

第3章 栈和队列

一. 填空题

- 1、栈
- 2、线性，栈顶，队尾，队头
- 3、栈顶，栈底
- 4、队列
- 5、先存入元素，后移动栈顶指针
- 6、`data[++top]=x;`
- 7、2 3 5 4 1，100CH
- 8、S X S S X S X X
- 9、数组
- 10、`HS=NULL`
- 11、当前位置
- 12、先移动队首指针，后取出元素
- 13、 $n-1$
- 14、 $(M+1) \% N$

二. 单项选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	C	C	C	BACCE	C	A	C	B	D	C
题号	11	12	13	14	15	16	17	18		
答案	D	B	A	C	A	D				

三. 判断题

题号	1	2	3	4	5	6
答案	√	√	√	√	×	×

四. 综合题

1. 解：用一整型变量 `top` 表示栈顶指针，`top` 为 0 时表示栈为空。如果栈不空，则从 `stack[1]` 开始存放元素。实现本题功能的函数如下：

入栈算法：

```
void Push(EleType x)
{
    if((top+length)>n)
        printf("上溢出\n");
    else
    {
        if(top==0) /*为空栈*/
        {
            top++;
            stack[top]=x;
        }
        else
        {
            top=top+length;
            stack[top]=x;
        }
    }
}
```

出栈算法：

```
void Pop(EleType x)
{
    if(top==0)
        printf("为空栈\n");
    else
    {
        if(top==1)
        {
```

```

        x=stack[top];
        top--;
    }
    else
    {
        x=stack[top];
        top=top-length;
    }
}
}

```

2. 解：实现本题功能的函数如下：

```

void travel(Queue, int front,rear)
{
    int i;
    for(i=front;i<=rear;i++)
    {
        printf("%4d",Queue[i]);
    }
}

```

3. 解:用一个循环数组 Queue[0,n-1]表示该循环队列,头指针为 front,计数器 count 用来记录队列中结点的个数。

入队算法如下：

```

void enqueue(int x)
{
    int temp;
    if(count==n)
        printf("队列上溢出\n");
    else

```

```
        {  
            count++;  
            temp = (front+count)%n;  
            Queue[temp]=x;  
        }  
    }
```

出队算法如下：

```
int dequeue()  
{  
    int temp;  
    if(count==0)  
        printf("队列下溢出\n");  
    else  
    {  
        temp=Queue[front];  
        front=(front+1)%n;  
        count--;  
        return temp;  
    }  
}
```

4. 解：设表达式在字符数组 $a[]$ 中，使用一堆栈 S 来帮助判断。实现本题功能的函数如下：

```
int correct(char a[])  
{  
    Stack S;  
    InitStack(S);  
    for(i=0;i<strlen(a);i++)  
        if(a[i]=='(')  
            Push(S,'(');  
}
```

```
        else if (a[i]=='')
        {
            if(StackEmpty(S))
                return 0;
            else
                Pop(S);
        }
        if(StackEmpty(S))
            return 1;          /*配对正确*/
        else
            return 0;          /*配对错误*/
    }
```