

实验 11 结构体

一、实验目的

- (1) 掌握结构体类型变量的定义和使用；
- (2) 掌握结构体类型数组的概念和使用；
- (3) 了解共用体的概念与使用。

二、实验内容

I.基础部分：理解结构体和共用体的存储

- (1) 给定程序 c11-1-1.c，请分析程序的运行结果，回答问题。

```
/* c11-1-1.c */
#include<stdio.h>
struct data
{
    int i;
    float a;
    char c[4];
};
int main ( )
{
    struct data u;
    printf("变量 u 所占的存储空间大小为: %d \n",sizeof(u));
    printf("请输入数据: \n");
    scanf("%d",&u.i);
    printf("i=%d, a=%f, c[0]=%c, c[1]=%c c[2]=%c c[3]=%c\n",
u.i,u.a,u.c[0],u.c[1],u.c[2],u.c[3]);
    return 0;
}
```

- ① 从键盘输入整数 100，分析运行结果。

- ② 将 scanf 语句改为：

```
scanf ("%f",&u.a);
```

程序另存为 c11-1-2.c，从键盘输入 3.5，分析程序的运行结果。

- ③ 将 scanf 语句改为：

```
scanf ("%d%f%c%c%c%c",&u.i,&u.a,&u.c[0],&u.c[1],&u.c[2],&u.c[3]);
```

程序另存为 c11-1-3.c 从键盘输入 100 3.5abcd，分析程序的运行结果。

提示：注意数据输入格式，3.5 后没有空格。

- (2) 给定程序 c11-1-4.c，请分析程序的运行结果，回答问题。

```

/* c11-1-4.c */
#include<stdio.h>
union data
{
    int i;
    float a;
    char c[4];
};
int main ( )
{
    union data u;
    printf("变量 u 所占的存储空间大小为: %d \n",sizeof(u));
    printf("请输入数据: \n");
    scanf("%d",&u.i);
    printf("i=%d, a=%f, c[0]=%c, c[1]=%c c[2]=%c c[3]=%c\n",
u.i,u.a,u.c[0],u.c[1],u.c[2],u.c[3]);
    return 0;
}

```

① 从键盘输入整数 100，分析运行结果。

② 然后将 scanf 语句改为：

```
scanf ("%f", &u. a);
```

程序另存为 c11-1-5.c，从键盘输入 3.5，分析运行结果。

③ 将 scanf 语句改为：

```
scanf ("%d%f%c%c%c%c", &u. i, &u. a, &u. c[0], &u. c[1], &u. c[2], &u. c[3]);
```

程序另存为 c11-1-6.c，从键盘输入 100 3.5abcd，分析程序的运行结果。

提示：注意数据输入格式，3.5 后没有空格。

II.提高部分

1、程序改错

注意：程序中的其它地方不要随意改动，不得增行或删行，也不得更改程序的结构！

（1）给定程序 c11-2-1.c 的功能是：定义一个结构体变量（包括年、月、日），计算给定日期在本年中是第几天。程序正确的运行结果如图 11-1 所示，程序中有 4 处错误，其中 3 处在 main()中。请改正程序中的错误，并运行出正确的结果。

<pre> Please input Y,M,D 2021 2 29 日期输入有误！ </pre>	<pre> Please input Y,M,D 2020 3 10 2020 3 10 is: 70 days </pre>
---	---

图 11-1 程序 c11-2-1.c 的运行结果示例

```

/*c11-2-1.c*/
#include<stdio.h>
#include<stdlib.h>
struct date
{
    int year ;

```

```

    int month ;
    int day ;
};
int days(int year,int month,int day ) ;
int check( int year,int month,int day);
int main( )
{
    struct date dt;
    printf ( "Please input Y,M,D\n" ) ;
    scanf ( "%d%d%d",&dt ) ;
    if (check(year,month,day)==0)
    {
        printf("日期输入有误 !\n");
        exit(0);
    }
    printf ( "\n%d %d %d is: %d days\n",dt.year,dt.month,dt.day,days(dt) );
}

/*check 函数判别输入的日期是否合法，合法返回 1，不合法返回 0*/
int check( int year,int month,int day)
{
    if(year<0||month<1||month>12||day<1) return 0;
    if((month%2==1&&month<=7)||(month%2==0&&month>=8))
    {
        if (day>31) return 0;
        else return 1;
    }
    else if((month==4)||(month==6)||(month==9)||(month==11))
    {
        if (day>30) return 0;
        else return 1;
    }
    else if(month==2)
    {
        if(((year%4==0&&year%100!=0)||year%400==0)//闰年
        {
            if(day>29) return 0;
            else return 1;
        }
        else
            if(day>28) return 0;
            else return 1;
    }
}

