

```

using System;
using System.Collections.Generic;
using System.IO;
using static System.Collections.Specialized.BitVector32;
using System.Security.Policy;

namespace ConsoleApp
{
    internal class Program
    {
        struct Student
        {
            public int StudentID;
            public string Name;
            public int Grade;
        }

        // Load student data from the file.
        static List<Student> LoadStudents(string filePath)
        {
            List<Student> students = new List<Student>();
            // Read all lines from the file
            string[] lines = File.ReadAllLines(filePath);
            // Skip the header line (StudentID, Name, Grade)
            for (int i = 1; i < lines.Length; i++)
            {
                string[] data = lines[i].Split(',');
                if (data.Length == 3)
                {
                    int studentID = int.Parse(data[0].Trim());
                    string name = data[1].Trim();
                    int grade = int.Parse(data[2].Trim());
                    Student student = new Student
                    {
                        StudentID = studentID,
                        Name = name,
                        Grade = grade
                    };
                    students.Add(student);
                }
            }
            return students;
        }

        // Bubble Sort to sort students by name
        static void BubbleSort(List<Student> students)
        {
            int n = students.Count;
            for (int i = 0; i < n - 1; i++)
            {
                for (int j = 0; j < n - i - 1; j++)
                {
                    if (string.Compare(students[j].Name, students[j + 1].Name) > 0)
                    {
                        // Swap students[j] and students[j+1]
                        Student temp = students[j];
                        students[j] = students[j + 1];
                        students[j + 1] = temp;
                    }
                }
            }
        }
    }
}

```

```

    }
}
// Display all students
static void DisplayStudents(List<Student> students)
{
    foreach (Student student in students)
    {
        Console.WriteLine($" Student Id: {student.StudentID} \n Name:
{student.Name} \n Grade: {student.Grade} ");
    }
}
// Find a student by name
static Student FindStudentByName(List<Student> students, string searchName)
{
    Student foundStudent = new Student();
    foreach (Student student in students)
    {
        if (student.Name.Equals(searchName,
StringComparison.OrdinalIgnoreCase))
        {
            foundStudent = student;
            break;
        }
    }
    return foundStudent;
}
static void Main(string[] args)
{
    // Path to the student_data.txt file
    string filePath = @"C:\Users\Annem
sucharitha\MPP\StudentFilesSorting\StudentFilesSorting\studentsfilesdata.txt";
    // Check if the file exists
    if (!File.Exists(filePath))
    {
        Console.WriteLine("Error: student_data.txt file not found.");
        return;
    }
    List<Student> students = LoadStudents(filePath);
    // Sort students by name using Bubble Sort..
    BubbleSort(students);
    // Display student data
    Console.WriteLine("Sorted Student Data:");
    DisplayStudents(students);
    bool shouldContinue = true;
    while (shouldContinue)
    {
        Console.WriteLine("Enter the name of the student to search or type
'quit' to exit: ");

        string userInput = Console.ReadLine();
        if (userInput.Equals("quit", StringComparison.OrdinalIgnoreCase))
        {
            shouldContinue = false;
        }
        else
        {
            Student foundStudent = FindStudentByName(students, userInput);

```

```
        if (foundStudent.Name != null)
        {
            Console.WriteLine($"Found: Student ID:
{foundStudent.StudentID} \nName: {foundStudent.Name}Grade: {foundStudent.Grade}");
        }
        else
        {
            Console.WriteLine("Student not found");
        }
        Console.Read();
    }
}
}
```