#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable: 4996)

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <Windows.h>

//频繁用到火车表的信息

#define HEADER1 "------------------------------------BOOK TICKET------------------------------------\n"

#define HEADER2 "| number |start city|reach city|takeoff time|receive time|price|ticket number|\n"

#define HEADER3 "|-------------|----------|----------|------------|------------|-----|-------------|\n"

#define FORMAT "|%-10s |%-10s|%-10s|%-10s |%-10s |%5d| %5d |\n"

#define DATA p->data.num,p->data.startcity,p->data.reachcity,p->data.takeofftime,p->data.receivetime,p->data.price,p->data.ticketnum

//火车票信息

struct train

{

char num[10];//列车号

char startcity[10];//出发城市

char reachcity[10];//达到城市

char takeofftime[10];//发车时间

char receivetime[10];//到达时间

int price;//票价

int ticketnum;//票数

};

//订票人的信息

struct man

{

char num[10];//ID

char name[10];//姓名

int bookNum;//订的票数

};

//火车票信息链表的结点结构

typedef struct node

{

struct train data;

struct node\* next;

}TrainNode,\*TrainLink;

//订票人链表的结点结构

typedef struct Man

{

struct man data;

struct Man\* next;

}BookNode,\*BookLink;

//全局变量是否保存信息

int saveflag;

//函数定义

void menu();//菜单显示

void TrainInfo(TrainLink trainHeader);//添加火车信息

void searchtrain(TrainLink trainHeader);//查询火车信息

void printHeader();//格式化输出表头

void printData(TrainLink trainNode);//格式化输出表中数据

void SaveTrainInfo(TrainLink trainHeader);//保存火车信息

void SaveBookInfo(BookLink trainHeader);//保存用户信息

void Bookticket(TrainLink trainHeader, BookLink bookHeader);//订票

void Modify(TrainLink trainHeader);//修改信息

void showTrain(TrainLink trainHeader);//显示列车信息

//#include "header.h"

/\*

主要功能：

按0，退出火车订票系统

按1，进入添加火车票信息界面，在提示下输入火车的车次、起点、终点、出发时间、到达时间、票价和订购的票数。

按2，查询火车票信息，两种查询方式，一种按车次查询，另一种按到达地点查询。

按3，进入订票系统，根据到底地点查询买票。

按4，进入修改火车票界面，根据提示修改火车票信息。

按5，展示所有的火车票。

按6，火车票信息和订票人信息的保存。

\*/

int main()

{

FILE\* fp1, \* fp2;

TrainLink trainTemp2, trainTemp1, trainHeader;//火车信息临时结点1，2，头结点

BookLink bookTemp2, bookTemp1, bookHeader;//预订信息临时结点1，2，头结点

char ch1 = 0, ch2 = 0;

int sel = 0;//选择

trainHeader = (TrainNode\*)malloc(sizeof(TrainNode));//创建头结点

if (trainHeader == NULL)

{

printf("Memory allocation error!\n");//内存分配失败

exit(1);

}

trainHeader->next = NULL;

trainTemp1 = trainHeader;

bookHeader = (BookNode\*)malloc(sizeof(BookNode));

if (bookHeader == NULL)

{

printf("Memory allocation error!\n");//内存分配失败

exit(1);

}

bookHeader->next = NULL;

bookTemp1 = bookHeader;

fp1 = fopen("train.txt", "ab+");//打开存储车票信息的文件

if (fp1 == NULL)

{

printf("can't open this file!\n");

exit(1);

}

while (!feof(fp1))

{

trainTemp2 = (TrainNode\*)malloc(sizeof(TrainNode));

if (trainTemp2 == NULL)

{

printf("Memory allocation error!\n");

exit(1);

}

if (fread(trainTemp2, sizeof(TrainNode), 1, fp1) == 1)//从磁盘文件读取数据，存入链表中

{

trainTemp2->next = NULL;

trainTemp1->next = trainTemp2;

trainTemp1 = trainTemp2;

}

}

fclose(fp1);

fp2 = fopen("man.txt", "ab+");//打开订票客户信息的文件

if (fp2 == NULL)

{

printf("can't open this file!\n");

exit(1);

}

while (!feof(fp2))

{

bookTemp2 = (BookNode\*)malloc(sizeof(BookNode));

if (bookTemp2 == NULL)

{

printf("Memory allocation error!\n");

exit(1);

}

if (fread(bookTemp2, sizeof(BookNode), 1, fp2) == 1)

{

bookTemp2->next = NULL;

bookTemp1->next = bookTemp2;

bookTemp1 = bookTemp2;

}

}

fclose(fp2);

while (1)

{

system("cls");

menu();

printf("\tplease choose (0-6): ");

scanf("%d", &sel);//输入选择

system("cls");

if (sel == 0)

