
一、写出下列程序段的运行结果（40 分）

1.（4 分）

```
char str[]="bXa!c2$3M?a2k",c;
int i=0,d=0,u=0;
while((c=str[i]) != '\0')
{
    i++;
    if(c>='A' && c<='Z')
        u++;
    else if(c>='0' && c<='9')
        d++;
}
printf("u=%d, d=%d",u,d);
```

2.（4 分）

```
int i=1, j=2, a=3, b=0;
if (++i>1 || j++>2)
    a= i && j-2;
b=!i || j+3;
printf("%d, %d, %d, %d",
        i, j, a, b);
```

3.（4 分）

```
void f(int n,int k)
{
    if(n>=k) f(n/k, k);
    printf("%d", n%k);
}
int main()
{
    f(200, 8); printf("\n");
    f(55, 2);
    return 0;
}
```

4.（4 分）

```
int d []={11, 57, 22, 16, 83, 24,
          89, 6, 73, 0};
int i=0,m=0,n=0;
```

```
while(d[i]/10)
```

```
{
    switch(d[i]%10)
    {
        case 1:case 2:case 3:
        case 4:case 5: m++;
                        break;
        case 6:case 7:case 8:
        case 9: n++; break;
    }
    i++;
}
printf("m=%d, n=%d",m,n);
```

5.（4 分）

```
char msg[]="abcdefghij";
char* p = msg;
int count=0,k=1;
while(*p)
{
    putchar(*p);
    count++;
    if(count==k)
    {
        count=0;
        k++; printf("\n");
    }
    p++;
}
```

6.（4 分）

```
enum {A=7, B, C, M=10, N,
X=1, Y, Z} c1=B, c2=M, c3=X, t;
printf("%d, %d, ", c1-c2,
        c2>c3?c2:c3);
if(c1 > c2) t=c1,c1=c2,c2=t;
if(c1 > c3) t=c1,c1=c3,c3=t;
if(c2 > c3) t=c2,c2=c3,c3=t;
```

```
printf("%d, %d", c1,c3);
```

7. (4 分)

```
int a[4][4]={1,2,3,4,5,6,7,8,9};
int i,j;
for(i=0;i<4;i++)
{
    for(j=0;j<=i;j++)
        printf("%d ",a[i][j]);
    printf("\n");
}
```

8. (4 分)

```
int f(int * p)
{
    static int a=5;
    int b ;
    b = ++a*5-2;
    *p = b;
    return a;
}
int main()
{
    int a=10;
    int b;
    b=f(&a);
    printf("a=%d, b=%d\n",a,b);
    b=f(&a);
    printf("a=%d, b=%d\n",a,b);
    return 0;
}
```

9. (4 分)

```
struct {
    int a, b, c;
} s[3]= {1,2,3,4,5,6,7,8,9};
```

二、改错题 (20 分)

某数据文件以二进制方式存有若干(小于 100)个学生的数据, 函数 **load** 把

```
int ave,sum;
```

```
ave=(s[1].a + s[1].b+s[1].c)/3;
sum=s[0].a + s[1].a+s[2].a;
printf("%d, %d", ave, sum);
```

10. (4 分)

```
FILE *fp;
int a = 123;
long pos;
float x = 2020.1291;
char msg[]="Welcome to
XMU!";
```

```
fp=fopen("save.dat","w");
fprintf(fp,"%o,%0.2f,%s",
        a,x,msg);
fclose(fp);
```

```
fp = fopen("save.dat","r");
fscanf(fp,"%d",&a);
fseek(fp,2L,1);
fscanf(fp,"%f",&x);
fseek(fp,1L,1);
fgets(msg,5,fp);
pos = ftell(fp);
printf("a=%d, pos=%d\n",
        a, pos);
printf("x=%0.1f, msg=%s",
        x, msg);
fclose(fp);
```

