数据结构实验报告——实验二

学号：__20201060330__姓名：___胡诚皓___得分：_____

一、实验目的

1. **复习结构体、数组、指针；**
2. **掌握数组的静态创建与动态创建；**
3. **了解顺序存储的基本访问方法。**

二、实验内容

1. **（必做题）学生成绩信息存储**

每个学生的成绩信息包括：学号、语文、数学、英语、总分、加权平均分；采用动态方法创建数组用于存储若干学生的成绩信息；输入学生的学号、语文、数学、英语成绩；计算学生的总分和加权平均分（语文占 30%，数学占 50%，英语占 20%）；输出学生的成绩信息。

2. **（必做题）追加和删除学生成绩信息**

可以在数组末尾追加新学生的成绩信息；可以根据学号，删除该学生的成绩信息。

3. **（选做题）对学生成绩信息进行排序**

可以根据学号或总分，升序排序学生的成绩信息。

三、数据结构及算法描述

1. **学生成绩信息存储**

使用结构体 Stu 来存储学生的成绩相关信息，其中 long long id 用于存储学生的学号，double chinese, math, english 分别用来存储学生的语文成绩、数学成绩、英语成绩，double total 用于存储学生的总分，double average 用于存储学生的加权平均分。

为了避免命令行输入时回车键对输入判断的干扰，先读取将要输入的学生的个数，再读取每个学生的信息，在每次读入一个学生的成绩信息后，立刻计算该学生的总分与加权平均分存入。

2. **追加和删除学生成绩信息**

在追加学生的成绩信息时，直接将新追加的学生信息放在动态数组的最后。由于需要先

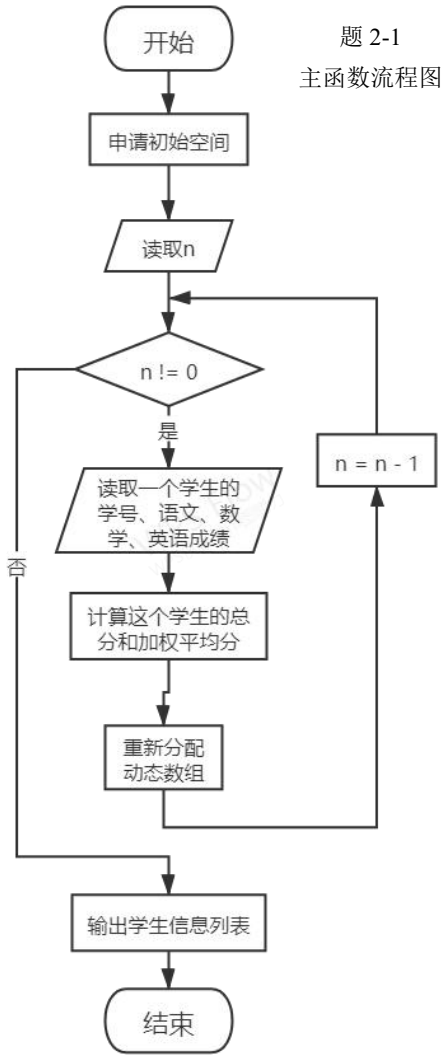
查找到要删除 id 的学生的位置，先用快速排序对 Stu 数组进行升序排序，再通过二分查找找到需要删除的学生的位置，然后再进行删除。若输入的 id 存在，删除该学生后会重新输出学生信息列表；若输入的 id 不存在，会提示“id Not Found!”

