

第10章 对文件的输入输出

1. 什么是文件型指针？通过文件指针访问文件有什么好处？

解：略。

2. 对文件的打开与关闭的含义是什么？为什么要打开和关闭文件？

解：略。

3. 从键盘输入一个字符串，将其中的小写字母全部转换成大写字母，然后输出到一个磁盘文件 test 中保存。输入的字符串以“!”结束。

解：

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main ()
{
    FILE * fp;
    char str[100];
    int i=0;
    if((fp=fopen("a1","w"))==NULL)
    {
        printf("can not open file\n");
        exit(0);
    }
    printf("input a string:\n");
    gets(str);
    while (str[i]!='!')
    {
        if(str[i]>='a'&& str[i]<='z')
            str[i]=str[i]-32;
        fputc(str[i],fp);
        i++;
    }
    fclose(fp);
    fp=fopen("a1","r");
    fgetc(str,strlen(str)+1,fp);
    printf("%s\n",str);
    fclose(fp);
    return 0;
}
```



成惠资料订购链接



找课后习题答案
下载「知否大学」APP

运行结果:

```
input a string:
i love china!
I LOVE CHINA
```

4. 有两个磁盘文件 A 和 B, 各存放一行字母, 今要求把这两个文件中的信息合并(按字母顺序排列), 输出到一个新文件 C 中去。

解: 先用第 3 题的程序分别建立两个文件 A 和 B, 其中内容分别是 "I LOVE CHINA" 和 "I LOVE BEIJING"。

在程序中先分别将 A、B 文件的内容读出放到数组 c 中, 再对数组 c 排序。最后将数组内容写到 C 文件中。流程图如图 10.1 所示。

程序如下:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;
    int i, j, n, il;
    char c[100], t, ch;
    if((fp=fopen("a1", "r")) == NULL)
        {printf("\ncan not open file\n");
        exit(0);
        }
    printf("file A : \n");
    for(i=0; (ch=fgetc(fp)) != EOF; i++)
    {
        c[i]=ch;
        putchar(c[i]);
    }
    fclose(fp);

    il=i;
    if((fp=fopen("b1", "r")) == NULL)
        {printf("\ncan not open file\n");
        exit(0);
        }
    printf("\nfile B: \n");
    for(i=il; (ch=fgetc(fp)) != EOF; i++)
    {c[i]=ch;
    putchar(c[i]);
    }
```

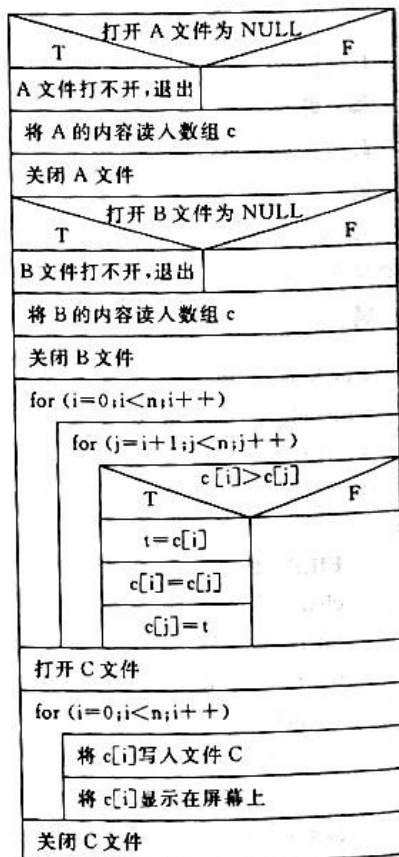


图 10.1

```

fclose(fp);

n=i;
for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if(c[i]>c[j])
            {t=c[i];
             c[i]=c[j];
             c[j]=t;
            }
printf("\nfile C :\n");
fp=fopen("c1","w");
for(i=0;i<n;i++)
    {putc(c[i],fp);
     putchar(c[i]);
    }
printf("\n");
fclose(fp);
return 0;
}

```

运行结果:

```

file A :
I LOVE CHINA
file B:
I LOVE BEIJING
file C :
ABCEEEGHIIIIJLLNNOUU

