

Lab 7

Question 1

Define a **struct** named `Date` with members `day`, `month`, and `year`. Then, create another struct named `Person` with:

- `name (char[10])`
- `dob (of type Date)`
- A function `display()` inside the struct to print the details.

Write a program that takes input for a `Person`, including their date of birth, and displays the details using the `display()` function.

Question 2 (Use the skeleton file)

Create a struct named `Subject` with:

- `subjectName (string)`
- `marks (int)`

Then, define another struct named `Student`, which contains:

- `name (string)`
- `rollNumber (int)`
- An array of `Subject` objects

Write a program that:

1. Takes input for one student along with their marks in subjects.
2. Computes and displays the total marks for each student.

Question 3

Define a program to store information about students. Each student should have a name, age, `rollNumber` and address. The address should include the street name and zip code. Create a nested structure to represent the address within the structure for a student. Use this structure to store information about a student and display their details.

```
struct Address {  
  
    string street;  
  
    int zipCode;  
  
};
```

```

struct Student {

    string name;

    string rollNumber;

    Address address; // Nested structure

// Add input() and display() functions

};

```

The ExamRoom structure has a seatingArrangement member, which is a dynamically allocated 2D array (matrix) of Student structures.

- Make a function to input Students data inputStudentRecords(int numOfStudents).
- The makeSeatingPlan(int rows, int columns) member function makes the seating arrangement according
- The displaySeatingPlan() member function displays the students name of the matrix.

```

struct ExamRoom {

    int numRows;

    int numCols;

    Student *students

    Student** seatingArrangement;

    inputStudentRecords(int n) {

        // allocate memory for *students

        // Initialize *students by taking user input for “n” number of student

    }

    makeSeatingPlan(int rows, int cols) {

        // set numRows & numColumns

        // Initialize the **seatingArrangement to arrange the students (*students) in ExamRoom
        (seating arrangement) in ascending order on roll Numbers

    }

```

```
displaySeatingPlan(){  
    // display the name of students in **seatingArrangement  
}
```