

## 专题（兼容类）：程序题（A场）

### 一、单选题（本大题 10 道小题，每小题 1.5 分，共 15 分）

具体答题在考试客户端中单项选择题部分

### 二、程序填空题（本大题 2 道小题，每空 2.5 分，共 20 分）。

具体答题在考试客户端中程序填空题部分

### 三、操作题

（一）程序调试题（3小题，第1、2小题各3处错，第3小题4处错，每改正1处得3分，共30分）

#### 1. 程序调试题 1

**调试要求：**下列程序包含3个错误，请按题中的功能要求，打开C:\KS\A\_3\_1.c，调试并修改该程序（在所修改语句后加“/\*\_\*\_”或“//\_”字样的注释），使其运行能得到正确的结果。修改后的程序仍保存为C:\KS\A\_3\_1.c。

**功能说明：**输入n，按以下公式计算并输出结果。

$$\sum_{k=1}^n k! + \sum_{k=1}^n k^2, 1 \leq n \leq 10$$

**运行示例：**

input n: 3	input n: 10
23	4038298

**带错误源程序：**

```
#include <stdio.h>
int main(void)
{
    int k, s=1, n;
    int sum;
    printf("input n: ");
    scanf("%d", &n);
    for(k=1; k<n; k++)
    {
        s=s*k;
        sum+=s;
    }
    for(k=1; k<=n; k++)
        s=s+k*k;
    printf("%d\n", sum);
    return 0;
}
```

#### 2. 程序调试题 2

**调试要求：**下列程序包含3个错误，请按题中的功能要求，打开C:\KS\A\_3\_2.c，调试并修改该程序（在所修改语句后加“/\*\_\*\_”或“//\_”字样的注释），使其运行能得到正确

的结果。修改后的程序仍保存为C:\KS\A\_3\_2.c。

**功能说明：**数组x有10个元素，计算x中连续元素之和。主函数main()中初始化了9个元素，从键盘上输入最后1个元素，再输入下标 i1和i2 (设i1≤i2)，调用函数sum()，计算下标 i1(含)到 i2(含)的元素之和，并输出结果。

**运行示例1：**

```
input a number: 40
Input i1 and i2 :8 9
Sum=85
```

**运行示例2：**

```
input a number: 20
Input i1 and i2 :0 9
Sum=234
```

**运行示例3：**

```
input a number: 30
Input i1 and i2 :8 10
输入数据出错
```

**带错误源程序：**

```
#include<stdio.h>
int sum(int x, int i1, int i2)
{
    int i, s;
    for(i=i1; i<=i2; i++)
        s=s+x[i];
    return s;
}
int main(void)
{
    int i, i1, i2, result;
    int x[10] = {12, 23, 21, 34, 40, 6, 15, 18, 45};
    printf("input a number: ");
    scanf("%d", &x[9]);
    printf("Input i1 and i2 :");
    scanf("%d%d", &i1, &i2);
    if(i1>=0 || i2<10)
    {
        result=sum(x, i1, i2);
        printf("Sum=%d\n", result );
    }
    else
        printf("输入数据出错\n");
    return 0;
}
```

### 3. 程序调试题 3

**调试要求:** 下列程序包含4个错误, 请按题中的功能要求, 打开C:\KS\A\_3\_3.c, 调试并修改该程序(在所修改语句后加“/\*\_\*/”或“//\_”字样的注释), 使其运行能得到正确的结果。修改后的程序仍保存为C:\KS\A\_3\_3.c。

**功能说明:** 家庭收支记录。某家庭每天对收入或支出记帐, 需要记录的信息包括明细(不超过五个汉字)、类型(收入或支出)、金额和日期(用整型表示年月日, 如2022年5月8日, 表示20220508)。结构体数组中已经保存6条记录, 要求再输入一个记录, 并统计收入和支出金额, 以及收入余额(收入金额-支出金额)。(注所有金额输出保留2位小数)