```

5. 有5个学生,每个学生有3门课程的成绩,从键盘输入学生数据(包括学号、姓名和3门课程成绩),计算出平均成绩,将原有数据和计算出的平均分数存放在磁盘文件“stud”中。

解:

解法一: N-S图如图10.2所示。

```

#include <stdio.h>
struct student
{
    char num[10];
    char name[8];
    int score[3];
    float ave;
} stu[5];

int main()
{
    int i, j, sum;
    FILE * fp;
    for(i=0; i<5; i++)

```

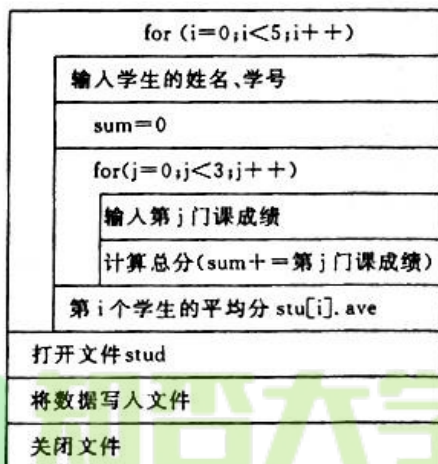


图 10.2

```
{printf("\ninput score of student %d:\n", i+1);
printf("NO. :");
scanf("%s", stu[i]. num);
printf("name:");
scanf("%s", stu[i]. name);
sum=0;
for(j=0; j<3; j++)
{printf("score %d:", j+1);
scanf("%d", &stu[i]. score[j]);
sum+=stu[i]. score[j];
}
stu[i]. ave=sum/3.0;
}
//将数据写入文件
fp=fopen("stud", "w");
for(i=0; i<5; i++)
if(fwrite(&stu[i], sizeof(struct student), 1, fp)!=1)
printf("file write error\n");
fclose(fp);
fp=fopen("stud", "r");
for(i=0; i<5; i++)
{fread(&stu[i], sizeof(struct student), 1, fp);
printf("\n%s, %s, %d, %d, %d, %6.2f\n", stu[i]. num, stu[i]. name, stu[i]. score[0],
stu[i]. score[1], stu[i]. score[2], stu[i]. ave);
}
return 0;
}
```

运行结果:

```
input score of student 1:
NO.:101
name:Li
score 1:98
score 2:89
score 3:88

input score of student 2:
NO.:128
name:Wang
score 1:88
score 2:79
score 3:78

input score of student 3:
NO.:138
name:Chen
score 1:78
score 2:69
score 3:68

input score of student 4:
NO.:148
name:Ma
score 1:108
score 2:99
score 3:98
```

知否大学

知否大学

— 微信公众号同名 —


```

input score of student 5:
NO.:150
name:Wei
score 1:60
score 2:59
score 3:58

```


```
101,Li,90,89,88, 89.00
```

```
120,Wang,80,79,78, 79.00
```

```
130,Chen,70,69,68, 69.00
```

```
140,Ma,100,99,98, 99.00
```

```
150,Wei,60,59,58, 59.00
```

 说明：在程序的第1个for循环中，有两个printf函数语句用来提示用户输入数据，即“printf(“input score of student %d:\n”,i+1);”和“printf(“score %d:”,j+1);”，其中用“i+1”和“j+1”而不是用i和j的用意是使显示提示时，序号从1起，即学生1和成绩1（而不是学生0和成绩0），以符合人们的习惯，但在内存中数组元素下标仍从0算起。

程序最后5行用来检查文件stud中的内容是否正确，从结果来看，是正确的。请注意：用fwrite函数向文件输出数据时不是按ASCII码方式输出的，而是按内存中存储数据的方式输出的（例如一个整数占2个或4个字节，一个实数占4个字节），因此不能用DOS的type命令输出该文件中的数据。

