

数据结构实验报告——实验二

学号：_____20201020425_____ 姓名：_____邓其_____ 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

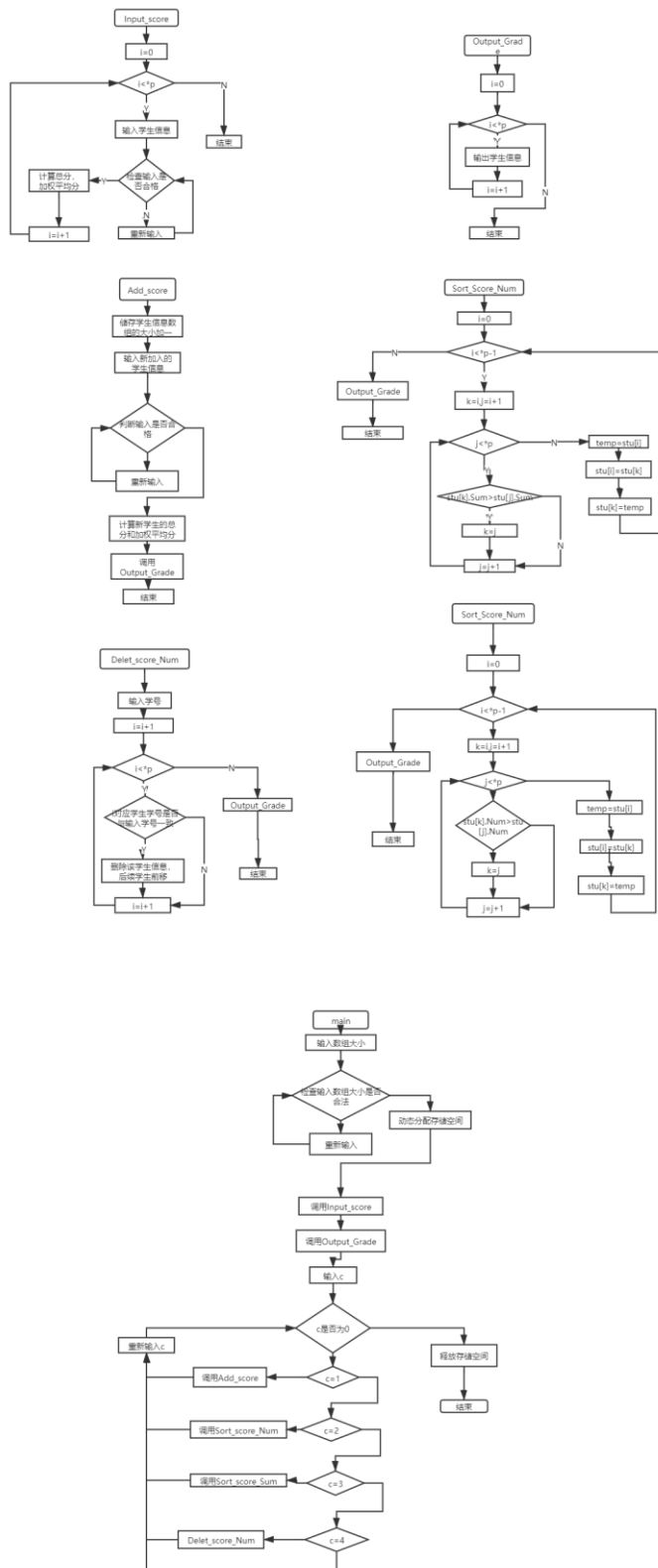
- 1、（必做题）每个学生的成绩信息包括：学号、语文、数学、英语、总分、加权平均分；采用动态方法创建数组用于存储若干学生的成绩信息；输入学生的学号、语文、数学、英语成绩；计算学生的总分和加权平均分（语文占 30%，数学占 50%，英语占 20%）；输出学生的成绩信息。
- 2、（必做题）可以在数组末尾追加新学生的成绩信息；可以根据学号，删除该学生的成绩信息。
- 3、（选做题）可以根据学号或总分，升序排序学生的成绩信息。

三、数据结构及算法描述

在本次实验中使用到了动态数组与结构体，动态数组具有数组的性质，并且可以在使用的过程中改变数组的大小，每个数组元素都是一个结构体，建立的时候使用内置函数 `malloc` 与 `realloc` 来实现空间的动态分配，在操作中可对动态数组进行删除，插入指定元素，可以使用基础操作对数组元素中进行排序，并输出。