#### 运行示例1:

输入一条记录: 明细、类型、金额和日期: 买书 支出 350.9 20220125				
序号	日期	明细	类型	金额
1	20220110	工资	收入	14572.80
2	20220105	交房租	支出	4000.00
3	20220115	买衣服	支出	320.00
4	20220120	奖金	收入	3000.00
5	20220120	交水电费	支出	235.60
6	20220125	买米面	支出	346.00
7	20220125	买书	支出	350.90
收入:		17572.80	支出:	5252.50
				收入余额: 12320.30

#### 运行示例2:

输入一条记录: 明细、类型、金额和日期: 买车票 支出 500.9 20220130				
序号	日期	明细	类型	金额
1	20220110	工资	收入	14572.80
2	20220105	交房租	支出	4000.00
3	20220115	买衣服	支出	320.00
4	20220120	奖金	收入	3000.00
5	20220120	交水电费	支出	235.60
6	20220125	买米面	支出	346.00
7	20220130	买车票	支出	500.90
收入:		17572.80	支出:	5402.50
				收入余额: 12170.30

#### 带错误源程序:

```
#include <stdio.h>
#include <string.h>
#define N 7
struct d_list           //家庭收支记录
{
    char detail[11];      //明细
    char ttype[10];       //类型: 收入或支出
    float count;          //金额
    long date;             //年月日
} ;
float income=0.0;

float in_out_count(type d_list p[], int n)
{
    int i;
    float out=0.0;
    for (i=1; i<n; i++)
        if (!strcpy(p[i].ttype, "收入"))
```

```

        income=income+p[i].count;
    else
        out=out+p[i].count;
    return out;
}

int main(void)
{
    struct d_list d_info[N]= {{"工资", "收入", 14572.80, 20220110},
        {"交房租", "支出", 4000, 20220105}, {"买衣服", "支出", 320, 20220115},
        {"奖金", "收入", 3000, 20220120}, {"交水电费", "支出", 235.60, 20220120 },
        {"买米面", "支出", 346.0, 20220125}
    };
    int i;
    float expend, total;
    printf("输入一条记录：明细、类型、金额和日期：");
    scanf("%s%s%f%d", d_info[N-1].detail, d_info[N-1].ttype, &d_info[N-
    1].count, &d_info[N-1].date);
    expend=in_out_count(d_info, N);
    total=income-expend;
    printf("%6s %6s %12s %8s %10s\n", "序号", "日期", "明细", "类型", "金额");
    for(i=0; i<N; i++)
        printf("%4d %10ld %12s %6s %12.2f\n", i+1, d_info[i].date, d_info[i].detail, d_info
        [i].ttype, d_info[i].count);
    printf("%10s:%12.2f\t %10s:%12.2f\t %10s:%12d\n", "收入", income, "支出",
    ", expend, "收入余额", total);
    return 0;
}

```

## (二) 编程题 (2小题, 第1小题15分, 第2小题20分, 共35分)

### 1. 编程题 1

**编程要求:** 按程序功能能要求, 编写程序并保存为C:\KS\A\_4\_1.c。

**程序功能:** 某电商平台商品信息如下:

商品号	商品名称	单价(元)/500克
1	生菜	4.2
2	西红柿	6.8
3	鸡胸肉	12.8
4	猪后腿肉	16.5

输入购买的商品号及购买数量。若商品号和数量均输入无误, 计算并输出购买的总价(保留1位小数)。

**运行示例1:**

```
请输入商品编号:1  
请输入商品数量:2  
购买1商品2件, 共计8.4元
```

运行示例2:

```
请输入商品编号:1  
请输入商品数量:-1  
输入商品号或数量错误!
```

运行示例3:

```
请输入商品编号:5  
请输入商品数量:2  
输入商品号或数量错误!
```

## 2. 编程题 2

**编程要求:** 按程序功能要求, 编写程序并保存为C:\KS\A\_4\_2.c。

**程序功能:** 输入字符串s和字符串sub, 判断字符串s中是否存在sub子串, 若存在, 将字符串s中第一个sub子串的所有字符替换成'\*', 并输出变换后的字符串, 否则输出"不存在该子串!"。

运行示例1: (一个子串测试)

```
请输入字符串:  
He is 12 years old.  
请输入子串:  
is  
处理后的字符串:  
He ** 12 years old.
```