解法二：也可以用下面的程序来实现：

```

#include <stdio.h>
#define SIZE 5
struct student
{
    char name[10];
    int num;
    int score[3];
    float ave;
}stud[SIZE];

int main()
{
    void save(void); //函数声明
    int i;
    float sum[SIZE];
    FILE *fp1;
    for (i=0;i<SIZE;i++) //输入数据,并求每个学生的平均分
    {
        scanf("%s %d %d %d",&stud[i].name,&stud[i].num,&stud[i].score[0],
            &stud[i].score[1],&stud[i].score[2]);
        sum[i]=stud[i].score[0]+stud[i].score[1]+stud[i].score[2];
        stud[i].ave=sum[i]/3;
    }
    save(); //调用 save 函数,向文件 stu.dat 输出数据
}

```

```

fp1=fopen("stu.dat","rb");           //用只读方式打开 stu.dat 文件
printf("\n name NO. score1 score2 score3 ave\n");
printf("-----\n");

//输出表头
for (i=0;i<SIZE;i++)                 //从文件读入数据并在屏幕输出
{fread(&stud[i],sizeof(struct student),1,fp1);
 printf("%-10s %3d %7d %7d %7d %8.2f\n",stud[i].name,stud[i].num,
 stud[i].score[0],stud[i].score[1],stud[i].score[2],stud[i].ave);
}
fclose(fp1);
return 0;
}

void save(void)                       //向文件输出数据的函数
{
FILE *fp;
int i;
if ((fp=fopen("stu.dat","wb"))==NULL)
{printf("The file can not open\n");
 return;
}
for(i=0;i<SIZE;i++)
if (fwrite(&stud[i],sizeof(struct student),1,fp)!=1)
{printf("file write error\n");
 return;
}
fclose(fp);
}

```

运行结果:

Zhang	101	77	78	98	
Li	102	67	78	88	
Wang	103	89	99	97	
Wei	104	77	76	98	
Tan	105	78	89	97	
name	NO.	score1	score2	score3	ave
Zhang	101	77	78	98	84.33
Li	102	67	78	88	77.67
Wang	103	89	99	97	95.00
Wei	104	77	76	98	83.67
Tan	105	78	89	97	88.00

本程序用 save 函数将数据写到磁盘文件上,再从文件读回,然后用 printf 函数输出,从运行结果可以看到文件中的数据是正确的。

6. 将第 5 题 stud 文件中的学生数据,按平均分进行排序处理,将已排序的学生数据存入一个新文件 stu_sort 中。

解:

解法一: N-S 图如图 10.3 所示。

```
include <stdio.h>
#include <stdlib.h>
#define N 10
struct student
{
    char num[10];
    char name[8];
    int score[3];
    float ave;
}st[N], temp;

int main()
{
    FILE * fp;
    int i, j, n;

    //读文件
    if((fp=fopen("stud", "r"))==NULL)
    {
        printf("can not open. \n");
        exit(0);
    }
    printf("File 'stud': ");
    for(i=0; fread(&st[i], sizeof(struct student), 1, fp)!=0; i++)
    {
        printf("\n%8s%8s", st[i]. num, st[i]. name);
        for(j=0; j<3; j++)
            printf("%8d", st[i]. score[j]);
        printf("%10.2f", st[i]. ave);
    }
    printf("\n");
    fclose(fp);
    n=i;

    //排序
    for(i=0; i<n; i++)
        for(j=i+1; j<n; j++)
            if(st[i]. ave < st[j]. ave)
            {
                temp=st[i];
                st[i]=st[j];
                st[j]=temp;
            }

    //输出
    printf("\nNow:");
```

