

# 2019转计算机机考

这三道题为2019年转专业的C语言上机真题，答对一道得60分，两道80，三道100，建议复习c语言时看看结构体、排序，有能力的可以再看看C++中的stl库，可作为上机时应急使用。复习时重点与难点是字符串、指针、结构体的查找与排序。答题限时：60分钟。

## 题目1-数组的应用

输入格式：

输入在一行中给出序列中整数个数N ( $0 < N \leq 1000$ )，以及N个整数。数字间以空格分隔。

输出格式：

在一行中输出出现次数最多的整数及其出现次数，数字间以空格分隔。如果有多个重复出现次数一样的数，输出值最大的那个。

输入样例：

10 3 2 -1 5 3 4 3 0 3 2

输出样例：

3 4

代码示例：

```
#include<stdio.h>
int main(){           //有些冗余，以后再想想怎样优化
    int n,i=0,j,tmp,count,max=0;
    scanf("%d",&n);
    if (n==1){
        scanf("%d",&tmp);
        printf("%d 1\n",tmp);
        return 0;
    }
    int s[n+1],l[n+1];    //s[]存储次数，下标表示出现了多少次，相对应的出现这么多次的
                           整数
    while(n>=1){
        count=1;
        scanf("%d",&tmp);
        l[i++]=tmp;        //l[]存储输入的整数，下标从0开始
        for(j=0;j<i-1;j++){
            if (tmp==l[j]){
                s[++count]=tmp;    //下标从2开始，因为目标数字唯一，其他循环时不会到
                s[max],即已有下标最大存储了值的空间
            }
        }
        if(count>max){
            max=count;    //更新max
        }
        n--;
    }
    printf("%d %d\n",s[max],max);
    return 0;
}
```

## 题目2-ASCII码的应用

### ISBN识别码判断

每一本正式出版的图书都有一个ISBN号码与之对应，ISBN码包括9位数字、1位识别码和3位分隔符，其规定格式如“x-xxx-xxxxx-x”，其中符号“-”就是分隔符（键盘上的减号），最后一位是识别码，例如0-670-82162-4就是一个标准的ISBN码。ISBN码的首位数字表示书籍的出版语言，例如0代表英语；第一个分隔符“-”之后的三位数字代表出版社，例如670代表维京出版社；第二个分隔符后的五位数字代表该书在该出版社的编号；最后一位为识别码。

识别码的计算方法如下：

首位数字乘以1加上次位数字乘以2……以此类推，用所得的结果mod 11，所得的余数即为识别码，如果余数为10，则识别码为大写字母X。例如ISBN号码0-670-82162-4中的识别码4是这样得到的：对067082162这9个数字，从左至右，分别乘以1, 2, ..., 9,再求和，即 $0\times 1+6\times 2+\dots+2\times 9=158$ ，然后取 $158 \bmod 11$ 的结果4作为识别码。

你的任务是编写程序判断输入的ISBN号码中识别码是否正确，如果正确，则仅输出“Right”；如果错误，则输出你认为是正确的ISBN号码。

程序运行结果示例1：

0-123-41562-4

Right

程序运行结果示例2：

0-123-41562-7

0-123-41562-4

输入格式：

输入只有一行，是一个字符序列，表示一本书的ISBN号码（保证输入符合ISBN的格式要求）。

输出格式：

输入的ISBN号码的识别码正确，输出信息：“Right”

输入的ISBN号码的识别码错误，按照规定的格式，输出正确的ISBN号码（包括分隔符“-”）

代码示例：

```
#include <stdio.h>
int main()
{
    char str[20];
    int i, j, sum=0, x=1;
    scanf("%s", str);
    for(i=0; i<11; i++)//11位和12位分别为-和识别码 不需要计算
    {
        if(str[i]=='-')//分隔符不计算
        {
            sum=sum+0;
        }
        else
        {
            sum=sum+(str[i]-48)*x;//主要-48 因为是ASCII码
            x++;
        }
    }
    sum=sum%11;
    if(sum==(str[12]-48) || (str[12]=='X'&&sum==10))//相等或者为x时
    {
        printf("Right\n");
    }
```

```

}
else
{
    if(sum==10)//当识别码为x时
    {
        for(j=0; j<12; j++)//输出
        {
            printf("%c",str[j]);
        }
        printf("x");
    }
    else
    {
        for(j=0; j<12; j++)//输出前面12个数字
        {
            printf("%c",str[j]);
        }
        printf("%d\n",sum);//输出正确的识别码
    }
}
return 0;
}

```

### 题目3-结构体排序

假设学生的基本信息包括学号、姓名、三门课程成绩以及个人平均成绩，定义一个能够表示学生信息的结构类型。输入n（n<50）个学生的成绩信息，按照学生的个人平均分从高到低输出他们的信息。如果平均分相同，按输入的先后顺序排列。

输入格式:

输入一个正整数n(n<50)，下面n行输入n个学生的信息，包括：学号、姓名、三门课程成绩（整数）。

输出格式:

输出从高到低排序后的学生信息，包括：学号、姓名、平均分（保留两位小数）。

输入样例:

```

3
101 Zhang 78 87 85
102 Wang 91 88 90
103 Li 75 90 84

```

输出样例:

```

102,Wang,89.67
101,Zhang,83.33
103,Li,83.00

```

代码示例:

```

#include<stdio.h>
struct student{//定义一个结构体分别放入学号姓名，三个成员的分数
    int num;
    char name[10];
    int s[3];
    double ave;
};

```

```

int main()
{
    struct student a[1000]; //建议将空间定义的大一些，以免造成空间储存不足
    int n, i, j;
    int sum=0;
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        scanf("%d %s %d %d %d", &a[i].num, &a[i].name, &a[i].s[0], &a[i].s[1], &a[i].s[2]);
        sum=a[i].s[0]+a[i].s[1]+a[i].s[2]; //求和
        a[i].ave=(sum*1.0)/3; //求三个数的平均数
    }
    struct student t; //结构类型要一致
    for(i=0; i<n-1; i++)
    {
        for(j=0; j<n-i-1; j++)
        {
            if(a[j].ave<a[j+1].ave)
            {
                t=a[j+1];
                a[j+1]=a[j];
                a[j]=t;
            }
        }
    }
    printf("-----输出-----\n");
    for(i=0; i<n; i++){
        printf("%d,%s,%.2lf\n", a[i].num, a[i].name, a[i].ave); //分别输出学号，姓名，以及平均成绩
    }
    return 0;
}

```