3. 对学生成绩信息进行排序

此处根据总分进行升序排序，使用的是优化的快速排序算法。

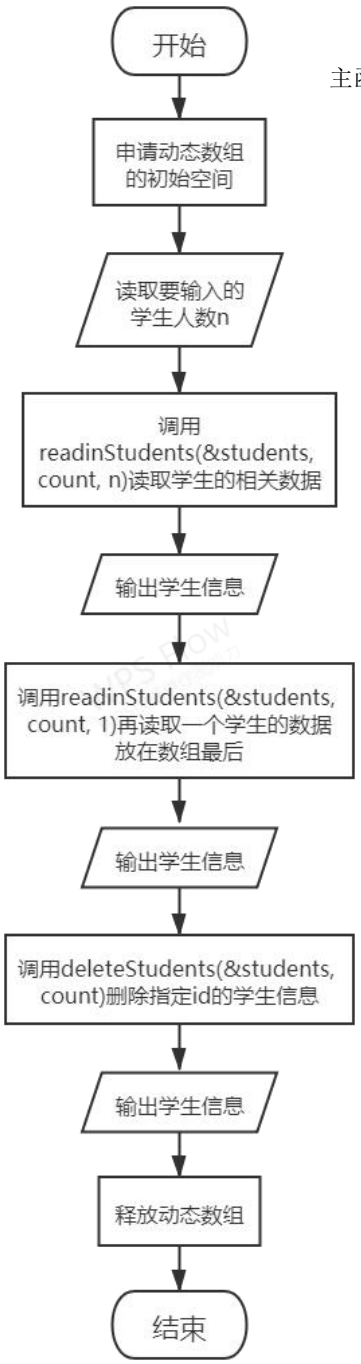
四、详细设计

1. 学生成绩信息存储

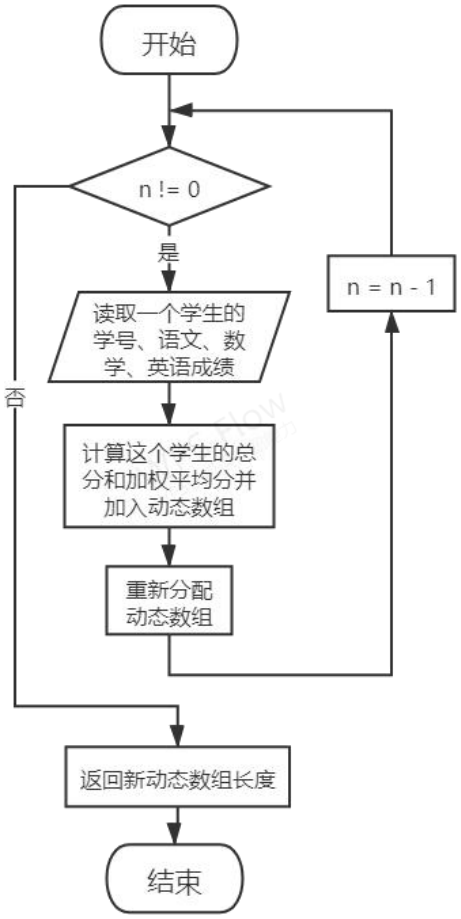


题 2-1
主函数流程图

2. 追加和删除学生成绩信息



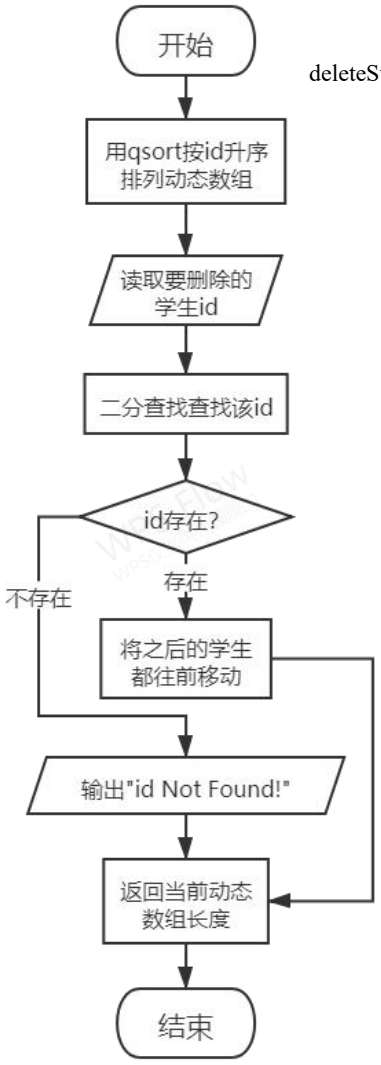
题 2-2
主函数流程图



题 2-2
readinStudents 函数流程图

题 2-2

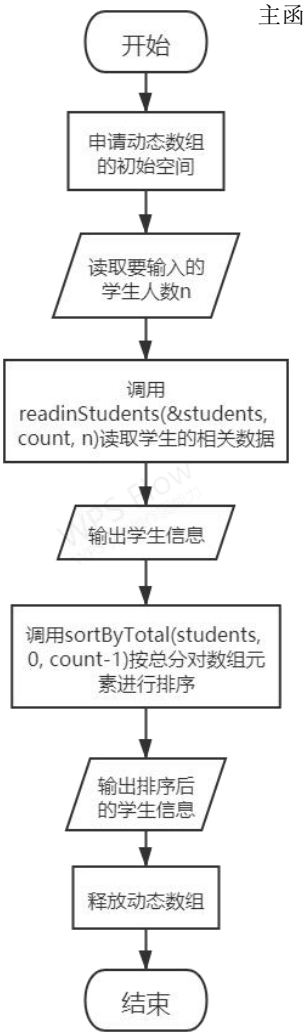
deleteStudents 函数流程图



3. 对学生成绩信息进行排序

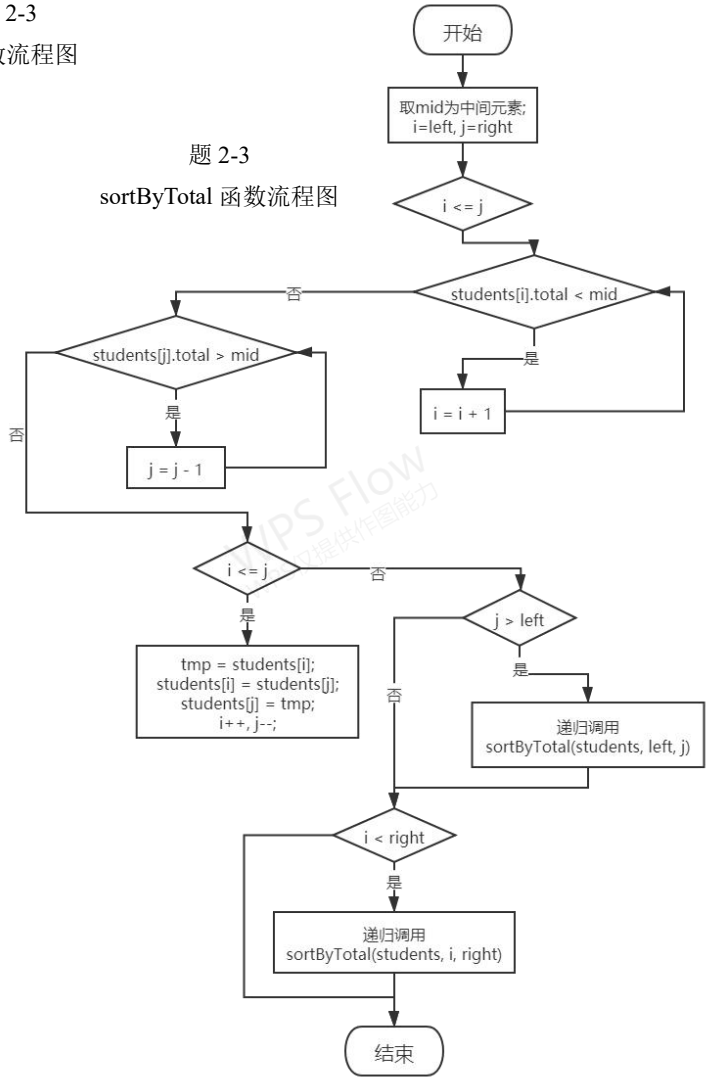
题 2-3

主函数流程图



题 2-3

sortByTotal 函数流程图



五、程序代码

1. 学生成绩信息存储

2-1.c

```
#include <stdio.h>
#include <stdlib.h>

//使用结构体来存储每个学生的学号及成绩信息
typedef struct {
    long long id;
    double chinese, math, english;
    double total, average;
} Stu;
```

```
int main() {
    Stu *students;
    int count=0;
    long long num;
    double chi, math, eng;
    int n;

    //申请动态数组的空间
    students = (Stu *) malloc(sizeof(Stu));


    //先读取将要输入的学生人数
    printf("How many students?\n");
    scanf("%d", &n);

    //读入每个学生的相关信息
    while (n--) {
        scanf("%lld %lf %lf %lf", &num, &chi, &math, &eng);
        (students+count)->id = num;
        (students+count)->chinese = chi;
        (students+count)->math = math;
        (students+count)->english = eng;
        (students+count)->total = chi + math + eng;
        (students+count)->average = 0.3*chi + 0.5*math + 0.2*eng;
        count++;
        //每次读入之后都重新多申请一个 Stu 结构体的空间，准备下次读入
        students = (Stu *) realloc(students, (count+1)*sizeof(Stu));
    }

    //规范格式更美观地输出
    printf("-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|\n");
    for (int i = 0; i < count; i++) {
        printf("%12lld|%11.2f|%8.2f|%11.2f|%9.2f|%7.2f|\n",           students[i].id,
students[i].chinese,   students[i].math,   students[i].english,   students[i].total,
students[i].average);
    }

