数据结构实验报告——实验 X

学号：_20201050331__姓名：_黄珀芝_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）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）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）可以根据学号或总分，升序排序

三、数据结构及算法描述

三个实验的算法合在一起讲了，因为只是构建与调用函数发生改变：

（1） 构造 student 结构体，定义学号、总分、语文成绩、数学成绩、英语成绩、加权平均分变量和类型；

（2） 用 malloc 函数为 stu 数组分配动态存储空间；

（3） 构建 Input_Score 函数，依次用 printf 和 scanf 提示用户输入学生的学号、三科成绩的信息；

（4） 构建 Count_Score 函数，用数学式子计算总分（sum = C+M+E）和加权平均分（average=C*0.3+M*0.5+E*0.2）；

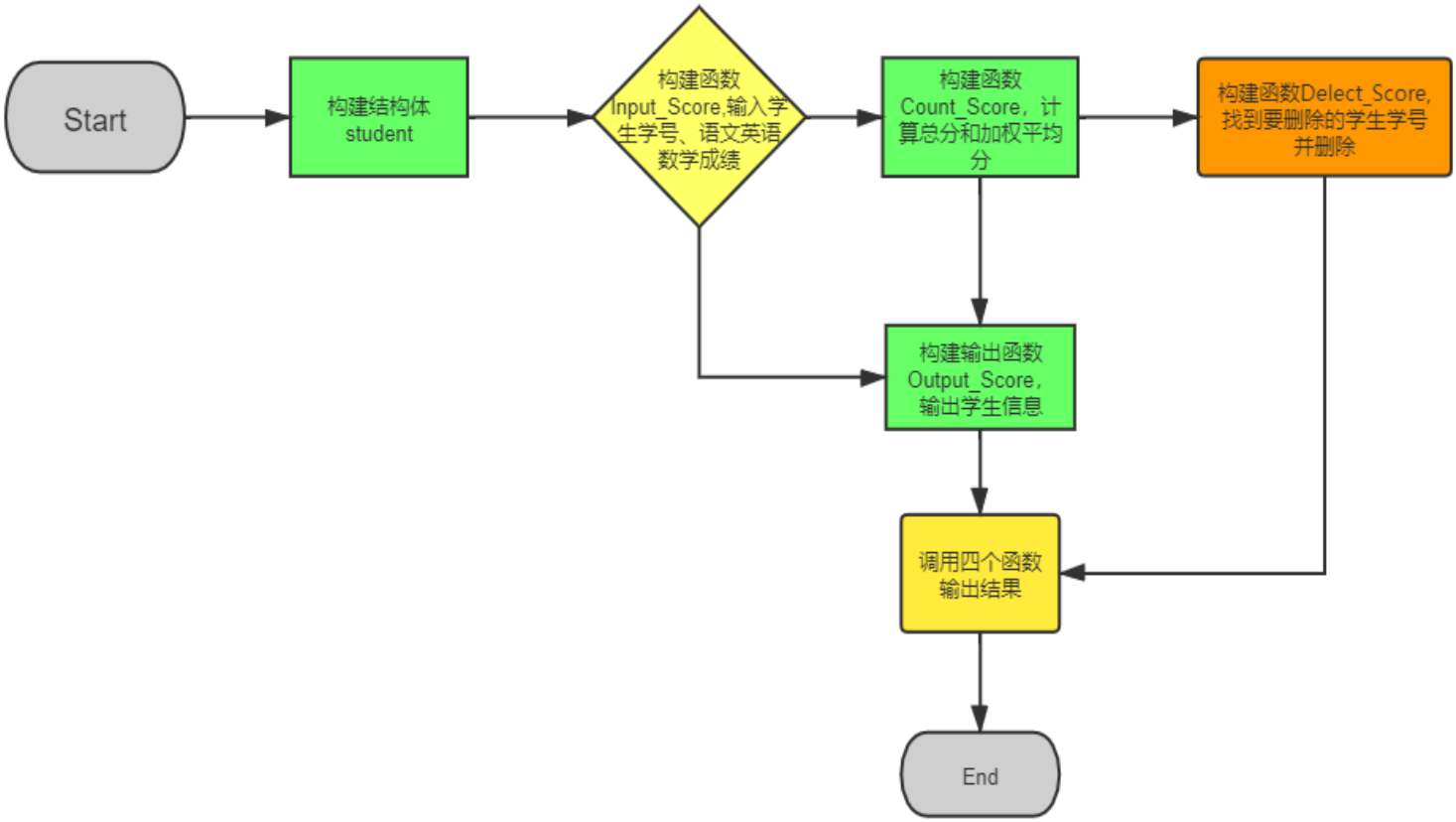
（5） 构建 Out_Score 函数，用一个 for 循环依次输出学生信息；

（6） 构建 Delect_Score 函数，用一个 for 来遍历数组，用 if 判断与读取需要被删除的学生信息的学号，若在遍历中寻找到了相同学号即删除该学号学生的信息；

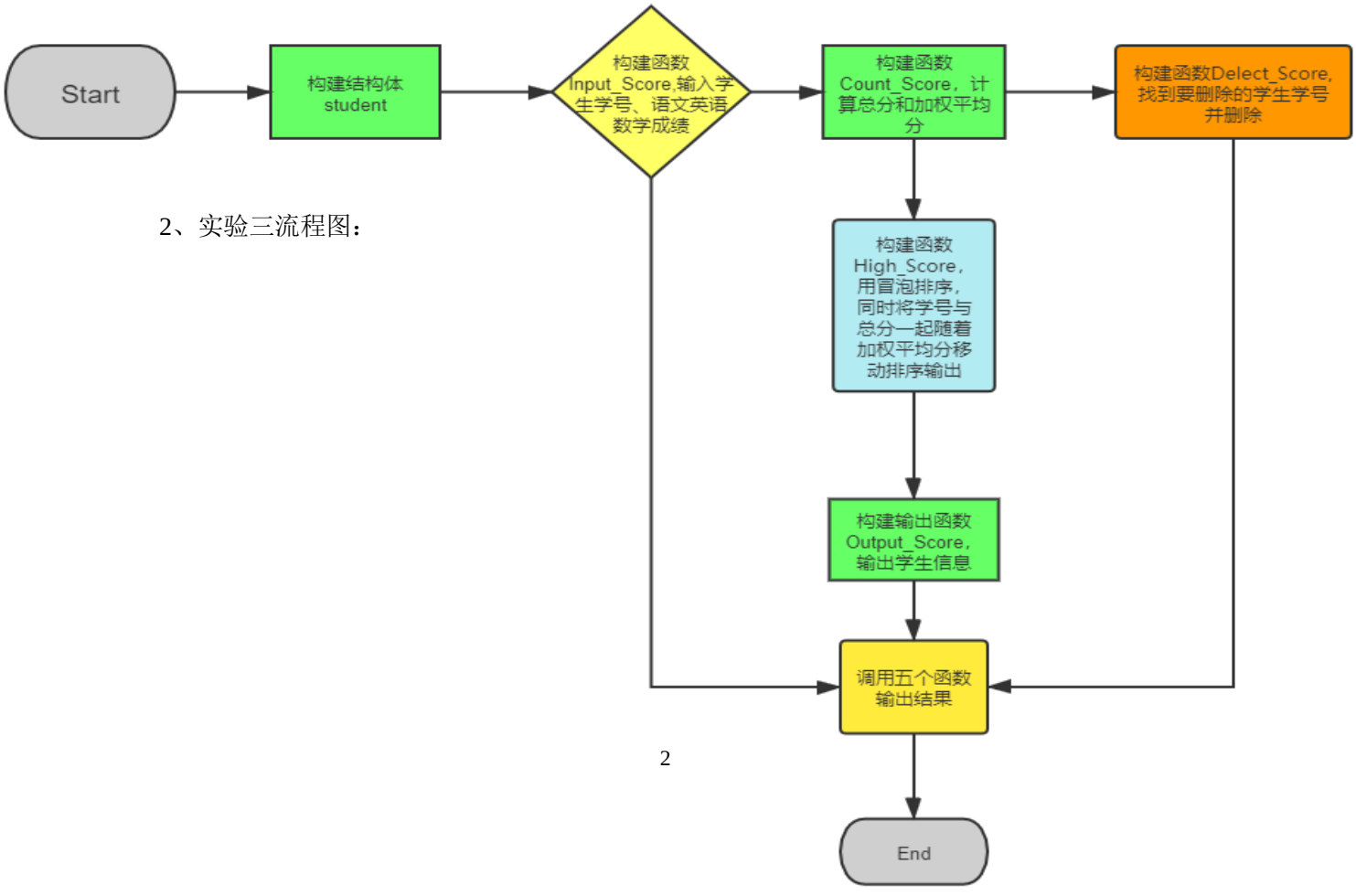
（7） 构建 High_Score 函数，用冒泡排序，用两个 for 循环遍历，若前一个数组元素比后一个数组元素大即交换。注意！同时也需要交换学号和总分这两个信息，因为每个学生的加权平均分和总分和学号是一一对应的！因此学号和总分需要跟着一起移动排序。只需要用三次交换即可完成。

