数据结构作业 3.2/3.3/3.12/3.14/3.22/3.23

- 姓名：牟鑫一
- 班级：191174班

3.2

3.3

3.12 十进制整数N转换为基为B的B进制数

```
template class<T>
int Convert(int N, int B) {
    int i = 0;   //用于保存转换后B进制数的各位数
    int s = 0;   //用于保存转换后的B进制数
    LinkedStack<int> P;
    while(N != 0) {
        i = N % B;
        N = N / B;
        P.Push(i);    //把二进制数逆序压入栈
    }
    while (!P.isEmpty()) {
        P.Pop(i);          //顺序出栈
```

```
        s = s * 10 + i;    //实际上s以十进制数的形式保存转换后的B进制数
    }
    return s;
}
```

3.14递归算法

（1）求数组A中的最大整数

```
int Max(int A[], int n) {  //传入数组A的地址，大小
    int temp;
    if (n == 1) return A[0];  //当n=1时不用比较直接返回值A[0]作为最大值
    else {
        temp = Max(A, n - 1);  //从数组尾开始向前递归
        if (temp > A[n - 1]) return temp;
        else return A[n - 1];
    }
}
```

（2）求n个整数的和

```
int Sum(int A[], int n) {
    int sum = 0;
    if (n == 1) return A[0];  //当n=1时返回A[0]，逐层返回sum
    else sum = A[n - 1] + Sum(A, n - 1); //从数组尾开始向前递归
    return sum;
}
```

（3）求n个整数的平均值

```
double Average(int A[],int n){
    double ave;
    if (n == 1) return A[0];   //当n=1时，平均值即为A[0]的值
    else ave = ((n - 1)*Average(A, n - 1) + A(n - 1)) / n; //递归
    return ave; //n个数的平均值=((前n-1个数的平均值)×(n-1)+(第n个数的值))÷n
}
```

3.22 循环队列（插入/删除 操作）

```
//注：rear队尾指针                    length队长
//注：elements[]存放队列的数组        maxSize最大可容纳元素个数

//插入
template class<T>
bool EnQueue<T>::EnQueue(T x) {
```

```
    if (ifFull == 1) return false;   //判断是否队满
    length++;   //插入元素，队长加1
    rear = (rear + 1) % maxSize;  //队尾插入，队尾位置后移一位
    elements[rear] = x;   //插入x
    return true;
}

//删除
template class<T>
bool DeQueue<T>::DeQueue(T &x) {
    if (isEmpty == 0) return false;  //判断是否队空
    length--;    //删除元素，队长减1
    x = elements((rear - length + maxSize) % maxSize);  //从队头删除
    return true;
}
```

3.23 带标志tag的循环队列（插入/删除 操作）

```
//注：rear队尾指针    front队头指针    tag队满标志
//注：elements[]存放队列的数组    maxSize最大可容纳元素个数

//插入
template class<T>
bool EnQueue<T>::EnQueue(T x) {
    if (ifFull == 1) return false;   //判断是否队满
    rear = (rear + 1) % maxSize;   ////队尾插入，队尾位置后移一位
    elements[rear] = x;    //插入x
    tag = 1;     //tag置1，表非空
    return true;
}

//删除
template class<T>
bool DeQueue<T>::DeQueue(T &x) {
    if (isEmpty == 0) return false;  //判断是否队空
    front = (front + 1) % maxSize;   //从队头删除，队头后移1位
    tag = 0;     //tag置0，表未满
    x = elements[front];   //返回队头元素
    return true;
}
```