**程序设计与实践**

**实验报告**

**实验五：停车场管理**

**实验时间：2022.04.13 ~ 2022.04.22**

**目录**

**实验名称………………………………………………………………P3**

**实验环境………………………………………………………………P3**

**实验目的………………………………………………………………P3**

**实验步骤………………………………………………………………P3**

**1.驶入…………………….…..……………..………………………………P3**

**2.驶离…………………….…..……………..………………………………P4**

**3.修改车位数…………………….…..……………..………………………P6**

**4.主函数与菜单…………………….…..……………..……………………P6**

**5.退出程序…………………….…..……………..…………………………P7**

**实验总结………………………………………………………………P7**

**参考资料………………………………………………………………P8**

**一：****实验名称：停车场管理**

**二：实验环境: 一台安装有C语言编译器的电脑。**

**三：实验目的**

（1）培养和锻炼对具有一定复杂度和规模的问题的分析与求解能力。

（2）掌握程序设计的综合方法，能综合应用各种数据类型实现较复杂数据的存储。

（3）培养良好的C程序设计风格与代码规范意识。

**四：实验内容**

　　设有一个可以停放n辆汽车的狭长停车场，它只有一个大门供车辆进出。车辆按到达停车场时间的先后次序依次从停车场最里面向大门口处停放。如果停车场已放满ｎ辆车，则以后到达的车辆只能在停车场大门外的便道上等待，一旦停车场内有车开走，则排在便道上的第一辆车可以进入停车场。停车场内如有某辆车要开走，则在它之后进入停车场的车都必须先退出停车场为它让路，待其开出停车场后，这些车辆再依原来次序进场。编写程序对该停车场进行管理。

输入数据时，先输入一个整数总车位n（n<=10），再输入若干组数据，每组数据包括3个数据项：汽车到达或离开的信息（A到达、D离开、E结束）、汽车号码、汽车到达或离开的时刻。当输入“E 0 0”时程序结束。

若有车辆到达，则输出该汽车的停车位置；若有车辆离开，则输出该汽车在停车场内停留的时间。

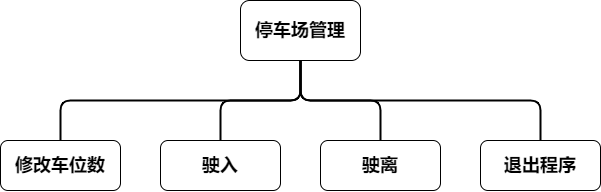


图4.1 停车场管理的功能图

**五：实验步骤**

**本次实验的基本要求是完成停车场管理的正常驶入和正常驶离。同时兼具计算停车时间、收取停车费、修改总车位数等功能。**

**注意：本程序在输入整数数字处，对程序做了保护措施。如果输入非法字符，会直接退出程序。**

1、驶入

输入 ’ A ’ 进入停车页面，然后输入“车牌号(char)”和“停车时间(int)”。程序会先检查停车场内是否有空位，有车位则停入，没有车位则停在便道上进行排队等待。最后将数据存入结构体数组并提示用户停车成功。

void arrive(int cars) //函数 - 入场

{

int flag=1; //测试cars内是否有空位

scanf("%s",num); //输入车牌

scanf("%d",&time); //输入时间

for(i=1;i<=cars;i++) //for循环查找空位

{

if(strlen(p1[i].number)==0) //查找方式是车牌号的char型的长度是否为'空(NULL)'

{

strcpy(p1[i].number,num); //保存输入的车牌号进入结构体

p1[i].ptime=time; //保存输入的入场时间进入结构体

printf("\n入场成功，您的车位号是%d号。",flag); //提示使用者成功停入

count++; //车辆总数增加

break;

}

else

{

flag++;

continue;

}

}

if(flag>cars) //停车场无空位的情况下

{

for(i=cars+1;;i++) //已知无空位，for循环从cars+1开始，因为一定会跳出循环，所以不设跳出参数

{

if(strlen(p1[i].number)==0) //查找方式是车牌号的char型的长度是否为'空(NULL)'

{

strcpy(p1[i].number,num); //保存输入的车牌号进入结构体

p1[i].ptime=time; //保存输入的入场时间进入结构体

printf("\n当前车位已满，请停留在便道上等待。\n",p1[i].number); //提示使用者成功停入

count++; //车辆总数增加

break; //跳出循环

}

}

}

printf("\n");