    //释放动态数组占用的空间
    free(students);

    return 0;
}
```

2. 追加和删除学生成绩信息

2-2.c

```
#include <stdio.h>
#include <stdlib.h>

//使用结构体来存储每个学生的学号及成绩信息
typedef struct {
    long long id;
    double chinese, math, english;
    double total, average;
} Stu;

void displayStudents(Stu *, int);
int readinStudents(Stu **, int, int);
int cmp(const void *, const void *);
int deleteStudents(Stu **, int);

int main() {
    Stu *students;
    int count=0;
    int n;

    //申请动态数组的空间
    students = (Stu *) malloc(sizeof(Stu));

    //先读取将要输入的学生人数
    printf("How many students?\n");
    scanf("%d", &n);

    //使用封装好的函数读取数据
    count = readinStudents(&students, count, n);
    displayStudents(students, count);

    //再读入一个学生的数据
    printf("Add one student at the end of the array\n");
    count = readinStudents(&students, count, 1);
    displayStudents(students, count);

    //使用封装好的函数删除指定 id 学生
    printf("Delete one student by id\n");
    count = deleteStudents(&students, count);
    displayStudents(students, count);
```

```
    //释放动态数组占用的空间
    free(students);

    return 0;
}

//规范格式更美观地输出
//给入需要输出的 Stu 数组首地址指针，和动态数组的元素个数 count
void displayStudents(Stu *students, int count) {
    printf("-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|\n");
    for (int i = 0; i < count; i++) {
        printf("%12lld|%11.2f|%8.2f|%11.2f|%9.2f|%7.2f|\n",           students[i].id,
students[i].chinese,   students[i].math,   students[i].english,   students[i].total,
students[i].average);
    }
}

//封装读入学生数据
//此处为了使用 realloc 不断调整动态数组的空间大小，也就是需要修改指向 Stu 数组首地址的指针
//所以需要传入指向 Stu 数组首地址指针的指针
//还要传入动态数组的元素个数 origin_num 及将要读入的学生的人数
int readinStudents(Stu **students, int origin_num, int n) {