{

if (saveflag == 1)//当退出时判断信息是否保存

{

getchar();

printf("\nthe file have been chagned!do you want to save it(y/n)?\n");

scanf("%c", &ch1);

if (ch1 == 'y' || ch1 == 'Y')

{

SaveBookInfo(bookHeader);

SaveTrainInfo(trainHeader);

}

}

printf("\nThank you!!!You are Welcome too\n");

Sleep(1000);

exit(0);

}

switch (sel)//输入不同的值选择对应的操作

{

case 1:

TrainInfo(trainHeader);//存入火车票信息

break;

case 2:

searchtrain(trainHeader);//查找火车票信息

break;

case 3:

Bookticket(trainHeader, bookHeader);//订火车票

break;

case 4:

Modify(trainHeader);//修改火车票信息

break;

case 5:

showTrain(trainHeader);//展示火车票

break;

case 6:

SaveTrainInfo(trainHeader);//保存火车票信息

SaveBookInfo(bookHeader);//保存订票信息

break;

case 0:

return 0;

}

printf("\nplease press any key to continue.......\n");

getch();

}

}

void menu()

{

printf("\n\n\n\n");

puts("\t\t----------------------------------------------------------------|");

puts("\t\t| Booking Tickets |");

puts("\t\t----------------------------------------------------------------|");

puts("\t\t| 0.quit the system |");

puts("\t\t| 1.Insert a train information |");

puts("\t\t| 2.Search a train information |");

puts("\t\t| 3.Book a train ticket |");

puts("\t\t| 4.Modify the train information |");

puts("\t\t| 5.Show the train information |");

puts("\t\t| 6.save information to file |");

puts("\t\t----------------------------------------------------------------|");

}

void TrainInfo(TrainLink trainHeader)

{

TrainLink p, r, s;

char num[10];

r = trainHeader;//头结点

s = trainHeader->next;//第一个结点

while (r->next != NULL)

{

r = r->next;

} //r为尾节点

while (1)

{

printf("please input the number of the train(0-return)");//0车号，返回主界面

scanf("%s", num);

if (strcmp(num, "0") == 0)

{

break;

}

while (s)//比较车号

{

if (strcmp(s->data.num, num) == 0)//车号存在

{

printf("the train '%s' is existing!\n", num);

return;

}

s = s->next;

}

p = (TrainNode\*)malloc(sizeof(TrainNode));//创建新结点

if (p == NULL)

{

printf("Memory allocation error!\n");//分配失败

exit(1);

}

strcpy(p->data.num, num);//复制车号

printf("Input the city where the train will start:");//输入出发城市

scanf("%s", p->data.startcity);

printf("Input the city where the train will reach:");//输入到达城市

scanf("%s", p->data.reachcity);

printf("Input the time which the train take off:");//输入到达时间

scanf("%s", p->data.takeofftime);

printf("Input the time which the train receive:");//输入到达时间

scanf("%s", p->data.receivetime);

printf("Input the price of ticket:");//输入火车票价

scanf("%d", &p->data.price);

printf("Input the number of booked tickets:");//输入预订票数

scanf("%d", &p->data.ticketnum);

p->next = NULL;//新结点指向NULL

r->next = p;//尾结点指向新结点

r = p;//更新尾结点

saveflag = 1;

}

}

void searchtrain(TrainLink trainHeader)

{

TrainLink s[10], r;

int sel = 0, k = 0, i = 0;

char str1[5], str2[10];

str1[0] = '\0';

str2[0] = '\0';

if (!trainHeader->next)

{

puts("There is no any record!");

return;

}

printf("Choose the way:\n\n1:according to the number of train;\n2:according to the city:\n");

scanf("%d", &sel);//输入选择信号

if (sel == 1)

{

printf("Input the number of train you want:");//车号查询

scanf("%s", str1);

r = trainHeader->next;

while (r != NULL)

{

if (strcmp(str1, r->data.num) == 0)

{

s[i++] = r;//存入记录

break;

}

else

{

r = r->next;

}

}

}

else if (sel == 2)

{

printf("Input the city you want:");//目的城市查询

scanf("%s", str2);

r = trainHeader->next;

while (r != NULL)

{

if (strcmp(str2, r->data.reachcity) == 0)

{

s[i++] = r;//存入记录

}

r = r->next;

}

}

if (i == 0)

{

puts("can not find!");//没有相应记录

return;

}

else

{

printHeader();//输出表头

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

{

printData(s[k]);//输出火车信息

}

}

}

void printHeader()

{

printf(HEADER1);

printf(HEADER2);

printf(HEADER3);

}

void printData(TrainLink trainNode)

{

TrainLink p;

p = trainNode;

printf(FORMAT, DATA);

}

void SaveTrainInfo(TrainLink trainHeader)

{

FILE\* fp;

TrainLink p;

int count = 0, flag = 1;

fp = fopen("train.txt", "wb");//打开只写二进制文件

if (fp == NULL)