数据存入数组 **a** 中，函数 **sort** 将学生数据按照成绩（**score** 项）从大到小排列，并将结果显示在屏幕上。

程序中存在不少于 10 个错误，指出错误所在位置并改正。

```
#include <stdio>                                /*第 1 行*/
#define SIZE 100                                /*第 2 行*/
struct Stu                                     /*第 3 行*/
{                                                /*第 4 行*/
    char name[20];                             /*第 5 行*/
    char sex;                                  /*第 6 行*/
    double score;                             /*第 7 行*/
};                                              /*第 8 行*/
int load(struct Stu student[]);                /*第 9 行*/
{                                                /*第 10 行*/
    FILE fp;                                  /*第 11 行*/
    int i;                                    /*第 12 行*/
    int count =0;                             /*第 13 行*/
    if((fp=fopen("d:\\data.dat","rb"))==NULL) /*第 14 行*/
    {                                           /*第 15 行*/
        printf("cannot open file\n");        /*第 16 行*/
        return 0;                             /*第 17 行*/
    }                                          /*第 18 行*/
    for(i=0; i<SIZE; i++)                     /*第 19 行*/
        if(fread(&student[i],sizeof(struct Stu),1,fp)!=1)/*第 20 行*/
            count++;                          /*第 21 行*/
        else                                  /*第 22 行*/
            if(feof(fp))                      /*第 23 行*/
                break;                       /*第 24 行*/
    fclose(fp);                               /*第 25 行*/
    return count;                             /*第 26 行*/
}                                              /*第 27 行*/
void sort(struct Stu student[],int num) /*第 28 行*/
{                                              /*第 29 行*/
    int i,j,k;                               /*第 30 行*/
```

```

for(i=0;i<num-1;i++)          /*第 31 行*/
{                               /*第 32 行*/
    k=i;                       /*第 33 行*/
    for(j=i+1;j<=num-1;j++)    /*第 34 行*/
        if(student[j].score<student[k].score) /*第 35 行*/
            k=student[j].score; /*第 36 行*/
    t=student[k];              /*第 37 行*/
    student[k]=student[i];     /*第 38 行*/
    student[i]=t;              /*第 39 行*/
}                               /*第 40 行*/
}                               /*第 41 行*/
int main()                     /*第 42 行*/
{                               /*第 43 行*/
    struct Stu a[SIZE];        /*第 44 行*/
    int i;                     /*第 45 行*/
    int count=0;               /*第 46 行*/
    load(a);                   /*第 47 行*/
    printf("Total:%d\n",count); /*第 48 行*/
    sort(a,SIZE);              /*第 49 行*/
    for(i=0; i<=count; i++)    /*第 50 行*/
    {                           /*第 51 行*/
        printf("name:%s\t",a[i].name); /*第 52 行*/
        printf("sex :%c\t",a[i].sex);   /*第 53 行*/
        printf("score :%.2ld\n",a[i].score); /*第 54 行*/
    }                           /*第 55 行*/
    return 0;                  /*第 56 行*/
}                               /*第 57 行*/

```

三、编程题（40 分） 注意：程序中请添加必要的注释

1.（12 分）键盘输入一个二进制格式的字符串，转换为无符号十进制数显示。

要求二进制输入串的格式为：必须有前缀（0b 或 0B）；最多 4 个字节；数符只有 0 和 1。例如输入 0b00111111 或 0B00111111 都符合要求，此时程序需显示转换结果 63；如果输入的字符串格式不符合以上要求，输出"error"。

提示：存在多种不符合要求的情况。

2. 为疫情防控的需要，需统计个人的行程信息。若已经获得某人的行程日志，每条日志包含到访日期和到访城市，顺序随机。编写程序，实现以下功能：

（1）设计结构体类型 Date 和 Log，Date 存储日期，包含年、月和日 3 个域；Log 存储一条日志，包含到访日期（Date 型）、到访城市 2 个域；（2 分）