本实验中并未使用特殊的算法，仅仅使用使用了分支语句，循环语句和指针，函数指针的使用。

四、详细设计



五、程序代码

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

typedef struct Num_score//定义了一个结构体用于储存每个学生的学号、各科成绩、总分、加权平均分，并取了一个别名 student

```
{
    int Num;
    int score[3];
    int sum;
    float aver;
} student;
```

void Input_score(int *p,student *stu){//读入已有学生信息

```
int i;
for(i=0;i<*p;i++){
    printf("请依次输入第%d 个学生的学号, 语文, 数学, 英语成绩(用空格分开): ",i+1);

    scanf("%d %d %d %d",&(stu+i)->Num,&(stu+i)->score[0],&(stu+i)->score[1],&(stu+i)->score[2]);

    while
((stu+i)->score[0]>100||(stu+i)->score[0]<0||(stu+i)->score[1]>100||(stu+i)->score[1]<0||(stu+i)->score[2]>100||(stu+i)->score[2]<0)
    {
        printf("请重新输入第%d 个学生的信息: ",i+1);
        scanf("%d %d %d %d",&(stu+i)->score[0],&(stu+i)->score[1],&(stu+i)->score[2]);
    }
    stu[i].sum=stu[i].score[0]+stu[i].score[1]+stu[i].score[2];
    stu[i].aver=stu[i].score[0]*0.3+stu[i].score[1]*0.5+stu[i].score[2]*0.2;
}
}
```

void Output_Grade(int *p,student *stu){//输出已存入的学生信息

```
int i;
printf("各学生成绩如下: \n");
for ( i = 0; i < *p; i++)
{
    printf("第个%d 学生的信息为: \t 学号%d\t 语文%d\t 数学%d\t 英语%d\t 总分%d\t 加\t\t\t\t\t 权\t\t\t\t\t 平\t\t\t\t\t 均\n",i+1,stu[i].Num,stu[i].score[0],stu[i].score[1],stu[i].score[2],stu[i].sum,stu[i].aver);
}
}
```

student* Add_score(int *p,student *stu){//添加学生信息

```
// student *p;
stu=(student*)realloc(stu,(*p+1)*sizeof(student));
if(!stu) exit(0);
printf("请依次输入新加入学生成绩: \n");
```

```

scanf("%d %d %d %d",&stu[*p].Num,&stu[*p].score[0],&stu[*p].score[1],&stu[*p].score[2
]);
while
(stu[*p].score[0]<0||stu[*p].score[0]>100||stu[*p].score[1]<0||stu[*p].score[1]>100||stu[*p].score[
2]<0||stu[*p].score[2]>100)
{
    printf("请重新依次输入新加入学生成绩: \n");
    scanf("%d %d %d %d",&stu[*p].score[0],&stu[*p].score[1],&stu[*p].score[2]);
}
stu[*p].sum=stu[*p].score[0]+stu[*p].score[1]+stu[*p].score[2];
stu[*p].aver=stu[*p].score[0]*0.3+stu[*p].score[1]*0.5+stu[*p].score[2]*0.2;
*p++;
Output_Grade(p,stu);
// return stu;
}

```

```

void Delete_score_Num(int *p,student *stu){
    int i,j,Num;
    printf("请输入要删除学生信息的学号: ");
    scanf("%d",&Num);
    for(i=0;i<*p;i++){
        if(stu[i].Num==Num){
            for(j=i;j<*p;j++) stu[j]=stu[j+1];
            *p=1;i--;
        }
    }
    Output_Grade(p,stu);
}

```

```

void Sort_Score_Num(int *p,student *stu){//根据学号升序排列
    int i,j,k;
    student temp;
    for(i=0;i<*p-1;i++){
        k=i;
        for(j=i+1;j<*p;j++){
            if (stu[k].Num>stu[j].Num) k=j;
        }
        temp=stu[i];stu[i]=stu[k];stu[k]=temp;
    }
    printf("根据学号排序如下: \n");
    Output_Grade(p,stu);
}

