//A卷

1. while(fread(&flightary[g\_num], sizeof(FLIGHT), 1, pfile) == 1 )
2. int adddata2arry()

{

if(g\_num >= SIZE)

{

printf("Reach max storage capability! Adding fail!\n");

return -1;

}

do{

puts("请输入出发城市<C for Chendu B for Beijing X for Xian>: ");

scanf("%c",&flightary[g\_num].cityfrom);

getchar();

}while(flightary[g\_num].cityfrom != 'C' && flightary[g\_num].cityfrom !='B'&& flightary[g\_num].cityfrom !='X');

do{

puts("请输入到达城市<C for Chendu B for Beijing X for Xian>: ");

scanf("%c",&flightary[g\_num].cityto);

getchar();

}while(flightary[g\_num].cityto != 'C' && flightary[g\_num].cityto !='B'&& flightary[g\_num].cityto !='X');

//24小时制，整点

do{

puts("请输入离港时间 <1 to 24 >: ");

scanf("%d",&flightary[g\_num].depttime);

}while(flightary[g\_num].depttime <1 || flightary[g\_num].depttime >24);

//飞行时长小于等于12小时，整点

do{

puts("请输入飞行时间 <1 to 12 >: ");

scanf("%d",&flightary[g\_num].flttime);

}while(flightary[g\_num].flttime <1 || flightary[g\_num].flttime >12);

g\_num++ ;

printf("添加成功\n");

system("pause");

return g\_num;

}

1. void deletedatafromarry()

{

int index=0;

int i ;

//调用显示所以航班信息及其在数组中的下标号

displayall();

//根据显示信息编号，删除信息，后续元素依次上移

do{

printf("输入需要删除的航班号：0~%d\t",g\_num-1);

scanf("%d",&index);

}while(index <0 || index >g\_num-1);

//删除，即：将后面的元素往前挪一步

for(i= index; i<g\_num-1; i++)

{

flightary[i] =flightary[i + 1] ;

}

g\_num--;

printf("删除成功\n");

system("pause");

}

1. void search()

{

int i,j;

char from,to;

do{

puts("请输入出发城市<C for Chendu B for Beijing X for Xian>: ");

scanf("%c",&from);

getchar();

}while(from != 'C' &&from != 'c' &&from!='b' && from !='B'&& from !='X' && from != 'x');

do{

puts("请输入到达城市<C for Chendu B for Beijing X for Xian>: ");

scanf("%c",&to);

getchar();

}while(to != 'C' && to != 'c' && to!='b' && to !='B'&& to !='X' && to != 'x');

//搜索直飞航线

printf("直飞航线：\n");

puts("No\t出发城市\t 到达城市\t离港时间\t到达时间\n");

for(i=0;i<g\_num;i++)

{

if(flightary[i].cityfrom==from && flightary[i].cityto==to)

{

displaysingle(i);

}

}

printf("经停航线：\n");

puts("No\t出发城市 经停城市 到达城市\t离港时间\t到达时间\n");

for(i=0;i<g\_num;i++)

{

//搜索经停

if(flightary[i].cityfrom==from && flightary[i].cityto!=to)

{

for(j=0;j<g\_num;j++)

{

if(flightary[j].cityfrom==flightary[i].cityto && flightary[j].cityto==to)

{

if(flightary[i].depttime+flightary[i].flttime+1<=flightary[j].depttime)

{

//printf("%2d\t%c\t%c\t%2d\t%2d--",i,flightary[i].cityfrom,flightary[i].cityto,flightary[i].depttime,flightary[i].flttime);

//printf("%2d\t%c\t%c\t%2d\t%2d\n",j,flightary[j].cityfrom,flightary[j].cityto,flightary[j].depttime,flightary[j].flttime);

int temphour=0;

printf("%2d-%2d\t\t%c\t%c\t%c\t\t%2d\t\t",i,j,flightary[i].cityfrom,flightary[i].cityto,flightary[j].cityto,flightary[i].depttime);

temphour=flightary[j].depttime+flightary[j].flttime;

if(temphour>24)

printf("%2d+1day\n",temphour%24);

else

printf("%2d\n",temphour);

}

}

}

}

}

system("pause");

}

//B卷

1. while(fread(&flightary[g\_num], sizeof(FLIGHT), 1, pfile) == 1 )
2. void bubblesortflight()

