

实验二 队列

班级:_____姓名:_____学号:_____

一、实验目的

1. 掌握循环队列的存储方式及基本操作的实现;
2. 掌握利用队列解决问题的方法;

二、实验题目与要求

题目: 循环队列的实现

要求:

- (1) 编程实现循环队列的类型定义及其初始化、入队、出队、取队头、遍历等操作;
- (2) 编写程序, 对所实现的循环队列进行验证。

三、实验内容

1. 循环队列的实现

完成循环队列的类型定义及其基本操作函数代码, 并保存为 “CirQueue.h”。设计完成空缺代码并上机调试。

```
typedef struct{
    QElemType *base;
    int front, rear;
    int queueSize;
}CirQueue;
Status InitCirQueue(CirQueue &Q, int QSize){
    //构造一个空间大小为 QSize 的初始空队列 Q
```

```

}
Status queueIsEmpty(CirQueue Q) {
    //判断循环队列 Q 是否为空
}

```

```
void clearQueue(CirQueue &Q){
    //将循环队列 Q 清空
}
int queueLength(CirQueue Q){
    //求循环队列 Q 的长度
}
Status getFront(CirQueue Q, QElemType &e){
    //取队列 Q 的队头元素值
}
Status insertQueue(CirQueue &Q, QElemType e){
    //在循环队列 Q 的队尾端插入元素 e
}
Status deleteQueue(CirQueue &Q, QElemType &e){
    //将队列 Q 的队头元素出队并由 e 返回其值
}
void queueTraverse(CirQueue Q,
    void (*visit)(QElemType e) )
{
    //遍历操作
    int i=Q.front;
    while(i!=Q.rear){
        visit(Q.base[i]);
        i=(i+1)%Q.queueSize;
    }
}

```

2. 循环队列测试程序设计

```
#include <stdio.h>
#include <stdlib.h>
#include "Status.h"
typedef int QElemType;
#include "CirQueue.h"
void visitElem(QElemType e){
    printf("%d  ", e);
}
int main() {
    //用于队列测试的主函数
    CirQueue Q;
    QElemType e;
    int i;
    InitCirQueue(Q,11);
    for(i=1;i<=7; i++){
        insertQueue(Q, 2*i+1);
        printf("队列 Q:");
        queueTraverse(Q, visitElem);
        printf("\n");
        for(i=1;i<=5;i++){
            _____; //出队, 用 e 保存出队元素
            printf("%d 已出队\n", e);
        }
        printf("队列 Q:");
        queueTraverse(Q, visitElem);
        printf("\n");
        for(i=1;i<=6; i++){
            _____; //将 2*i 入队
            printf("队列 Q:");
            _____; //利用遍历输出队列
            printf("\n");
            _____; //取队头元素给 e
            printf("当前队长为:%d, 当前队头元素为:%d\n",
                queueLength(Q), e );
            _____; //队列清空
            printf("队列已清空, 队长为:%d\n",
                queueLength(Q) );
            system("pause");
            return 0;
        }
    }
```

3. 程序运行结果