

41#244

也是暴力模拟即可。后缀算术表达式，如样例中的“1 2+8 2-7 4-/*=”=”“(1+2) * ((8-2) / (7-4))=”答案
为6。规则是a b op，a和b是两个数字，op是计算符号，根据规则就可以还原算式计算。

42#245

模拟序列判断是否合法即可。

43#246

循环链表需要注意在原来普通链表的末尾加入指向链表首元的指针。

44#247

如题意，操作的时候增加对标志的判定再正常操作和输出即可。

45#248

循环队列和循环链表类似，用front和rear标记前后元素，需要注意的就是在末尾元素加入和首元的连接
再正常操作。

46#249

Ackermann函数定义如下：

$$A(m, n) = \begin{cases} n+1 & \text{si } m = 0 \\ A(m-1, 1) & \text{si } m > 0 \text{ et } n = 0 \\ A(m-1, A(m, n-1)) & \text{sinon} \end{cases}$$

若m=0，返回n+1。

若m>0且n=0，返回Ackermann(m-1,1)。

若m>0且n>0，返回Ackermann(m-1,Ackermann(m,n-1))

按照此规则书写递归函数即可。

47#250

加入数组A[m][n]对Ackermann函数进行记忆化，对每组Ack(m,n)，直接输出在数组中对应的值即可。

48#251

在Max_L函数中递归调用自己即可。

49#252

同上，递归调用函数直到指针指向NULL返回。

50#253

同样是基础操作，只是将循环改成了递归遍历。