```
/*
Student Database Management
Write a program to create a structure Student with fields: rollNumber, name, and marks (array for marks of 3 subjects). Implement functions to:
Add a new student.
Display details of all students.
Calculate the average marks of a student.

Enhance the program to include the following features:
Dynamic Memory Allocation:
Use malloc to dynamically allocate memory for the student database.
Support adding more students dynamically using realloc.
Recursive Features:
Recursively display all student details.
Recursively calculate the total marks and percentage for each student.
Sorting Students:
Sort students by their totalMarks in descending order.
Display the sorted list.
Highest and Lowest Scorers:
Identify and display the student(s) with the highest and lowest percentages.
Grade Assignment:
Assign grades to students based on their percentage:
A for 90% and above.
B for 75%–89%.
C for 50%–74%.
D for below 50%.
Include the grade as a field in the Student structure.
Search Feature:
Search for a student by their rollNumber or name using recursion.
Display their details if found.
Update Student Details:
```

```c
Allow updating a student's name or marks.
Recalculate their total marks, percentage, and grade automatically.
Menu-Driven System:
Implement a menu for the following actions:
Add a new student.
Display all students.
Sort students by total marks.
Find the highest and lowest scorers.
Search for a student by roll number or name.
Update student details.
Exit.
*/

#include <stdio.h>
#include <stdlib.h>
#include<string.h>
typedef struct
{
    int rollno;
    char name[20];
    float marks[3];
    float totalmark;
    float average;
    char grade;
} student;
student Students[];

void add_student(int);
void calculate_grade(student *);
void display_student(int);
void sort_student(int);
void min_max(int);
void search_student(int, char *, int);
void update_student(int, int);
int main()
{
    int n;

    printf("Enter the number of student");
    scanf("%d", &n);

    while (1)
    {
        int ch;
        printf("Enter the option\n");
        printf("1.Add a new student\n2.Display all students\n3.Sort students by total marks\n4.Find the highest and lowest scorers\n5.Search for a student by roll number or name\n6.Update student details\n7.Exit.");
        scanf("%d", &ch);

        switch (ch)
        {
        case 1:
            add_student(n);
            break;
        case 2:
            display_student(n);
            break;
        case 3:
            sort_student(n);
            break;
        case 4:
            min_max(n);
            break;
        case 5:
        {
            int rollno;
            char name[20];
            printf("Enter roll number (enter -1 to skip): ");
            scanf("%d", &rollno);
            printf("Enter name (enter 'none' to skip): ");
            scanf("%s", name);
            search_student(rollno, name,n);
            break;
        }
        case 6:
        {
            int rollno;
            printf("Enter roll number of the student to update: ");
            scanf("%d", &rollno);
            update_student(rollno,n);
            break;
        }
        case 7:


            printf("Exiting the program. \n");
            exit(0);
        default:
            printf("Invalid choice. Please try again.\n");


            break;
        }
    }
}
```

```
void add_student(int n)
{
    student *ptr = (student *)malloc(n * sizeof(student));
    for (int i = 0; i < n; i++)
    {
        printf("Student Rollno. : ");
        scanf("%d", &ptr->rollno);
        printf("Name : ");
        scanf("%s", ptr->name);
        printf("Enter marks of 3 subject(out of 100) : ");
        for (int j = 0; j < 3; j++)
        {
            scanf("%f", &ptr->marks[j]);
        }
        calculate_grade(ptr);

        Students[i] = *ptr;
        printf("\nDetails added successfully\n");
    }
}
void calculate_grade(student *ptr)
{
    float sum = 0;
    for (int i = 0; i < 3; i++)
    {
        sum = sum + ptr->marks[i];
    }
    ptr->totalmark = sum;
    ptr->average = sum / 3;
    if (ptr->average >= 90 && ptr->average <= 100)
    {
        ptr->grade = 'A';
    }
    else if (ptr->average >= 75 && ptr->average <= 89)
    {
        ptr->grade = 'B';
    }
    else if (ptr->average >= 50 && ptr->average <= 74)
    {
        ptr->grade = 'C';
    }
    else if (ptr->average < 50 && ptr->average > 0)
    {
        ptr->grade = 'D';
    }
    else
    {
        printf("check marks entered");
    }
}

void display_student(int n)
{

    printf("\nStudent Details\n");
    printf("--------------------------\n\n");
    printf("RollNo\tName\tMarks\t\t\tTotal mark\tAverage\t\tGrade\n");
    for (int i = 0; i < n; i++)
    {
        printf("%d\t%s\t", Students[i].rollno, Students[i].name);
        for (int j = 0; j < 3; j++)
        {
            printf("%.2f,", Students[i].marks[j]);
        }
        printf("\t%.2f\t\t%.2f\t\t%c\n\n", Students[i].totalmark, Students[i].average, Students[i].grade);
    }
}

void sort_student(int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (Students[j].average < Students[j + 1].average)
            {
                // Swap the entire structure
                student temp = Students[j];
                Students[j] = Students[j + 1];
                Students[j + 1] = temp;
            }
        }
    }