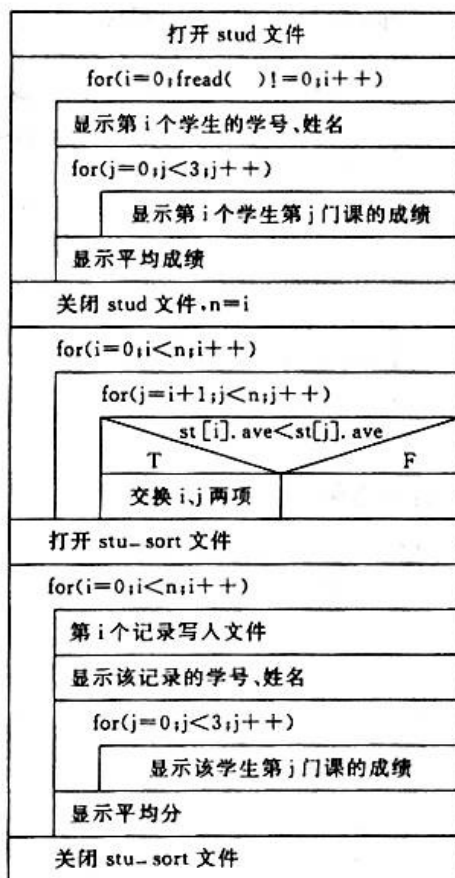


图 10.3

```

fp=fopen("stu_sort","w");
for(i=0;i<n;i++)
{fwrite(&st[i],sizeof(struct student),1,fp);
printf("\n%8s%8s",st[i].num,st[i].name);
for(j=0;j<3;j++)
printf("%8d",st[i].score[j]);
printf("%10.2f",st[i].ave);
}
printf("\n");
fclose(fp);
return 0;
}

```

运行结果:

File 'stud':					
110	Li	90	89	88	89.00
120	Wang	80	79	78	79.00
130	Chen	70	69	68	69.00
140	Ma	100	99	98	99.00
150	Wei	60	59	58	59.00
Now:					
140	Ma	100	99	98	99.00
110	Li	90	89	88	89.00
120	Wang	80	79	78	79.00
130	Chen	70	69	68	69.00
150	Wei	60	59	58	59.00

解法二: 与第 5 题解法二相应, 可以接着使用下面的程序来实现本题要求。

```

#include <stdio.h>
#include <stdlib.h>
#define SIZE 5
struct student
{
    char name[10];
    int num;
    int score[3];
    float ave;
}stud[SIZE],work;
int main()
{
    void sort(void);
    int i;
    FILE *fp;
    sort();
    fp=fopen("stud_sort.dat","rb");
    printf("sorted student's scores list as follow\n");
    printf("-----\n");
    printf("NAME NO. SCORE1 SCORE2 SCORE3 AVE\n");
    printf("-----\n");
}

```



知否大学
- 微信公众号同名 -


```

for(i=0;i<SIZE;i++)
{
    fread(&stud[i],sizeof(struct student),1,fp);
    printf("%-10s %3d %8d %8d %8d %9.2f\n",stud[i].name,stud[i].num,
        stud[i].score[0],stud[i].score[1],stud[i].score[2],stud[i].ave);
}
fclose(fp);
return 0;
}

void sort(void)
{
    FILE *fp1,*fp2;
    int i,j;
    if((fp1=fopen("stu.dat","rb"))==NULL)
    {
        printf("The file can not open\n\n");
        exit(0);
    }
    if((fp2=fopen("stud_sort.dat","wb"))==NULL)
    {
        printf("The file write error\n");
        exit(0);
    }
    for(i=0;i<SIZE;i++)
    {
        if(fread(&stud[i],sizeof(struct student),1,fp1)!=1)
        {
            printf("file read error\n");
            exit(0);
        }
        for(j=i+1;j<SIZE;j++)
        {
            if(stud[i].ave<stud[j].ave)
            {
                work=stud[i];
                stud[i]=stud[j];
                stud[j]=work;
            }
        }
        fwrite(&stud[i],sizeof(struct student),1,fp2);
    }
    fclose(fp1);
    fclose(fp2);
}

```

运行结果:

NAME	NO.	SCORE1	SCORE2	SCORE3	AUE
Wang	103	89	99	97	95.00
Tan	105	78	89	97	88.00
Zhang	101	77	78	98	84.33
Wei	104	77	76	98	83.67
Li	102	67	78	88	77.67

找课后习题答案
「知否大学」APP

7. 将第 6 题已排序的学生成绩文件进行插入处理。插入一个学生的 3 门课程成绩, 程序先计算新插入学生的平均成绩, 然后将它按成绩高低顺序插入, 插入后建立一个新文件。

