**C语言 第八次作业**

**刘泓尊 2018011446**

1. **词典合并**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void swap(char\*a,char\*b)

{

char tmp[10];

strcpy\_s(tmp,10,a);

strcpy\_s(a,10,b);

strcpy\_s(b,10,tmp);

}

int main()

{

FILE\*fp1,\*fp2,\*fp3;

char s[20][10],ch;

int i=0,j=0,count,len;

if((fp1=fopen("8-1-dict1.txt","r"))==NULL){

printf("Cannot open file '8-1-dict1.txt'\n");

exit(0);

}

if((fp2=fopen("8-1-dict2.txt","r"))==NULL){

printf("Cannot open file '8-1-dict1.txt'\n");

exit(0);

}

if((fp3=fopen("8-1-dict3.txt","w+"))==NULL){

printf("Cannot open file '8-1-dict3.txt'\n");

exit(0);

}

//将文件中的信息存在数组s中

while((ch=fgetc(fp1))!=EOF){

s[i][j++]=ch;

if(ch=='\n'){s[i][j]='\0';i++;j=0;}

}

s[i][j]='\0';i++;j=0;

while((ch=fgetc(fp2))!=EOF){

s[i][j++]=ch;

if(ch=='\n'){s[i][j]='\0';i++;j=0;}

}

s[i][j]='\0';count=i;

//进行排序

for(i=count;i>=0;i--){

for(j=0;j<i;j++){

if(strcmp(s[j],s[j+1])>0)swap(s[j],s[j+1]);

}

}

//进行删除

for(i=0;i<count;i++){

if(strcmp(s[i],s[i+1])==0){s[i+1][0]='\0';i++;}

}

for(i=0;i<count;i++){

len=strlen(s[i]);

if(s[i][len-1]!='\n'&&strcmp(s[i],"\0")!=0){s[i][len]='\n';s[i][len+1]='\0';}

}

//向新文件输出字符串内容

for(i=0;i<=count;i++){

if(strcmp(s[i],"\0")!=0){

fwrite(s[i],strlen(s[i]),1,fp3);

}

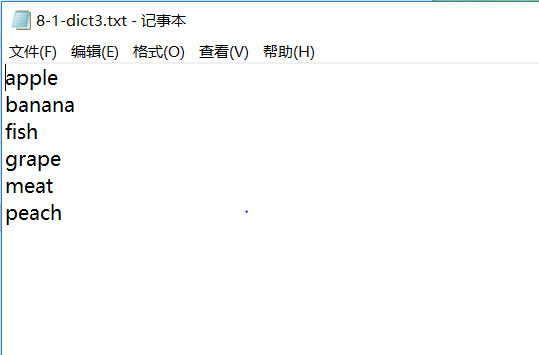
}

fclose(fp1);fclose(fp2);fclose(fp3);

fp1=fp2=fp3=NULL;

return 0;

}



1. **产品销售**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct sell{

char dm[5];

char mc[11];

int dj;

int sl;

long je;

};

void ReadData(sell\*record)

{

FILE\*fp;

int i;

if((fp=fopen("8-2-IN.DAT","rb"))==NULL){

printf("Cannot open file '8-2-IN.DAT'\n");

exit(0);

}

for(i=0;i<100;i++){

if((fread(record+i,sizeof(sell),1,fp))!=1)

printf("file write error\n");

record[i].je=(record[i].dj)\*(record[i].sl);

}

fclose(fp);

}

void SortData(sell\*record)

{

int i,j;

for(i=99;i>=0;i--){

for(j=0;j<i;j++){

if((record[j].je)>(record[j+1].je)){

sell tmp;

tmp=record[j];

record[j]=record[j+1];

record[j+1]=tmp;

}

}

}

}

void WriteData(sell\*record)

{

FILE\*fp1;

char tmp\_dm[20],tmp\_mc[20];

if((fp1=fopen("8-2-OUT.DAT","wb+"))==NULL){

printf("Cannot open file '8-2-OUT.DAT'\n");

exit(0);

}

printf("金额最高的10个记录:\n产品代码 |产品名称 |单价 |数量 |金额 |\n");

for(int i=0;i<100;i++){

fwrite(record+i,sizeof(sell),1,fp1);

//递减顺序输出金额最高的10个记录

if(i>=90){

strcpy(tmp\_dm,record[189-i].dm);tmp\_dm[5]='\0';

strcpy(tmp\_mc,record[189-i].mc);tmp\_mc[11]='\0';//在结尾加结束符

printf("%-10s|%-20s|%-10d|%-10d|%-8ld|\n",tmp\_dm,tmp\_mc,record[189-i].dj,record[189-i].sl,record[189-i].je);

}

}

fclose(fp1);

}

int main()

{

struct sell record[100],\*record\_p=record;