    printf("Students sorted by average marks in descending order.\n");
    display_student(n); // Display after sorting
}

void min_max(int n)
{
    student max = Students[0];
    student min = Students[0];
    for (int i = 0; i < n; i++)
    {
        if (Students[i].totalmark > max.totalmark)
        {
            max = Students[i];
        }
        if (Students[i].totalmark < min.totalmark)
        {
            min = Students[i];
        }
    }

    printf("Highest score details\n");
    printf("------------------------------\n\n");
    printf("Highest mark = %.2f\n", max.totalmark);
    printf("Name: %s\n", max.name);
    printf("roll number : %d\n", max.rollno);
    printf("Average mark : %.2f\n", max.average);
    printf("Grade : %c\n\n", max.grade);

    printf("Lowest score details\n");
    printf("------------------------------\n\n");
    printf("Lowest mark = %.2f\n", min.totalmark);
    printf("Name: %s\n", min.name);
    printf("roll number : %d\n", min.rollno);
    printf("Average mark : %.2f\n", min.average);
    printf("Grade : %c\n\n", min.grade);
}
void search_student(int rollno, char *name, int n)
{
    int found = 0;
    for (int i = 0; i < n; i++)
    {
        if ((rollno != -1 && Students[i].rollno == rollno) ||
            (strcmp(name, "none") != 0 && strcmp(Students[i].name, name) == 0))
        {
            printf("\nStudent found:\n");
            printf("RollNo: %d, Name: %s, Marks: %.2f, %.2f, %.2f, Total: %.2f, Average: %.2f, Grade: %c\n",
                   Students[i].rollno, Students[i].name,
                   Students[i].marks[0], Students[i].marks[1], Students[i].marks[2],
                   Students[i].totalmark, Students[i].average, Students[i].grade);
            found = 1;
            break;
        }
    }

    if (!found)
    {
        printf("Student not found.\n");
    }
}

void update_student(int rollno, int n)
{
    int found = 0;
    for (int i = 0; i < n; i++)
    {
        if (Students[i].rollno == rollno)
        {
            found = 1;

            printf("\nCurrent details of the student:\n");
            printf("RollNo: %d, Name: %s, Marks: %.2f, %.2f, %.2f, Total: %.2f, Average: %.2f, Grade: %c\n",
                   Students[i].rollno, Students[i].name,
                   Students[i].marks[0], Students[i].marks[1], Students[i].marks[2],
                   Students[i].totalmark, Students[i].average, Students[i].grade);

            printf("\nEnter new name (or type 'same' to keep the current name): ");
            char new_name[20];
            scanf("%s", new_name);
            if (strcmp(new_name, "same") != 0)
            {
               strcpy(Students[i].name, new_name);
            }

            printf("Enter new marks for 3 subjects (or enter -1 to keep the current marks):\n");
            for (int j = 0; j < 3; j++)
            {
               float new_mark;
               printf("Subject %d: ", j + 1);
               scanf("%f", &new_mark);
               if (new_mark != -1)
               {
                   Students[i].marks[j] = new_mark;
               }
            }

            calculate_grade(&Students[i]);
            printf("\nUpdated details of the student:\n");
            printf("RollNo: %d, Name: %s, Marks: %.2f, %.2f, %.2f, Total: %.2f, Average: %.2f, Grade: %c\n",
                   Students[i].rollno, Students[i].name,
                   Students[i].marks[0], Students[i].marks[1], Students[i].marks[2],
                   Students[i].totalmark, Students[i].average, Students[i].grade);
```

```
            break;
        }
    }

    if (!found)
    {
        printf("Student with roll number %d not found.\n", rollno);
    }
}
```

OUTPUT

```
Enter the number of student5
Enter the option
1.Add a new student
2.Display all students
3.Sort students by total marks
4.Find the highest and lowest scorers
5.Search for a student by roll number or name
6.Update student details
7.Exit.1
Student Rollno. : 1
Name : ammu
Enter marks of 3 subject(out of 100) : 60
90
80

Details added successfully
Student Rollno. : 2
Name : aju
Enter marks of 3 subject(out of 100) : 60
80
```

```
60

Details added successfully
Student Rollno. : 3
Name : kuku
Enter marks of 3 subject(out of 100) : 99
98
97

Details added successfully
Student Rollno. : iju
Name : Enter marks of 3 subject(out of 100) : 70
90
90

Details added successfully
Student Rollno. : 5
Name : raju
Enter marks of 3 subject(out of 100) : 45
34
56

Details added successfully
Enter the option
1.Add a new student
2.Display all students
3.Sort students by total marks
4.Find the highest and lowest scorers
5.Search for a student by roll number or name
6.Update student details
7.Exit.2
```

```
Student Details

---------------------------


RollNo  Name    Marks                Total mark     Average       Grade

1       ammu    60.00,90.00,80.00,      230.00        76.67         B


2       aju     60.00,80.00,60.00,      200.00        66.67         C


3       kuku    99.00,98.00,97.00,      294.00        98.00         A


3       iju     70.00,90.00,90.00,      250.00        83.33         B


5       raju    45.00,34.00,56.00,      135.00        45.00         D

Enter the option

1.Add a new student

2.Display all students

3.Sort students by total marks

4.Find the highest and lowest scorers

5.Search for a student by roll number or name

6.Update student details

7.Exit.3

Students sorted by average marks in descending order.


