Makara Ramoabi

20240045

C++

1. Scenario Analysis: 2D Array in an IT System

A 2D array can be used in an IT system to store a school timetable. In this scenario, rows represent days of the week (Monday to Friday), and columns represent time slots or periods during the day. Each cell in the 2D array contains information about the subject taught, the teacher, or the classroom assigned for that specific day and time. For example, timetable[2][3] could represent the class scheduled on Wednesday during the fourth period. Using a 2D array allows the system to organize and access timetable data efficiently. Administrators can easily update schedules, detect conflicts, and display timetables for students and teachers. This structure mirrors real-world scheduling and makes it easier for the system to manage and retrieve structured data accurately.

2. Concept Research: Array Bounds Checking

Array bounds checking is the process of ensuring that a program only accesses valid indexes within an array's defined size. Its importance lies in preventing errors such as accessing memory outside the array, which can cause crashes, incorrect results, or security vulnerabilities. In languages like C++, failing to check array bounds may lead to undefined behavior. Proper bounds checking improves program stability, reliability, and safety. It helps developers catch logical errors early and protects systems from data corruption or malicious exploits, especially in critical IT systems.

3. Tool Practice: 2D Array Program Reflection

Writing a 2D array program in Visual Studio Code helped me understand how data can be stored in rows and columns. I used nested loops to input and display values from the array. This practice showed how 2D arrays are useful for handling structured data such as tables and matrices. Visual Studio Code made it easy to write, run, and debug the program using syntax highlighting and error messages. The exercise improved my understanding of memory organization and how indexes work together in multidimensional arrays within real IT applications.

4. Application Practice: 1D Array Processing Function

A function designed to process a 1D array can be used to calculate the average of student marks in an IT system. The function receives an array of numbers and its size as parameters. It loops through the array, adds all values, and divides the total by the number of elements. The purpose of this function is to automate calculations, reduce errors, and improve efficiency. Such a function can be reused for different datasets, making the system more modular, maintainable, and reliable.