解: N-S 图如图 10.4 所示。

```
#include <stdio.h>
#include <stdlib.h>
struct student
{
    char num[10];
    char name[8];
    int score[3];
    float ave;
}st[10],s;

int main()
{
    FILE *fp,*fp1;
    int i,j,t,n;
    printf("\nNO. :");
    scanf("%s",s.num);
    printf("name:");
    scanf("%s",s.name);
    printf("score1, score2, score3:");
    scanf("%d,%d,%d",&s.score[0],&s.score[1],&s.score[2]);
    s.ave=(s.score[0]+s.score[1]+s.score[2])/3.0;

    //从文件读数据
    if((fp=fopen("stu_sort","r"))==NULL)
    {
        printf("can not open file.");
        exit(0);
    }
    printf("original data:\n");
    for(i=0;fread(&st[i],sizeof(struct student),1,fp)!=0;i++)
    {
        printf("\n%8s%8s",st[i].num,st[i].name);
        for(j=0;j<3;j++)
            printf("%8d",st[i].score[j]);
        printf("%10.2f",st[i].ave);
    }

    n=i;
    for(t=0;st[t].ave>s.ave && t<n;t++);

    //向文件写数据
    printf("\nNow:\n");
    fp1=fopen("sort1.dat","w");
    for(i=0;i<t;i++)
        fwrite(&st[i],sizeof(struct student),1,fp1);
```

输入待插入的学生的数据
计算其平均分
打开 stu_sort 文件
从该文件读入数据并显示出来
确定插入的位置 t
向文件输出前面 t 个学生的数据并显示
向文件输出待输入的学生数据并显示
向文件输出 t 后面的学生数据并显示
关闭文件

图 10.4

```

printf("\n %8s%8s",st[i]. num,st[i]. name);
for(j=0;j<3;j++)
    printf("%8d",st[i]. score[j]);
printf("%10.2f",st[i]. ave);
}
fwrite(&s,sizeof(struct student),1,fp1);
printf("\n %8s %7s %7d %7d %7d%10.2f",s. num,s. name,s. score[0],
    s. score[1],s. score[2],s. ave);

for(i=t;i<n;i++)
{fwrite(&st[i],sizeof(struct student),1,fp1);
printf("\n %8s%8s",st[i]. num,st[i]. name);
for(j=0;j<3;j++)
    printf("%8d",st[i]. score[j]);
printf("%10.2f",st[i]. ave);
}
printf("\n");
fclose(fp);
fclose(fp1);
return 0;
}

```

运行结果:

NO.:168					
name:Ian					
score1,score2,score3:98,97,98					
original data:					
140	Ma	100	99	98	99.00
101	Li	90	89	88	89.00
120	Wang	80	79	78	79.00
130	Chen	70	69	68	69.00
150	Wei	60	59	58	59.00
Now:					
140	Ma	100	99	98	99.00
160	Ian	98	97	98	97.67
101	Li	90	89	88	89.00
120	Wang	80	79	78	79.00
130	Chen	70	69	68	69.00
150	Wei	60	59	58	59.00

为节省篇幅,本题和第8题不再给出第6题“解法二”的程序,请读者自己编写程序。

8. 将第7题结果仍存入原有的 stu_sort 文件而不另建立新文件。

解:

```

#include <stdio.h>
#include <stdlib.h>

struct student
{
    char num[10];
    char name[8];
    int score[3];
}

```

```
float ave;  
}st[10],s;
```

```
int main()
```

```
{FILE * fp, * fp1;  
int i,j,t,n;  
printf("\nNO. :");  
scanf("%s",s.num);  
printf("name:");  
scanf("%s",s.name);  
printf("score1,score2,score3:");  
scanf("%d,%d,%d",&s.score[0],&s.score[1],&s.score[2]);  
s.ave=(s.score[0]+s.score[1]+s.score[2])/3.0;
```

```
//从文件读数据
```

```
if((fp=fopen("stu_sort","r"))==NULL)  
{printf("can not open file.");  
exit(0);  
}
```

```
printf("original data:\n");
```

```
for(i=0;fread(&st[i],sizeof(struct student),1,fp)!=0;i++)  
{printf("\n%8s%8s",st[i].num,st[i].name);  
for(j=0;j<3;j++)  
printf("%8d",st[i].score[j]);  
printf("%10.2f",st[i].ave);  
}
```

```
n=i;
```

```
for(t=0;st[t].ave>s.ave && t<n;t++);
```

```
//向文件写数据
```

```
printf("\nNow:\n");
```

```
fp1=fopen("sort1.dat","w");
```

