

C语言程序设计样卷讲评

以某年期中测试为例

题型

- 一、写出下列程序段的运行结果（10题，40分）
- 二、改错题（1题，共20分）
- 三、编程题（3题，40分。注意：程序中添加必要的注释）

19-3-3

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。

19-3-3

输入样例：

32*(62-5)

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，1分

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，2分
    int i=0,n=0,j=0;

    gets(s); //读取表达式，1分
    while (s[i]) { //循环，1分
        while (s[i]>='0'&& s[i]<='9' || s[i]=='.')
            t[j++]=s[i++]; //识别一个操作数，3分
        if (j) {
            t[j]='\0'; //在串尾放上结束标志，1分
            strcpy(Ops[n++],t); //存储到Ops中，2分
            //在此做输出也可
            j=0;
        }
        if (s[i])
            i++; //跳过非操作数，1分
    }
    for (i=0;i<n;i++) { //输出，2分
        printf("%s\n",Ops[i]);
    }
    return 0;
}
```

19-2-1

本题中给定的程序预期实现如下功能：

从键盘出入一串字符，统计这一串字符中各个数字出现的次数，将统计结果存入数组中；然后出现的频率对按各个数进行降序排序，频率相同的数字则根据数字自身从小到大排序；最后依次输出统计结果，每个数字单独一行显示，次数为0的数字不显示。例如：输入12x&34ac2345*C(+3456，程序输出如下：

本程序前10行包含了4个错误，其余行包含了若干错误，请指出错的行并改正。

数字3出现了：3次
数字4出现了：3次
数字2出现了：2次
.....

19-2-1

```
#include <stdio.h>
int main()
{
    int count[ ];
    char c,tmpd, d[]="0123456789";
    int i,j,tmp;
    while(c=getchar()!='\n')
    {
        if(0<= c <=9)
            count[c]++;
    }
    for(i =0; i<9; i++)
    {
        for(j=i; j<9; j++)
        {
```

```
/* 第 1 行 */
```

```
/* 第 2 行 */
```

```
/* 第 3 行 */
```

```
/* 第 4 行 */
```

```
/* 第 5 行 */
```

```
/* 第 6 行 */
```

```
/* 第 7 行 */
```

```
/* 第 8 行 */
```

```
/* 第 9 行 */
```

```
/* 第 10 行 */
```

```
/* 第 11 行 */
```

```
/* 第 12 行 */
```

```
/* 第 13 行 */
```

```
/* 第 14 行 */
```

```
/* 第 15 行 */
```

```
        if((count[j]<count[j+1] ) || /* 第 16
行 */
        (count[j]==count[j+1]) &(d[j] < d[j+1]))
/* 第 17 行 */
        {
            tmp=count[j], count[j+1]=count[j],
count[j] = tmp; /* 第 18 行 */
            tmpd = d[j], d[j+1]= d[j] ,d[j]=tmpd; /* 第 19 行 */
        } /* 第 20 行 */
    } /* 第 21 行 */
    } /* 第 22 行 */
    for(i=0;i<=10;i++) /* 第 23 行 */
    { /* 第 24 行 */
        if(count[i]=0) /* 第 25 行 */
            break; /* 第 26 行 */
        printf(" 数 字 %d 出 现 了 :%d 次
\n",d[i],count[i]); /* 第 27 行 */
    } /* 第 28 行 */
} /* 第 29 行 */
/* 第 30 行 */
```

19-2-1

```
#include <stdio.h>
int main()
{
    int count[ ];
    char c,tmpd, d[]="0123456789";
    int i,j,tmp;
    while(c=getchar()!='\n')
    {
        if(0<= c <=9)
            count[c]++;
    }
    for(i =0; i<9; i++)
    {
        for(j=i; j<9; j++)
        {
```

/* 第 1 行 */

/* 第 2 行 */

/* 第 3 行 */

/* 第 4 行 */

/* 第 5 行 */

/* 第 6 行 */

/* 第 7 行 */

/* 第 8 行 */

/* 第 9 行 */

/* 第 10 行 */

/* 第 11 行 */

/* 第 12 行 */

/* 第 13 行 */

/* 第 14 行 */

/* 第 15 行 */

```
        if((count[j]<count[j+1] ) ||
行 */
        (count[j]==count[j+1]) &(d[j] < d[j+1]))
/* 第 17 行 */
        {
            tmp=count[j],
count[j] = tmp;
            tmpd = d[j], d[j+1]= d[j] ,d[j]=tmpd;
        }
    }
    for(i=0;i<=10;i++)
    {
        if(cout[i]=0)
            break;
        printf(" 数 字 %d 出 现 了 :%d 次
\n",d[i],count[i]);
    }
}
```

/* 第 16 行 */

/* 第 17 行 */

/* 第 18 行 */

/* 第 19 行 */

/* 第 20 行 */

/* 第 21 行 */

/* 第 22 行 */

/* 第 23 行 */

/* 第 24 行 */

/* 第 25 行 */

/* 第 26 行 */

/* 第 27 行 */

/* 第 28 行 */

/* 第 29 行 */

/* 第 30 行 */

20-1-4

- 4. (4分)

```
int a=1;  
while (a++<=1)  
while (a++<=2);  
printf("%d", a);
```

5

20-2-1

- 以下程序实现：从键盘出入N个互不相同的字符串（字符数小于80且不含空格），若输入的字符串曾经输入，则给出提示信息后重新输入。最后将字符串按输入顺序逆序输出。
- 程序中存在若干错误（第10行前包含4个错误），请找到错误所在位置，并修改。

20-2-1

```
#include <stdio.h> /*第1行*/
#define 10 N /*第2行*/
int main() /*第3行*/
{ /*第4行*/
    char str[N][80]={0}; /*第5行*/
    char strtemp[80]={0}; /*第6行*/
    int j; /*第7行*/
    int count; /*第8行*/
    while(count<N) /*第9行*/
    { /*第10行*/
        scanf("%s",&strtemp); /*第11行*/
        for(i=0;i<=count;i++) /*第12行*/
        { /*第13行*/
            if(strtemp==str[i]) /*第14行*/
            { /*第15行*/
```

20-2-1

```
        printf("already exists!\n"); /*第16行*/
        continue;                  /*第17行*/
    }                               /*第18行*/
}                                   /*第19行*/
if(i==count+1)                    /*第20行*/
{                                  /*第21行*/
    strcpy(str[i],strtemp); /*第22行*/
    ++count;                /*第23行*/
}                             /*第24行*/
};                             /*第25行*/
for (i=0;i<N;i++)             /*第26行*/
{                               /*第27行*/
    printf("NO.%d string is: %s \n",i,str[N-i]); /*第28行*/
}                               /*第29行*/
return;                        /*第30行*/
}                               /*第31行*/
```