```

```

int days ( int year,int month,int day)
{
    int i,day_sum=0 ,leap;
    int day_tab[13]={ 0,31,28,31,30,31,30,31,31,30,31,30};
    if (year%4==0 && year% 100 !=0 || year%400==0 )
        leap=1;
    else
        leap=0;
    for(i=0;i<month ;i++)
        day_sum= day_sum +day_tab[i] ;
    day_sum= day_sum +day ;
    if (leap==0 && month>=3)
        day_sum= day_sum+1;
    return day_sum;
}

```

(2) 给定程序 c11-2-2.c 的功能是：结构体数组中存放有若干个人的姓名和年龄，以下程序可以输出其中年龄最小者的姓名和年龄，程序正确的运行结果如图 11-2 所示，程序中包含 3 个错误，请改正程序中的错误并运行出正确的结果。

年龄最小者为：lin ,其年龄为 17

图 11-2 程序 c11-2-2.c 的运行结果示例

```

/* c11-2-2.c */
#include <stdio.h>
#define N 5
struct man
{
    char name[20];
    int age;
}
int main()
{
    struct person[]={{"Li",20},{"Yang",21},{"Sun",18},{"zhao",23},{"lin",17}};
    int i,kmin;
    kmin=0;
    for(i=1;i<N;i++)
    {
        if(person[i].age <person[kmin].age ) kmin=i;
    }
    printf("年龄最小者为： %s ,其年龄为 %d\n", person[i].name ,person[i].age );
    return 0;
}

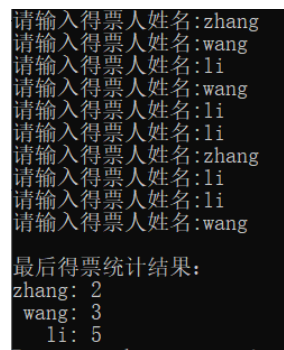
```

2、程序填空

说明：程序有多个空 (1)、(2) ……需要补充完整。请将程序中相应的位置填入正确答案并调试直到得到正确结果为止。

注意：不要随意改动程序，不得增行或删行，也不得更改程序的结构！

(1) 给定程序 c11-2-3.c 的功能是：设有三个候选人，Zhang，Wang 和 Li，设有 10 个人参加投票，每次输入一个得票的候选人的名字，要求最后输出各候选人的得票结果。程序运行结果示例如图 11-3 所示，请把程序补充完整。



```
请输入得票人姓名:zhang
请输入得票人姓名:wang
请输入得票人姓名:li
请输入得票人姓名:wang
请输入得票人姓名:li
请输入得票人姓名:li
请输入得票人姓名:zhang
请输入得票人姓名:li
请输入得票人姓名:li
请输入得票人姓名:wang

最后得票统计结果:
zhang: 2
wang: 3
li: 5
```

图 11-3 程序 c11-2-3.c 的运行结果示例

```
/* c11-2-3.c */
#include <stdio.h>
#include <string.h>
struct person /*候选人信息结构体，这是一个全局的结构体数组*/
{
    char name[20]; /*姓名*/
    int count; /*得票数*/
};
int main()
{
    struct person stu[3]={ "zhang",0,"wang",0,"li",0};
    int i,j;
    char leader_name[20];
    for(i=1;i<=10;i++) /*设有 10 个人参加投票*/
    {
        printf("请输入得票人姓名:");
        scanf("%s", leader_name); /*输入得票人姓名*/
        for(j=0;j<3;j++)
            if(strcmp(leader_name,stu[j].name)==0)
                ____ (1) ____;
    }
    printf("\n 最后得票统计结果: \n");
    for(i=0; i<3 ; i++)
        printf("%5s: %d\n", ____ (2) ____ );
    return 0;
}
```

(2) 给定程序 c11-2-4.c 的功能是：每个学生包括学号、姓名和成绩数据，要求找出成绩最高者的姓名和成绩。程序运行结果示例如图 11-4 所示，请把程序补充完整。

```

请输入人数:4
请输入学生信息:
1001 赵晓丽 90
1002 钱红 95
1003 王晓雷 87
1004 李丽 80

最高分:
学号: 1002  姓名: 钱红  成绩: 95.0

```

图 11-4 程序 c11-2-3.c 的运行结果示例

```

#include <stdio.h>
#include<stdlib.h>
struct student
{
    int num;
    char name[20];
    float score;
};
int main()
{
    struct student *stu;
    int i,imax=0,n;
    printf(" 请输入人数:");
    scanf("%d",&n);
    stu=____(1)____;    //生成动态数组
    printf(" 请输入学生信息:\n");
    for(i=0;i<n;i++)
        scanf ("%d%s%f", ____ (2) ____);
    imax=0;
    for(i=1;i<n;i++)
    {
        if(____(3)____)    imax=i;
    }
    printf("\n 最高分:\n");
    printf("学号: %d  姓名: %s  成绩: %4.1f\n",
    stu[imax].num,stu[imax].name,stu[imax].score);
    return 0;
}

```

3、程序设计

编写程序程序 c11-2-5.c，实现学生成绩管理系统(V4.0 版)

文件 score.txt 中存放若干同学的学号及高数、英语、C 语言 3 门课的成绩，格式如下：

1001 张晓莉 90 80 70

1002 赵 龙 85 78 80
1003 李大有 60 70 76

.....