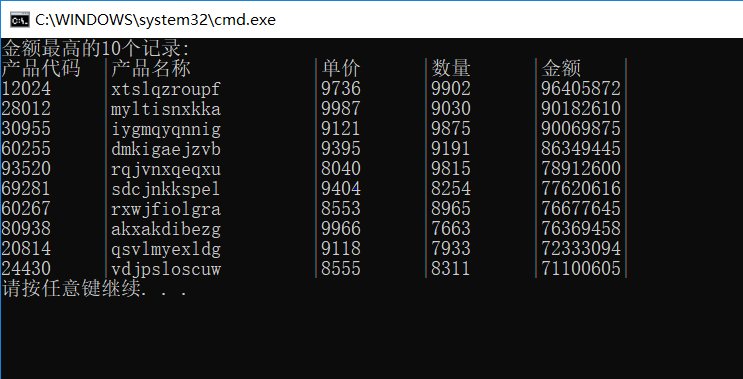
ReadData(record\_p);

SortData(record\_p);

WriteData(record\_p);

return 0;

}



**3.股票收益统计**

#include<stdio.h>

#include<string.h>

#include<malloc.h>

struct node{

char time[11];

char name[9];

int num;

float price\_in;

float price\_out;

float rate;

float interest\_rate;

};

void main()

{

node\*info;int i=0,count,j=0,num=0,k,flag=0;

float sum[10]={0.0};char ch,name[10][10];

info=(node\*)malloc(1000\*sizeof(node));

FILE\*fp1,\*fp2;

if((fp1=fopen("8-3-stock.txt","r"))==NULL){

printf("cannot open file!\n");

return;

}

ch=fgetc(fp1);

while(!feof(fp1)){

j=0;

while(ch!=' '){

info[i].time[j++]=ch;

ch=fgetc(fp1);

}

info[i].time[j]='\0';

ch=fgetc(fp1);j=0;

while(ch!=' '){

info[i].name[j++]=ch;

ch=fgetc(fp1);

}

info[i].name[8]='\0';

fscanf(fp1,"%d",&info[i].num);

ch=fgetc(fp1);

fscanf(fp1,"%f",&info[i].price\_in);

ch=fgetc(fp1);

fscanf(fp1,"%f",&info[i].price\_out);

ch=fgetc(fp1);

ch=fgetc(fp1);

if(ch==EOF)break;

i++;

}

count=i;

strcpy\_s(name[++num],10,info[0].name);

for(i=0;i<=count;i++){

info[i].interest\_rate=(info[i].price\_out-info[i].price\_in)/info[i].price\_in;

for(k=1;k<=num;k++){

if(strcmp(info[i].name,name[k])==0){

sum[k]+=(info[i].num)\*(info[i].price\_in);

flag=1;

break;

}

}

if(flag==0){

strcpy\_s(name[++num],12,info[i].name);

sum[num]+=(info[i].num)\*(info[i].price\_in);

}

flag=0;

}

for(i=0;i<=count;i++){

for(k=1;k<=num;k++){

if(strcmp(info[i].name,name[k])==0){

info[i].rate=(info[i].num)\*(info[i].price\_in)/sum[k];

break;

}

}

}

if((fp2=fopen("8-3-stat.txt","w+"))==NULL){

printf("cannot open file!\n");

return ;

}

i=0;

while(i<=count){

fputs(info[i].time,fp2);

fputc(' ',fp2);

fputs(info[i].name,fp2);

fputc(' ',fp2);

fprintf(fp2,"%f%% ",100\*(info[i].rate));

fprintf(fp2,"%f%%\n",100\*(info[i].interest\_rate));

i++;

}

fclose(fp1);

fclose(fp2);

free(info);

}



**4.链表与文件**

#include<stdio.h>

#include<string.h>

#include<malloc.h>

struct node{

char serial\_num[20];

node\*next;

};

void PrintList(node\*head,int n)//输出链表内容

{

node\*p1=head;//为了简便，这里用head代表线性链表的head和循环链表的rear

printf("当前链表内容:\n");

switch(n){//用1代表线性链表，用2代表循环链表

case 1:

while(p1!=NULL){

printf("%s -> ",p1->serial\_num);

p1=p1->next;

}

break;

case 2:

while(p1->next!=head){

p1=p1->next;

printf("%s -> ",p1->serial\_num);

}

printf("%s -> ",p1->next->serial\_num);

break;

}

putchar(10);

}

node\*LineInsert(node\*head,node\*p0)//线性链表插入

{

node\*p1,\*p2;

p1=p2=head;

if(head==NULL){head=p0;p0->next=NULL;}

else{

//关键字递增有序，所以应该先比较高位数字，因此采用strcmp比较大小，注意：10<2

while(strcmp(p0->serial\_num,p1->serial\_num)>0&&p1->next!=NULL){

p2=p1;

p1=p1->next;

