NAME: NATHAN SIKALO
REG NO: SCT212-0189/2022
UNIT NAME: DATA STRUCTURES AND ALGORITHM

LAB [1] ASSIGNMENT

TASK 1

For task one, you need to define two functions that take an array of integers as a parameter and return either the sum or the maximum of the elements. You can use a loop to iterate over the array and update a variable that stores the current sum or maximum. You also need to declare an array of a given size and fill it with user input. You can use the cin and cout streams to read and write data from the console . Here is a possible code for this task:

```
#include <iostream>

// Function to calculate summation of array elements
int summation(int arr[], int n) {
    int sum = 0;
    for (int i = 0; i < n; ++i) {
        sum += arr[i];
    }
    return sum;
}

// Function to find the maximum element in the array
int maximum(int arr[], int n) {
    int max = arr[0];
    for (int i = 1; i < n; ++i) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    return max;
}

int main() {
    int n;

    // Get the size of the array from the user
    std::cout << "Enter the size of the array: ";
    std::cin >> n;

    // Declare an array of size n
    int arr[n];

    // Input array elements from the user
    std::cout << "Enter " << n << " integers:" << std::endl;
    for (int i = 0; i < n; ++i) {
        std::cin >> arr[i];
    }
    // Call summation and maximum functions
    int sum = summation(arr, n);
    int max = maximum(arr, n);

    // Display the results
    std::cout << "Summation: " << sum << std::endl;
    std::cout << "Maximum: " << max << std::endl;

    return 0;
}
```

Task 2

For task two, you need to use arrays and structures to store and manipulate the data of students, courses, and grades.

A structure is a user-defined data type that can hold different types of variables as its members.

You can use the struct keyword to define a structure. You can also use the dot operator (.) to access the members of a structure.

You can declare an array of structures to store multiple instances of the same type.

You can also use the const keyword to make a variable or a member immutable, meaning that it cannot be changed after initialization. Here is a possible code for this task:

```
           #include <iostream>
#include <string>

// Define the Course struct
struct Course {
    std::string course_code;
    std::string course_name;
};

// Define the Grade struct
struct Grade {
    int mark;
    char the_grade;
};

// Define the Student struct
struct Student {
    std::string registration_number;
    std::string name;
    int age;
    Course course;
    Grade grades;
};

int main() {
    const int max_students = 40;
    Student students[max_students];
    // Add at most 40 students
    // (You can add more functionality for user input or read from a file)

    // Edit a student's details
    students[0].name = "New Name";
    students[0].age = 21;

    // Add marks and calculate grades
    students[0].grades.mark = 75;
    if (students[0].grades.mark > 69) {
        students[0].grades.the_grade = 'A';
    } else if (students[0].grades.mark > 59) {
        students[0].grades.the_grade = 'B';
    } else if (students[0].grades.mark > 49) {
        students[0].grades.the_grade = 'C';
    } else if (students[0].grades.mark > 39) {
        students[0].grades.the_grade = 'D';
    } else {
        students[0].grades.the_grade = 'E';
    }

    // Ensure grades, once calculated, cannot be altered

    return 0;
}
```

Task 3

For task three the C++ code defines three classes: `Course`, `Grade`, and `Student`. Each class encapsulates relevant attributes for representing information about a course, grade, and student, respectively. The `main` function initializes an array of `Student` objects to store information for up to 40 students. The code is intended to be extended to implement functionalities related to managing and manipulating student information, such as adding, editing, and displaying details for individual students.

```
#include <iostream>
#include <string>

class Course {
public:
    std::string course_code;
    std::string course_name;
};

class Grade {
public:
    int mark;
    char the_grade;
};
```

```
class Student {
public:
    std::string registration_number;
    std::string name;
    int age;
    Course course;
    Grade grades;
};

int main() {
    const int max_students = 40;
    Student students[max_students];

    // Implement the required functionalities using the Student class

    return 0;
}
```

Task 4

Task four goes with sql

```
+---------------------+
|         List        |
+---------------------+
| - size: int         |
| + insert(item)      |
| + remove(item)      |
| + search(item)      |
| + isEmpty(): bool   |
+---------------------+
```