}

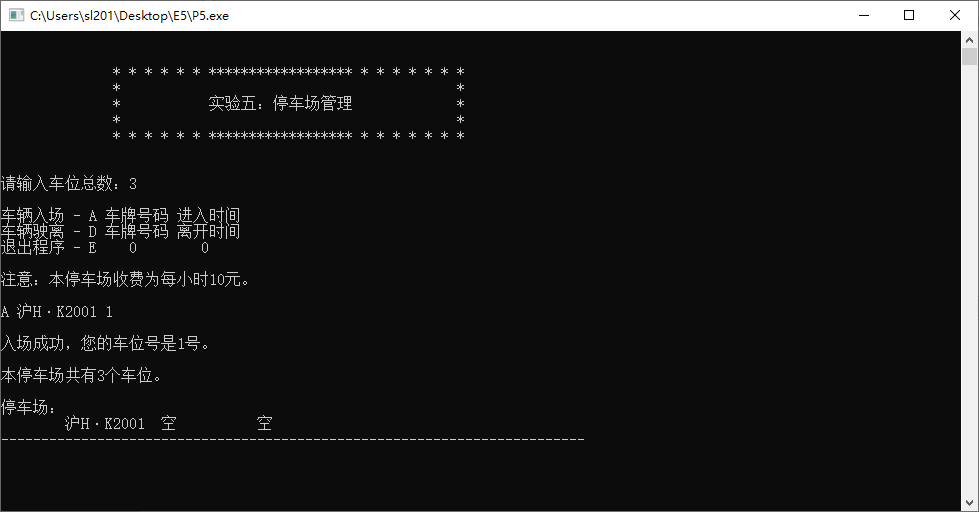
 

图5.1-1 驶入停车场、便道上

2、驶离

输入 ’ D ’ 进入驶离页面，然后输入“车牌号(char)”和“停车时间(int)”。程序先寻找与输入的车牌号相一致的结构体数据。如果原车辆停留在便道上，分别有2种方案：第一是继续等待车位、第二是离开排队等待。如果是离开排队的话，等候区其他的车辆将进行重新排列。如果原车辆停留在停车场，计算停留时间，对应车辆驶离停车场，最后调整其他车位。如果停留时间小于0，作为非法驶离，需要重新输入离开时间。

void change(int time,int cars,int i) //函数 - 移动车位

{

for(int j=i;j<cars;j++){ //调整停车场车位

strcpy(p1[j].number,p1[j+1].number);

p1[j].ptime=p1[j+1].ptime;

}

strcpy(p1[cars].number,"\0"); //末位为空

p1[cars].ptime=time; //便道上第一辆车的进场时间

for(int q=cars+1;q<=count;q++) //停车场末位+等候区车位、时间顺序重置

{

if(strlen(p1[q].number)!=0)

{

strcpy(p1[cars].number,p1[q].number);

p1[q].ptime=time;

printf("车牌号【%s】成功停入%d号车位。\n\n",p1[cars].number,cars);

strcpy(p1[q].number,"\0");

for(int t=cars+1;t<=count;t++)

{

if(strlen(p1[t].number)==0)

{

strcpy(p1[t].number,p1[t+1].number);

strcpy(p1[t+1].number,"\0");

}

}

break;

}

}

}

void depart(int cars) //函数 - 驶离

{

int flag=0,flag2=0;

scanf("%s",num);

scanf("%d",&time);

for(i=1;i<=count;i++){

if(strcmp(p1[i].number,num)==0&&i>cars) //便道上驶离两种情况

{

printf("车号【%s】当前在便道上等待，是否准备离开？\n1 - 离开 2 - 等待(默认)\n请输入：",p1[i].number);

int request;

if(scanf("%d",&request)!=1)

{

system("CLS");

printf("\n输入非法字符，自动退出程序. . .");

system("pause");

exit(0);

}

if(request==1)

{

strcpy(p1[i].number,"\0");

p1[i].ptime=0;

for(i==cars;i<=count;i++){ //更改便道其他车辆的位置和总数

if(strlen(p1[i].number)==0){

strcpy(p1[i].number,p1[i+1].number);

strcpy(p1[i+1].number,"\0");

count--;

}

}

flag2=1;

break;

}

else

{

printf("继续等待中. . .\n");

flag2=1;

break;

}

}

else if(strcmp(p1[i].number,num)==0&&i<=cars) //停车场驶离

{

flag2=1;

int exit;

exit=time-p1[i].ptime;

if(exit<0) //离开时间差值必须大于等于0

{

printf("\n驶离时间非法，请重新输入. . .\n\n");

break;

}

else

{

printf("\n车牌号【%s】停留时间为%d小时，\n需缴付停车费%d元整。欢迎下次光临。\n\n",p1[i].number,exit,exit\*10);

flag++;

strcpy(p1[i].number,"\0"); //驶离车辆设为空和调整时间

p1[i].ptime=0;

change(time,cars,i); //引入更改车位的函数

break;

}

}

else

{

continue;

}

}

count=count-flag; //修改总数

if(flag2==0) printf("\n无该车号的车，请重新输入. . .\n");