（8） 根据题目要求，有需要的调用以上函数。

四、详细设计（1、实验一和实验二流程图）



2、实验三流程图：



五、程序代码

1、#include <stdio.h>

#include <stdlib.h>

```
typedef struct score_grade
{ long studentnum;//学生学号只能输入十位数字以内（用户要求详细说明）
  int Chinese;
  int Math;
  int English;
  int score_sum;
  double average;
} student;

void Input_Score(int n,student stu[])
{
      int i;
      for(i = 0;i < n;i++){
            printf("请输入第%d 个学生的学号：\n",i+1);
            scanf("%d",&stu[i].studentnum);
            printf("请输入第%d 个学生的语文成绩：\n",i+1);
            scanf("%d",&stu[i].Chinese);
            printf("请输入第%d 个学生的数学成绩：\n",i+1);
            scanf("%d",&stu[i].Math);
            printf("请输入第%d 个学生的英语成绩；\n",i+1);
            scanf("%d",&stu[i].English);
      }
}


void Count_Score(int n,student stu[])
```

```
{
    int i;
    for(i = 0;i < n;i++){
        stu[i].score_sum = 0;
        stu[i].average = 0.00;
        stu[i].score_sum = stu[i].Chinese + stu[i].Math + stu[i].English;
        stu[i].average = stu[i].Chinese*0.3+stu[i].Math*0.5+stu[i].English*0.2;
    }
}


void Output_Score(int n,student stu[])
{
    int i;
    for(i = 0;i < n;i++){
        printf("第%d 个学生的学号是：%ld; 总分是：%d;加权平均分是：%f\n",i+1,stu[i].studentnum,stu[i].score_sum,stu[i].average);
}
}

void main()
{

    int n;
    printf("请输入学生人数 n:\n");
    scanf("%d",&n);
    student*stu;
    stu = (student*)malloc(n*sizeof(student));
    if(!stu)exit(-1);
    Input_Score(n,stu);
```

```
    Count_Score(n,stu);
    Output_Score(n,stu);
}
```

2、

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

typedef struct score_grade
{ long student_num;//学生学号
  int Chinese;
  int Math;
  int English;
  int score_sum;
  double average;
} student;

void Input_Score(int n,student stu[])
{
    int i;
    for(i = 0;i < n;i++)
    {

        printf("请输入第%d 个学生的学号：\n",i+1);
        scanf("%ld",&stu[i].student_num);
        printf("请输入第%d 个学生的语文成绩：\n",i+1);
        scanf("%d",&stu[i].Chinese);
        printf("请输入第%d 个学生的数学成绩：\n",i+1);
        scanf("%d",&stu[i].Math);
        printf("请输入第%d 个学生的英语成绩：\n",i+1);
        scanf("%d",&stu[i].English);
```