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

```
{fwrite(&st[i],sizeof(struct student),1,fp1);  
printf("\n %8s%8s",st[i].num,st[i].name);  
for(j=0;j<3;j++)  
printf("%8d",st[i].score[j]);  
printf("%10.2f",st[i].ave);  
}
```

```
fwrite(&s,sizeof(struct student),1,fp1);
```

```
printf("\n %8s %7s %7d %7d %7d%10.2f",s.num,s.name,s.score[0],  
s.score[1],s.score[2],s.ave);
```

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



```

{fwrite(&st[i], sizeof(struct student), 1, fp1);
printf("\n %8s%8s", st[i]. num, st[i]. name);
for(j=0; j<3; j++)
    printf("%8d", st[i]. score[j]);
printf("%10.2f", st[i]. ave);
}
printf("\n");
fclose(fp);
fclose(fp1);
return 0;
}

```

运行结果:

```

NO.:168
name:Hua
score1,score2,score3:78,89,91
original data:
    140      Ma      100      99      98      99.00
    101      Li       98      89      88      89.00
    120     Wang      80      79      78      79.00
    130     Chen      70      69      68      69.00
    150     Wei       60      59      58      59.00
Now:
    140      Ma      100      99      98      99.00
    101      Li       98      89      88      89.00
    160     Hua       78      89      91      86.00
    120     Wang      80      79      78      79.00
    130     Chen      70      69      68      69.00
    150     Wei       60      59      58      59.00

```

9. 有一磁盘文件 employee, 其中存放职工的数据。每个职工的数据包括职工姓名、职工号、性别、年龄、住址、工资、健康状况、文化程度。今要求将职工名、工资的信息单独抽出来另建一个简明的职工工资文件。

解: N-S 图如图 10.5 所示。

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct employee
{char num[6];
char name[10];
char sex[2];
int age;
char addr[20];
int salary;
char health[8];
char class[10];
}em[10];

```

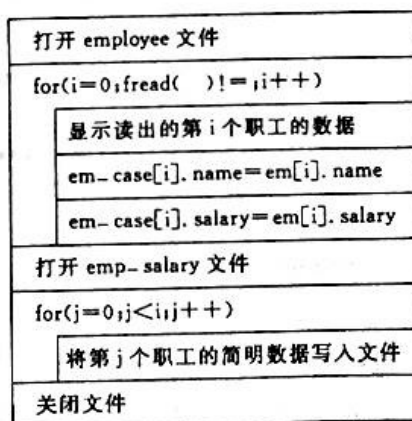


图 10.5

```

struct emp
{
    char name[10];
    int salary;
} em_case[10];

