

专题 1（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。

程序功能：输入 n 的值，根据下列表达式求解 An 的值。

$$A1 = 1, A2 = \frac{1}{1 + A1}, A3 = \frac{1}{1 + A2}, \dots, An = \frac{1}{1 + A(n-1)}$$

运行示例1：

```
input n: 10
A10=0.617977
```

运行示例2：

```
input n: 11
A11=0.618056
```

【带错误的源代码】

```
#include <stdio.h>
float fun (int n)
{
    float A=0;
    int i;
    for (i=1; i<n; i++)
        A = 1.0/(1+A);
    return A ;
}

int main(void)
{
    int n ;
    printf("input n: ") ;
    scanf("%d", n ) ;
    printf("A%d=%f\n", n, fun(int n) );
    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。

程序功能：输入字符串，将小写字母转换为大写字母，其它字符不变，输出转换后的结果并保存到文件test.txt。

运行示例：

```
输入:hello 123 C language
HELLO 123 C LANGUAGE
```



【带错误的源代码】

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    FILE *fp;
    char str[100];
    int i=0;
    if((fp=fopen("test.txt", "r"))==NULL)
    {
        printf("can't open this file.\n");
        exit(0);
    }
    printf("输入:");
    gets(str);
    while (str)
    {
        if(str[i]>='a' && str[i]<='z')
            str[i]=str[i]+32;
        fputc(str[i], fp);
        i++;
    }
    printf("%s\n", str);
    fclose(fp);
    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。

程序功能：已知奥运会奖牌信息如下：

```

struct Medal
{
    char country[20]; //国家名
    int gold; //金牌数
    int silver; //银牌数
    int bronze; //铜牌数
    int total; // 奖牌总数
};

```

输入奖牌信息，调用sort()函数按奖牌总数降序排序。

运行示例：

```

请输入国家个数：5
请输入5个国家、金、银、铜牌个数：
英国 8 16 17
印度 2 4 5
中国 21 19 11
法国 11 18 17
澳大利亚 6 12 14

按奖牌总数由多到少排序后：
国家      金牌      银牌      铜牌      奖牌总数
中国      21        19        11        51
法国      11        18        17        46
英国      8         16        17        41
澳大利亚  6         12        14        32
印度      2         4         5         11

```

【带错误的源代码】

```

#define N 100
#include <stdio.h>
struct Medal
{
    char country[80];
    int gold;
    int silver;
    int bronze;
    int total;
};
void sort(struct Medal m[],int n)
{
    int i,j;
    struct temp;
    for(i=0; i<n-1; i++)
        for(j=0; j<n-1-i; j++)
            if(m[j].total>m[j+1].total)
            {
                temp=m[j];
                m[j]=m[j+1];
                m[j+1]=temp;
            }
}

```

```

}
int main(void)
{
    int i,n;
    struct Medal m[N];
    printf("请输入国家个数: ");
    scanf("%d",&n);
    printf("请输入%d个国家、金、银、铜牌个数: \n",n);
    for(i=0; i<n; i++)
    {
        scanf("%s%d%d%d",m[i].country,&m[i].gold,&m[i].silver,&m[i].bronze);
        total=m[i].gold+m[i].silver+m[i].bronze;
    }
    sort(m[N],n);
    printf("\n按奖牌总数由多到少排序后: \n");
    printf("国家\t\t金牌\t银牌\t铜牌\t奖牌总数\n");
    for(i=0; i<n; i++)
        printf("%-16s%d\t%d\t%d\t%d\n", m[i].country, m[i].gold, m[i].silver,
            m[i].bronze,m[i].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: 按程序功能要求, 编写程序并保存为 **C:\KS\A_4_1.c**。

程序功能: 某市企业管理测评中, 若企业得分低于70为较差企业, 以 “*” 表示; 若企业得分大于等于70且小于80为中等企业, 以 “**” 表示; 若企业得分大于等于80且小于90为良好企业, 以 “***” 表示; 若企业得分大于等于90为优秀企业, 以 “****” 表示。编写程序, 输入企业得分, 输出对应的符号表示。若输入小于0或者大于100, 则提示 “输入错误”。

运行示例 1:

```

请输入企业得分:
65
*

```

运行示例 2:

```

请输入企业得分:
88.5
***

```

运行示例 3:

```

请输入企业得分:
120
输入错误

```

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: 函数 `int isPalindrome(char str[])` 的功能是: 删除字符串 `str` 中的字母和数字字符之外的所有字符; 将所有大写字母转换为小写字母; 判断转换后的字符串是否是回

文串。如果是回文串，返回 1；否则返回 0。例如 aba、m2chhc2m 是回文串。

程序要求：(1)编写 isPalindrome 函数；

(2)编写主函数，输入一个字符串，调用此自定义函数并输出结果。

运行示例 1：

```
请输入字符串: A man, a plan, a canal: Panama
删除后: AmanaplanacanalPanama
转换后: amanaplanacanalpanama
是回文串
```

