

1、写出下面程序段的时间复杂度，用大 O 表示。

```
Sum = 0;
For (i=0; i<n; i++)
    For (j = i; j < n * n; j++)
        Sum++;
```

2. 设有 a、b、c、d、e 五个元素依次进栈（进栈后可立刻出栈），问能否得到下列的出栈序列。  
若能得到，则写出相应的 push 和 pop 序列；若不能，说明理由。

a、b、c、d、e

b、c、a、d、e

a、b、e、c、d

3、循环队列实现时，通常 front 指向队头元素前的一个位置，而且该位置永远不能存放数据，rear 指向队尾元素。假如换一种做法，让 front 指向队头元素，rear 指向队尾元素后的一个位置，并且 rear 指向的位置不能存放数据。如果存放队列的数组名为 arr，规模为 size，试写出入队(void enqueue(T x))、出队(T dequeue())、判队空(bool isEmpty())的函数。

4、以下是带头尾结点的双链表类的定义。在双链表类中重载一个减法函数。减法函数的结果是在第一个双链表中删除在第二个双链表中也出现的元素。例如，A 表的内容为 1, 2, 3, 4, 5, B 表的内容为 1, 3, 5, 则 A-B 的结果为 2, 4。

```
template <class elemType>
class linkList: public list<elemType> {
    friend linkList< elemType> operator-(const linkList< elemType> &,
                                          const linkList< elemType> &);

private:
    struct node { //双链表中的结点类
        elemType data;
        node *prev, *next;
        node(const elemType &x, node *p = NULL, node *n = NULL)
            { data = x; next = n; prev = p;}
        node():next(NULL), prev(NULL) {}
        ~node() {}
    };

    node *head, *tail; //头尾指针

public:
    .....
};
```