计算机与信息工程学院实验报告

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

姓名：杨佳森 学号：2112080106\_ 专业：\_数据科学与大数据技术\_年级：\_2021\_\_

课程： 数据结构 主讲教师：\_\_刘成\_\_\_\_\_\_ 辅导教师：\_\_\_无\_\_\_\_

实验时间：2023\_年 \_6\_月 \_6\_日 下午\_17\_时至19\_时，实验地点 计算机大楼213\_

实验题目： 基于学生信息系统的查找排序

实验目的： 两个查找两排序

实验环境（硬件和软件） vs code 和PC

实验内容：

实验步骤：

1、显示学生信息表

void displayStudentInfo(const Student& student) {

    cout << "学生姓名: " << student.name << endl;

    cout << "学生学号: " << student.studentId << endl;

    cout << "学生成绩: ";

    for (int i = 0; i < 5; i++) {

        cout << student.marks[i] << " ";

    }

    cout << endl;

}

2、插入学生信息

void insertStudent(Student students[], int& numStudents) {

    if (numStudents >= 10) {

        cout << "学生信息已满，无法插入更多学生。" << endl;

        return;

    }

    Student newStudent;

    cout << "请输入学生姓名：";

    cin >> newStudent.name;

    cout << "请输入学生学号：";

    cin >> newStudent.studentId;

    cout << "请输入学生成绩（五门，用空格分隔）：";

    for (int i = 0; i < 5; i++) {

        cin >> newStudent.marks[i];

    }

    students[numStudents] = newStudent;

    numStudents++;

    cout << "学生信息插入成功。" << endl;

}

3、修改学生信息

void modifyStudent(Student students[], int numStudents, int targetId) {

    for (int i = 0; i < numStudents; i++) {

        if (students[i].studentId == targetId) {

            cout << "请输入修改后的学生姓名：";

            cin >> students[i].name;

            cout << "请输入修改后的学生成绩（五门，用空格分隔）：";

            for (int j = 0; j < 5; j++) {

                cin >> students[i].marks[j];

            }

            cout << "学生信息修改成功。" << endl;

            return;

        }

    }

    cout << "未找到学号为" << targetId << "的学生，无法修改。" << endl;

}

4、删除学生信息

void deleteStudent(Student students[], int& numStudents, int targetId) {

    for (int i = 0; i < numStudents; i++) {

        if (students[i].studentId == targetId) {

            for (int j = i; j < numStudents - 1; j++) {

                students[j] = students[j + 1];

            }

            numStudents--;

            cout << "学生信息删除成功。" << endl;

            return;

        }

    }

    cout << "未找到学号为" << targetId << "的学生，无法删除。" << endl;

}

5、冒泡排序

void bubbleSort(Student students[], int n) {

    for (int i = 0; i < n - 1; i++) {

        for (int j = 0; j < n - i - 1; j++) {

            if (students[j].studentId > students[j + 1].studentId) {

                swap(students[j], students[j + 1]);

            }

        }

    }

}

6、选择排序

void selectionSort(Student students[], int n) {

    for (int i = 0; i < n - 1; i++) {

        int minIndex = i;

        for (int j = i + 1; j < n; j++) {

            if (students[j].studentId < students[minIndex].studentId) {

                minIndex = j;

            }

        }

        swap(students[i], students[minIndex]);

    }

}

7、线性查找

int linearSearch(const Student students[], int numStudents, int targetId) {

    for (int i = 0; i < numStudents; i++) {

        if (students[i].studentId == targetId) {

            return i;

        }

    }

    return -1;

}

8、二分查找

int binarySearch(const Student students[], int left, int right, int targetId) {

    if (right >= left) {

        int mid = left + (right - left) / 2;

        if (students[mid].studentId == targetId) {

            return mid;

        }

        if (students[mid].studentId > targetId) {

            return binarySearch(students, left, mid - 1, targetId);

        }

        return binarySearch(students, mid + 1, right, targetId);

