

实验 队列的基本运算

1、实验目的

掌握链队列的基本操作，初始化、入队、出队、链队列的销毁以及显示等运算在链式存储结构上的实现。

2、实验内容

- (1) 链队列的初始化;
- (2) 链队列入队操作的实现;
- (3) 链队列出队操作的实现;
- (4) 链队列销毁操作的实现;
- (5) 显示链队列中各个元素。

3、实验要求

- (1) 能够熟练在 Visual C++6.0 环境中进行程序的编辑、编译和调试;
- (2) 会书写类 C 语言的算法，并将算法转变为程序实现。

4、程序运行框架

```
#include <stdio.h>
#include <malloc.h>
```

```
typedef char QElemType;
typedef struct QNode{
    QElemType data;
    struct QNode *next;
}QNode,*QueuePtr;
```

```
typedef struct{
    QueuePtr front;
    QueuePtr rear;
}LinkQueue;
```

```
int InitQueue(LinkQueue &Q) {return 0; }
int EnQueue(LinkQueue &Q,QElemType e) {return 0; }
int DeQueue(LinkQueue &Q,QElemType &e) {return 0; }
int DestroyQueue(LinkQueue &Q) {return 0; }
void DispQueue(LinkQueue Q){ }
void main(){}
```

5、测试数据

补充程序，在链队列中依次将元素 A、B、C、D、E 入队，之后出队，显示队列元素后将全部队列销毁，最后出队并观察运行结果。

实验内容可执行程序：

```
#include <stdio.h>
#include <malloc.h>
```

```

typedef char QElemType;
typedef struct QNode{
    QElemType data;
    struct QNode *next;
}QNode,*QueuePtr;

```

```

typedef struct{
    QueuePtr front;
    QueuePtr rear;
}LinkQueue;

```

```

int InitQueue(LinkQueue &Q) {
    Q.front=Q.rear=(QueuePtr)malloc(sizeof(QNode));
    if(!Q.front)
        return 0;
    _____;
    return 1;}

```

```

int EnQueue(LinkQueue &Q,QElemType e) {
    QueuePtr p;
    p=(QueuePtr)malloc(sizeof(QNode));
    if(!p)
        return 0;
    _____;
    _____;
    Q.rear->next=p;
    _____;
    return 1;}

```

```

int DeQueue(LinkQueue &Q,QElemType &e) {
    QueuePtr p;
    if(_____)
        return 0;
    _____;
    e=p->data;
    _____;
    if(_____)
        _____;
    free(p);
    return 1;}

```

```

int DestroyQueue(LinkQueue &Q) {
    while(Q.front)
    {
        _____;
        free(Q.front);
        _____;
    }
    return 1; }

```

```

void DispQueue(LinkQueue Q)
{
    QueuePtr p;
    if(_____)
        printf("队列为空! \n");
    p=Q.front->next;
    while(p!=NULL)
    {
        printf("%c",_____);
        p=p->next;
    }
}

```

```

void main()
{
    LinkQueue Q;
    QElemType e;
    _____;          /*初始化队列*/
    EnQueue(Q,'A');
    printf("队列为:");
    DispQueue(Q);
    printf("\n");
    _____;          /*入队数据元素 B*/
    printf("队列为:");
    _____;          /*显示队列*/
    printf("\n");
    EnQueue(Q,'C');
    printf("队列为:");
    DispQueue(Q);
    printf("\n");
    EnQueue(Q,'D');
    printf("队列为:");
    DispQueue(Q);
    printf("\n");
}

```

```

        _____;    /*入队数据元素 E*/
printf("队列为:");
DispQueue(Q);
printf("\n");
        _____;    /*出队*/
printf("队列为:");
DispQueue(Q);
printf("\n");
        _____;    /*出队*/
printf("队列为:");
DispQueue(Q);
printf("\n");
        _____;    /*出队*/
printf("队列为:");
DispQueue(Q);
printf("\n");
        _____;    /*出队*/
DispQueue(Q);
        _____;    /*销毁队列*/
}

```