Project 2 实验报告

16340284 张丰露

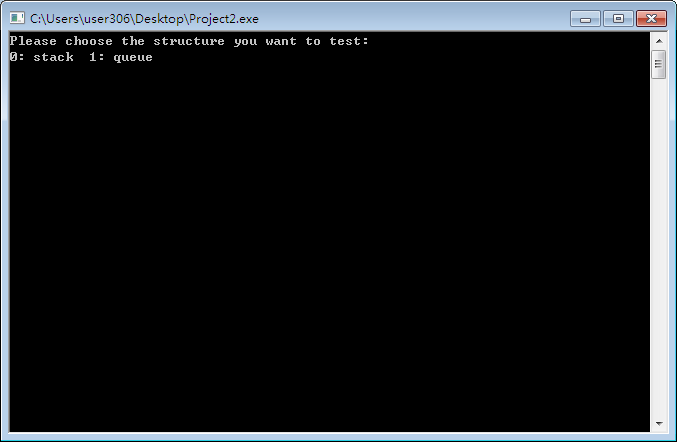
# 实现方法

栈使用链表实现，队列使用顺序储存结构实现。

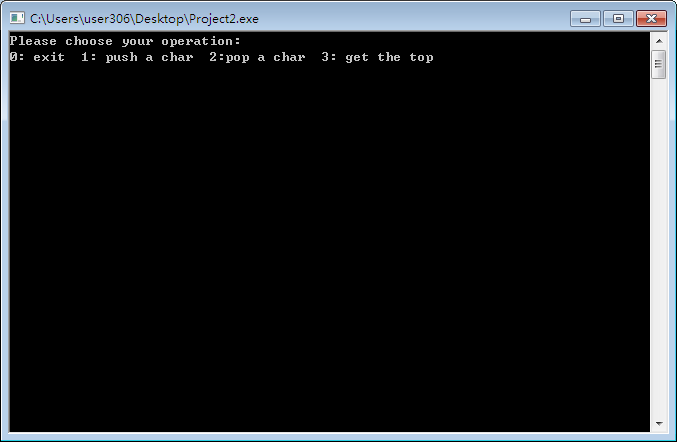
# 运行方法

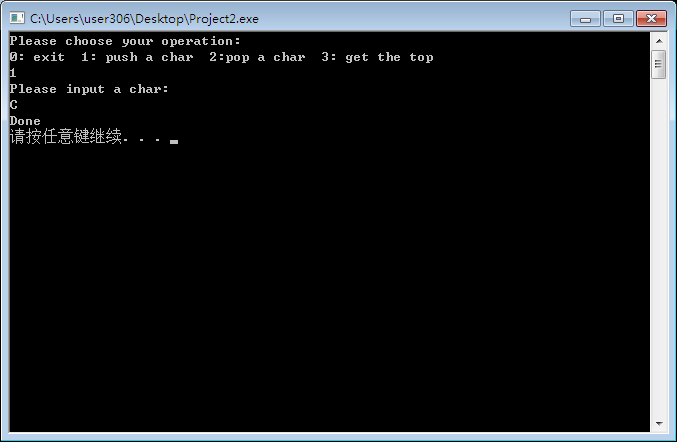
请按程序提示执行。

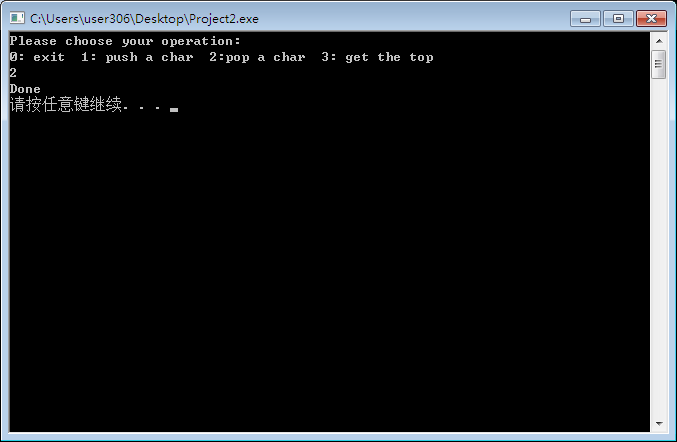
# 运行结果

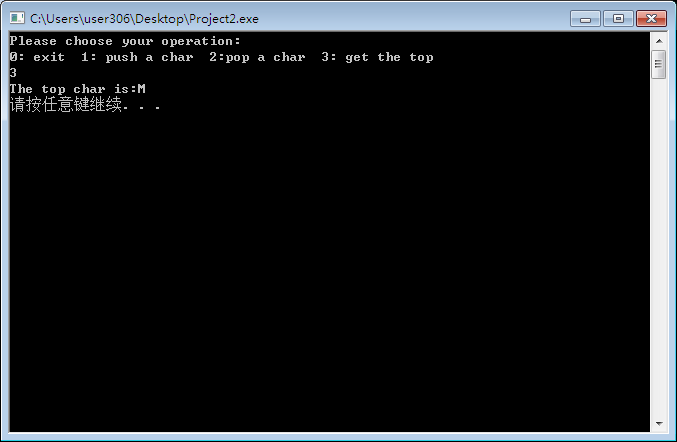


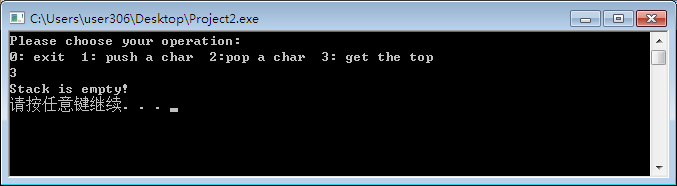
测试栈：



