苏州城市学院实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算科学与人工智能学院 | | 年级专业 | | 22物联网 | | 姓名 | 王子超 | 学号 | 2200443011 |
| 课程名称 | | 数据结构 | | | | | | | 成绩 |  |
| 指导教师 | |  | | 同组实验者 | | 无 | | 实验日期 | 2024/3/4 | |

|  |  |
| --- | --- |
| 实 验 名 称 | 实验二 实现顺序线性表 |

代码：

#include<iostream>

#include<string>

using namespace std;

#define N 20

typedef char change;

typedef struct {

change a[N];

int front;

int rear;

}sqQueue;

void init(sqQueue &L){

L.front = L.rear = 0;

}

bool EnQueue(sqQueue &L,change x){

if((L.rear+1) % N == L.front)

return false;

L.a[L.rear] = x;

L.rear = (L.rear+1) % N;

return true;

}

change DeQueue(sqQueue &L,change &x){

if(L.front == L.rear)

return false;

x = L.a[L.front];

L.front = (L.front + 1) % N;

return true;

}

int judge(string s){

sqQueue A;

init(A);

int index=0;

while( s[index]!=':' && s[index]!='\0'){

EnQueue(A,s[index]);

index++;

}

if(s[index] != ':'){

cout<<"N"<<endl;

return 0;

}

index++;

int leftlen = (A.rear - A.front) % N;

int rightlen = 0;

int result = 1;

while(s[index]!='\0') {

change rightchar;

DeQueue(A,rightchar);

if(s[index] != rightchar){

result = 0;

}

rightlen++;

index++;

}

if(leftlen > rightlen){

cout<<"L"<<endl;

}else if(rightlen > leftlen){

cout<<"R"<<endl;

}else if(leftlen == rightlen){

if(result)

cout<<"S"<<endl;

else

cout<<"D"<<endl;

}

return 1;

}

int main(){

string s;

cout<<"请输入理想字符:( abcd:bedw )"<<endl;

getline(cin,s);

judge(s);

return 0;

}

运行结果:

