实验3 队列操作实现及应用

1. 实验目的

（1）能实现队列的顺序存储和链式存储下的基本操作；

（2）能设计队列结构并用队列基本操作解决实际问题。

2. 实验内容

（1）第一关：队列数据结构设计及基本操作实现：包括队列初始化、判断队列是否为空、入队、出队运算和求队头元素等操作；

（2）第二关：利用栈和队列操作实现回文判断。

详见educoder

**任务描述**

本关任务： （1）编写实现队列，至少包括建立空队列、判断队列是否为空、入队、出队和求队头元素等功能， （2）设计主程序调用队列和自己在前面实验中实现的栈实现回文判断算法。

**相关知识**

队列与栈相关知识请参考课本。 使用队列和栈进行回文判断的算法较为简单，此处仅仅是为了练习栈与队列的使用而设计的场景，具体算法如下：

（1）读入字符串到一维数组

（2）将字符串中的字符依次复制到队列中

（3）将字符串中的字符依次压入栈中

（4）循环，出队一个字符，出栈一个字符，进行比较字符是否相等。

**输入提示**

（1）scanf函数一般格式为scanf(“%s”,st),但scanf默认回车和空格是输入不同组之间的间隔和结束符号，所以输入带空格，tab或者回车的字符串是不可以的。 （2）可以用scanf(“%[^/n]”,str)//遇到回车结束，/n换成c则表示遇到字符c结束 （3）这里假设所输入的字符串没有空格，tab或者回车之类的字符

**编程要求**

根据提示，在右侧编辑器完整自己编写代码实现。

**测试说明**

平台会对你编写的代码进行测试：

测试输入：123abc 预期输出：no

测试输入：abcdcba 预期输出：yes

#include <iostream>

using namespace std;

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef char DataType;

//队列数据元素结构

typedef struct node

{

   DataType info;

   struct node \*next;

}QueueData;

typedef struct queueRecord{

   QueueData  \*front, \*rear ;

 }LINKQUEUE;

typedef struct queueRecord \*PLinkQueue;

PLinkQueue createEmptyQueue\_link( )

{

   //创建一个空队列，实质：生成一个LINKQUEUE类型的结点，并给front 和 rear 成员赋值

    //请在此处填写代码，完成相应功能

    /\*-------begin----------\*/

    /\*-------end----------\*/

}

int isEmptyQueue\_link(PLinkQueue  queue)

{ //判定队列是否为空，实质： 看队列的front指针是否为空，若为空，则队列为空

  //请在此处填写代码，完成相应功能

    /\*-------begin----------\*/

    /\*-------end----------\*/

}

 void enQueue\_link(DataType x, PLinkQueue queue)

 {

   //将数据元素x插入队尾。实质：生成一个struct node类型的结点，并给相应成员赋值后插入队尾

    //请在此处填写代码，完成相应功能

    /\*-------begin----------\*/

    /\*-------end----------\*/

 }

DataType deQueue\_link(PLinkQueue Q)

 {

     //出队，实质： 取出Q队列的队首结点，返回该结点的数据元素，并将该结点使用enQueue\_link(QueueData \*p,PLinkQueue Q)插入队尾

     //本函数为针对本实验特殊设计，可实现秘钥的循环使用

     //请在此处填写代码，完成相应功能

    /\*-------begin----------\*/

    /\*-------end----------\*/

 }

 //下面放自己在有关栈的实验中写过的栈的代码，或者如果会用stl中的stack也可以用

 //注意需要根据具体情况做必要修改

/\*-------begin----------\*/

/\*-------end----------\*/

/\*

函数功能：输入字符串，使用栈和队列，判断该字符串是否是回文，是回文则返回1，否则返回0

输入参数： src 放的是要判断是否是回文的字符串

返回值：1：回文；0：不是回文

\*/

int ishuiwen(char src[])

 {

/\*-------begin----------\*/

/\*-------end----------\*/

 }

 int main(void)

 {

 //请自行补充完整主函数，调用函数ishuiwen进行判断。

 /\*-------begin----------\*/

/\*-------end----------\*/

    return 0;

 }