利用结构体编写程序，实现如下功能：

- ① 从 score.txt 中逐个读入每个同学的成绩，用一个函数实现；
- ② 计算每个同学平均分，用一个函数实现；
- ③ 成绩按平均分排序，用一个函数实现；
- ④ 屏幕输出结果并写入文件 score_avg.txt 中，用一个函数实现。

文件 score_avg.txt 中的记录应该为：

学号	姓名	高数	英语	C 语言	平均分
1002	赵 龙	85	78	80	81.00
1001	张晓莉	90	80	70	80.00
1003	李大有	60	70	76	68.67

.....

III.拓展部分:通过结构体实现班车预约系统

很多高校都有多个校区，校区之间有通勤班车，坐班车需要预先预约。本题目的是编写一个简单的班车预约系统。包含如下功能：

- ① 师生可以预约班车；
- ② 师生可以查询已经预约的班车信息。

假设有 10 趟免费班车，用编号 1-10 表示，不同的编号表示不同的时间。

(1) 结构体及数据文件设计

根据需求，师生预约的结构体可以设计如下：

```
typedef struct {  
    int year; //年  
    int month; //月  
    int day; //日  
} Date;  
  
typedef struct  
{  
    char No[9]; //8 位学号或工号  
    char name[20];  
    int trav_line_no; //10 条免费班车线路，编号从 1 到 10  
    Date order_date; //预约的日期  
} bus_order;
```

所有师生的

预约信息保存在文本文件 order.txt 中，文件示例如图 11-5 所示，其中数据依次为工号/学号、姓名、班车编号、预约的日期。

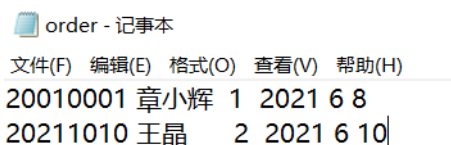


图 11-5 文本文件 order.txt 内容示例

(2) 模块设计

根据功能需求，本系统可以分为 3 大模块。

表 11-1 游客班车预约系统主要模块

序号	函数名（功能模块）	功能
1	main()	主控函数
2	order()	班车预约
3	query ()	预约信息查询

(3) 总体设计思想

所有操作通过文件或数组完成，当游客预约班车时，将预约信息追加到 order.txt 的末尾；当师生查询信息时，从 order.txt 读取所有信息到结构体数组，再逐个进行信息比较。

① 调用 order() 函数预约班车时需要输入师生工号/学号、姓名、班车线路号、预约日期，并且要确保输入的预约日期不早于当前日期。

为了保证预约的日期不早于当前日期，需要读取系统当前日期，代码段如下：

```
time_t now ;
struct tm *tm_now ;
time(&now) ;
tm_now = localtime(&now) ;
其中，time 函数返回 1970-1-1, 00:00:00 以来经过的秒数
localtime 函数将时间数值变换成本地时间，考虑到本地时区和夏令时标志；
struct tm *结构体，原型如下：
struct tm {
    tm_sec; /* 秒 - 取值区间为[0,59] */
    tm_min; /* 分 - 取值区间为[0,59] */
    int tm_hour; /* 时 - 取值区间为[0,23] */
    int tm_mday; /* 一个月中的日期 - 取值区间为[1,31] */
    int tm_mon; /* 月份（从一月开始，0 代表一月） - 取值区间为[0,11] */
    int tm_year; /* 年份，其值等于实际年份减去 1900 */
    int tm_wday; /* 星期 - 取值区间为[0,6]，其中 0 代表星期天，1 代表星期一 */
    int tm_yday; /* 从每年 1 月 1 日开始的天数 - 取值区间[0,365]，其中 0 代表 1 月 1 日 */
    int tm_isdst; /* 夏令时标识符，夏令时 tm_isdst 为正；不实行夏令时 tm_isdst 为 0 */
};
```

例如，下列程序可以获取当前计算机系统日期：

```
#include <stdio.h>
#include <time.h>
int main()
{
    time_t now ;
    struct tm *tm_now ;
    time(&now) ;
    tm_now = localtime(&now) ;
    printf("%d %d %d\n", tm_now->tm_year+1900, tm_now->tm_mon, tm_now->tm_mday);
    return 1;
}
```


- ② 查询函数 `query()` 查询预约信息时, 需要把数据文件信息读入数组, 因此需要再设计一个函数 `read()` 用于读取数据文件。
- ③ 需要设计一个菜单实现功能的选择。