
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=3e+5, y=0.85;
int a1 = x||i&&j-3, a2 = !!!x;
int a3 = i+j+k;
int a4 = (j=8)&&c&&i==5;
printf("%d, %d, %d, %d",
       a1, a2, a3, a4);
```

2.（4 分）

```
int a=0,i;
for(i=1;i<5;i++)
{
    switch(i)
    {
        case 0:
        case 3: a+=2;
        case 1:
        case 2: a+=3;
        default: a+=5;
    }
}
printf("%d", a);
```

3.（4 分）

```
int i, k=5, p[3];
int a[10]={2,2,9,2,0,1,9,2,2,0};
for(i=0; i<3; i++)
    p[i]=a[i*(i+1)];
for(i=0; i<3; i++)
    k+=p[i]*2;
printf("%d", k);
```

4.（4 分）

```
int f()
{
    static int i=0;
```

```
    int s=1;
    s+=i++;
    return s;
}

int main()
{
    int i,a=0;
    for(i=0; i<3; i++)
        a+=f();
    printf("%d", a);
    return 0;
}
```

5.（4 分）

```
char f(char x, char y)
{
    if(x<y)
        return x;
    return y;
}

int main()
{
    char a='9', b='8', c='7';
    printf("%c", f(f(a,b), f(b,c)));
    return 0;
}
```

6.（4 分）

```
char b1[12] = "Hello world";
char b2[12], *pb=b1+11;
while(*(--pb)>=*b1)
    strcpy(b2,pb);
printf("%d, %s",
       strlen(b2), b2);
```

7. (4 分)

```
int f(int a)
{
    if(a>6)
        return a;
    return f(a+=2);
}
void g(int a)
{
    if (a>=3)
        g(a-3);
    printf("%d, ", a);
    if (a>=3)
        g(a-3);
}
int main()
{
    printf("%d\n", f(2));
    g(6);
    return 0;
}
```

8. (4 分)

```
union un {
    char a[5];
    char b,c,d,e;
}f={"1234"};
printf("%c, %c, ", f.a[0], f.c);
char *q=&f.c;
*(q+1)='A';
printf("%c, %s ", *q, f.a);
```

9. (4 分)

```
struct st{
    int a;
    char b[4];
```

```
}arr[3]={1, "abc",
          2, "def",
          3, "ghi"};
struct st *p=arr;
printf("%d, %s, ",p->a, p->b);
printf("%s, %c",
        ((p+1)->b)+1,
        *((p+2)->b));
```

10. (4 分)

```
FILE *fp;
char c=50, s[10];
int d=20, e, f;
long len;
fp=fopen("a.tmp", "w");
fprintf(fp, "%c0%d.1.10", c, d);
fclose(fp);

fp=fopen("a.tmp", "r");
fscanf(fp, "%d", &e);
len=ftell(fp);
printf("%ld, ", len);
fgets(s, 4, fp);
fscanf(fp, "%d", &f);
printf("%d, %d, ", e, f);
fseek(fp, 0L, 2);
printf("%ld", ftell(fp)-len);
fclose(fp);
```

二、改错题 (20 分)

数据文件“d:\data.dat”以二进制方式存有若干个商品数据，函数 load 将

数据读出，从键盘输入指定商品名，将指定商品名（item）的价格（price）增加 10%，并将记录显示出来，若没有相关记录，则给出提示信息。程序中包含若干错误，指出错误所在位置并改正。

```
#include <stdio.h> /*第 1 行*/
#include <string.h> /*第 2 行*/
#define SIZE 50; /*第 3 行*/
struct Goods /*第 4 行*/
{ /*第 5 行*/
    char item[30]; /*第 6 行*/
    double price; /*第 7 行*/
    int count; /*第 8 行*/
}; /*第 9 行*/
int main() /*第 10 行*/
{ /*第 11 行*/
    struct Goods a[SIZE]; /*第 12 行*/
    char strtemp[30]; /*第 13 行*/
    int flag; /*第 14 行*/
    int i; /*第 15 行*/
    load(a[SIZE]); /*第 16 行*/
    scanf("%s",&strtemp); /*第 17 行*/
    for(i=0; i<SIZE; i++) /*第 18 行*/
    { /*第 19 行*/
        if(strtemp==a[i].item) /*第 20 行*/
        { /*第 21 行*/
            flag =1; /*第 22 行*/
            a[i].price+=1.1; /*第 23 行*/
            printf("item is:%s\t",a[i].item ); /*第 24 行*/
            printf("price is:%lf\t",a[i].price ); /*第 25 行*/
            printf("count is:%d\n",a[i].count ); /*第 26 行*/
        } /*第 27 行*/
    } /*第 28 行*/
    if(!flag) /*第 29 行*/
        printf("there is no such item!\n"); /*第 30 行*/
}
```