运行示例2: (多个子串测试)

```
请输入字符串:  
This is a book.  
请输入子串:  
is  
处理后的字符串:  
Th** is a book.
```

运行示例3: (无子串测试)

```
请输入字符串:  
I am a teacher.  
请输入子串:  
is  
不存在该子串!
```

## (三) 编程题 (第 5 问, 20 分)

**编程要求:** 打开C:\KS\A\_5\_1.c, 按程序功能要求, 补充完善程序, 修改后的程序仍保存为C:\KS\A\_5\_1.c。

**应用背景:** 某厨艺大赛的比拼项目分为“热菜、雕刻和拼盘”三项, 所有选手皆需参与所有三项比拼, 每项比拼获得一个名次, 每个名次对应一个不同的积分。根据三项比拼的总积分进行最后的总排名。

**程序功能:** 根据每个选手不同项目的排名求得该项目上获得的积分, 并根据三项比拼结果获得的总积分由高到低排出最终名次。数据定义与原始代码见后, 请按要求完成如下设计:

(1) 设计getpoints() 函数, 根据各选手的项目排名计算其在该项目上的积分, 如果有n名

选手，则第一名积分为 $2*n$ ，后面名次每降一名，积分少2分。

函数声明： `void getpoints(struct contestant c[], int n);`

参数说明： `c`为参赛选手结构体数组， `n`为参赛人数。

(2) 设计 `gettotalpoints()` 函数，求各选手3个项目的总积分。

函数声明： `void gettotalpoints(struct contestant c[], int n);`

参数说明： `c`为参赛选手结构体数组， `n`为参赛人数。

(3) 修改 `main()` 主函数，调用自定义函数，实现程序功能。

```
#include <stdlib.h>
#include <stdio.h>
struct event
{ //选手参赛信息
    char cname[20]; //比赛项目名称
    int ranking; //该项目中名次
    int points; //积分
};
struct contestant
{ //选手信息
    char pname[20]; //选手姓名
    struct event e[3]; //选手参赛表
    int totalpoints; //总积分
};
void getpoints(struct contestant c[], int n)
{ //根据各选手排名计算对应积分，第1名 $2*n$ 分，每降一名积分少2分
    //补充填写代码.....
}
void gettotalpoints(struct contestant c[], int n)
{ //计算各选手总积分
    //补充填写代码.....
}
int main(void)
{
    int i, j;
    struct contestant c[5] = {
        {"zhao", {{ "cooking", 2, 0}, {"carving", 3, 0}, {"cold_dishes", 4, 0}}, 0},
        {"qian", {{ "cooking", 3, 0}, {"carving", 4, 0}, {"cold_dishes", 5, 0}}, 0},
        {"sun", {{ "cooking", 5, 0}, {"carving", 5, 0}, {"cold_dishes", 3, 0}}, 0},
        {"li", {{ "cooking", 4, 0}, {"carving", 1, 0}, {"cold_dishes", 2, 0}}, 0},
        {"zhou", {{ "cooking", 1, 0}, {"carving", 2, 0}, {"cold_dishes", 1, 0}}, 0}
    };
    //补充填写代码.....要求调用函数实现功能
    printf("序号\t姓名\t积分\n");
    for (i = 0; i < 5; i++)
        printf("%d\t%s\t%d\n", i+1, c[i].pname, c[i].totalpoints);
    return 0;
}
```

**运行示例：**

序号	姓名	积分
1	zhao	18
2	qian	12
3	sun	10
4	li	22
5	zhou	28

**(四) 编程题 (第 6 问, 30 分)**

**编程要求：** 打开C:\KS\A\_6\_1.c，按程序功能要求，补充完善程序，修改后的程序仍保存为C:\KS\A\_6\_1.c。

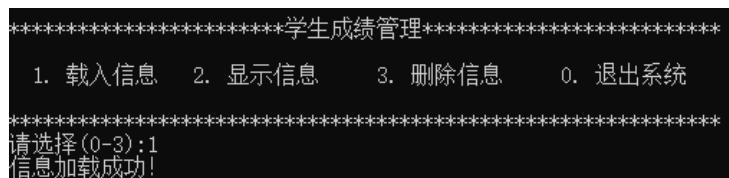
**程序功能：** 本程序是通过链表实现学生个人信息（包括学号、姓名和分数）的载入（从磁盘文件中读入）、显示和删除等功能。其中：