（2）设计 sort 函数，从键盘读入行程日志，按日期先后顺序存储在 Log 型结构体数组中并通过参数返回给主函数，返回值为日志条数 N；（6 分）

（3）设计 print 函数，在屏幕上按日期先后顺序列出到访的不同城市，返回值为列出的城市个数。（6 分）

输入输出的格式说明如下：

【输入格式】第一行为 1 个整数，即日志条数 N， $N \leq 1000$ 。后继 N 行，每行对应一条日志，包含表示到访日期的年月日的 3 个整数 y、m、d 和 1 个表示到访城市的字符串 s，s 不超过 20 个字符。

【输出格式】按时间先后顺序输出，每个城市一行。

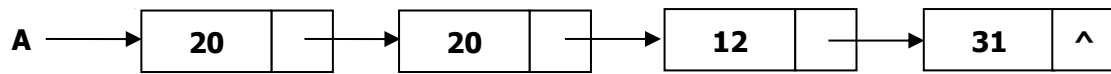
【输入样例】

```
5
2020 12 17 FJ_Xiamen
2020 12 22 FJ_Xiamen
2020 12 25 ZJ_Hangzhou
2020 12 18 FJ_Quanzhou
2020 12 30 FJ_Quanzhou
```

【输出样例】

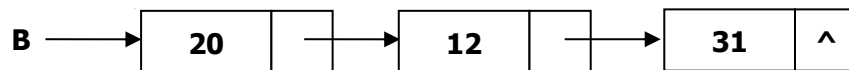
```
FJ_Xiamen
FJ_Quanzhou
ZJ_Hangzhou
```

3. 用链表存储一组类型为int的整数。下图所示是4个整数20， 20， 12， 31被存储在链表A的情形：



根据以上描述，完成以下问题：

- (1) 规定两个成员名必须为d和next，写出链表结点struct N的定义；（2分）
- (2) 编写函数：int cnt(struct N* head,int n)，返回链表head存储的所有整数中，出现整数n的次数。例如：cnt(A, 20)返回2，cnt(A, 21)返回0；（4分）
- (3) 编写函数cp，形参是A，功能是按结点出现顺序复制链表A，但对其中重复的数字，只保留第一个，函数返回复制的新链表B，原链表A不允许做任何修改。例如执行cp(A)后，得到的新链表B如下：（8分）



提示：允许调用(2)的 cnt 函数。

参考答案

一、(40 分) 评分标准：每小题 4 分

题号	答案	备注
1	u=2, d=3	各 2 分，逗号不计分
2	2, 2, 0, 1	各 1 分，逗号不计分
3	310 110111	每行 2 分
4	m=4, n=3	各 2 分，逗号不计分
5	a bc def ghij	每行 1 分
6	-2, 10, 1, 10	各 1 分，逗号不计分
7	1 5 6 9 0 0 0 0 0 0	每行 1 分
8	a=28, b=6 a=33, b=7	每行 2 分，逗号不计分
9	5, 12	各 2 分，逗号不计分
10	a=173, pos=16 x=20.1, msg=Welc	每行 2 分，逗号不计分

二、改错题 (20 分)

共 11 个错，找到给 1 分，修改正确给 1 分，满分 20，即允许一个错误没找到
仅仅找到错误，未修改或未说明错误原因，不给分。

1. 第 1 行，**stdio** 改为 **stdio.h**
2. 第 9 行，删除;
3. 第 11 行，**fp** 改为 ***fp**
4. 第 20 行，**!=** 改为 **==** (或 1 改为 0)
5. 第 30 行 (或 29 行)，增加 **struct Stu t;**
6. 第 36 行，**k=student[j].score;** 改为 **k=j**
7. 第 35 行，**<** 改为 **>**
8. 第 47 行，改为 **count=load(a)**
9. 第 49 行，**SIZE** 改为 **count**
10. 第 50 行，**<=** 改为 **<** (或 **count** 改为 **count-1**)
11. 第 54 行，**ld** 改为 **lf**