```

    return;                                /*第 31 行*/
}                                           /*第 32 行*/
void load(struct Goods a[])               /*第 33 行*/
{                                           /*第 34 行*/
    FILE *fp;                             /*第 35 行*/
    int i;                                /*第 36 行*/
    if((fp=fopen("d:\data.dat","rb"))==NULL)/*第 37 行*/
    {                                       /*第 38 行*/
        printf("cannot open file\n");    /*第 39 行*/
        return;                           /*第 40 行*/
    }                                       /*第 41 行*/
    for(i=0; i<SIZE; i++)                 /*第 42 行*/
        if(fread(a[i],sizeof(struct Goods),1,fp)!=1) /*第 43 行*/
            printf("file read error\n");    /*第 44 行*/
    return;                               /*第 45 行*/
}                                           /*第 46 行*/

```

三、编程题（40 分） 注意：程序中请添加必要的注释

1. （12 分）

输入由 n ($0 < n < 100$) 个十进制数字构成的编码序列，将该序列从后向前每 k ($0 < k < 8$) 位分割成一组并来回折叠后，分别看作一个 k 位十进制数，然后对齐相加并舍弃最高位的进位后，输出得到这个 k 位十进制数。

输入格式：第一行输入两个数 n 和 k （用空格隔开）， n 表示该序列的长度， k 为所求的十进制数的位数；第二行输入 n 个十进制数字，中间没有任何分隔符。

输出格式：折叠求和后得到的 k 位十进制数。

输入样例：

14 4

04401042205864

输出样例：

138

输出说明：

5864

0224

4010

+ 40

10138

2. 某晚会节目组要给所有出场嘉宾合影留念，为了保证拍照效果较佳，摄影师要求按照身高对在场的嘉宾进行对称排序，排序规则是：高的在中间，矮的在两边，左右两边对称的位置上要求左边不比右边高（对称位上的嘉宾若身高相同，左边嘉宾的名字字典序小于等于右边）。

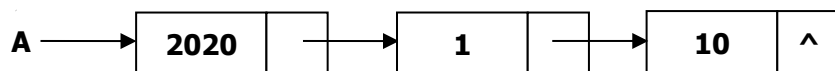
例如：有五个嘉宾：(Zhang, 170)、(Li, 182)、(Wang, 170)、(Zhao, 175)、(Qian, 176)，则最后拍照时的排列顺序是：

(Wang, 170) (Zhao, 175) (Li, 182) (Qian, 176) (Zhang, 170)

其中：Wang 的对称位置是 Zhang，Zhao 的对称位置是 Qian。

请用 C 语言设计一个程序，为节目组实现如下功能：从键盘读入嘉宾的个数 N ($N \leq 20$)，然后依次输入 N 个嘉宾信息，嘉宾信息要求用结构体存储，每个嘉宾信息包括：名字和身高，嘉宾名字不包含空格且长度不超过 31 个字符；然后对输入的嘉宾根据身高和名字进行对称排序，最后将排序结果打印输出，输出格式参照上方划线部分所示。（14 分）

3. 用链表存储一组字符串($1 \leq \text{串长} \leq 80$)。下图所示是3个字符串"2020", "1", "10"被存储在链表A的情形:

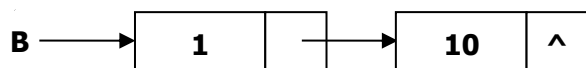


定义: 给定一个字符串s, 指标t(s)定义为s中ASCII码最小的字符的出现次数。

例如, 当s="2020", 最小字符为'0', 出现次数为2, 所以t(s)=2。

根据上述定义, 完成以下问题:

- (1) 规定两个成员名必须为s和next, 写出链表结点struct N的定义; (2分)
- (2) 编写函数: `int t(char *s)`, 计算字符串s的t值; (5分)
- (3) 利用(2), 编写函数ex, 形参是A和n, 功能是将输入的链表A, 按照结点出现顺序将t值正好为n的结点复制出来, 组成一个新链表B并返回, 原链表A不允许做任何修改。例如执行ex(A, 1)后, 所得到的新链表B如下: (7分)



参考答案

一、(40 分) 评分标准：每小题 4 分

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

- 第 **3** 行删除分号
- 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。
- 第 **37** 行增加一个 ****，即 **d:** 改为 **d:**
- 第 **44** 行后或 **45** 行前，增加 **fclose(fp);**
- 第 **43** 行 **a[i]** 前增加 **&**
- 第 **14** 行增加初始化 **flag=0**
- 第 **17** 行删除 **&**
- 第 **20** 行更改为 **if(strcmp(strtemp,a[i].item)==0)**
- 第 **16** 行删除 **[size]**
- 第 **31** 行增加 **0**，或者写更改为 **return 0;**
- 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;
}
```