```
    }

}

void Count_Score(int n,student stu[])
{
    int i;
    for(i = 0;i < n;i++){
        stu[i].score_sum = 0;
        stu[i].average = 0.00;
        stu[i].score_sum = stu[i].Chinese + stu[i].Math + stu[i].English;
        stu[i].average = stu[i].Chinese*0.3+stu[i].Math*0.5+stu[i].English*0.2;
    }
}


void Output_Score(int n,student stu[])
{
    int i;
    for(i = 0;i < n;i++){
        printf("第%d 个学生的学号是： %ld; 总分是： %d;加权平均分是： %f\n",i+1,stu[i].student_num,stu[i].score_sum,stu[i].average);
}
}
```

（即加入了这个函数）

```
void Delet_Score(int n,student stu[])
{
    int j;
    long NUM;
```

```
        printf("请输入需要删除信息的学生学号：\n");
        scanf("%ld",NUM);

        if(stu[j].student_num != NUM)
        printf(" 第 %d 个学生的学号是： %ld; 总分是： %d; 加权平均分是： %f\n",j+1,stu[j].student_num,stu[j].score_sum,stu[j].average);
}
void main()
{
   int n;
        printf("请输入学生人数 n:\n");
        scanf("%d",&n);
        student*stu;
        stu = (student*)malloc(n*sizeof(student));
        if(!stu)exit(-1);
        Input_Score(n,stu);
        Count_Score(n,stu);
        Output_Score(n,stu);
        Delet_Score(n,stu);
        free(stu);

}
```

3、在原有基础上加入这个函数并调用（源代码里面是完整的）

```
void High_Num(int n,student stu[])
{
        int t;
        int i,j,k;//定义整型变量
   printf("成绩由大到小排序：\n");//提示语句
   for(i=0;i<n;i++)//外层 for 循环
```

```
    {
        k=i;//把 i 的值赋给 k
        for(j=i+1;j<n;j++)//内层 for 循环
        {
            if(stu[j].average>stu[k].average)//挑出分数高的
            {
                k=j;//把相应的 j 赋值给 k
            }
        }
        t=stu[k].average; //把成绩高的放到前面
        stu[k].average=stu[i].average;
        stu[i].average=t;

        t=stu[k].student_num; //改变学号的排布
        stu[k].student_num=stu[i].student_num;
        stu[i].student_num=t;

        t=stu[k].score_sum; //改变总分的排布
        stu[k].score_sum=stu[i].score_sum;
        stu[i].score_sum=t;
    }
    for(i=0;i<n;i++)//循环输出 5 个人的成绩
    {
      printf(" 第 %d 个学生的学号是： %ld; 总分是： %d; 加权平均分是： %f\n",i+1,stu[i].student_num,stu[i].score_sum,stu[i].average);//输出结果
    }
}
High_Num(n,stu);
```

六、测试和结果

1.实验一：

```
请输入学生人数n:
2
请输入第1个学生的学号:
2020105098
请输入第1个学生的语文成绩:
122
请输入第1个学生的数学成绩:
123
请输入第1个学生的英语成绩;
111
请输入第2个学生的学号:
2020106035
请输入第2个学生的语文成绩:
89
请输入第2个学生的数学成绩:
90
请输入第2个学生的英语成绩;
56
第1个学生的学号是: 2020105098; 总分是: 356;加权平均分是: 120.300000
第2个学生的学号是: 2020106035; 总分是: 235;加权平均分是: 82.900000
```

2.实验二（增加删除）：

