

## Assessment 2

### 1. Grading System

Specifications:

Variables: Student name, roll number, marks, and grades.

Static & Const: Static variable for total students; const for maximum subjects.

Switch Case: Menu for entering marks, viewing grades, and calculating averages.

Looping Statements: Loop to input and calculate grades.

Pointers: Pointer for marks and grade calculation.

Functions: Separate functions for marks entry and grade calculation.

Arrays: Store marks and grades.

Structures: Structure for student and academic details.

Nested Structures: Nested structures for personal and academic information.

Unions: Union for grade representation.

Nested Unions: Nested union for different grading schemes.

Output Expectations: Display student grades and class averages.

Menu Example:

1. Enter Marks
2. View Grades
3. Calculate Average
4. Exit

Sol: `#include <stdio.h>`

`#define MAX_SUBJECTS 5 // Maximum number of subjects`

`#define MAX_STUDENTS 10 // Maximum number of students`

```
// Structure to store student details
struct Student {
    char name[50];
    int rollNumber;
    int marks[MAX_SUBJECTS];
    char grade[MAX_SUBJECTS];
    float average;
};

// Declare an array to store multiple students
struct Student students[MAX_STUDENTS];
int totalStudents = 0; // Track the number of students

// Function declarations
void enterMarks();
void viewGrades();
void calculateAverage();

int main() {
    int choice;

    // Main menu loop
    while (1) {
        printf("\nGrading System Menu:\n");
        printf("1. Enter Marks\n");
        printf("2. View Grades\n");
        printf("3. Calculate Average\n");
        printf("4. Exit\n");
```

```
printf("Enter your choice: ");
scanf("%d", &choice);

switch (choice) {
    case 1:
        enterMarks();
        break;
    case 2:
        viewGrades();
        break;
    case 3:
        calculateAverage();
        break;
    case 4:
        printf("Exiting the program...\n");
        return 0;
    default:
        printf("Invalid choice! Please try again.\n");
}
}

return 0;
}

// Function to enter marks for a student
void enterMarks() {
```

```
if (totalStudents >= MAX_STUDENTS) {  
    printf("Maximum number of students reached. Cannot add more students.\n");  
    return;  
}
```

```
struct Student newStudent;
```

```
printf("Enter student name: ");  
scanf("%s", newStudent.name);
```

```
printf("Enter roll number: ");  
scanf("%d", &newStudent.rollNumber);
```

```
// Enter marks for each subject  
printf("Enter marks for %d subjects:\n", MAX_SUBJECTS);  
for (int i = 0; i < MAX_SUBJECTS; i++) {  
    printf("Subject %d: ", i + 1);  
    scanf("%d", &newStudent.marks[i]);  
}
```

```
// Store the student data in the array  
students[totalStudents] = newStudent;  
totalStudents++;
```

```
printf("Marks entered successfully for %s (Roll Number: %d)\n",  
newStudent.name, newStudent.rollNumber);
```

```
}
```

```
// Function to view grades for each student
```

```
void viewGrades() {
```

```
    if (totalStudents == 0) {
```

```
        printf("No students data available.\n");
```

```
        return;
```

```
    }
```

```
for (int i = 0; i < totalStudents; i++) {
```

```
    printf("\nStudent Name: %s\n", students[i].name);
```

```
    printf("Roll Number: %d\n", students[i].rollNumber);
```

```
    printf("Grades:\n");
```

```
for (int j = 0; j < MAX_SUBJECTS; j++) {
```

```
    int marks = students[i].marks[j];
```

```
    // Assign grade based on marks
```

```
    if (marks >= 90) {
```

```
        students[i].grade[j] = 'A';
```

```
    } else if (marks >= 80) {
```

```
        students[i].grade[j] = 'B';
```

```
    } else if (marks >= 70) {
```

```
        students[i].grade[j] = 'C';
```

```
    } else if (marks >= 60) {
```

```
        students[i].grade[j] = 'D';
    } else {
        students[i].grade[j] = 'F';
    }

    printf("Subject %d Grade: %c\n", j + 1, students[i].grade[j]);
}
}
```

// Function to calculate the average marks for each student

```
void calculateAverage() {
    if (totalStudents == 0) {
        printf("No students data available.\n");
        return;
    }

    for (int i = 0; i < totalStudents; i++) {
        int totalMarks = 0;

        // Calculate total marks for the student
        for (int j = 0; j < MAX_SUBJECTS; j++) {
            totalMarks += students[i].marks[j];
        }
    }
}
```

```
// Calculate average marks
students[i].average = totalMarks / (float)MAX_SUBJECTS;

// Display average for the student
printf("\nStudent Name: %s\n", students[i].name);
printf("Roll Number: %d\n", students[i].rollNumber);
printf("Average Marks: %.2f\n", students[i].average);
}
}
```

O/p:

Grading System Menu:

1. Enter Marks
2. View Grades
3. Calculate Average
4. Exit

Enter your choice: 1

Enter student name: Likitha

Enter roll number: 69

Enter marks for 5 subjects:

Subject 1: 87

Subject 2: 90

Subject 3: 60

Subject 4: 75

Subject 5: 77

Marks entered successfully for Likitha (Roll Number: 69)

Grading System Menu:

1. Enter Marks
2. View Grades
3. Calculate Average
4. Exit

Enter your choice: 1

Enter student name: John

Enter roll number: 70

Enter marks for 5 subjects:

Subject 1: 56

Subject 2: 89

Subject 3: 67

Subject 4: 89

Subject 5: 87

Marks entered successfully for John (Roll Number: 70)

Grading System Menu:

1. Enter Marks
2. View Grades
3. Calculate Average
4. Exit

Enter your choice: 2

Student Name: Likitha



Roll Number: 69

Grades:

Subject 1 Grade: B

Subject 2 Grade: A

Subject 3 Grade: D

Subject 4 Grade: C

Subject 5 Grade: C

Student Name: John

Roll Number: 70

Grades:

Subject 1 Grade: F

Subject 2 Grade: B

Subject 3 Grade: D

Subject 4 Grade: B

Subject 5 Grade: B

Grading System Menu:

1. Enter Marks

2. View Grades

3. Calculate Average

4. Exit

Enter your choice: 3

Student Name: Likitha

Roll Number: 69

Average Marks: 77.80

Student Name: John

Roll Number: 70

Average Marks: 77.60

Grading System Menu:

1. Enter Marks
2. View Grades
3. Calculate Average
4. Exit

Enter your choice: 4

Exiting the program...