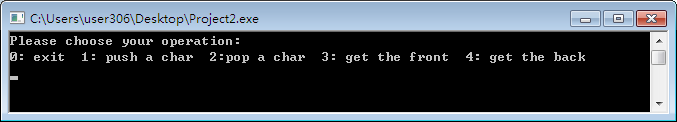


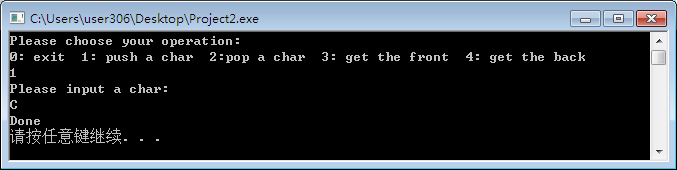


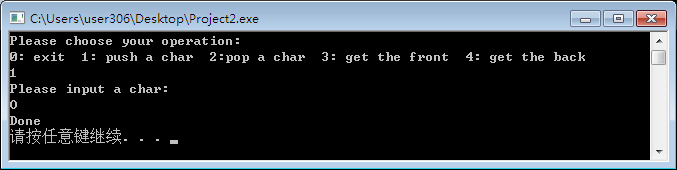


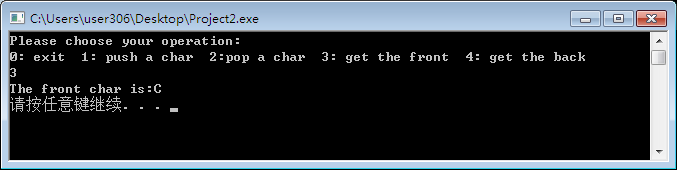


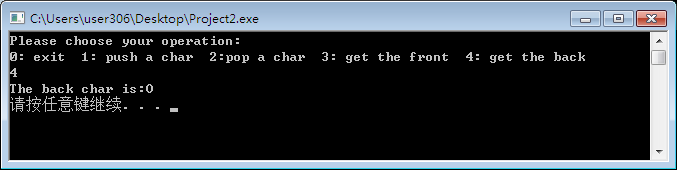
测试队列：

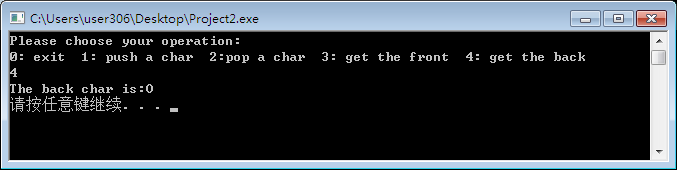












# 源代码

## Stack.hpp

#include <iostream>

using namespace std;