Student Details

---------------------------


RollNo  Name    Marks                Total mark     Average       Grade

3       kuku    99.00,98.00,97.00,      294.00        98.00         A
```

```
3      iju     70.00,90.00,90.00,      250.00         83.33          B


1      ammu    60.00,90.00,80.00,      230.00         76.67          B


2      aju     60.00,80.00,60.00,      200.00         66.67          C


5      raju    45.00,34.00,56.00,      135.00         45.00          D

Enter the option
1.Add a new student
2.Display all students
3.Sort students by total marks
4.Find the highest and lowest scorers
5.Search for a student by roll number or name
6.Update student details
7.Exit.4
Highest score details
------------------------------


Highest mark = 294.00
Name: kuku
roll number : 3
Average mark : 98.00
Grade : A


Lowest score details
------------------------------


Lowest mark = 135.00
Name: raju
roll number : 5
```

```
Average mark : 45.00
Grade : D

Enter the option
1.Add a new student
2.Display all students
3.Sort students by total marks
4.Find the highest and lowest scorers
5.Search for a student by roll number or name
6.Update student details
7.Exit.5
Enter roll number (enter -1 to skip): 9
Enter name (enter 'none' to skip): mk
Student not found.
Enter the option
1.Add a new student
2.Display all students
3.Sort students by total marks
4.Find the highest and lowest scorers
5.Search for a student by roll number or name
6.Update student details
7.Exit.5
Enter roll number (enter -1 to skip): 2
Enter name (enter 'none' to skip): none

Student found:
RollNo: 2, Name: aju, Marks: 60.00, 80.00, 60.00, Total: 200.00, Average: 66.67, Grade: C
Enter the option
1.Add a new student
2.Display all students
3.Sort students by total marks
```

```
4.Find the highest and lowest scorers
5.Search for a student by roll number or name
6.Update student details
7.Exit.5
Enter roll number (enter -1 to skip): -1
Enter name (enter 'none' to skip): ammu

Student found:
RollNo: 1, Name: ammu, Marks: 60.00, 90.00, 80.00, Total: 230.00, Average: 76.67, Grade: B
Enter the option
1.Add a new student
2.Display all students
3.Sort students by total marks
4.Find the highest and lowest scorers
5.Search for a student by roll number or name
6.Update student details
7.Exit.6
Enter roll number of the student to update: 1

Current details of the student:
RollNo: 1, Name: ammu, Marks: 60.00, 90.00, 80.00, Total: 230.00, Average: 76.67, Grade: B

Enter new name (or type 'same' to keep the current name): ammu
Enter new marks for 3 subjects (or enter -1 to keep the current marks):
Subject 1: 100 100 100
Subject 2: Subject 3:
Updated details of the student:
RollNo: 1, Name: ammu, Marks: 100.00, 100.00, 100.00, Total: 300.00, Average: 100.00, Grade: A
Enter the option
1.Add a new student
2.Display all students
```

```
3.Sort students by total marks

4.Find the highest and lowest scorers

5.Search for a student by roll number or name

6.Update student details

7.Exit.2


Student Details

---------------------------


RollNo  Name    Marks                Total mark     Average       Grade

3      kuku    99.00,98.00,97.00,     294.00         98.00         A


3      iju     70.00,90.00,90.00,     250.00         83.33         B


1      ammu    100.00,100.00,100.00,   300.00         100.00        A


2      aju     60.00,80.00,60.00,     200.00         66.67         C


5      raju    45.00,34.00,56.00,     135.00         45.00         D


Enter the option

1.Add a new student

2.Display all students

3.Sort students by total marks

4.Find the highest and lowest scorers

5.Search for a student by roll number or name

6.Update student details

7.Exit.3

Students sorted by average marks in descending order.


Student Details
```

```
---------------------------


RollNo  Name    Marks                 Total mark      Average        Grade

1       ammu    100.00,100.00,100.00,   300.00         100.00         A


3       kuku    99.00,98.00,97.00,      294.00         98.00          A


3       iju     70.00,90.00,90.00,      250.00         83.33          B


2       aju     60.00,80.00,60.00,      200.00         66.67          C


5       raju    45.00,34.00,56.00,      135.00         45.00          D

Enter the option
1.Add a new student
2.Display all students
3.Sort students by total marks
4.Find the highest and lowest scorers
5.Search for a student by roll number or name
6.Update student details
7.Exit.7
Exiting the program.
PS C:\Users\amrit\Desktop\Quest_Cprogramming\Day18_viva2>
```