**暨南大学本科实验报告专用纸**

课程名称 数据结构实验 成绩评定

实验项目名称 停车场管理 指导教师 刘波、郭科芩

实验项目编号 8060154906 实验项目类型 设计性 实验地点 N116

学生姓名 阮炜霖 学号 2020101603

学院 信息科学技术学院 系 计算机系 专业 网络工程

实验时间 2021 年 9 月 9 日 上 午～ 9 月 9 日 上 午

**（一）实验目的和要求；**

目的：熟练栈和队列的结构特性，掌握在实际问题背景下的应用。

要求：任选一种高级程序语言编写源程序，并调试通过，测试正确。

**（二）实验主要内容；**

* 问题描述：

设停车场是一个可停放n 辆汽车的狭长通道，且只有一个大门可供汽车进出。汽车在停车场内按车辆达到时间的先后顺序，依次由北向南排列（大门在最南端，最先达到的第一辆车停放在车场的最北端），若车场内已停满n辆汽车，则后来的汽车只能在门外的便道上等候，一旦有车开走，则排在便道上的第一辆车即可开入；当停车场内某辆车要离开时，在它之后进入的车辆必须先退车车场为它让路，待赶辆车开出大门外，其它车辆在按原次序进入车场，每辆停放在车场的车在它离开停车场时必须按它停留的时间长短叫纳费用。试为停车场编制按上述要求进行管理的模拟程序。

* 基本要求

以栈模拟停车场，以队列模拟车场外的便道，按照从终端读入的输入数据序列进行模拟管理。每一组输入数据包括三个数据项：汽车“达到”或“离去”信息、汽车牌照号码以及达到或离去的时刻。对每一组输入数据进行操作后的输出信息为：若是车辆达到、则输出汽车在停车场内或便道上停车位置；若是车辆离去，则输出汽车在停车场内停留的时间和应交纳的费用（在便道上停留的时间不收费）。栈以顺序结构实现，队列以链表结构实现。

* 测试数据

设n=2，输入数据为(“A”,1,5), (“A”,2,10), (“D”,1, 15), (“A”,3,20),(“D”,2, 35), (“E”,0,0),

(“A”,1,5), 其中：“A”表示达到（Arrival）;“D”表示离去(Departure); “E”表示输入结束(End)。

**（三）实验原理**

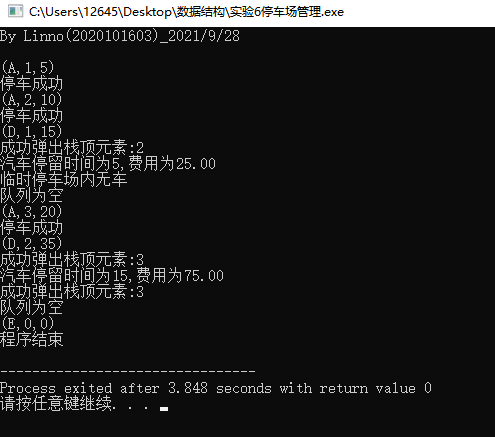
用顺序表实现两个栈，其中一个表示停车场，另外一个表示临时停车场。用链表实现一个表示排队停车的队列。针对每个每个数据结构编写弹入和弹出函数，并完成相应操作。

**（四）实验步骤及调试分析；**

先按照实验要求构造出数据类型，后定义函数实现相关功能，通过输入数据得到的答案

与正确答案进行对比，对程序进行不断调试。

**（五）实验结果及分析；**

通过实验结果，我们得出结论：该程序基本实现了实验内容中的各项要求，并使用栈和队列成功模拟了停车场管理系统。

**（六）附录：源程序**

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<cstring>

#define N 2

using namespace std;

struct X{

int card,tim;

}s[10005];

X stk1[N],stk2[N<<2];

int top1=-1,top2=-1; //设置栈顶指针

typedef struct QNode{

X data;

struct QNode \* nxt;

}QNode;

QNode \*queue,\*front,\*rear;

QNode \* initQueue(){

QNode \* que=(QNode\*)malloc(sizeof(QNode));

que->nxt=NULL;

return que;

}

QNode \* enqueue(QNode \*rear,X data){

QNode \*enElem=(QNode\*)malloc(sizeof(QNode));

enElem->data=data;

enElem->nxt=NULL;

rear->nxt=enElem;

rear=enElem;

return rear;

}

void push1(X data){ //进栈

if(top1==N){

printf("停车场已满\n");

return;

}

top1++;

stk1[top1].card=data.card;

stk1[top1].tim=data.tim;

printf("停车成功\n");

}

void dequeue(QNode \*front,QNode \*rear){

QNode \*p=NULL;

if(front->nxt==NULL){

printf("队列为空\n");

return;

}

p=front->nxt;

push1(p->data); //将它放在停车场内

printf("弹出%d\n",p->data.card);

front->nxt=p->nxt;

if(rear==p){

rear=front;

}

free(p);

}

void push2(X data){ //进栈

if(top2==N){

printf("临时停车场已满\n");

return;

}

top2++;

stk2[top2].card=data.card;

stk2[top2].tim=data.tim;

}

void pop1(){ //出栈

if(top1<=0){

printf("停车场内无车\n");

return;

}

printf("成功弹出栈顶元素:%d\n",stk1[top1].card);

top1--;

}

void pop2(){ //出栈

if(top2<=0){

printf("临时停车场内无车\n");

return;

}

printf("成功弹出栈顶元素:%d\n",stk2[top2].card);

top2--;

}

signed main(){

char op;

int idx=1;

queue=front=rear=initQueue();

puts("By Linno(2020101603)\_2021/9/28\n");

while(1){

scanf("(%c,%d,%d)",&op,&s[idx].card,&s[idx].tim);

getchar();

if(op=='E') break; //结束程序

if(op=='A'){

if(top1==N) rear=enqueue(rear,s[idx]); //排队

else push1(s[idx]); //停车成功

}

if(op=='D'){

while(top1!=0){ //直到所有车辆都被弹出

push2(stk1[top1]);

pop1();

}

int time=s[idx].tim-stk2[top2].tim;

double fee=time\*5.0;

printf("汽车停留时间为%d,费用为%.2lf\n",time,fee);

pop2();//将最里面的车辆驶出

while(top2!=0){ //将临时停放的车辆重新驶入停车场

push1(stk2[top2]);

pop2();

}

dequeue(front,rear);

}

idx++;

}

cout<<"程序结束\n";

return 0;

}

**暨南大学本科实验报告专用纸(附页)**