int main()
{
    FILE * fp1, * fp2;
    int i, j;
    if ((fp1 = fopen("employee", "r")) == NULL)
    {
        printf("can not open file. \n");
        exit(0);
    }
    printf("\n NO. name sex age addr salary health class\n");
    for (i = 0; fread(&em[i], sizeof(struct employee), 1, fp1) != 0; i++)
    {
        printf("\n%4s%8s%4s%6d%10s%6d%10s%8s", em[i]. num, em[i]. name, em[i]. sex,
            em[i]. age, em[i]. addr, em[i]. salary, em[i]. health, em[i]. class);
        strcpy(em_case[i]. name, em[i]. name);
        em_case[i]. salary = em[i]. salary;
    }
    printf("\n\n*****");
    if ((fp2 = fopen("emp_salary", "wb")) == NULL)
    {
        printf("can not open file\n");
        exit(0);
    }
    for (j = 0; j < i; j++)
    {
        if (fwrite(&em_case[j], sizeof(struct emp), 1, fp2) != 1)
            printf("error!");
        printf("\n %12s%10d", em_case[j]. name, em_case[j]. salary);
    }
    printf("\n*****");
    fclose(fp1);
    fclose(fp2);
    return 0;
}


```

运行结果:

NO.	name	sex	age	addr	salary	health	class
101	Li	m	23	Beijing	670	good	F.H.D.
102	Wang	f	45	Shanghai	780	bad	master
103	Ma	m	32	Taijin	650	good	univ.
104	Liu	f	56	Xian	540	pass	college

Li	670
Wang	780
Ma	650
Liu	540

Press any key to continue.

 说明：数据文件 employee 是事先建立好的，其中已有职工数据，而 emp_salary 文件则是由程序建立的。

建立 employee 文件的程序如下：

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

struct employee
{
    char num[6];
    char name[10];
    char sex[2];
    int age;
    char addr[20];
    int salary;
    char health[8];
    char class[10];
}em[10];

int main()
{
    FILE *fp;
    int i;
    printf("input NO. , name, sex, age, addr,salary,health,class\n");
    for (i=0;i<4;i++)
        scanf("%s %s %s %d %s %d %s %s",em[i].num,em[i].name,em[i].sex,
            &em[i].age,em[i].addr,&em[i].salary,em[i].health,em[i].class);

    //将数据写入文件
    if((fp=fopen("employee","w"))==NULL)
    {
        printf("can not open file.");
        exit(0);
    }
    for (i=0;i<4;i++)
        if(fwrite(&em[i],sizeof(struct employee),1,fp)!=1)
            printf("error\n");
    fclose(fp);
    return 0;
}
```

在运行此程序时从键盘输入 4 个职工的数据，程序将它们写入 employee 文件。在运行前面一个程序时从 employee 文件中读出数据并输出到屏幕，然后建立一个简明文件，同时在屏幕上输出。

10. 从第 9 题的“职工工资文件”中删去一个职工的数据，再存回原文件。

解: N-S 图如图 10.6 所示。

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

struct employee
{
    char name[10];
    int salary;
} emp[20];

int main()
{
    FILE *fp;
    int i, j, n, flag;
    char name[10];
    if((fp=fopen("emp_salary", "rb")) == NULL)
    {
        printf("can not open file.\n");
        exit(0);
    }
    printf("\noriginal data:\n");
    for(i=0; fread(&emp[i], sizeof(struct employee), 1, fp) != 0; i++)
        printf("\n %8s %7d", emp[i].name, emp[i].salary);
    fclose(fp);
    n=i;
    printf("\ninput name deleted:\n");
    scanf("%s", name);
    for(flag=1, i=0; flag && i<n; i++)
    {
        if(strcmp(name, emp[i].name) == 0)
        {
            for(j=i; j<n-1; j++)
            {
                strcpy(emp[j].name, emp[j+1].name);
                emp[j].salary = emp[j+1].salary;
            }
            flag=0;
        }
    }
    if(!flag)
        n=n-1;
    else
        printf("\nnot found!");
    printf("\nNow, The content of file:\n");
    if((fp=fopen("emp_salary", "wb")) == NULL)
    {
        printf("can not open file.\n");
        exit(0);
    }
    for(i=0; i<n; i++)
```

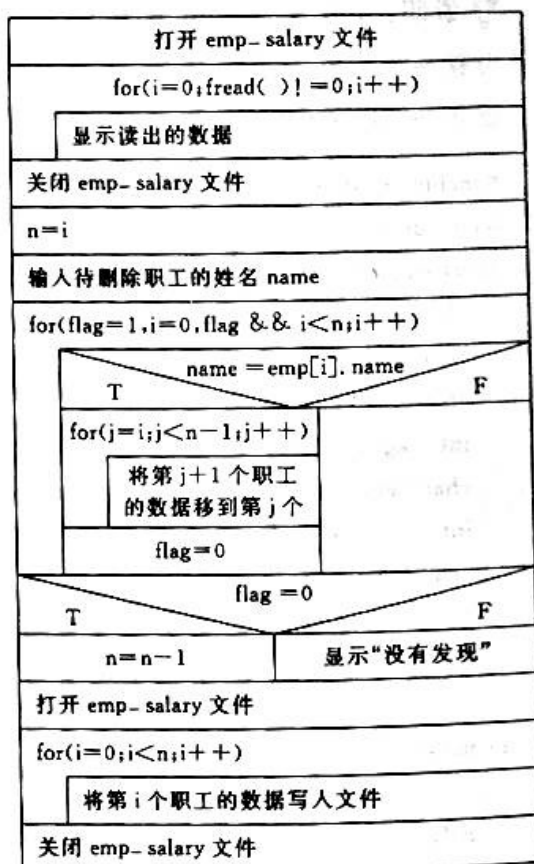


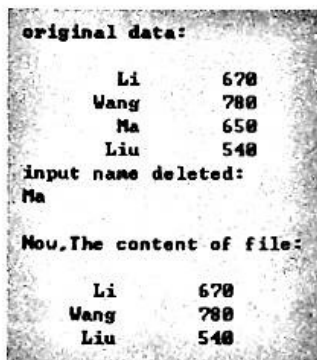
图 10.6


