数据结构实验报告实验三-循环队列

一、实验环境：

Windows 10 64位 MinGW GCC 4.8.1

IDEA：CLoin

二、实验内容：

编写程序实现一个循环队列，并实现基本的出队入队功能。

三、算法流程：

显示菜单

输入1：

输入一个元素将它入队并显示当前队列元素。

输入2：

将队首元素出队并打印值，并显示当前队列元素。

输入3：

退出程序。

四、程序代码：

#include**<stdio.h>**#include**<stdlib.h>**#define **MAXSIZE** 5  
#define **ERROR** 0  
#define **OK** 1  
**typedef struct**{  
 **int** \*data;  
 **int** front,rear; *// 头、尾指针*}SqQueue;  
**int** Init\_Q(SqQueue \*Q){  
 Q->data=(**int** \*)malloc(**MAXSIZE**\***sizeof**(**int**));  
 **if**(!Q->data) **return ERROR**;  
 Q->front=Q->rear=0;  
 **return OK**;  
}  
**void** Print\_Q(SqQueue Q){  
 **int** j;  
 printf(**"element in queue:"**);  
 **for**(j=0;j<(**MAXSIZE**+Q.rear-Q.front)%**MAXSIZE**;j++){  
 printf(**"%d"**,Q.data[(Q.front+j)%**MAXSIZE**]);  
 }  
}  
**int** EnQueue(SqQueue \*Q,**int** e){  
 **if**((Q->rear+1)%**MAXSIZE**==Q->front)  
 **return ERROR**;  
 Q->data[Q->rear]=e;  
 Q->rear=(Q->rear+1)%**MAXSIZE**;  
 **return OK**;  
}  
  
**int** DeQueue(SqQueue \*Q){  
 **int** e;  
 **if**(Q->front==Q->rear) **return ERROR**;  
 e=Q->data[Q->front];  
 Q->front=(Q->front+1)%**MAXSIZE**;  
 **return** e;  
}  
**int** main(){  
 **int** k; **int** e,x; **char** ch; SqQueue Q1;  
 Init\_Q(&Q1);  
 **do**{  
 printf(**"\n\n 1. 数据元素e进队列"**);  
 printf(**"\n\n 2. 出队一个元素，返回其值"**);  
 printf(**"\n\n 3. 结束程序运行"**);  
 printf(**"\n===================================="**);  
 printf(**"\n 请输入您的选择(1,2,3)"**);  
 scanf(**"%d"**,&k);  
 **switch**(k){  
 **case** 1:  
 {  
 printf(**"\n 进队e=?"**);  
 scanf(**"%d"**,&e);  
 EnQueue(&Q1,e);  
 Print\_Q(Q1);  
 }  
 **break**;  
 **case** 2:{  
 x= DeQueue(&Q1);  
 printf(**"\n出队元素: %d"**, x);  
 Print\_Q(Q1);  
 }  
 **break**;  
 **case** 3: exit(0);  
 }  
 printf(**"\n ----------------"**);  
 }**while**(k>=1 && k<3);  
 printf(**"\n 再见！"**);  
 printf(**"\n 打回车键，返回。"**);  
 ch=getchar();  
 **return** 0;  
}

五、运行截图：





