

**数据结构实验报告**

**实验报告题目：** **顺序表基本算法的实现**

**学生姓名**：  **刘凌杰**

**专 业： 软件工程**

**学 号**： 201720180411

**指导教师： 邹 国 华**

**2018年 10 月 28 日**

**一. 实验内容：**

* **队的初始化**
* **队的清除**
* **队是否为空**
* **队是否为满**
* **获取队头元素**
* **取出队头元素**
* **队尾添加元素**
* **队头删除元素**
* **队的长度**

**二. 实验算法实现：**

**1.结构体**

typedef int DataType;

typedef struct SeqQueue{

DataType data[QUEUESIZE];

int front; //队头（删除）

int rear; //队尾（插入）

}SeqQueue;

**2.队的初始化**

//队的初始化

void lnitQueue(SeqQueue \*q)

{

q->front=0;

q->rear=0;

}

**3.队的清除**

//队的清除

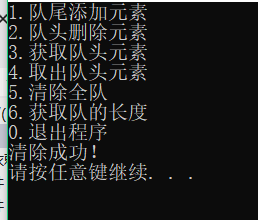
void clearQueue(SeqQueue \*q)

{

q->front=0;

q->rear=0;

}



**4.判断队是否为空**

//判断队是否为空

bool queueEmpty(SeqQueue \*q)

{

return q->front==q->rear;

}

**5.判断队是否为满**

//判断队是否为满

bool queueFull(SeqQueue \*q)

{

return q->front==(q->rear+1)%QUEUESIZE;

}

**6.获取队头元素**

//获取队头元素

bool getHead(SeqQueue \*q,DataType \*e)

{

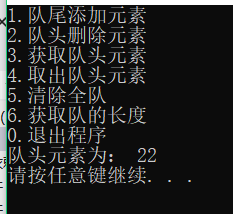
if(queueEmpty(q))

return false;

\*e=q->data[q->front];

return true;

}



**7.取出队首元素**

//取出队首元素

bool deQueue(SeqQueue \*q,DataType \*e)

{

if(queueEmpty(q))

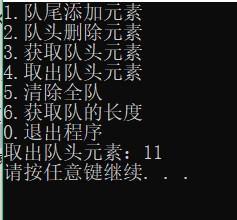
return false;

\*e=q->data[q->front];

q->front=(q->front+1)%QUEUESIZE; //当front溢出数组时，进行数组位置还原，形成循环队列

return true;

}



**8.队尾添加元素**

//队尾添加元素

bool enQueue(SeqQueue \*q,DataType e)

{

if(queueFull(q))

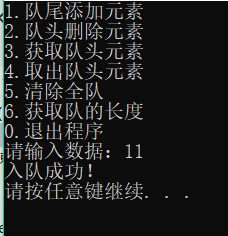
return false;

q->data[q->rear]=e;

q->rear=(q->rear+1)%QUEUESIZE;

return true;

}



**9.求队长度**

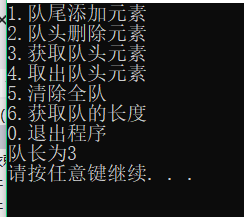
//求队元素个数

int queueLength(SeqQueue \*q)

{

return (q->rear)-(q->front);

}



**10.队头删除元素**

//队头删除元素

bool deleteQueue(SeqQueue \*q)

{

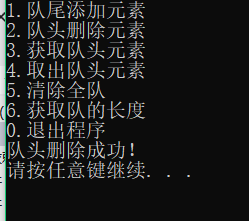
if(queueEmpty(q))

return false;

(q->front+1)%QUEUESIZE;

return true;

}

****

**11.main函数**

void main()

{

SeqQueue queue;

int i;

char ch;

DataType e;

lnitQueue(&queue);

while(1)

{

cout<<"1.队尾添加元素"<<endl;

cout<<"2.队头删除元素"<<endl;

cout<<"3.获取队头元素"<<endl;

cout<<"4.取出队头元素"<<endl;

cout<<"5.清除全队"<<endl;

cout<<"6.获取队的长度"<<endl;

cout<<"0.退出程序"<<endl;

ch=getch();

switch(ch)

{

case'1':

cout<<"请输入数据：";

cin>>e;

if(enQueue(&queue,e))

cout<<"入队成功！"<<endl;

else

cout<<"队已满！"<<endl;

system("pause");

system("cls");

break;

case'2':

if(deleteQueue(&queue))

cout<<"队头删除成功！"<<endl;

else

cout<<"空队！"<<endl;

system("pause");

system("cls");

break;

case'3':

if(getHead(&queue,&e))

cout<<"队头元素为： "<<e<<endl;

else

cout<<"空队！"<<endl;

system("pause");

system("cls");

break;

case'4':

if(deQueue(&queue,&e))

cout<<"取出队头元素："<<e<<endl;

else

cout<<"空队！"<<endl;

system("pause");

system("cls");

break;

case'5':

clearQueue(&queue);

cout<<"清除成功！"<<endl;

system("pause");

system("cls");

break;

case'6':

cout<<"队长为"<<queueLength(&queue)<<endl;

system("pause");

system("cls");

break;

case'0':

exit(0);

default:

cout<<"输入序号有误！"<<endl;

system("pause");

system("cls");

}

}

}

**三．实验总结：**

经过一段时间的学习，队算法的实现基本掌握了，就是代码打得还不太熟练，还有待提高，有些功能不太熟要问同学或查资料，尤其在做插入、删除运算时队的长度忘记改变。以后在这方面我会多花点时间，让自己更加熟练。