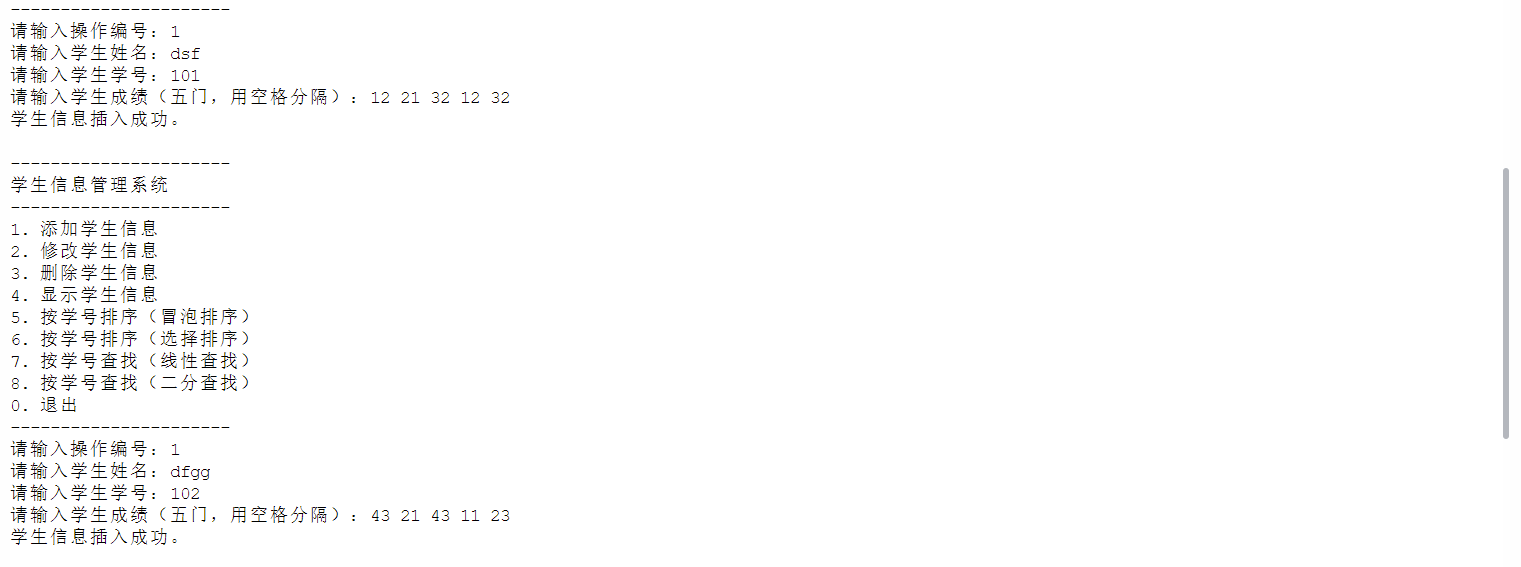
    }

    return -1;

}

实验数据记录：

1、乱序插入三个数据



2、显示排序前的学生信息表



3、冒泡排序



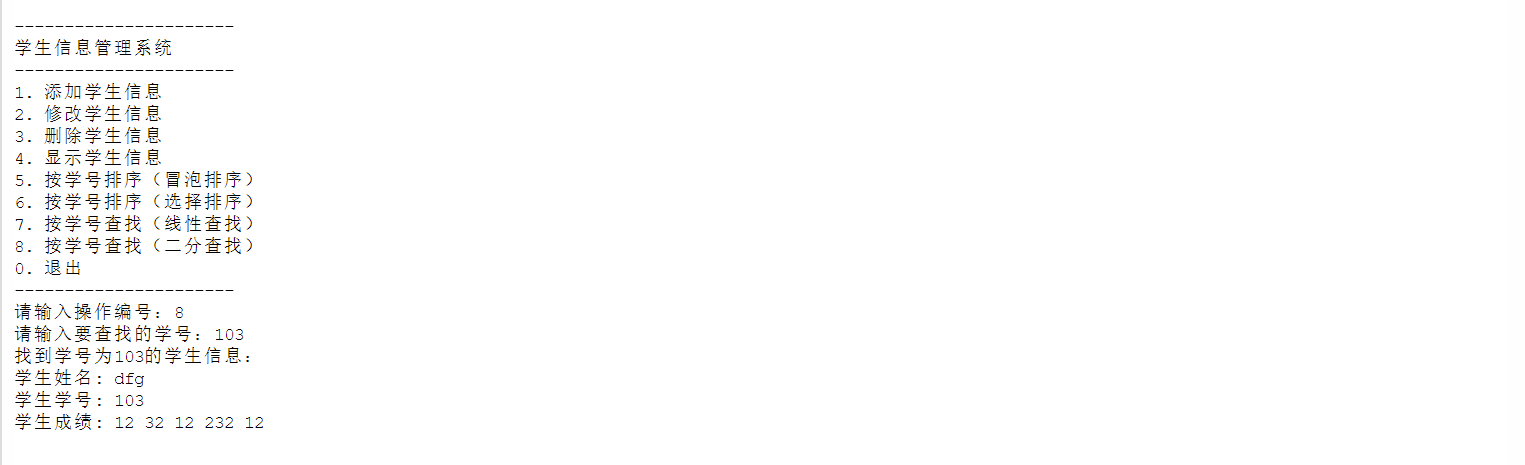
4、选择排序



5、线性查找



6、二分查找



问题讨论：

源代码

#include <iostream>

#include <string>

using namespace std;

struct Student {

    string name;

    int studentId;

    int marks[5];

};

void displayStudentInfo(const Student& student) {

    cout << "学生姓名: " << student.name << endl;

    cout << "学生学号: " << student.studentId << endl;

    cout << "学生成绩: ";

    for (int i = 0; i < 5; i++) {

        cout << student.marks[i] << " ";

    }

    cout << endl;

}

void insertStudent(Student students[], int& numStudents) {

    if (numStudents >= 10) {

        cout << "学生信息已满，无法插入更多学生。" << endl;

        return;

    }

    Student newStudent;

    cout << "请输入学生姓名：";

    cin >> newStudent.name;

    cout << "请输入学生学号：";

    cin >> newStudent.studentId;

    cout << "请输入学生成绩（五门，用空格分隔）：";

    for (int i = 0; i < 5; i++) {

        cin >> newStudent.marks[i];

    }

    students[numStudents] = newStudent;

    numStudents++;

    cout << "学生信息插入成功。" << endl;

}

void modifyStudent(Student students[], int numStudents, int targetId) {

    for (int i = 0; i < numStudents; i++) {

        if (students[i].studentId == targetId) {

            cout << "请输入修改后的学生姓名：";

            cin >> students[i].name;

            cout << "请输入修改后的学生成绩（五门，用空格分隔）：";

            for (int j = 0; j < 5; j++) {

                cin >> students[i].marks[j];

            }

            cout << "学生信息修改成功。" << endl;

            return;

        }

    }

    cout << "未找到学号为" << targetId << "的学生，无法修改。" << endl;

}

void deleteStudent(Student students[], int& numStudents, int targetId) {

    for (int i = 0; i < numStudents; i++) {

        if (students[i].studentId == targetId) {

            for (int j = i; j < numStudents - 1; j++) {

                students[j] = students[j + 1];

            }

            numStudents--;

            cout << "学生信息删除成功。" << endl;

            return;

        }

    }

    cout << "未找到学号为" << targetId << "的学生，无法删除。" << endl;

}

void bubbleSort(Student students[], int n) {

    for (int i = 0; i < n - 1; i++) {

        for (int j = 0; j < n - i - 1; j++) {

            if (students[j].studentId > students[j + 1].studentId) {

                swap(students[j], students[j + 1]);

            }

        }

    }

}

void selectionSort(Student students[], int n) {

    for (int i = 0; i < n - 1; i++) {

        int minIndex = i;

        for (int j = i + 1; j < n; j++) {

            if (students[j].studentId < students[minIndex].studentId) {

                minIndex = j;

            }

        }

        swap(students[i], students[minIndex]);

    }

}

int linearSearch(const Student students[], int numStudents, int targetId) {

    for (int i = 0; i < numStudents; i++) {

        if (students[i].studentId == targetId) {

            return i;

        }

    }

    return -1;

}

int binarySearch(const Student students[], int left, int right, int targetId) {

    if (right >= left) {

        int mid = left + (right - left) / 2;

        if (students[mid].studentId == targetId) {

            return mid;

        }

        if (students[mid].studentId > targetId) {

            return binarySearch(students, left, mid - 1, targetId);

        }

        return binarySearch(students, mid + 1, right, targetId);

    }

    return -1;

}

int main() {

    Student students[10];

    int numStudents = 0;

    while (true) {

        cout << "----------------------" << endl;

        cout << "学生信息管理系统" << endl;

        cout << "----------------------" << endl;

        cout << "1. 添加学生信息" << endl;

        cout << "2. 修改学生信息" << endl;

        cout << "3. 删除学生信息" << endl;

        cout << "4. 显示学生信息" << endl;

        cout << "5. 按学号排序（冒泡排序）" << endl;

        cout << "6. 按学号排序（选择排序）" << endl;

        cout << "7. 按学号查找（线性查找）" << endl;

        cout << "8. 按学号查找（二分查找）" << endl;

        cout << "0. 退出" << endl;

        cout << "----------------------" << endl;

        cout << "请输入操作编号：";

        int choice;

        cin >> choice;

        switch (choice) {

            case 1:

                insertStudent(students, numStudents);

                cout << endl;

                break;

            case 2:

                int modifyTargetId;

                cout << "请输入要修改的学号：";

                cin >> modifyTargetId;

                modifyStudent(students, numStudents, modifyTargetId);

                cout << endl;

                break;

            case 3:

                int deleteTargetId;

                cout << "请输入要删除的学号：";

                cin >> deleteTargetId;

                deleteStudent(students, numStudents, deleteTargetId);

                cout << endl;

                break;

            case 4:

                cout << "学生信息表：" << endl;

                for (int i = 0; i < numStudents; i++) {

                    displayStudentInfo(students[i]);

                }

                cout << endl;

                break;

            case 5:

                bubbleSort(students, numStudents);

                cout << "使用冒泡排序按学号排序后的学生信息：" << endl;

                for (int i = 0; i < numStudents; i++) {

                    displayStudentInfo(students[i]);

                }

                cout << endl;

                break;

            case 6:

                selectionSort(students, numStudents);

                cout << "使用选择排序按学号排序后的学生信息：" << endl;

                for (int i = 0; i < numStudents; i++) {

                    displayStudentInfo(students[i]);

                }

                cout << endl;

                break;

            case 7:

                int linearSearchTargetId;

                cout << "请输入要查找的学号：";

                cin >> linearSearchTargetId;

                {

                    int linearSearchIndex = linearSearch(students, numStudents, linearSearchTargetId);

                    if (linearSearchIndex != -1) {

                        cout << "找到学号为" << linearSearchTargetId << "的学生信息：" << endl;

                        displayStudentInfo(students[linearSearchIndex]);

                    } else {

                        cout << "未找到学号为" << linearSearchTargetId << "的学生。" << endl;

                    }

                }

                cout << endl;

                break;

            case 8:

                int binarySearchTargetId;

                cout << "请输入要查找的学号：";

                cin >> binarySearchTargetId;

                {

                    int binarySearchIndex = binarySearch(students, 0, numStudents - 1, binarySearchTargetId);

                    if (binarySearchIndex != -1) {

                        cout << "找到学号为" << binarySearchTargetId << "的学生信息：" << endl;

                        displayStudentInfo(students[binarySearchIndex]);

                    } else {

                        cout << "未找到学号为" << binarySearchTargetId << "的学生。" << endl;

                    }

                }

                cout << endl;

                break;

            case 0:

                cout << "程序已退出。" << endl;

                return 0;

            default:

                cout << "无效的操作编号，请重新输入。" << endl;

                cout << endl;

                break;

        }

    }

    return 0;

}