```

```

void Sort_Score_Sum(int *p,student *stu){//根据总分升序排列

```

```

    int i,j,k;
    student temp;
    for(i=0;i< *p-1;i++){
        k=i;
        for(j=i;j< *p;j++){
            if(stu[k].sum>stu[j].sum) k=j;
        }
        temp=stu[i];stu[i]=stu[k];stu[k]=temp;
    }
    printf("根据总成绩排序如下： \n");
    Output_Grade(p,stu);
}

int main(){
    int n,Num,c,*p;
    p=&n;
    student *stu;
    printf("请输入数组大小 n (1-100): ");
    scanf("%d",p);
    while (n<0||n>100)
    {
        printf("请重新输入数组大小 (1-100): ");
        scanf("%d",p);
    }
    stu=(student*)malloc(n*sizeof(student));
    if(!stu) exit(0);
    Input_score(p,stu);
    // Sort_Score_Num(n,stu);
    Output_Grade(p,stu);
    printf("执行增加学生信息请输入 1，按学号排序请输入 2，按总成绩排序输入 3，删除
    学生信息请输入 4，退出请输入 0: ");//再初始化学生信息后，选择将要进行的操作
    scanf("%d",&c);
    while (c)
    {
        if(c==1)
            Add_score(p,stu);
        else if (c==2)
            Sort_Score_Num(p,stu);
        else if(c==3)
            Sort_Score_Sum(p,stu);
        else if(c==4)
            Delete_score_Num(p,stu);
        else printf("输入有误，请重新输入！ \n");
        printf("执行增加学生信息请输入 1，按学号排序请输入 2，按总成绩排序输入 3，

```

删除学生信息请输入 4，退出请输入 0: ");

```
scanf("%d",&c);
```

```
}
```

```
}
```

六、测试和结果

(给出测试用例，并给出测试结果)

```
请输入数组大小n (1-100): 3
请依次输入第1个学生的学号，语文，数学，英语成绩 (用空格分开): 10104 90 98 78
请依次输入第2个学生的学号，语文，数学，英语成绩 (用空格分开): 10105 90 89 87
请依次输入第3个学生的学号，语文，数学，英语成绩 (用空格分开): 10101 98 98 78
各学生成绩如下:
第1个学生的信息为: 学号10104 语文90 数学98 英语78 总分266 加权平均分91.60
第2个学生的信息为: 学号10105 语文90 数学89 英语87 总分266 加权平均分88.90
第3个学生的信息为: 学号10101 语文98 数学98 英语78 总分274 加权平均分94.00
执行增加学生信息请输入1，按学号排序请输入2，按总成绩排序请输入3，删除学生信息请输入4，退出请输入0: 1
请依次输入新加入学生成绩:
10102 98 78 67
各学生成绩如下:
第1个学生的信息为: 学号10104 语文90 数学98 英语78 总分266 加权平均分91.60
第2个学生的信息为: 学号10105 语文90 数学89 英语87 总分266 加权平均分88.90
第3个学生的信息为: 学号10101 语文98 数学98 英语78 总分274 加权平均分94.00
第4个学生的信息为: 学号10102 语文98 数学78 英语67 总分243 加权平均分81.80
执行增加学生信息请输入1，按学号排序请输入2，按总成绩排序请输入3，删除学生信息请输入4，退出请输入0: 2
根据学号排序如下:
各学生成绩如下:
第1个学生的信息为: 学号10101 语文98 数学98 英语78 总分274 加权平均分94.00
第2个学生的信息为: 学号10102 语文98 数学78 英语67 总分243 加权平均分81.80
第3个学生的信息为: 学号10104 语文90 数学98 英语78 总分266 加权平均分91.60
第4个学生的信息为: 学号10105 语文90 数学89 英语87 总分266 加权平均分88.90
执行增加学生信息请输入1，按学号排序请输入2，按总成绩排序请输入3，删除学生信息请输入4，退出请输入0: 3
根据总成绩排序如下:
各学生成绩如下:
第1个学生的信息为: 学号10102 语文98 数学78 英语67 总分243 加权平均分81.80
第2个学生的信息为: 学号10104 语文90 数学98 英语78 总分266 加权平均分91.60
第3个学生的信息为: 学号10105 语文90 数学89 英语87 总分266 加权平均分88.90
第4个学生的信息为: 学号10101 语文98 数学98 英语78 总分274 加权平均分94.00
执行增加学生信息请输入1，按学号排序请输入2，按总成绩排序请输入3，删除学生信息请输入4，退出请输入0: 4
请输入要删除学生信息的学号: 10105
各学生成绩如下:
第1个学生的信息为: 学号10102 语文98 数学78 英语67 总分243 加权平均分81.80
第2个学生的信息为: 学号10104 语文90 数学98 英语78 总分266 加权平均分91.60
第3个学生的信息为: 学号10101 语文98 数学98 英语78 总分274 加权平均分94.00
执行增加学生信息请输入1，按学号排序请输入2，按总成绩排序请输入3，删除学生信息请输入4，退出请输入0: 0
```

七、用户手册

(告诉用户如何使用程序，使用注意事项等)

根据提示输入对应命令即可。