    printf("Input  students\'  information(id,  Chinese  score,  Math  score,  English
score)\n");
    int count = origin_num;
    double chi, math, eng;
    long long num;

    while (n--) {
        scanf("%lld %lf %lf %lf", &num, &chi, &math, &eng);
        (*students)[count].id = num;
        (*students+count)->chinese = chi;
        (*students+count)->math = math;
        (*students+count)->english = eng;
        (*students+count)->total = chi + math + eng;
        (*students+count)->average = 0.3*chi + 0.5*math + 0.2*eng;
        count++;
        *students = (Stu *) realloc(*students, (count+1)*sizeof(Stu));
    }

    return count;
}
//封装从动态数组中删除学生的功能
```

```
//与 readinStudents 函数相同，由于需要修改指向 Stu 数组首地址的指针
//需要传入指向 Stu 数组首地址指针的指针
//还需要传入动态数组中的元素个数 origin_num
int deleteStudents(Stu **students, int origin_num) {
    long long id;
    int left, right, mid;
    int new_num=origin_num;
    qsort(*students, origin_num, sizeof(Stu), cmp);
    scanf("%lld", &id);
    left = 0;
    right = origin_num - 1;
    while (left <= right) {
        mid = (left + right)/2;
        if ((*students)[mid].id == id)
            break;
        else if ((*students)[mid].id > id)
            right = mid - 1;
        else if ((*students)[mid].id < id)
            left = mid + 1;
    }
    if (left > right)
        printf("id Not Found!\n");
    else {
        for (int i = mid; i < new_num-1; i++)
            (*students)[i] = (*students)[i+1];
        new_num--;
        *students = (Stu *) realloc(*students, (new_num+1)*sizeof(Stu));
    }
    printf("input id:\n");

    return new_num;
}

//用于 qsort 的比较函数
int cmp(const void *a, const void *b) {
    const Stu *first=(const Stu *) a;
    const Stu *second=(const Stu *) b;

    return first->id - second->id;
}
```

