栈和队列

1.栈

```
#include <stdio.h>
#include "stdlib.h"

#define MaxSize 100
#define Element int

typedef struct stack{
    Element data[MaxSize];          // 栈元素的最大个数
    int top;        // 栈顶
} *SqStack,Stack;

void InitStack(SqStack s){
    s = (SqStack) malloc(sizeof(Stack));
    s->top = -1;
}

int StackEmpty(SqStack s){
    if(s->top == -1){
        return 1;        // 空栈
    } else{
        return 0;       // 不为空
    }
}

int PushStack(SqStack s,Element e){
    if(s->top == MaxSize - 1){
        return 0;
    }
    s->data[++s->top] =  e;
    return 1;
}

Element Pop(SqStack s){
    if(s->top == -1){
        return 0;
    }
    Element e = s->data[s->top--];
    return e;
}

Element getTop(SqStack s){
    if(s->top == -1){
        return 0;
    }
    Element e = s->data[s->top];
    return e;
```

```
}
int main() {
    SqStack s = (SqStack) malloc(sizeof(Stack));
    s->top = -1;
//    InitStack(s);
//    SqStack s = NULL;
    InitStack(s);
    PushStack(s,20);
    PushStack(s,30);
    PushStack(s,40);
    PushStack(s,50);
    Pop(s);
    PushStack(s,70);
    Element e = getTop(s);
    printf("%d\n",e);

    for (;s->top != -1;) {
        printf("%d ", Pop(s));
    }
    return 0;
}
```

```
输出:
70
70 40 30 20
```

2.队列

```
#include <stdio.h>
#include "stdlib.h"

#define MaxSize 100
#define Element int

typedef struct {
  Element data[MaxSize];
  int fornt,rear;
} *SqQueue;

void InitQueue(SqQueue Q){
    Q->rear = Q->fornt;
}

int isEmpty(SqQueue Q){
    if(Q->rear == Q->fornt){
        return 1;
    } else{
        return 0;
    }
```

```
}

int EnQueue(SqQueue Q,Element e){
    if((Q->rear+1) % MaxSize == Q->fornt){
        return 0;
    }
    Q->data[Q->rear] = e;
    Q->rear = (Q->rear+1) % MaxSize;
    return 1;
}

Element DeQueue(SqQueue Q){
    if(Q->rear == Q->fornt) return 0;
    Element e = Q->data[Q->fornt];
    Q->fornt = (Q->fornt+1) % MaxSize;
    return e;
}

int main() {
    SqQueue q = (SqQueue *)malloc(sizeof (SqQueue));
    EnQueue(q,288);
    EnQueue(q,35);
    EnQueue(q,98);
    EnQueue(q,3);
    EnQueue(q,83);
    EnQueue(q,99);
    EnQueue(q,555);

    Element e = DeQueue(q);
    printf("%d\n",e);

    for (; q->rear != q->fornt ; ) {
        printf("%d ", DeQueue(q));
    }
    printf("\n");
    int res = isEmpty(q);
    if (res){
        printf("为空");
    } else{
        printf("不为空");
    }

    return 0;
}
```

```
输出:
288
35 98 3 83 99 555
为空
```