{

unsigned int pass;

unsigned int i;

for (pass = 1; pass < g\_num; ++pass) {

// loop to control number of comparisons per pass

for (i = 0; i < g\_num - 1; ++i) {

// compare adjacent elements and swap them if first

// element is greater than second element

if (flightary[i].cityfrom > flightary[i + 1].cityfrom ) {

FLIGHT hold = flightary[i];

flightary[i] = flightary[i + 1];

flightary[i + 1] = hold;

}

if (flightary[i].cityfrom == flightary[i + 1].cityfrom && flightary[i].cityto > flightary[i + 1].cityto ) {

FLIGHT hold = flightary[i];

flightary[i] = flightary[i + 1];

flightary[i + 1] = hold;

}

if (flightary[i].cityfrom == flightary[i + 1].cityfrom && flightary[i].cityto == flightary[i + 1].cityto && flightary[i].depttime > flightary[i + 1].depttime ) {

FLIGHT hold = flightary[i];

flightary[i] = flightary[i + 1];

flightary[i + 1] = hold;

}

}

}

}

1. void search()

{

int i,j;

char from,to;

do{

puts("请输入出发城市<C for Chendu B for Beijing X for Xian>: ");

scanf("%c",&from);

getchar();

}while(from != 'C'&& from !='B'&& from !='X');

do{

puts("请输入到达城市<C for Chendu B for Beijing X for Xian>: ");

scanf("%c",&to);

getchar();

}while(to != 'C' && to !='B'&& to !='X');

printf("\n");

puts("No 航班号 出发城市 到达城市 离港时间 到达时间\t\t余票额 单价\n");

for(i=0;i<g\_num;i++)

{

if(flightary[i].cityfrom==from && flightary[i].cityto==to)

{

displaysingle(i);

}

}

system("pause");

}

1. int bookticket()

{

int i=0,flightno=0;

int booknum = 0;

do{

printf("请输入航班编号: ");

scanf("%d",&flightno);

}while(flightno <10 || flightno > 99);

for(i=0;i<g\_num;i++)

{

if(flightno == flightary[i].fltno && flightary[i].tcktnum > 0)

{

printf("目前余票%d，请输入购票数<xx>: ",flightary[i].tcktnum);

scanf("%d",&booknum);

if(booknum > flightary[i].tcktnum)

{

printf("购票数目大于余额，本次购票失败\n");

system("pause");

return -1;

}

else

{

printf("您已成功购票%d张：%c到%c，起飞时间%d，应付总金额%d\n",booknum,flightary[i].cityfrom,flightary[i].cityto,flightary[i].depttime,booknum\*flightary[i].price);

flightary[i].tcktnum -= booknum;

system("pause");

return booknum;

}

}

}

}

//C卷

1. unsigned int CalcChksum(unsigned char \*p, int num)

{

unsigned int nChkSum = 0;

int i;

for (i=0; i<num; i++)

nChkSum += \*(p+i);

return nChkSum;

}

1. int RecAdd\_GotoWork()

{

records recTemp;

if(gNumRecords >= RECORDS\_MAX)

{

printf("达到存储上限，无法增加\n");

return FALSE;

}

printf("请输入姓名: ");

gets(recTemp.name);

printf("请输入日期(20171201格式): ");

scanf("%d", &recTemp.date);

if (RecSearch\_byDateName(recTemp.date, recTemp.name) >= 0)

{

printf("该日期已经存在该人的上班记录，添加无效\n");

return FALSE;

}

printf("请输入上班时间(1001格式): ");

scanf("%d", &recTemp.goto\_time);

recTemp.getoff\_time = FALSE;

recTemp.attendence = FALSE;

gRecords[gNumRecords] = recTemp;

gNumRecords++ ;

return TRUE;

}

3.

void RecSort\_byDateName()

{

int i, j=0;

for (i=0; i<gNumRecords-1; i++) /\*只需要循环n-1次\*/

{

for (j=0; j<gNumRecords-i-1; j++) /\*需要循环n-i-1次\*/

{ /\*一次循环后，大的值就到了最后\*/

if (gRecords[j].date > gRecords[j+1].date ) /\*前面的值大于后面的就交换，是从小到大排序\*/

{

records recTemp;

recTemp = gRecords[j];

gRecords[j] = gRecords[j+1];

gRecords[j+1] = recTemp;

}

else if (strcmp(gRecords[j].name, gRecords[j+1].name) > 0 ) /\*前面的值大于后面的就交换，是从小到大排序\*/

{

records recTemp;

recTemp = gRecords[j];

gRecords[j] = gRecords[j+1];

gRecords[j+1] = recTemp;

}

}

}

}