struct node{

char data;

node\* next;

node(char d,node\* n = NULL){

data = d;

next = n;

}

};

class Stack{

node\* top;

public:

Stack(){

top = NULL;

}

~Stack(){

while(top!=NULL){

node\* temp =(\*top).next;

delete top;

top = temp;

}

}

void Push(char c){

if(top == NULL){

top = new node(c);

}

else{

top = new node(c,top);

}

}

void Pop(){

if(top == NULL){

return;

}

node\* temp =(\*top).next;

delete top;

top = temp;

}

char Top(){

if(top == NULL)

return 0;

else

return (\*top).data;

}

};

## Queue.hpp

#include <iostream>

#include <cstring>

using namespace std;

const int maxn = 10000;

class Queue{

char arr[maxn];

int head;

int tail;

public:

Queue(){

head = 0;

tail = 0;

memset(arr,0,sizeof(arr));

}

void Push(char c){

if(head-tail == 1 || head == 0 && tail == maxn-1)

return;

arr[tail] = c;

tail++;

if(tail>=maxn)

tail = 0;

}

void Pop(){

if(head == tail)

return;

head++;

if(head>=maxn)

head = 0;

}

char Front(){

if(head == tail)

return 0;

return arr[head];

}

char Back(){

if(head == tail)

return 0;

//cout<<tail<<endl;

int ttail = tail-1;

if(ttail >= 0)

return arr[ttail];

return arr[maxn-1];

}

};

## main.cpp

#include "Stack.hpp"

#include "Queue.hpp"

#include <cstdlib>

using namespace std;

int main(){

int op =-1;

cout<<"Please choose the structure you want to test:"<<endl;

cout<<"0: stack 1: queue"<<endl;

cin>>op;

system("cls");

if(op == 0){

Stack s;

op = -1;

do{

if(op == 1){

cout<<"Please input a char:"<<endl;

char c;

cin>>c;

s.Push(c);

cout<<"Done"<<endl;

}

else if(op == 2){

cout<<"Done"<<endl;

s.Pop();

}

else if(op == 3){

char c = s.Top();

if(c == 0){

cout<<"Stack is empty!"<<endl;

}

else{

cout<<"The top char is:"<<c<<endl;

}

}

if(op!=-1)

system("pause");

system("cls");

cout<<"Please choose your operation:"<<endl;

cout<<"0: exit 1: push a char 2:pop a char 3: get the top"<<endl;

cin>>op;

}

while(op!=0);

}

else{

Queue q;

op = -1;

do{

if(op == 1){

cout<<"Please input a char:"<<endl;

char c;

cin>>c;

q.Push(c);

cout<<"Done"<<endl;

}

else if(op == 2){

cout<<"Done"<<endl;

q.Pop();

}

else if(op == 3){

char c = q.Front();

if(c == 0){

cout<<"Queue is empty!"<<endl;

}

else{

cout<<"The front char is:"<<c<<endl;

}

}

else if(op == 4){

char c = q.Back();

if(c == 0){

cout<<"Queue is empty!"<<endl;

}

else{

cout<<"The back char is:"<<c<<endl;

}

}

if(op!=-1)

system("pause");

system("cls");

cout<<"Please choose your operation:"<<endl;

cout<<"0: exit 1: push a char 2:pop a char 3: get the front 4: get the back"<<endl;

cin>>op;

}

while(op!=0);

}

system("cls");

cout<<"Thank you!\n";

return 0;

}