3. 对学生成绩信息进行排序

2-3.c

```c
#include <stdio.h>
#include <stdlib.h>

//使用结构体来存储每个学生的学号及成绩信息
typedef struct {
    long long id;
    double chinese, math, english;
    double total, average;
} Stu;

void displayStudents(Stu *, int);
int readinStudents(Stu **, int, int);
void sortByTotal(Stu *, int, int);

int main() {
    Stu *students;
    int count=0;
    int n;

    //申请动态数组的空间
    students = (Stu *) malloc(sizeof(Stu));

    //先读取将要输入的学生人数
    printf("How many students?\n");
    scanf("%d", &n);

    //使用封装好的函数读取数据
    count = readinStudents(&students, count, n);
    displayStudents(students, count);

    //进行排序并输出
    printf("After sorting:\n");
    sortByTotal(students, 0, count-1);
    displayStudents(students, count);

    //释放动态数组占用的空间
    free(students);

    return 0;
}

//规范格式更美观地输出
```

```
//给入需要输出的 Stu 数组首地址指针，和动态数组的元素个数 count
void displayStudents(Stu *students, int count) {
    printf("-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|\n");
    for (int i = 0; i < count; i++) {
        printf("%12lld|%11.2f|%8.2f|%11.2f|%9.2f|%7.2f|\n",           students[i].id,
students[i].chinese,   students[i].math,   students[i].english,   students[i].total,
students[i].average);
    }
}

//封装读入学生数据
//由于需要修改指向 Stu 数组首地址的指针，所以需要传入指向 Stu 数组首地址指针的指针
//还要传入动态数组的元素个数 origin_num 及将要读入的学生的人数
int readinStudents(Stu **students, int origin_num, int n) {
    printf("Input  students\'  information(id,  Chinese  score,  Math  score,  English
score)\n");
    int count = origin_num;
    double chi, math, eng;
    long long num;

    while (n--) {
        scanf("%lld %lf %lf %lf", &num, &chi, &math, &eng);
        (*students)[count].id = num;
        (*students+count)->chinese = chi;
        (*students+count)->math = math;
        (*students+count)->english = eng;
        (*students+count)->total = chi + math + eng;
        (*students+count)->average = 0.3*chi + 0.5*math + 0.2*eng;
        count++;
        *students = (Stu *) realloc(*students, (count+1)*sizeof(Stu));
    }

    return count;
}

//使用优化的快速排序对 Stu 数组进行排序
//由于需要修改指向 Stu 数组首地址的指针，所以需要传入指向 Stu 数组首地址指针的指针
//left, right 为当前进行处理的数组区间，且为闭区间
void sortByTotal(Stu *students, int left, int right) {
    int i=left, j=right;
    double mid;
    Stu tmp;
    mid = students[(left+right)/2].total;
    while (i <= j) {
```

```
        while (students[i].total < mid) i++;
        while (students[j].total > mid) j--;
        if (i <= j) {
            tmp = students[i];
            students[i] = students[j];
            students[j] = tmp;
            i++;
            j--;
        }
    }
    if (j > left) sortByTotal(students, left, j);
    if (i < right) sortByTotal(students, i, right);
}
```

