
— 计算机程序设计基础 (1) —

第13次作业

本学期作业提交说明

- 作业建议提交实验报告（如果当次作业要求则必须提交）。报告可包含但不限于：对作业的简单思路分析、实验结果的截图、代码、分析总结等。**如果提交的作业不包含足够说明信息，造成作业评判困难的，不给予相应题目分数，且不接受复议补交！**
- 对于需要写代码的题目，要求同时提交源代码；源代码可以直接拷贝到实验报告里，代码较长的话可以另附源代码文件提交。
- 实验报告可以提交word或pdf格式，建议提交pdf版。如果提交代码文件，注意仅提交*.h/c/cpp/hpp等源代码文件和代码运行所必须的依赖项即可，Visual Studio或Xcode等IDE产生的项目解决方案（如.sln）等文件不要提交！
- 实验报告、代码文件等都放在一个文件夹内，压缩成*.zip/rar等压缩文件，按时提交到网络学堂。
- **作业严禁抄袭！一旦发现并被判定为抄袭，无论抄与被抄，当次作业直接按照零分处理！**

本次作业提交说明

- 本次作业必做题3道，共10分；选做题4道。
- 选做题附加分0.5分，视完成情况给分，但本次作业分数不超过10分。
- **本次作业需要撰写实验报告，必须包含的内容详见各题。**
- 截止时间：**第15周周日（2018.12.30） 23:59**，缓交扣除当次作业分数的20%！

必做题

第1题

学生成绩系统（3分）

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。

本题仅前3题计分，带*的题目分数计入下次作业，可以在下次作业中完成。

说明：自行设计输入输出测试样例并给出结果截图。

第2题

星期输出（3分）

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。

说明：自行设计输入输出测试样例并给出结果截图。

第3题

分数计算（4分）

现有结构体类型 `fraction`，用于存放分数：

```
struct fraction
{
    long long int numerator; // 分子
    unsigned long long int denominator; // 分母
};
```

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。

说明：需要在实验报告中给出算法思路说明。

选做题

第1题

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。

说明：自行在main函数中建立初始链表，然后测试insert和delete_by_index函数。函数参数中的PNODE &是引用语法，如果不会的话，可以改为PNODE *指针，再完成此题。

第2题

将下列程序的输出改为COUT语句，并分析结果_____。

```
#include <stdio.h>
typedef struct {
    char name[9];
    char sex;
    int score[3];
} STU;
void f(STU *a)
{
    STU b={"Zhao", 'm', {85, 90} },*p=&b;
    a->score[2] = a->score[0] + a->score[1];
    a = p;
    a->sex = 'f';
    a->score[2] = a->score[0] + a->score[1];
```

```

    printf("%s,%c,%d,%d,%d,", a->name, a->sex, a->score[0], a->score[1], a->score[2]);
}
void main()
{
    STU c={"Qian", 'f', {95, 92}}, *d=&c;
    f(&c);
    printf("%s,%c,%d,%d,%d\n", d->name, d->sex, d->score[0], d->score[1], d->score[2]);
}

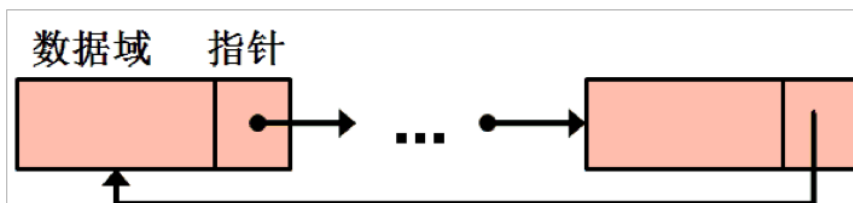
```

第3题

设有一个链表，其结点值均为正整数。编写一个c/cpp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。

第4题

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。