**Lab Exercises:**

1. You are tasked with implementing a dynamic matrix class in C++ that supports the following operations: Dynamic Matrix Creation: Create a dynamic 2D array (matrix) with rows and columns specified by the user. Matrix Resizing: Implement a method to resize the matrix. The new size should be provided as input (new rows and columns). If the new size is larger, initialize the new elements with a given value. If the new size is smaller, truncate the matrix. Matrix Transposition: Implement a method to transpose the matrix (rows become columns and vice versa). Matrix Printing: Implement a method to print the matrix. After add 2 to each odd index then print the array. Memory Deallocation: Ensure proper memory management, including deallocation of the dynamic matrix when no longer needed.

1. Implement Jagged arrays from the above pseudocode by taking a size of 5 and resizing it to 10 in each index.
2. Create a header file called matrix\_multuply.h that takes two arrays as input and multiplies them and outputs a multiplied array.

[HINT: Use 2D arrays to accomplish this]

1. Write a program that creates a 2D array of 5x5 values of type Boolean. Suppose indices represent people and the value at row i, column j of a 2D array is true just in case i and j are friends and false otherwise. You can use initializer list to instantiate and initialize your array to represent the following configuration: (\* means “friends”)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i/j | 0 | 1 | 2 | 3 | 4 |
| 0 |  | \* |  | \* | \* |
| 1 | \* |  | \* |  | \* |
| 2 |  | \* |  |  |  |
| 3 | \* |  |  |  | \* |
| 4 | \* | \* |  | \* |  |

Write a method to check whether two people have a common friend. For example, in the example above, 0 and 4 are both friends with 3 (so they have a common friend), whereas 1 and 2 have no common friends.

1. You are tasked with developing a program to manage and display the Grade Point Average (GPA) for the core courses offered in the first semester of four departments: Software Engineering (SE), Artificial Intelligence (AI), Computer Science (CS), and Data Science (DS). Each department offers a distinct number of core courses for this semester: SE has 3 core courses, AI has 4 core courses, CS has 2 core courses, and DS has 1 core course. To efficiently store and present this data, which type of array structure would you employ? implement a solution using the chosen array structure to display the GPAs of the core courses for each department.
2. You're developing a program to manage a seating chart for a conference held in a hall with multiple rows of seats. Each row has a different seat capacity. To efficiently handle the seating arrangements, you decide to use a dynamic array. Implement a C++ code that allocates memory for the seating chart and allows attendees' names to be inputted for each seat. Choose and implement the appropriate type of dynamic array for this scenario.