DSA (CSE102) Lab 3: Sorting Algorithms and File I/O

24-01-25, Time: 2 Hours, Total Points: 100

Instructions

- All code must be written in **C** only.
- There will be no partial marks unless specified.
- You have to submit only the `.c` files. **Do not submit .exe or binary files. Submission of binary files will result in 0 marks in the LAB.**
- Submit all the C files without zipping them.
- You cannot use any **inbuilt function Except for strcmp()** ; any function you want to use, you should implement from scratch.
- You can use any **documentation** for the C programming language: https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html.

Question 1: Student Record Management System

Write a C program to implement a simple Student Record Management System using file I/O. The program should perform the following operations:

1. **Add a new student record to a file. (10 points)**
2. **Display all student records stored in the file. (10 points)**
3. **Search for a student record by roll number. (10 points)**
4. **Update Name Or Marks for a student record by roll number. (10 points)**
5. **Delete record for a student by roll number. (10 points)**
6. **Sort the records by Name using MergeSort or Insertion sorting algorithm taught in class (30 points)**
7. **Sort the records by Marks using MergeSort or Insertion sorting algorithm taught in class (20 points)**

Each student record should contain:

- **Roll Number** (integer)
- **Name** (string)
- **Marks** (float)

File Structure

The records file named `students.txt` is provided, where each record is stored on a new line in the format: `Roll Number Name Marks`

Explanation of Subparts

1. Add a New Student Record: In this subpart, the user will input a student's Roll Number, Name, and Marks. The program will then add this information as a new record into the file `students.txt`.

- The user will be prompted to enter the roll number, name, and marks for the new student.
- The program will write the student data to the file in the format: `Roll Number Name Marks`.

Example Input/Output:

```
Enter your choice: 1
Enter Roll Number: 101
Enter Name: John
Enter Marks: 85.5

Record added successfully.
```

2. Display All Records: This subpart involves displaying all the student records stored in the file `students.txt`.

- The program will read the file and display the records in a tabular format: Roll Number, Name, and Marks.
- It should display all records stored in the file.

Example Input/Output:

```
Enter your choice: 2

All Student Records:
Roll Number    Name    Marks
101            John    85.50
102            Alice   90.00
```

3. Search for a Record: In this subpart, the user can search for a student record by entering the roll number. If the roll number is found, the program will display the corresponding student's details.

- The program will prompt the user to enter the roll number.
- It will search for the roll number in the file and display the record if found.

Example Input/Output:

```
Enter your choice: 3
Enter Roll Number to search: 102

Record Found: Roll Number: 102, Name: Alice, Marks: 90.00
```

4. Update Name or Marks: This subpart allows the user to update the name or marks for a student based on their roll number. The program will search for the roll number in the file and update the details if found.

- The user will enter the roll number of the student whose record needs to be updated.
- The user can then choose to update either the name or the marks.
- The program will overwrite the old record with the updated details.

Example Input/Output:

```
Enter your choice: 4
Enter Roll Number to update: 101
Enter 1 to update Name or 2 to update Marks: 1
Enter new Name: John Doe

Record updated successfully.
```

5. Delete a Record: In this subpart, the user will delete a student's record based on their roll number. The program will search for the roll number and remove the corresponding record from the file.

- The user will enter the roll number of the student whose record needs to be deleted.
- The program will remove that record from the file, shifting the rest of the records as needed.

Example Input/Output:

```
Enter your choice: 5
Enter Roll Number to delete: 102

Record deleted successfully.
```

6. Sort Records by Name: In this subpart, the program will sort the student records by Name in alphabetical order. You have to use MergeSort or Insertion sorting algorithm.

- The program will sort the student records by the Name field in ascending alphabetical order.
- After sorting, the records should be displayed in the sorted order.

Example Input/Output:

```
Enter your choice: 6

All Student Records Sorted by Name:
Roll Number    Name    Marks
101            Alice   90.00
102            John    85.50
```

7. Sort Records by Marks: In this subpart, the program will sort the student records by Marks in descending order. You have to use MergeSort or Insertion sorting algorithm.

- The program will sort the student records by the Marks field in descending order.
- After sorting, the records should be displayed in the sorted order.

Example Input/Output:

```
Enter your choice: 7

All Student Records Sorted by Marks:
Roll Number    Name    Marks
102            Alice   90.00
101            John    85.50
```