

重点关注

- 结构化变量名不代表该变量的地址, 这有区别于数组名
- 结构体变量作函数的参数。将结构体的全部成员值传送给函数, 值传送, 单向传递…..

第12次课外练习

本次课外练习必做题共3题，第15周末之前交。

1. 现有某班有10名学生，每个学生的信息包括：学号（ID）、姓名（name）、性别（sex）、出生日期（birthday）和成绩（score，浮点数）。编写程序，实现如下功能
 - (1) 在主函数main()中设计一个结构体数组，将从键盘输入10名学生信息保存在该数组中；
 - (2) 编写一个求平均值的函数，能对10名学生的成绩求平均值；
 - (3) 编写一个排序和显示函数，能对10名学生按照成绩的高低排序，并在屏幕显示排序结果；
 - (4 *) 编写一个写文件函数，将10名学生的成绩保存在1个数据文件中；
 - (5 *) 编写一个查询函数，输入学号就可以从数据文件中查找到与输入学号对应学生的信息，并显示。

说明：带*可以在下次作业中完成。

2. 采用枚举型数据类型，编写程序，要求通过键盘输入1-7，能分别输出相对应的星期英文单词。如输入1，则输出：Monday

3. 现有结构体类型 `fraction`，用于存放分数：

```
struct fraction
{
    long long int numerator;
    unsigned long long int denominator;
};
```

请编写函数 `Fra_Add()`、`Fra_Sub()`，分别实现两个 `fraction` 类型变量的加、减：

```
struct fraction Fra_Add(struct fraction f1, struct fraction f2);
struct fraction Fra_Sub(struct fraction f1, struct fraction f2);
```

要求：

- 1) 函数中要检查输入变量的合法性，当参数 `f1`、`f2` 中出现分母为 0 时，应返回结果 `{0, 0}`，表示运算非法；
- 2) 使用你编写的函数，计算以下数列的和：

$$4 \times \left[1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots + \frac{(-1)^{N-1}}{2N-1} \right]$$

其中，取 `N=10`，函数输出结果为一个 `fraction` 类型变量，分别打印结果的分数值“xxx/xxx”和浮点值。

选做题4道

1. 将课件中的单链表改成双向链表，即按以下定义：

```
typedef struct tagNode  
{  
    int index; // 非 0 元素的下标,应为非负整数  
    int value; // 非 0 元素的值  
    tagNode * prev; // 指向前一个元素的指针  
    tagNode * next; // 指向下一个元素的指针  
}NODE, *PNODE;
```

根据定义的双链表，重写以下函数：

```
PNODE insert(PNODE & head, int index, int value);  
void delete_by_index(PNODE & head, int index);
```

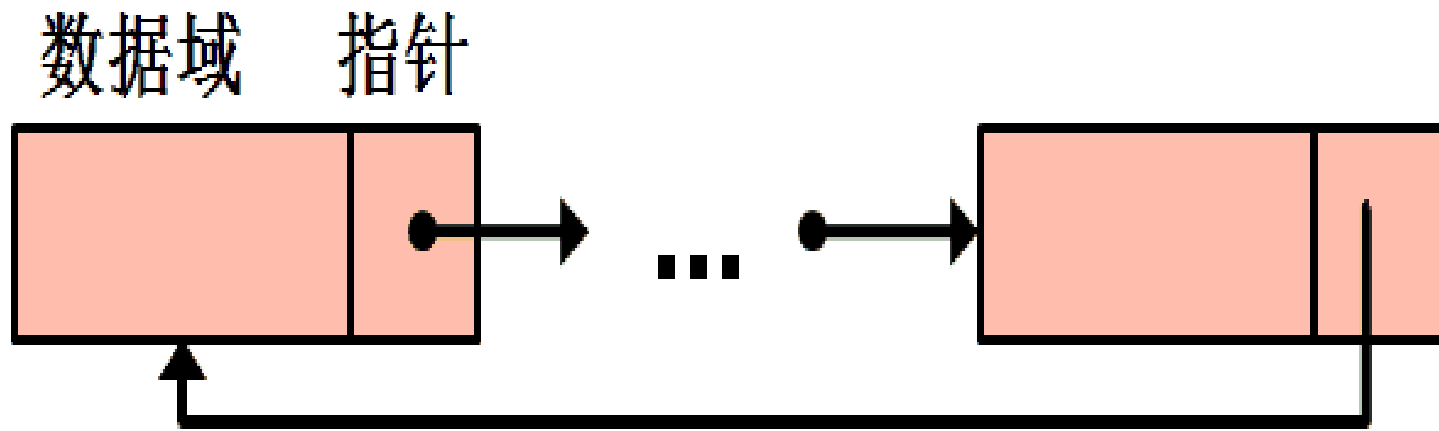
注意：元素插入时要能够识别下标重复的情况，自动覆盖已有下标的元素值。 并选择一个数据来测试的程序，输入若干组下标、值，然后建立链表，打印输出链表中所有元素的下标、值。

2. 设有一个链表，其结点值均为正整数。编写一个C函数，反复找出链表中结点值最小的结点，并输出该值，然后将该结点从链表中删除，直到链表空为止。

3 执行下列程序的输出结果是_____。

```
#include <stdio.h>
typedef struct {
    char name[9];
    char sex;
    int score[3];
} STU;
void f(STU *a)
{
    STU b={"huang", 'm', 81, 92}, *p=&b;
    *a = *p;
    a->sex = 'f';
    a->score[2] = a->score[0] + a->score[1];
}
void main()
{
    STU c={"Qian", 'f', 93, 97}, *d=&c;
    f(&c);
    printf("%s,%c,%d,%d,%d\n", d->name,d->sex,d->score[0],d->score[1],d->score[2]);
}
```

4. 使用循环链表编写约瑟夫问题的模拟程序：



39 个犹太人与约瑟夫及他的朋友躲到一个洞中，39 个犹太人决定宁死也不要被敌人抓到，于是决定了一个自杀方式，41 个人排成一个圆圈，由第 1 个人开始报数，每报到第 3 人该人就必须自杀，然后再由下一个重新报数，直到所有人都自杀身亡为止。然而约瑟夫和他的朋友并不想遵从，他将朋友与自己安排在两个位置上，最终逃过了这场死亡游戏。试问：这两个位置是哪两个？请使用循环链表模拟整个游戏过程中各参与者的出局顺序。