}

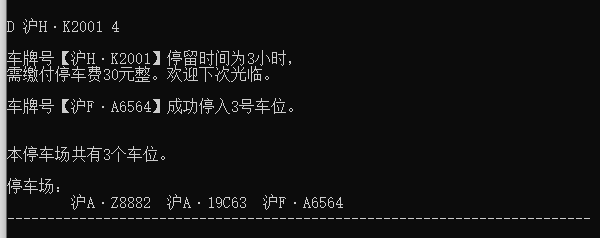


图5.2.1 停车场内正常驶离+新车进入停车场内车位

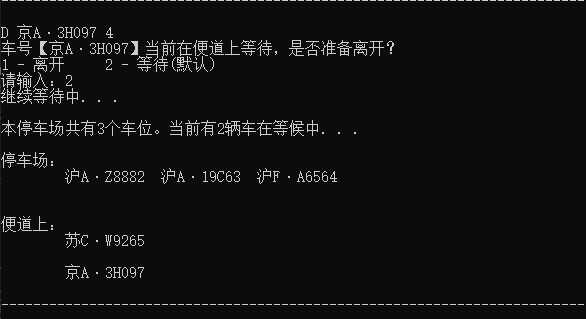
 

图5.2.2 便道上车辆离开或者继续等待

3、修改车位数

输入 ’ C ’ 进入修改页面，然后输入需要修改的车辆数。修改后会重新展示车辆的位置，并保证驶入驶离正常运行。但是扩大停车场，在便道上的车辆的停车时间将从0小时记起。

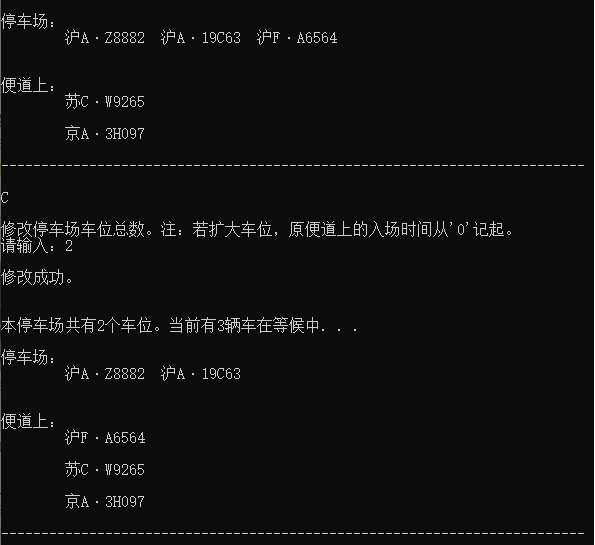


图5.3.1 修改车位数

4、主函数与菜单

菜单部分使用简易的页面（如图）。主函数使用while函数和if作为判断条件，直到break跳出循环。

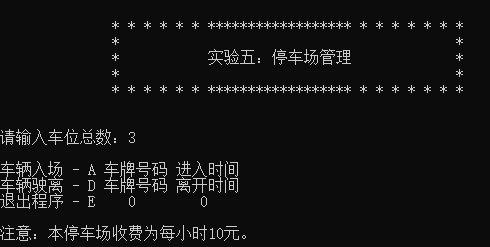


图5.4.1 菜单

5、退出程序

退出程序需要输入 ‘E 0 0‘，输入错误的数字会请求重新输入。

{

scanf("%s",num);

if(scanf("%d",&time)!=1)

{

system("CLS");

printf("\n输入非法字符，自动退出程序. . .");

system("pause");

exit(0);

}

if(num[0]=='0'&&strlen(num)==1&&time==0)

{

printf("\n已成功退出程序. . .");

system("pause");

exit(0);

}

else

{

printf("\n输入错误，请重新输入. . .\n");

}

}

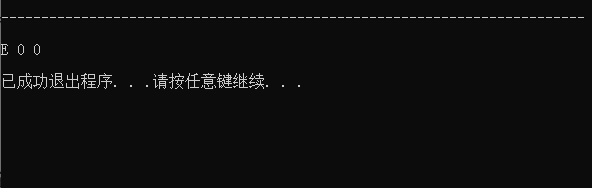


图5.5.1 退出程序

**六：实验小结**

完成情况： 基本完成实验的基本要求，优化了程序的用户提示页面。

需改进处： 在车位移动等方面没用使用数据结构中的链表、队列等功能，导致程序的阅读和功能调整较为繁琐。对驶离的车辆的数据做了完全删除，没用记录保留。

难处： 新的车辆驶入后，结构体数组中要对每一个数据和编号进行查找、修改才能保证程序的稳定运行。

收获： 有了一个完全属于自己的停车场管理的小程序和完整代码。

**七：参考资料**：

[1] C语言程序设计（第4版），何钦铭，颜晖主编。