- (1) 头文件head.h包含有stdio.h和其他系统头文件，以及结构体的定义等内容。
- (2) 菜单函数mainMenu()和主函数main()已经完整给出，无需修改。
- (3) 请编写载入函数loadInfo()、显示函数displayInfo()和删除函数delInfo()，其形参和返回值类型见程序中的函数声明。
- (4) 载入函数loadInfo()实现将给定文件C:\KS\A\_6\_1.txt中的学生信息读入内存（链表）中。
- (5) 显示函数displayInfo()实现完整显示链表中学生信息。
- (6) 删除函数delInfo()将内存链表中指定学号的信息删除。
- (7) 设计一个链表结点查找函数，供delInfo()函数在删除前查找定位使用，函数名称、参数表与返回类型自主设计，要求不能包含任何输入输出功能。

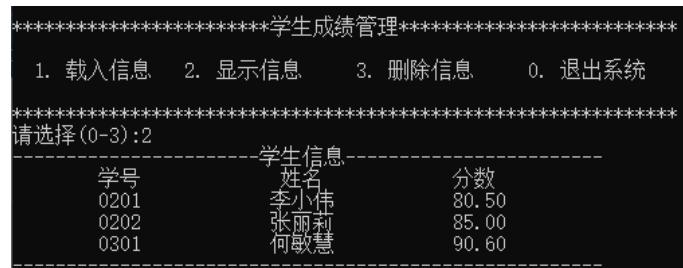
**图示 (A\_6\_1.txt) :**



**运行示例1：** 成功加载A\_6\_1.txt文件。



**运行示例2：** 显示载入（链表）的信息。



**运行示例3：** 未查找到学号，删除失败。

```
*****学生成绩管理*****
1. 载入信息 2. 显示信息 3. 删除信息 0. 退出系统
*****
请选择(0-3):3
-----删除信息-----
待删信息学号:0302
查无此号!
删除失败!
```

**运行示例4:** 信息删除成功。

```
*****学生成绩管理*****
1. 载入信息 2. 显示信息 3. 删除信息 0. 退出系统
*****
请选择(0-3):3
-----删除信息-----
待删信息学号:0202
    学号      姓名      分数
    0202    张丽莉    85.00
删除确认(Y/N): y
删除成功!
```

**提供的源程序代码:**

```
//head.h
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
/*学生个人信息结点定义*/
typedef struct Student {
    char no[5];           /* 学号 */
    char name[20];        /* 姓名 */
    float score;          /* 分数 */
    struct Student *next; /* 下一结点指针 */
} sNode;

//原A_6_1.c
#include "head.h"
void mainMenu(void);           /* 显示主菜单 */
sNode * loadInfo(sNode* h);    /* 加载信息函数 */
void displayInfo(sNode* h);    /* 显示信息函数 */
sNode * delInfo(sNode* h);    /* 删除信息函数 */
int main(void)
{
    sNode * head = NULL;        /* 链表结构头指针，作为头指针 */
    int ch;
    while (1)
    {
        mainMenu();
    }
}
```

```

printf("请选择(0-3):");
scanf("%d", &ch);
switch (ch)
{
    case 0:      //命令 0
        return 0;
    case 1:      //命令 1
        head = loadInfo(head);
        break;
    case 2:      //命令 2
        displayInfo(head);
        break;
    case 3:      //命令 3
        head = delInfo(head);
        break;
    default:     //无效命令, 过滤至换行
        scanf("%*[^\n]*c");
}
return 0;
}

void mainMenu(void)
{
    printf("\n*****学生成绩管理\n*****\n");
    printf(" 1. 载入信息\t2. 显示信息\t3. 删除信息\t0. 退出系统\n");
    printf("\n*****\n");
}

```