

## 参考答案

### 一、(40 分) 评分标准：每小题 4 分

题号	答案	备注
<b>1</b>	<b>1,0,6,0</b>	各 <b>1</b> 分，逗号不计分
<b>2</b>	<b>31</b>	<b>4</b> 分
<b>3</b>	<b>45</b>	<b>4</b> 分
<b>4</b>	<b>6</b>	<b>4</b> 分
<b>5</b>	<b>7</b>	<b>4</b> 分
<b>6</b>	<b>5, world</b>	各 <b>2</b> 分，逗号不计分
<b>7</b>	<b>8</b> <b>0, 3, 0, 6, 0, 3, 0,</b>	第 <b>1</b> 行， <b>2</b> 分，换行不计分 第 <b>2</b> 行， <b>2</b> 分，逗号不计分
<b>8</b>	<b>1, 1, 1, 1A34</b>	各 <b>1</b> 分，逗号不计分
<b>9</b>	<b>1, abc, ef, g</b>	各 <b>1</b> 分，逗号不计分
<b>10</b>	<b>4, 2020, 10, 5</b>	各 <b>1</b> 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

**1、** 第 **3** 行删除分号

**2、** 第 **10** 行之前（或写第 **3** 行后等都行）增加函数 **void load(struct Goods a[])**  
声明

或写将函数 **load** 移动至 **main** 之前也行。

**3、** 第 **37** 行增加一个 **\**，即 **d:\** 改为 **d:\\**

**4、** 第 **44** 行后或 **45** 行前，增加 **fclose(fp);**

**5、** 第 **43** 行 **a[i]** 前增加 **&**

**6、** 第 **14** 行增加初始化 **flag=0**

**7、** 第 **17** 行删除 **&**

**8、** 第 **20** 行更改为 **if(strcmp(strtemp,a[i].item)==0)**

**9、** 第 **16** 行删除 **[size]**

**10、** 第 **31** 行增加 **0**，或者写更改为 **return 0;**

**11、** 第 **23** 行 **+** 改为 **\***，或者写 **+=** 改为 **\*=**

### 三、编程题

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

```
int main()
```

```
{
```

```
    int a[100],n,k,i,j,flag=1;
```

```
    long num,sum=0;
```

```
    scanf("%d %d",&n,&k);    //读取数据，2 分
```

```
    for (i=0;i<n;i++)
```

```
        scanf("%1d",&a[i]);
```

```
    for (i=n-1;i>=0;i-=k) //外层循环，2 分
```

```
    {
```

```
        num=0;
```

```
        if (flag<0)    //逆序，转化为 k 位十进制数，2 分
```

```
        {
```

```
            for (j=0;j<k && i-j>=0;j++) //注意越界判断
```

```
                num=num*10+a[i-j];
```

```
        }
```

```
        else    //正序，转化为 k 位十进制数，2 分
```

```
        {
```

```
            j=i-k+1;
```

```
            if (j<0) j=0;    //注意越界判断
```

```
            for (;j<=i;j++)
```

```
                num=num*10+a[j];
```

```
        }
```

```
        flag*=-1;    //修改正逆序标志，1 分
```

```
        sum+=num;    //求和，1 分
```

```
    }
```

```
    for (num=1,i=0;i<k;i++)    //去掉最高位的进位后输出，2 分
```

```
        num*=10;
```

```
    printf("%d\n",sum%num);
```

```
    return 0;
```

```
}
```

2. (14 分)

```
#include<stdio.h>
#include<string.h> //头文件正确 1 分
int main()
{
    struct Guest
    {
        char name[32];
        int height;
    }g[20],tg[20],t; //结构体和变量定义正确 2 分

    int i, j, N, min;
    scanf("%d",&N); //1 分, 输入嘉宾个数

    //1.输入所有嘉宾信息 1 分,
    for(i=0;i<N;i++)
        scanf("%s%d",g[i].name, &g[i].height);
    //2.对嘉宾按照身高和名字升序排序 共 5 分, 其他排序方法也可得分
    for(i=0;i<N-1;i++) //1 分
    {
        min = i; //1 分
        for(j=i+1;j<N;j++) //1 分
            if(g[j].height < g[min].height //1 分
                || (g[j].height == g[min].height
                    && strcmp(g[j].name,g[min].name)<0))
                min = j;
        if(min != i) //1 分
        {
            t = g[i];
            g[i] = g[min];
            g[min] = t;
        }
    }
    //3.调整位置, 共 3 分
    for(i=0,j=0;i<N;i+=2,j++) //1 分
    {
        tg[j] = g[i]; //1 分
        if(i+1<N) //1 分
            tg[N-1-j]= g[i+1];
    }
    //4.输出所有嘉宾, 1 分
    for(i=0;i<N;i++)
        printf(" (%s, %d) ",tg[i].name,tg[i].height);
}
```

3. (14 分)

```
//(1)
struct N {
    char s[81]; //数组大小>=81, 1 分
    struct N *next; //next 加指针, 1 分
};

//(2)
int t(char *s)
{
    char *q=s;
    char min=*q; //找最小字符, 2 分
    while (*q)
    {
        if (min>*q) min=*q;
        q++;
    }
    int n=0; //统计出现次数, 2 分
    q=s;
    while (*q)
    {
        if (*q==min) n++;
        q++;
    }
    return n;
}

//(3)
struct N* ex(struct N* A, int n) //返回值 1 分, 形式参数 1 分
{
    struct N* B = NULL, *p1, *p;
    while (A != NULL)
    {
        if (t(A->s) == n) //筛选结点逻辑, 1 分
        {
            //malloc 1 分, sizeof 1 分
            p=(struct N*)malloc(sizeof(struct N));
            strcpy(p->s, A->s); //串复制, 1 分
            p->next = NULL;
            if (B == NULL) //插入结点逻辑, 2 分
                B = p;
            else
                p1->next = p;
            p1 = p;
        }
        A = A->next;
    }
    return B;
}
```