{

printf("the file can't be opened!");

return;

}

p = trainHeader->next;

while (p)

{

if (fwrite(p, sizeof(TrainNode), 1, fp) == 1)//向磁盘文件写入数据块

{

p = p->next;

count++;

}

else

{

flag = 0;

break;

}

}

if (flag)

{

printf("save %d train records\n", count);

saveflag = 0;

}

fclose(fp);

}

void SaveBookInfo(BookLink trainHeader)

{

FILE\* fp;

BookLink p;

int count = 0, flag = 1;

fp = fopen("man.txt", "wb");

if (fp == NULL)

{

printf("the file can't be opened!\n");

return;

}

p = trainHeader->next;

while (p)

{

if (fwrite(p, sizeof(p), 1, fp) == 1)

{

p = p->next;

count++;

}

else

{

flag = 0;

break;

}

}

if (flag)

{

printf("saved %d booking records\n", count);

saveflag = 0;

}

fclose(fp);

}

void Bookticket(TrainLink trainHeader, BookLink bookHeader)

{

TrainLink r[10], p;

char ch[2], tnum[10], str[10], str1[10], str2[10];

ch[0] = '\0';

tnum[0] = '\0';

str1[0] = '\0';

str2[0] = '\0';

BookLink q, h;

int i = 0, t = 0, flag = 0, dnum;

q = bookHeader;

while (q->next != NULL)

{

q = q->next;

}

printf("Input the city you want to go:");

scanf("%s", &str);//输入要到达的城市

p = trainHeader->next;

while (p != NULL)//遍历p指针

{

if (strcmp(p->data.reachcity, str) == 0)//检查是否匹配

{

r[i++] = p;//记录存于数组r

}

p = p->next;

}

printf("\n\nthe number of record have %d\n", i);

printHeader();

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

{

printData(r[t]);

}

if (i == 0)

{

printf("\nSorry!Can't find the train for you!\n");

}

else

{

printf("\ndo you want to book it?<y/n>\n");

scanf("%s", ch);

if (strcmp(ch, "Y") == 0 || strcmp(ch, "y") == 0)

{

h = (BookNode\*)malloc(sizeof(BookNode));

if (h == NULL)

{

printf("Memory allocation error!\n");

exit(1);

}

printf("Input your name:");

scanf("%s", str1);

strcpy(h->data.name, str1);

printf("Input your id:");

scanf("%s", str2);

strcpy(h->data.num, str2);

printf("please input the number of the train:");

scanf("%s", tnum);

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

{

if (strcmp(r[t]->data.num, tnum) == 0)

{

if (r[t]->data.ticketnum < 1)

{

printf("sorry,no ticket!\n");

Sleep(2000);

return;

}

printf("remain %d tickets\n", r[t]->data.ticketnum);//显示剩余火车票

flag = 1;

break;

}

}

if (flag == 0)

{

printf("input error!\n");

Sleep(2000);

return;

}

printf("Input your bookNum:");

scanf("%d", &dnum);//输入需要预订的票数

r[t]->data.ticketnum = r[t]->data.ticketnum - dnum;//更新火车票

h->data.bookNum = dnum;

h->next = NULL;

q->next = h;

q = h;

printf("\nLucky!you have booked a ticket!\n");

saveflag = 1;

}

}

}

void Modify(TrainLink trainHeader)

{

showTrain(trainHeader);

TrainLink p;

char tnum[10], ch;

tnum[0] = '\0';

p = trainHeader->next;

if (!p)

{

printf("\nthere isn't record for you to modify\n");

return;

}

else

{

printf("\nDo you want to modify it?(y/n)\n");

getchar();

scanf("%c", &ch);

if (ch == 'y' || ch == 'Y')

{

printf("\nInput the number of the train:");

scanf("%s", tnum);

int isFind = 0;

while (p != NULL)

{

if (strcmp(p->data.num, tnum) == 0)

{

printf("Input new number of train:");

scanf("%s", p->data.num);

printf("Input new city the train will start:");

scanf("%s", p->data.startcity);

printf("Input new city the train will receive:");

scanf("%s", p->data.reachcity);

printf("Input new time the train take off:");

scanf("%s", p->data.takeofftime);

printf("Input new time the train reach:");

scanf("%s", p->data.receivetime);

printf("Input new price of the ticket:");

scanf("%d", &p->data.price);

printf("Input new number of people who have booked ticket:");

scanf("%d", &p->data.ticketnum);

saveflag = 1;

isFind = 1;

break;

}

else

{

p = p->next;

}

}

if (!isFind)

{

printf("\tCan't find the record!\n");

}

}

}

}

void showTrain(TrainLink trainHeader)

{

TrainLink p;

p = trainHeader->next;

printHeader();

if (p == NULL)

{

printf("no records!\n");

}

else

{

while (p != NULL)

{

printData(p);

p = p->next;

}

}

}