运行示例 2：

```
请输入字符串: blue SKY
删除后: blueSKY
转换后: bluesky
不是回文串
```

运行示例 3：

```
请输入字符串: Ye 'x' in121 Mixey
删除后: Yexin121Nixey
转换后: yexin121nixey
是回文串
```

(三)编程题（第 5 问，20 分）

编程要求：预提供的程序已经保存在 C:\KS\A_5.c 中，编写的程序继续保存在 C:\KS\A_5.c 中。

程序功能：世界杯小组赛采用积分制，每个小组 4 个球队，共进行 6 场比赛，每个球队胜一场球积 3 分，平一场球积 1 分，负一场球得 0 分，最后小组内根据各球队的积分排名决定进入下一轮比赛的球队。

本程序的任务是简单模拟 2022 世界杯小组赛的积分统计系统，统计每个球队的进球数、失球数、胜利场数、平局场数、失败场数、积分，然后计算排名。

结构体 match 的数据可以存储比赛的赛事，包括所在小组、比赛的 2 支球队及进球，声明如下：

```
struct match
{
    char group;           //小组
    char team1[20];       //球队1
    char team2[20];       //球队2
    int goal1;            //球队1的进球数
    int goal2;            //球队2的进球数
};
```

球队比赛成绩存放在结构体 team 类型的数组中，结构体 team 类型的定义为：

```
struct team
{
    char group;           //小组
    char team[20];        //球队
    int goal;             //进球数
    int lost;             //失球数
    int win;              //胜利场数
    int tie;              //平局场数
};
```

```

int defeat;           //失败场数
int score;           //积分，胜一场球积3分，平一场球积1分，负一场球得0分
int rank;            //排名
};

```

所有的球队信息和比赛信息已经在 main()函数中声明：

```

struct team t[]={{'A',"Qatar"},{'A',"Senegal"},{'A',"Holland"},{'A',"Ecuador"},
                 {'B',"Iran"},{'B',"England"},{'B',"USA"},{'B',"Wales"}};
struct match m[]={{'A',"Qatar","Ecuador",0,2}, {'A',"Senegal","Holland",0,2},
                  {'A',"Qatar","Senegal",1,3}, {'A',"Holland","Ecuador",1,1},
                  {'A',"Holland","Qatar",2,0}, {'A',"Ecuador","Senegal",1,2},
                  {'B',"England","Iran",6,2}, {'B',"USA","Wales",1,1},
                  {'B',"Wales","Iran",0,2}, {'B',"England","USA",0,0},
                  {'B',"Wales","England",0,3}, {'B',"Iran","USA",0,1}};

```

编程完成如下功能：

(1) 文件C:\KS\A_5.c中已经给出了 get_team() 函数的部分代码。请完善 get_team() 函数，通过 get_team() 获取球队的进/失球数、胜/平/负局数、以及积分。函数原型为：

```
void get_team(struct team t[], int n1, struct match m[], int n2);
```

其中，结构体数组 t 中已经保存 n1 个球队的小组号和球队名，结构体数组 m 中已经保存 n2 场赛事。

(2) 通过函数 display_match() 输出赛事信息。函数原型为：

```
void display_match(struct match m[], int n);
```

其中，结构体数组 m 中已经保存 n 场赛事信息。

(3) 通过函数 display_team() 输出球队比赛成绩（进/失球数、胜/平/负局数、积分）。函数原型为：

```
void display_team(struct team t[], int n);
```

其中，结构体数组 t 中已经保存 n 个球队的比赛成绩。

(4) 编写 main() 函数，调用上述各功能。main() 函数只起声明数据和调用函数的功能。

输出结果示例：

小组	赛事	比分
A	Qatar:Ecuador	0:2
A	Senegal:Holland	0:2
A	Qatar:Senegal	1:3
A	Holland:Ecuador	1:1
A	Holland:Qatar	2:0
A	Ecuador:Senegal	1:2
B	England:Iran	6:2
B	USA:Wales	1:1
B	Wales:Iran	0:2
B	England:USA	0:0
B	Wales:England	0:3
B	Iran:USA	0:1

小组	球队	进/失	胜/平/负	积分
A	Qatar	1/7	0/0/3	0
A	Senegal	5/4	2/0/1	6
A	Holland	5/1	2/1/0	7
A	Ecuador	4/3	1/1/1	4
B	Iran	4/7	1/0/2	3
B	England	9/2	2/1/0	7
B	USA	2/1	1/2/0	5
B	Wales	1/6	0/1/2	1

(四)编程题（第 6 问，30 分）

编程要求：在第 5 问的基础上，完成下列功能，编写的程序保存在 **C:\KS\A_6.c** 中。

程序功能：

(1)世界杯赛事信息可以通过读取文本文件 match.txt 得到,假设 match.txt 中包含**不超过 100 个**赛事信息。文件中的数据和结构体 match 对应, 分别为: 小组、球队 1、球队 2、球队 1 的进球数、球队 2 的进球数。文件数据如下:

```
match - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
A Qatar Ecuador 0 2
A Senegal Holland 0 2
A Qatar Senegal 1 3
A Holland Ecuador 1 1
A Holland Qatar 2 0
A Ecuador Senegal 1 2
B England Iran 6 2
B USA Wales 1 1
B Wales Iran 0 2
B England USA 0 0
B Wales England 0 3
B Iran USA 0 1
```