}

if(strcmp(p0->serial\_num,p1->serial\_num)<=0){

if(p1==head){

head=p0;

p0->next=p1;

}else{

p2->next=p0;

p0->next=p1;

}

}else{

p1->next=p0;

p0->next=NULL;

}

}

return head;

}

node\*CircleInsert(node\*rear,node\*p0)//循环链表插入,返回尾指针

{

node\*p1,\*p2;

if(rear==NULL){rear=p0;p0->next=rear;}

p1=p2=rear->next;

if(rear!=NULL){

while(strcmp(p0->serial\_num,p1->serial\_num)>0&&p1!=rear){

p2=p1;

p1=p1->next;

}

if(strcmp(p0->serial\_num,p1->serial\_num)<=0){

if(p1==rear->next){

rear->next=p0;

p0->next=p1;

}else{

p2->next=p0;

p0->next=p1;

}

}else{

p0->next=rear->next;

p1->next=p0;

rear=p0;

}

}

return rear;

}

void create(node\*\*rear,node\*\*head2)

{

node\*p1,\*p2;

int n1,n2,i;

p1=\*rear;p2=\*head2;

printf("please input number of elements of aList:");

scanf\_s("%d",&n1);

printf("please input the elements of aList:\n");

for(i=0;i<n1;i++){//创建环形链表

p1=(node\*)malloc(sizeof(node));

scanf\_s("%s",p1->serial\_num,20);

\*rear=CircleInsert(\*rear,p1);

PrintList(\*rear,2);

fflush(stdin);

}

printf("\nplease input number of elements of bList:");

scanf\_s("%d",&n2);

printf("please input the elements of sList:\n");

for(i=0;i<n2;i++){//创建线性链表

p2=(node\*)malloc(sizeof(node));

scanf\_s("%s",p2->serial\_num,20);

\*head2=LineInsert(\*head2,p2);

PrintList(\*head2,1);

fflush(stdin);

}

}

node\*combin(node\*rear,node\*head2)

{

node\*head;

head=rear->next;

rear->next=head2;

return head;

}

void SaveList(node\*head,int n,char\*\*fname1)//将链表的数据输入到硬盘文件

{

FILE\*fp1;

node\*p1;

p1=head;

//同样为了简便，这里用head既代表顺序链表的head又用rear代表循环链表的rear

printf("\n请输入新建.txt文件名：");

scanf\_s("%s",\*fname1,20);

if((fp1=fopen(\*fname1,"w+"))==NULL){

printf("cannot open file %s\n",\*fname1);

return;

}

switch(n){

case 2:

do{

p1=p1->next;

if((fwrite(p1,20\*sizeof(char),1,fp1)!=1))//仅仅存入数据，不存入指针

printf("file1 write error!\n");

}while(p1!=head);

break;

case 1:

do{

if((fwrite(p1,20\*sizeof(char),1,fp1))!=1)

printf("file2 write error!\n");

p1=p1->next;

}while(p1!=NULL);

break;

}

fclose(fp1);

}

void LoadList(char\*\*fname\_p)//从硬盘文件读入链表

{

int n,i;node tmp[20];

char fname[20];FILE\*fp;

printf("\n请输入要打开的.txt文件名:\n");

scanf\_s("%s",fname,20);

if(strcmp(fname,\*fname\_p)==0||strcmp(fname,\*(fname\_p+2))==0)n=2;

else if(strcmp(fname,\*(fname\_p+1))==0)n=1;

else {printf("error!\n");return ;}

if((fp=fopen(fname,"r"))==NULL){

printf("cannot open file!\n");

return ;

}

printf("当前链表内容:\n");

switch(n){

case 1:

i=0;

rewind(fp);

while(!feof(fp)){

fread(&tmp[i].serial\_num,20\*sizeof(char),1,fp);

if(feof(fp))break;

printf("%s -> ",tmp[i].serial\_num);

i++;

};

break;

case 2:

i=0;

rewind(fp);

while(!feof(fp)){

fread(&tmp[i].serial\_num,20\*sizeof(char),1,fp);

if(feof(fp))break;

printf("%s -> ",tmp[i].serial\_num);

i++;

};

break;

}

fclose(fp);

return;

}

int main()

{

node\*head1,\*head2;

char fname[3][20];

char\*fname\_p[3]={fname[0],fname[1],fname[2]};

head1=NULL;head2=NULL;

create(&head1,&head2);

SaveList(head1,2,&fname\_p[0]);

SaveList(head2,1,&fname\_p[1]);

LoadList(fname\_p);

LoadList(fname\_p);//读入内存并输出到屏幕

head1=combin(head1,head2);

SaveList(head1,1,&fname\_p[2]);

printf("合并后的新链表:\n");

PrintList(head1,1);

return 0;

}

