[**循环队列的实现(出队，入队，遍历等)**](https://www.cnblogs.com/fengberlin/p/5973404.html)

队列的抽象数据类型定义为：

类型名称：队列。

数据对象集：一个有0个或多个元素的有穷线性表。

操作集：对于一个长度为正整数MaxSize的队列Q∈Queue， 记队列中的任一元素item∈ElementType，有：

　　　　（1）Queue CreateQueue(int MaxSize)：创建一个长度为MaxSize的空队列；

　　　　（2）bool isEmpty(Queue Q)：判断队列是否为空，若不空返回true（1），否则返回false（0）；

　　　　（3）void AddQ(Queue Q, ElementType item)：若队列满，返回已满的信息；否则，把这个元素入队；

　　　　（4）bool isFull(Queue Q)：判断队列是否满，若满返回true（1），否则返回false（0）；

　　　　（5）ElementType DeleteQ(Queue Q)：若队列为空，返回队列为空的信息；否则，把队头元素先用临时变量存储起来并从队列中删去，最后返回队头元素。

为了解决队尾溢出（假溢出）而实际上数组仍然有多余空间的问题，我们运用循环队列解决问题。

此时需要定义一个front和rear分别指向队列的头元素的前一个位置和尾元素，并且开始时都初始化成0；

当插入和删除操作的作用单元达到数组的末端后，用公式“rear(或front)%数组长度”取余运算就可以实现折返到起始单元。

队满的条件是：“(rear+1)%数组长度”等于front；队空的条件为：rear等于front。

循环队列的顺序存储结构的定义可以如下：

[复制代码](javascript:void(0);)

// 实现循环队列

#define MaxSize 21

typedef int ElementType;

typedef struct {

ElementType data[MaxSize];

int rear; // 队尾指针

int front; // 队头指针

}Queue;

[复制代码](javascript:void(0);)

循环队列的插入操作如下：

[复制代码](javascript:void(0);)

// 元素入队

void AddQ(Queue \*PtrQ, ElementType item)

{

if( (PtrQ->rear+1)%MaxSize == PtrQ->front )

{

printf("队列满.\n");

return;

}

PtrQ->rear = (PtrQ->rear+1) % MaxSize;

PtrQ->data[PtrQ->rear] = item;

}

[复制代码](javascript:void(0);)

循环队列的删除操作如下：

[复制代码](javascript:void(0);)

// 删除队头元素并把队头元素返回

ElementType DeleteQ( Queue \*PtrQ )

{

if( PtrQ->front == PtrQ->rear )

{

printf("队列空.\n");

return -1;

}

else {

PtrQ->front = (PtrQ->front+1) % MaxSize;

return PtrQ->data[PtrQ->front];

}

}

[复制代码](javascript:void(0);)

循环队列的遍历，这个是我自己摸索了几遍才出来的，所以想记录一下，具体操作如下：

[复制代码](javascript:void(0);)

// 队列元素的遍历

void print(Queue \*PtrQ)

{

int i = PtrQ->front;

if( PtrQ->front == PtrQ->rear )

{

printf("队列空.");

return;

}

printf("队列存在的元素如下:");

while( i != PtrQ->rear)

{

printf("%d ", PtrQ->data[i+1]);

i++;

i = i % MaxSize;

}

return;

}

[复制代码](javascript:void(0);)