prompt the user to enter a student ID to delete
- Check if the student exists
- Delete the student and related lessons from the database
- Print a success message
- Commit changes to the database

function updateStudent(cursor, connection):

- Prompt the user to enter a student ID to update
- Check if the student exists
- Call getStudentInputForUpdate to get updated information
- Update the student and lessons data in the database
- Print a success message
- Commit changes to the database

function showStudent(cursor):

- Prompt the user to enter a student ID to display
- Check if the student exists
- Retrieve and display the student information from the database

function main():

- Call connectDatabase to establish a connection
- Loop until the user chooses to exit:
  - Display a menu for operations (add, delete, update, show, exit)
  - Call the corresponding function based on the user's choice

End Program