六、测试和结果

1. 学生成绩信息存储

Input:

```
4
20201060001 86.5 98 90
20201060002 87.5 88 70.2
20201060003 90.5 87 88
20201060004 91 65 78
```

Output:

```
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|     86.50|   98.00|     90.00|   274.50|  92.95|
 20201060002|     87.50|   88.00|     70.20|   245.70|  84.29|
 20201060003|     90.50|   87.00|     88.00|   265.50|  88.25|
 20201060004|     91.00|   65.00|     78.00|   234.00|  75.40|
```

```
D:\Documents\YNU文件及资料\大二上\课程相关\courses-of-2nd-year\data-structure\experiment 2>2-1.exe
How many students?
4
20201060001 86.5 98 90
20201060002 87.5 88 70.2
20201060003 90.5 87 88
20201060004 91 65 78
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|     86.50|   98.00|     90.00|   274.50|  92.95|
 20201060002|     87.50|   88.00|     70.20|   245.70|  84.29|
 20201060003|     90.50|   87.00|     88.00|   265.50|  88.25|
 20201060004|     91.00|   65.00|     78.00|   234.00|  75.40|
```

2. 追加和删除学生成绩信息

Input:

```
4
20201060001 86.5 98 90
```

```
20201060002 87.5 88 70.2
20201060003 90.5 87 88
20201060004 91 65 78

20201060005 90 87 60

20201060002
```

Output:

```
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|      86.50|   98.00|      90.00|   274.50|  92.95|
 20201060002|      87.50|   88.00|      70.20|   245.70|  84.29|
 20201060003|      90.50|   87.00|      88.00|   265.50|  88.25|
 20201060004|      91.00|   65.00|      78.00|   234.00|  75.40|

-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|      86.50|   98.00|      90.00|   274.50|  92.95|
 20201060002|      87.50|   88.00|      70.20|   245.70|  84.29|
 20201060003|      90.50|   87.00|      88.00|   265.50|  88.25|
 20201060004|      91.00|   65.00|      78.00|   234.00|  75.40|
 20201060005|      90.00|   87.00|      60.00|   237.00|  82.50|

-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|      86.50|   98.00|      90.00|   274.50|  92.95|
 20201060003|      90.50|   87.00|      88.00|   265.50|  88.25|
 20201060004|      91.00|   65.00|      78.00|   234.00|  75.40|
 20201060005|      90.00|   87.00|      60.00|   237.00|  82.50|
```

```
D:\Documents\YNU文件及资料\大二上\课程相关\courses-of-2nd-year\data-structure\experiment 2>2-2.exe
How many students?
4
Input students' information(id, Chinese score, Math score, English score)
20201060001 86.5 98 90
20201060002 87.5 88 70.2
20201060003 90.5 87 88
20201060004 91 65 78
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|      86.50|   98.00|      90.00|   274.50|  92.95|
 20201060002|      87.50|   88.00|      70.20|   245.70|  84.29|
 20201060003|      90.50|   87.00|      88.00|   265.50|  88.25|
 20201060004|      91.00|   65.00|      78.00|   234.00|  75.40|
Add one student at the end of the array
Input students' information(id, Chinese score, Math score, English score)
20201060005 90 87 60
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|      86.50|   98.00|      90.00|   274.50|  92.95|
 20201060002|      87.50|   88.00|      70.20|   245.70|  84.29|
 20201060003|      90.50|   87.00|      88.00|   265.50|  88.25|
 20201060004|      91.00|   65.00|      78.00|   234.00|  75.40|
 20201060005|      90.00|   87.00|      60.00|   237.00|  82.50|
Delete one student by id
20201060002
input id:
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|      86.50|   98.00|      90.00|   274.50|  92.95|
 20201060003|      90.50|   87.00|      88.00|   265.50|  88.25|
 20201060004|      91.00|   65.00|      78.00|   234.00|  75.40|
 20201060005|      90.00|   87.00|      60.00|   237.00|  82.50|
```

