Lab 02 Tasks

Task 1: Dynamic Array Manipulation

You are developing a program to manage a dynamic array of integers. The program should:

1. Allow the user to input the size of the array.
2. Dynamically allocate memory for the array.
3. Populate the array with user input.
4. Find and display the sum, average, and maximum value in the array.
5. Deallocate the memory after use.

Task 2: 2D Matrix Operations

You are building a matrix manipulation tool. The program should:

1. Allow the user to input the dimensions of a 2D matrix.
2. Dynamically allocate memory for the matrix.
3. Populate the matrix with user input.
4. Perform matrix addition, subtraction, and multiplication (if possible).
5. Display the results and deallocate memory.

Task 3: Employee Management System

Create a program to manage employee records using structures. The program should:

1. Define a structure Employee with field like **employeeID**, **name**, **department**, and **salary**.
2. Allow the user to input details for multiple employees.
3. Store the employees in a **dynamically allocated array**.
4. Display all employees and allow searching by **employeeID**.
5. **Deallocate** memory after use.

Task 4: Student Grade Calculator

Develop a program to calculate the grades of students using structures. The program should:

1. Define a structure **Student** with fields like **name**, **rollNumber**, and an array to store marks in 5 subjects.
2. Allow the user to input details for multiple students.
3. Calculate the average marks and grade for each student.
4. Display the results and store them in a **dynamically allocated array**.
5. **Deallocate** memory after use.

Task 5: Dynamic String Manipulation

Create a program to manipulate strings dynamically. The program should:

1. Allow the user to input two strings.
2. Dynamically allocate memory for the strings.
3. Perform operations like concatenation, comparison, and finding the length.
4. Display the results and deallocate memory.

Task 6: Library Management System

Build a library management system using structures. The program should:

1. Define a structure **Book** with fields like **bookID**, **title**, **author**, and **isAvailable**.
2. Allow the user to add, remove, and search for books.
3. Store the books in a **dynamically allocated array**.
4. Display all available books and **deallocate** memory after use.

Task 7: Pointers and Functions

Create a program to demonstrate the use of pointers with functions. The program should:

1. Define a function to swap two integers using pointers.
2. Define a function to reverse an array using pointers.
3. Allow the user to input an array and demonstrate both functions.
4. Display the results and deallocate memory.

Task 8: Real-World Use Case - Inventory Management

Develop an inventory management system using structures. The program should:

1. Define a structure **Product** with fields like **productID**, **name**, **quantity**, and **price**.
2. Allow the user to add, remove, and update products.
3. Store the products in a **dynamically allocated array**.
4. Display the total value of the inventory (**sum of price * quantity** for all products).
5. **Deallocate** memory after use.

Task 9: Dynamic Memory Allocation for Records

Create a program to manage records of students using dynamic memory allocation. The program should:

1. Define a structure **Student** with fields like **name**, **rollNumber**, and **marks**.
2. Allow the user to input the number of students and their details.
3. **Dynamically allocate** memory for the students' records.
4. Display the student with the highest marks.
5. **Deallocate** memory after use.

Task 10: Matrix Transpose Using Pointers

Write a program to find the transpose of a matrix using pointers. The program should:

1. Allow the user to input the dimensions of a matrix.
2. **Dynamically allocate** memory for the matrix.
3. Populate the matrix with user input.
4. Compute and display the transpose of the matrix using pointer arithmetic.
5. **Deallocate** memory after use.