1. #include <stdio.h>

#include <malloc.h>

#define MaxSize 100

typedef char ElemType;

typedef struct

{

ElemType data[MaxSize];

int front, rear; //队头和队尾指针

} SqQueue;

void InitQueue(SqQueue\*& q)

{

q = (SqQueue\*)malloc(sizeof(SqQueue));

q->front = q->rear = -1;

}

void DestroyQueue(SqQueue\*& q) //销毁队列

{

free(q);

}

bool QueueEmpty(SqQueue\* q) //判断队列是否为空

{

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

}

bool enQueue(SqQueue\*& q, ElemType e) //进队

{

if (q->rear == MaxSize - 1) //队满上溢出

return false; //返回假

q->rear++; //队尾增1

q->data[q->rear] = e; //rear位置插入元素e

return true; //返回真

}

bool deQueue(SqQueue\*& q, ElemType& e) //出队

{

if (q->front == q->rear) //队空下溢出

return false;

q->front++;

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

return true;

}

int main()

{

SqQueue\* q; //创建队列q

ElemType e;

InitQueue(q); //初始化队

enQueue(q, 'a');

enQueue(q, 'b');

enQueue(q, 'c'); //依次进队a,b,c

deQueue(q, e);

printf("%c\n", e); //出队元素a

deQueue(q, e);

printf("%c\n", e); //出队元素b

deQueue(q, e);

2

如果rear<front结果是rear-front+maxsize

如果rear>front结果是rear-front

为了用一个表达式同时表达两者，用(rear-front+maxsize)%maxsize

假设maxsize=10

rear=1 front=9，那么结果是2

rear=9 front=1，那么结果是8

printf("%c\n", e); //出队元素c

DestroyQueue(q); //销毁队

return 0;

}

