Lab 02 Tasks

Task 1: Dynamic Array Structures

Create a program to manage a dynamic array of structures representing books in a library. Each book has a **title**, **author**, and **year**. The program should:

1. Allow the user to input the number of books.
2. **Dynamically allocate** memory for the books.
3. Populate the array with user input.
4. Display all books published after a specific year.
5. **Deallocate memory** after use.

Task 2: Pointer to Pointer for Matrix Operations

Write a program to perform matrix operations (addition and subtraction) using pointer-to-pointer (double pointer) for dynamic memory allocation. The program should:

1. Allow the user to input the dimensions of two matrices.
2. **Dynamically allocate** memory for the matrices.
3. Perform addition and subtraction.
4. Display the results and **deallocate memory**.

Task 3: Employee Salary Calculator

Develop a program to calculate the total salary of employees using structures. Each employee has a **name**, **hoursWorked**, and **hourlyRate**. The program should:

1. Allow the user to input details for multiple employees.
2. **Dynamically allocate** memory for the employees.
3. Calculate the total salary for each employee (**hoursWorked * hourlyRate**).
4. Display the results and **deallocate memory**.

Task 4: Dynamic String Array

Create a program to manage a dynamic array of strings. The program should:

1. Allow the user to input the number of strings.
2. **Dynamically allocate** memory for the strings.
3. Populate the array with user input.
4. Sort the strings alphabetically and display them.
5. **Deallocate** memory after use.

Task 5: Pointers and Arrays

Write a program to demonstrate the relationship between pointers and arrays. The program should:

1. Allow the user to input an array of integers.
2. Use pointer arithmetic to traverse and display the array.
3. Find the sum of the array elements using pointers.
4. Display the results and **deallocate** memory.

Task 6: Dynamic Memory Allocation for Student Records

Develop a program to manage student records using dynamic memory allocation. Each student has a **name**, **rollNumber**, and **marks** in 3 subjects. The program should:

1. Allow the user to input details for multiple students.
2. **Dynamically allocate** memory for the students.
3. Calculate the average marks for each student.
4. Display the results and **deallocate** memory.

Task 7: Matrix Multiplication Using Pointers

Write a program to perform matrix multiplication using pointers. The program should:

1. Allow the user to input the dimensions of two matrices.
2. **Dynamically allocate** memory for the matrices.
3. Perform matrix multiplication.
4. Display the result and **deallocate** memory.

Task 8: Dynamic Array of Pointers

Create a program to manage a dynamic array of pointers to integers. The program should:

1. Allow the user to input the size of the array.
2. **Dynamically allocate** memory for the array and its elements.
3. Populate the array with user input.
4. Display the array and **deallocate** memory.

Task 9: Real-World Use Case - Product Inventory

Develop a program to manage a product inventory using structures. Each product has a **productID**, **name**, **quantity**, and **price**. The program should:

1. Allow the user to input details for multiple products.
2. **Dynamically allocate** memory for the products.
3. Display the total value of the inventory (**sum of quantity * price**).
4. **Deallocate** memory after use.

Task 10: Dynamic Memory Allocation for 3D Arrays

Write a program to manage a 3D array using dynamic memory allocation. The program should:

1. Allow the user to input the dimensions of the 3D array.
2. **Dynamically allocate** memory for the array.
3. Populate the array with user input.
4. Display the array and **deallocate** memory.