
一、写出下列程序段的运行结果（40 分）

1.（4 分）

```
char c='k';
int i=1, j=2, k=3;
float x=0.998;
int a1 =i+1 || j+1;
int a2 = !!!k;
int a3 = (int)x;
int a4 = (j=8)&& c&& i==1;
printf("%d, %d, %d, %d",
        a1, a2, a3, a4);
```

2.（4 分）

```
float a,b;
a=6.7;
if (a==6.7)
    printf("A, ");
else
    printf("B, ");
b=1.5+1/5;
printf("%.2f", b);
```

3.（4 分）

```
int a;
a=100>90>80;
printf("%d, ",a);
unsigned char b=-1;
printf("%d", b);
```

4.（4 分）

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

5.（4 分）

```
int i, s=0;
for(i=1;i<10;i++)
{
    if (i==3 || i==5) continue;
    if (i==6) break;
    s+=i;
}
printf("%d", s);
```

6.（4 分）

```
int a=1,b=2,t,c;
if (a<b)
    t=a; a=b; b=t;
printf("%d, %d, %d, ", a, b, t);
a=2; b=-1; c=2;
if(a<b)
if(b<0) c=0;
else c+=1;
printf("%d ", c);
```

7.（4 分）

```
int a[][3]={5,4,3,2,1};
printf("%d, ", a[1][1]);

int b[5]={1,2,3,4,5};
int i=0,k=0;
while (i<5 && b[++i]%2)
    k=k+b[i];
printf("%d", k);
```

8. (4 分)

```
int s;  
scanf("%d", &s);  
while (s>0)  
{  
    switch(s)  
    {  
        case 1: printf("%d, ", s+1);  
                break;  
        case 2: printf("%d, ", s+2);  
                break;  
        default: printf("%d, ", s+3);  
                 break;  
    }  
    scanf("%d", &s);  
}
```

运行时，若输入 **1 2 0**<回车>，则输出结果是？

9. (4 分)

```
char a[10]="123456";  
printf("%c, ", a[0]);  
printf("%d\n", strlen(a));  
char b[5][2]={'\0'};  
strcpy(b[0], "hello");  
strcpy(b[1], "world");  
b[2][0]=0;  
printf("%s", b);
```

10. (4 分)

```
int a[]={1,2,3,4,5,6,7,8,9,10};  
int low=0,high=9,mid;  
int k=4;  
while (low<=high)  
{  
    mid=(low+high)/2;  
    if (a[mid]==k)  
        break;  
    else if (k>a[mid])  
        low=mid+1;  
    else  
        high=mid-1;  
}  
printf("%d", mid);
```

二、改错题 (20 分)

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。

```

#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*/
                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*/

```

三、编程题（40 分） 注意：程序中请添加必要的注释

- 1.（12 分）回文素数（质数）指的是：素数 n 从左向右和从右向左读其数值相同。求出所有的三位回文素数，以每行 4 个数的形式显示在屏幕上，并统计它们的数量 $count$ 。输出结果如下图所示：

```
101   131   151   181
191   313   353   373
383   727   757   787
797   919   929
count=15
```

- 2.（14 分）校门口有 100 个车位（编号为 1, 2, ..., 100）停满了共享单车，为了移走部分单车以方便通行，学校派了 N 个志愿者去移车，每个志愿者的任务单有两个数字 L 和 R ($1 \leq L \leq R \leq 100$)，表示他负责将编号处于区间 $[L, R]$ 之间的单车全部移走，由于工作上的疏忽，志愿者领到的区间可能有重叠。编写程序，计算当所有志愿者都完工的时候，车位上还剩下几辆车。

输入样例：2

输出样例：10

1 60

注：2 名志愿者分别移走 1~60, 4~90

4 90

号车，还剩下 91~100 号车，余 10 辆

- 3.（14 分）国际标准书号 ISBN 共 17 位，由 13 位数字和 4 位连字符（“-”）组成。最后 1 位数字表示校验位，校验规则为：对从左至右前 12 位数字，分别计算奇位数之和 s_1 与偶位数之和 s_2 ；将 s_2 乘以 3，与 s_1 相加，结果的个位数记为 sum ，即 $sum = (s_2 * 3 + s_1) \bmod 10$ ；则校验位 c 为： $c = 10 - sum$ 。

例如，C 语言课本的 ISBN 号为：978-7-302-48144-7。

其奇位数之和 $s_1 = 9 + 8 + 3 + 2 + 8 + 4 = 34$ ，偶位数之和 $s_2 = 7 + 7 + 0 + 4 + 1 + 4 = 23$

$sum = (23 * 3 + 34) \bmod 10 = 3$ ，校验位 $c = 10 - sum = 7$

编写程序，读入一个标准格式的 ISBN 号，判断其校验位是否正确，正确输出 1，错误输出 0。

输入样例 1： 978-7-302-48144-7

输出样例 1： 1

输入样例 2： 978-92-95055-02-7 （注：校验位应该为 5）

输出样例 2： 0