```

    fwrite(&emp[i], sizeof(struct employee), 1, fp);
fclose(fp);
fp = fopen("emp_salary", "r");
for(i=0; fread(&emp[i], sizeof(struct employee), 1, fp) != 0; i++)
    printf("\n%8s %7d", emp[i].name, emp[i].salary);
printf("\n");
fclose(fp);
return 0;
}

```

运行结果:



11. 从键盘输入若干行字符(每行长度不等),输入后把它们存储到一磁盘文件中。再从该文件中读入这些数据,将其中小写字母转换成大写字母后在显示屏上输出。

解: N-S图如图 10.7 所示。

```

#include <stdio.h>
int main()
{
    int i, flag;
    char str[80], c;
    FILE *fp;
    fp = fopen("text", "w");
    flag = 1;
    while(flag == 1)
    {
        printf("input string:\n");
        gets(str);
        fprintf(fp, "%s", str);
        printf("continue?");
        c = getchar();
        if((c == 'N') || (c == 'n'))
            flag = 0;
        getchar();
    }
    fclose(fp);
    fp = fopen("text", "r");
    while(fscanf(fp, "%s", str) != EOF)

```

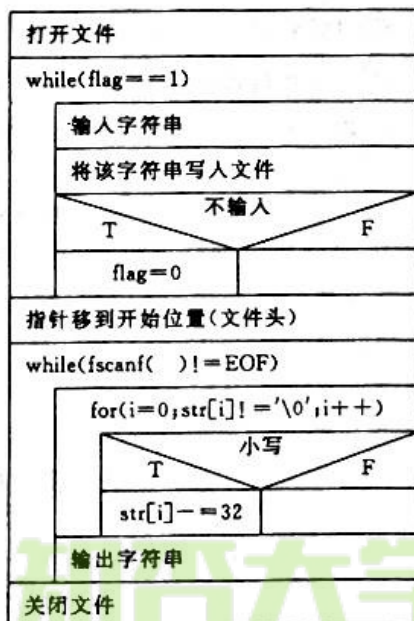


图 10.7

```

    for(i=0;str[i]!='\0';i++)
        if((str[i]>='a') && (str[i]<='z'))
            str[i]-=32;
    printf("%s\n",str);
}

fclose(fp);
return 0;
}

```

运行结果:

```

input string:
abcdef.
continue?y
input string:
ghijkl.
continue?y
input string:
mnopqrst.
continue?n
ABCDEF.
GHIJKL.
MNOPQRST.

```

此程序运行结果是正确的,但是如果输入的字符串中包含了空格,就会发生一些问题,例如输入:

```

input string:
i am a student.

```

得到的结果是:

```

I
AM
A
STUDENT.

```

把一行分成几行输出。这是因为用 `fscanf` 函数从文件读入字符串时,把空格作为一个字符串的结束标志,因此把该行作为 4 个字符串来处理,分别输出在 4 行上。请读者考虑怎样解决这个问题。

