

一、写出下列程序段的运行结果（40 分,每小题 4 分）

题号	参考答案
1	21,10,1,21,10,1
2	5 4 3 2 1
3	f=123.00,c=0.00,t=5.14,v=0.5142
4	Equal! k=11,k=13,k=b
5	2
6	u u n university *****
7	3, 1, 1
8	3 3 8 3
9	0.67,2
10	-1,6

参考答案：

序号	行号	改正后的代码
1	第 4 行	int count[10];
2	第 7 行	while((c=getchar())!='\n')
3	第 9 行	if(c>='0' && c<='9')
4	第 10 行	count[c-'0']++;
5	第 14 行	for(j=0;j<9-i;j++)
6	第 17 行	(count[j]==count[j+1]) && (d[j] > d[j+1]))
7	第 19 行	tmp=count[j],count[j]=count[j+1],count[j+1] = tmp;
8	第 20 行	tmpd = d[j], d[j]= d[j+1] ,d[j+1]=tmpd;
9	第 24 行	for(i=0;i<10;i++)
10	第 26 行	if(count[i] == 0)
11	第 28 行	printf("数字%c 出现了:%d 次\n",d[i],count[i]);

1. (12分) 请编写一个程序，对给定的一个字符串（仅含字母，长度不超过100），统计其中连续出现的相同字符个数，并转换为“字母+数量”格式输出。例如，输入“AAABBBCCCab”，输出“A3 B3 C3 a1 b1”。

```
#include <stdio.h>
int main()
{ char s[101]; //字符数组至少要有101个单元
  char ch=0;   //变量定义及初始化
  int i=0,n;
  gets(s); //scanf("%s",s); //正确读取
  do
  { if (ch!=s[i]) //与前一个字符不同
    { if (ch) //输出前一个字符的编码
      printf("%c%d ",ch,n);
      ch=s[i]; //新字符
      n=1;
    }
    else //与前一个字符相同，计数
      n++;
  } while (s[i++]); //循环
  return 0;
}
```

2. (14分) 请编写一个程序, 对给定的若干个0 ~ 9数字, 进行排列使得到的正整数尽可能小(0不能做首位数字)。

输入格式: 第一行给出数字个数 n ($n \leq 100$); 第二行给出0 ~ 9范围内的 n 个数字(至少含1个非0数字), 中间用一个空格分隔。

输出格式: 给出可构成的最小正整数。

输入样例:

10
2 4 0 5 0 3 0 2 1 0

输出样例:

1000022345

```
#include <stdio.h>
int main()
{
    int a[100];    //数组至少要有100个单元
    int i,j,n,flag=1,t;    //变量定义及初始化

    scanf("%d",&n);    //读取数字个数
    for (i=0;i<n;i++) //读取各个数字
        scanf("%d",&a[i]);
```

```
    for (i=0;flag && i<n-1;i++) //冒泡排序(可不设flag)
    {
        flag=0;
        for (j=0; j<n-1-i;j++)
            if (a[j]>a[j+1])
            {
                t=a[j]; a[j]=a[j+1]; a[j+1]=t;
                flag=1;
            }
    }
    for (i=0;!a[i];i++); //i指向最小的那个非0数字
    if (!a[0]) //与第一个0对换
    {
        a[0]=a[i];
        a[i]=0;
    }
    for (i=0;i<n;i++) //输出
        printf("%d",a[i]);
    return 0;
}
```

3. (14分) 假定四则运算表达式由操作数(数字及至多1个小数点)、运算符(仅+、-、*、/)以及小括号(和)组成。请编写一个程序,对于给定的一个四则运算表达式,识别出其中的操作数,顺序存储在二维字符数组 Ops[10][10] 中(每个一行,不会越界),并依次输出在屏幕上。

输入格式: 在一行中给出长度不超过 100 个字符的四则运算表达式,仅由上述字符组成无空格,无需做正确性检查。

输出格式: 依次输出式中各操作数,每个一行。

输入样例:

32*(62-5)

Ops 数组的存储样例如下所示:

3	2	\0	...
6	2	\0	...
5	\0	...	
...			

输出样例:

32

62

5

```
#include <stdio.h>
```

```
#include <string.h>
```

```
//如果用到字符串处理函数的话
```

```
int main()
```

```
{ char s[101]; //数组至少要有101个单元
```

```
char Ops[10][10], t[10]; //变量定义及初始化
```

```
int i=0,n=0,j=0;
```

```
gets(s); //读取表达式
```

```
while (s[i]) //循环
```

```
{ while (s[i]>='0'&& s[i]<='9' || s[i]=='.') //识别一个操作数
```

```
t[j++]=s[i++];
```

```
if (j)
```

```
{ t[j]='\0'; //在串尾放上结束标志
```

```
strcpy(Ops[n++],t); //存储到Ops中
```

```
//在此做输出也可
```

```
j=0;
```

```
}
```

```
if (s[i]) i++; //跳过非操作数
```

```
}
```

```
for (i=0;i<n;i++) //输出
```

```
{
```

```
printf("%s\n",Ops[i]);
```

```
}
```

```
return 0;
```

```
}
```