读取数据文件 match.txt 的 read()函数原型为:

```
int read(struct match m[]);
```

其中, 结构体数组 m 用于存放读取到的赛事信息, 函数返回读取到的信息记录数。

提示:如果采用 fscanf 语句,因为输入语句中第一个格式符为%c,需要在格式符中加入'\n', 具体格式为: fscanf(fp, "%c%s%s%d%d\n",)。

说明: 球队初始信息仍然通过结构体数组初始化获取。

(2)通过函数 score_sort()把球队按照小组升序和积分降序排序: 首先按小组的升序排序, 同

一小组内按照积分降序排序。函数原型为：

```
void score_sort(struct team t[], int n);
```

其中，结构体数组 t 中已经保存 n 个球队的信息。

(3) 通过函数 get_rank() 得到各球队在小组内的排名。函数原型为：

```
void get_rank(struct team t[], int n);
```

其中，结构体数组 t 中已经保存 n 个球队的信息。排名结果如下图所示：

小组	球队	进/失	胜/平/负	积分	排名
A	Holland	5/1	2/1/0	7	1
A	Senegal	5/4	2/0/1	6	2
A	Ecuador	4/3	1/1/1	4	3
A	Qatar	1/7	0/0/3	0	4
B	England	9/2	2/1/0	7	1
B	USA	2/1	1/2/0	5	2
B	Iran	4/7	1/0/2	3	3
B	Wales	1/6	0/1/2	1	4

提示：在排序的基础上求排名。

(4) 修改第 5 问中的函数 display_team(), 输出球队比赛成绩（包括排名）。函数原型仍为：

```
void display_team(struct team t[], int n);
```

其中，结构体数组 t 中已经保存 n 个球队的信息。

(5) 通过函数 search_match() 查询某球队的赛事信息。函数原型为：

```
int search_match(struct match m[], int n, struct match m_work[], char team[ ]);
```

其中，结构体数组 m 中已保存 n 条赛事信息，函数功能是查询球队名称为 team 的赛事信息，并保存至结构体数组 m_work 中，函数返回值为查询到的赛事信息数。

说明：该函数中不输出查找到的比赛信息，查找到的球队比赛信息在 main 函数中通过调用 display_match() 函数输出。

(6) 设置菜单，包含如下功能：

1. 赛事信息
2. 积分排名
3. 按球队查询赛事信息
0. 退出

程序开始、显示菜单前，需要调用 read() 函数从文件读入信息以获取全部赛事信息，并调用 get_team() 函数获取所有球队的信息（包括球队的进/失球数、胜/平/负局数、积分、排名）。

输出结果示例1：

输出结果示例2：


```

-----
1. 赛事信息
2. 积分排名
3. 按球队查询赛事信息
0. 退出
-----
请输入你的选择(0-3):1
  小组      赛事      比分
  A          Qatar:Ecuador  0:2
  A          Senegal:Holland 0:2
  A          Qatar:Senegal  1:3
  A          Holland:Ecuador 1:1
  A          Holland:Qatar   2:0
  A          Ecuador:Senegal 1:2
  B          England:Iran   6:2
  B          USA:Wales       1:1
  B          Wales:Iran      0:2
  B          England:USA     0:0
  B          Wales:England   0:3
  B          Iran:USA        0:1

-----
1. 赛事信息
2. 积分排名
3. 按球队查询赛事信息
0. 退出
-----
请输入你的选择(0-3):2
  小组  球队  进/失  胜/平/负  积分  排名
  A     Holland  5/1    2/1/0    7      1
  A     Senegal  5/4    2/0/1    6      2
  A     Ecuador  4/3    1/1/1    4      3
  A     Qatar    1/7    0/0/3    0      4
  B     England  9/2    2/1/0    7      1
  B     USA      2/1    1/2/0    5      2
  B     Iran     4/7    1/0/2    3      3
  B     Wales   1/6    0/1/2    1      4

```

```

-----
1. 赛事信息
2. 积分排名
3. 按球队查询赛事信息
0. 退出
-----
请输入你的选择(0-3):3
请输入想要查询的球队:USA
  小组      赛事      比分
  B          USA:Wales       1:1
  B          England:USA     0:0
  B          Iran:USA        0:1

-----
1. 赛事信息
2. 积分排名
3. 按球队查询赛事信息
0. 退出
-----
请输入你的选择(0-3):6
输入错误
-----
1. 赛事信息
2. 积分排名
3. 按球队查询赛事信息
0. 退出
-----
请输入你的选择(0-3):0
谢谢，再见!

```