三、编程题

2. (12 分)

```
#include <stdio.h>
#include <string.h>      //字符串库函数, 1 分
int main()
{
    char binx[35];        //字符串长度>=35, 1 分
    unsigned int decx;    //定义为无符号整数, 1 分
    scanf("%s",binx);     //输入, 1 分, 用 gets 也可以

    if (strlen(binx)>34)   //长度判断, 1 分
    {
        printf("error");
        return 0;
    }

    if (binx[0]!='0' || (binx[1]!='b'&&binx[1]!='B')) //前缀判断, 1 分
    {
        printf("error");
        return 0;
    }

    for (int i=2; binx[i]!='\0'; i++)                //数符判断, 2 分
    {
        if ((binx[i]!='0')&&(binx[i]!='1'))
        {
            printf("error");
            return 0;
        }
    }

    decx=0;                                           //初值 1 分
    for (int i=2; binx[i]!='\0'; i++)                //二进制转换为十进制逻辑, 2 分
    {
        decx=decx*2+(binx[i]-'0');
    }

    printf("%u", decx);                               //无符号数显示, 1 分

    return 0;
}
```


2. (14 分)

```
#include <stdio.h>
#include <string.h>
struct Date {    //结构体类型 Date 定义, 1 分
    int y,m,d;
};
struct Log {    //结构体类型 Log 定义, 1 分
    struct Date date;
    char city[21]; //长度大于等于 21 均可
};

int sort(struct Log r[]) { //函数首部, 1 分
    int N,i,j,changed=1;
    struct Log temp;

    scanf("%d",&N);    //读入数据, 2 分
    for (i=0;i<N;i++)
        scanf("%d %d %d %s",&(r[i].date.y),&(r[i].date.m),
&(r[i].date.d),r[i].city);
    for (i=0;i<N-1 && changed;i++) { //排序算法, 2 分
        changed=0;
        for (j=0;j<N-2-i;j++) {
            if (r[j].date.y>r[j+1].date.y
|| r[j].date.y==r[j+1].date.y && r[j].date.m>r[j+1].date.m
|| r[j].date.y==r[j+1].date.y && r[j].date.m==r[j+1].date.m
&& r[j].date.d>r[j+1].date.d) { //日期早晚比较, 1 分
                temp = r[j];
                r[j]=r[j+1];
                r[j+1]=temp;
                changed=1;
            }
        }
    }
    return N;
}

int print(struct Log r[],int N) { //函数首部, 1 分
    char citylist[1000][21];
    int i,j,count=0;
    for (i=0;i<N;i++) {    //除重, 2 分
        for (j=0;j<count;j++)
            if (strcmp(r[i].city,citylist[j])==0)
                break;
        if (j==count)
```

```

        strcpy(citylist[count++],r[i].city);
    }
    for (j=0;j<count;j++) //输出, 1 分
        printf("%s\n",citylist[j]);
    return count;
}

int main() { //主函数, 2 分
    struct Log route[1000];
    int N;
    N=sort(route);
    print(route,N); //或者合并写成: print(route, sort(rount));
    return 0;
}

```

3. (14 分)

```
//(1)
struct N {
    int d; //1 分
    struct N *next; //next 加指针, 1 分
};

//(2)
int cnt(struct N* head, int n)
{
    struct N *p=head;
    int m=0; //初值 1 分
    while (p != NULL) //终止条件 1 分
    {
        if (p->d == n) //计数 1 分
            m++;
        p=p->next; //后续 1 分
    }
    return m;
}

//(3)
struct N* cp(struct N* A) //返回值类型, 1 分
{
    struct N* B = NULL, *temp1, *temp;
    struct N* p = A;
    while (p != NULL)
    {
        if (cnt(B, p->d)==0) //判断结点是否在 B 出现, 2 分
        {
            temp = (struct N*)malloc(sizeof(struct N)); //malloc 1 分
                                                    //sizeof 1 分, 强制转换不计分
            temp->d = p->d;
            temp->next = NULL; //链表尾置空, 1 分

            //插入结点逻辑, 2 分
            if (B == NULL)
                B = temp;
            else
                temp1->next = temp;
            temp1 = temp;
        }
        p = p->next;
    }
    return B;
}
```