3. 对学生成绩信息进行排序

Input:

```
4
20201060001 86.5 98 90
20201060002 87.5 88 70.2
20201060003 90.5 87 88
20201060004 91 65 78
```

Output:

```
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|     86.50|   98.00|     90.00|   274.50|  92.95|
 20201060002|     87.50|   88.00|     70.20|   245.70|  84.29|
 20201060003|     90.50|   87.00|     88.00|   265.50|  88.25|
 20201060004|     91.00|   65.00|     78.00|   234.00|  75.40|
After sorting:
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060004|     91.00|   65.00|     78.00|   234.00|  75.40|
 20201060002|     87.50|   88.00|     70.20|   245.70|  84.29|
 20201060003|     90.50|   87.00|     88.00|   265.50|  88.25|
 20201060001|     86.50|   98.00|     90.00|   274.50|  92.95|
```

```
D:\Documents\YNU文件及资料\大二上\课程相关\courses-of-2nd-year\data-structure\experiment 2>2-3.exe
How many students?
4
Input students' information(id, Chinese score, Math score, English score)
20201060001 86.5 98 90
20201060002 87.5 88 70.2
20201060003 90.5 87 88
20201060004 91 65 78
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060001|     86.50|   98.00|     90.00|   274.50|  92.95|
 20201060002|     87.50|   88.00|     70.20|   245.70|  84.29|
 20201060003|     90.50|   87.00|     88.00|   265.50|  88.25|
 20201060004|     91.00|   65.00|     78.00|   234.00|  75.40|
After sorting:
-----id-----|--Chinese--|--Math--|--English--|--Total--|--Avg--|
 20201060004|     91.00|   65.00|     78.00|   234.00|  75.40|
 20201060002|     87.50|   88.00|     70.20|   245.70|  84.29|
 20201060003|     90.50|   87.00|     88.00|   265.50|  88.25|
 20201060001|     86.50|   98.00|     90.00|   274.50|  92.95|
```

七、用户手册

1. 学生成绩信息存储

先输入想要录入的学生人数，再依次输入每个学生的信息，按顺序分别为学号、语文成绩、数学成绩、英语成绩。学号为不超过长整型大小的整数，成绩既可以以整数形式输入，也可以以小数形式输入。

2. 追加和删除学生成绩信息

学生的相关信息的录入的输入规则与第 1 题中相同。在显示第一次输入的学生的相关信息后，会要求输入要添加的学生的相关信息，输入格式与之前相同。回车确认后，会显示添加了新的学生之后的学生信息列表。接着会要求输入想要删除的学生的 id，若 id 存在，删除该学生后会重新输出学生信息列表；若输入的 id 不存在，会提示“id Not Found!”

3. 对学生成绩信息进行排序

学生的相关信息的录入的输入规则与第 1 题中相同。输入完成后，会依次输出排序前和排序后的学生信息列表。