```
请输入学生人数n:
2
请输入第1个学生的学号:
1
请输入第1个学生的语文成绩:
23
请输入第1个学生的数学成绩:
34
请输入第1个学生的英语成绩;
45
请输入第2个学生的学号:
2
请输入第2个学生的语文成绩:
67
请输入第2个学生的数学成绩:
112
请输入第2个学生的英语成绩;
34
第1个学生的学号是: 1; 总分是: 102;加权平均分是: 32.900000
第2个学生的学号是: 2; 总分是: 213;加权平均分是: 82.900000
请输入需要删除信息的学生学号:
1

Process returned -1073741819 (0xC0000005)   execution time : 12.690 s
Press any key to continue.
```

3.实验三（按加权成绩来升序排序）（冒泡排序的笨方法）：

```
请输入学生人数n:
3
请输入第1个学生的学号：
1
请输入第1个学生的语文成绩：
123
请输入第1个学生的数学成绩：
111
请输入第1个学生的英语成绩；
148
请输入第2个学生的学号：
2
请输入第2个学生的语文成绩：
56
请输入第2个学生的数学成绩：
78
请输入第2个学生的英语成绩；
89
请输入第3个学生的学号：
3
请输入第3个学生的语文成绩：
45
请输入第3个学生的数学成绩：
90
请输入第3个学生的英语成绩；
120
第1个学生的学号是：1；总分是：382;加权平均分是：122.000000
第2个学生的学号是：2；总分是：223;加权平均分是：73.600000
第3个学生的学号是：3；总分是：255;加权平均分是：82.500000
成绩由大到小排序：
第1个学生的学号是：1；总分是：382;加权平均分是：122.000000
第2个学生的学号是：3；总分是：255;加权平均分是：82.000000
第3个学生的学号是：2；总分是：223;加权平均分是：73.000000

Process returned 3 (0x3)   execution time : 22.793 s
Press any key to continue.
```

七、用户手册

要求前提：在实验一中发现了一个问题：关于学号，我们云大的学号是 20201050331 十一位的，因此我在测试用例时自然的选择了输入十一位的数字来测试，结果发生了这种错误：

```
请输入学生人数n:
1
请输入第1个学生的学号：
20201050331
请输入第1个学生的语文成绩：
122
请输入第1个学生的数学成绩：
111
请输入第1个学生的英语成绩；
123
第1个学生的学号是：-1273786149；总分是：356;加权平均分是：116.700000
```

即输出的学号是编译错误的。我在寻找错误的时候，发现自己用的是 long 型（long 的取值范围为（-9223372036854774808~9223372036854774807），占用 8 个字节（-2 的 63 次方到 2 的 63 次方-1）。）的来定义学号，而在用 scanf 输入和用 printf 输出学号的时候，却用的是%d，这当然是不对的，因此我认为错误就是这个，应该改为%lld 或者%ld 就可以编译正确。但是在我修改了错误之后再 run，却发现输入十一位以上学号的时候编译错误依旧存在。但是输

```
请输入学生人数n:
2
请输入第1个学生的学号:
2020105098
请输入第1个学生的语文成绩:
122
请输入第1个学生的数学成绩:
123
请输入第1个学生的英语成绩;
111
请输入第2个学生的学号:
2020106035
请输入第2个学生的语文成绩:
89
请输入第2个学生的数学成绩:
90
请输入第2个学生的英语成绩;
56
第1个学生的学号是: 2020105098; 总分是: 356;加权平均分是: 120.300000
第2个学生的学号是: 2020106035; 总分是: 235;加权平均分是: 82.900000
```

入十位数字内的学号却能编译正确，正确显示输入的学号：

然后，我更加深入的查了一下 long 型范围：int、unsigned、long、unsigned long 、double 的数量级最大都只能表示为 10 亿，即它们表示十进制的位数不超过 10 个，即可以保存所有 9 位整数。而 short 只是能表示 5 位；
所以说，在这里应该使用 long long 型更为适合，或者用一个 char 来定义学号。

考虑到本次实验的学生号默认在十位数字以内，因此用户要求是：学号只能输入十位以内数字。