作业1 　　　　　　朱首赫 　　　　　　　　Date　　/　　/

P8　1.8 (1) $n-1$

(2) $n-1$

(3) $n-1$

(4) $n+(n-1)+\cdots+2+1 = \dfrac{(n+1)n}{2}$

(5) $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{i} j = \sum_{i=1}^{n}\dfrac{(1+i)i}{2} = \dfrac{n^3+3n^2+2n}{6}$

(6) n

(7) $\lfloor \sqrt{n}-1 \rfloor +1 = \lfloor \sqrt{n} \rfloor$

(8) 　$x: 91 \to 101$　10次

　　　$101 \to 91$, $y--$　1次

　　　$11 \times 100 = 1100$

1.9　$2 \cdot 2^{count} < n/2$

$count +1 < \log_2 n -1$

$\Rightarrow count = \lfloor \log_2 n \rfloor -2 = \log_2 n -2$ （由于n为2的来幂）
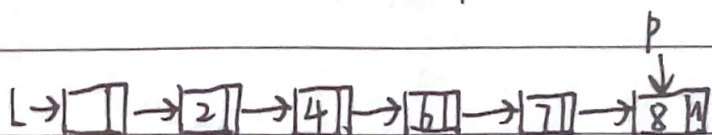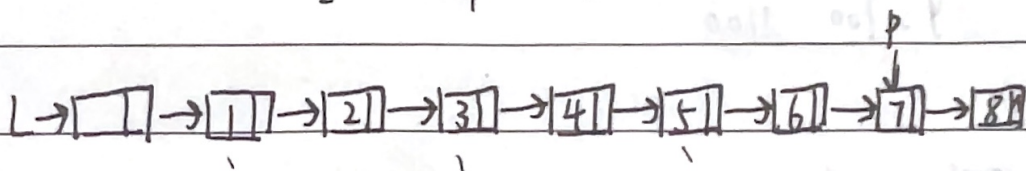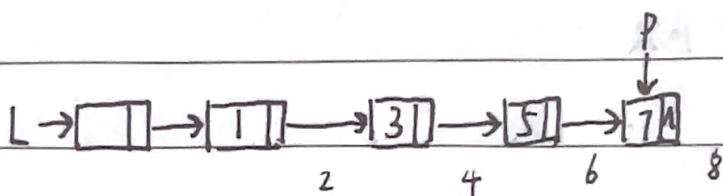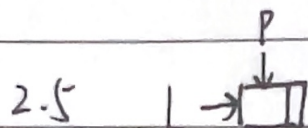
$T(n) = O(\log_2 n)$

1.12　(1) 是　　　(4) 是

(2) 是　　　(5) 否

(3) 否

分别执行

P13　2.4 (1) L→2→5→7→3→8→...
　　　　　　　　　↑P　↑Q　↑R　↑S

(2) 2→5→4→3→8→...　|　(4) L→2→5→7→7→8
　　↑P　↑Q　↑R　↑S　|　　　　↑P　↑Q　↑R　↑S

(3) L→2→5→7→5→8　|　(5) L→2→5→7→3→5
　　　↑P　↑Q　↑R　↑S　|　　　↑P　↑Q　↑R　↑S

(6) L →[2]→[10]→[14]→[6]→[6]→ ...

    P   Q   R   S

(7) L →[2]→[10]→[14]→[13]→[6]→ ...

    P   Q   R   S

2.5   L →[|]   P

L→[|]→[1]→[3]→[5]→[7]   P
      2   4   6   8

L→[|]→[1]→[2]→[3]→[4]→[5]→[6]→[7]→[8]   P

L→[|]→[2]→[4]→[6]→[7]→[8]   P
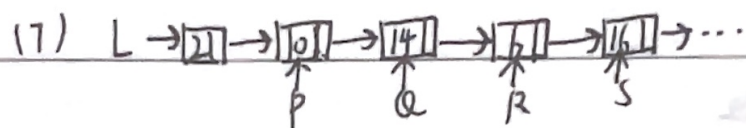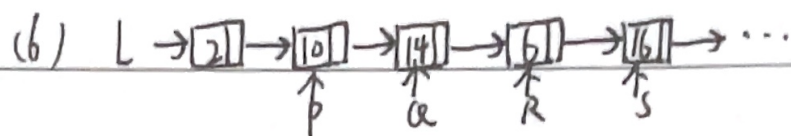
2.9 (1) 如果L的长度大于等于2，将L的首结点变成尾结点.

    (2) 将单循环链表从给点的结点处斩开成两段,并分别连接成两个单循环链表

3.3   stack

3.7 序号 OPTR栈 OPND栈 输入 主要操作

序号	OPTR栈	OPND栈	输入	主要操作
1	#		A-BxC/D+E↑F	Push(OPND,'A')
2	#	A	-BxC/D+E↑F	Push(OPTR,'-')
3	#-	A	BxC/D+E↑F	Push(OPND,'B')
4	#-	AB	xC/D+E↑F	Push(OPTR,'x')
5	#-x	AB	C/D+E↑F	Push(OPND,'C')

6.	#-x	ABC	∠D+E↑F#	operate(B,'*',C)
7.	#-	A B·C	∠D+E↑F#	Push(OPTR,'/')
8.	#-/	A B·C	D+E↑F#	Push(OPND,'D')
9.	#-/	A B·C D	+E↑F#	operate(B·C,'/',D)
10.	#-	A B·C/D	+E↑F#	operate(A,'-',B·C/D) (设差为G)
11.	#	G	+E↑F#	Push(OPTR,'+')
12.	#+	G	E↑F#	Push(OPND,'E')
13.	#+	GE	↑F#	Push(OPTR,'↑')
14.	#+↑	GE	F#	Push(OPND,'F')
15.	#+↑	GEF	#	operate(E,'↑',F) (设H=E^F)
16.	#+	GH	#	operate(G,'+',E) (设I=G+H)
17.	#	I	#	RETURN

3.10 第一个 0 前的数 倒序累加，并边加 边输出中间结果

```
void test(int &sum){                  | sum += x;
    Stack s;                          |     cout << sum; }
    InitStack(s);                     | DestoryStack(s);
    int x; cin>>x;                    | }
    while(x){
        Pust(s,x);
        cin>>x; }
    do{ cin>>x;
        Push(s,x);
    } while(x);
    while(!StackEmpty(s)){
        Pop(s, &x);
```