用两个栈实现队列

用两个栈实现一个队列。队列的声明如下，请实现它的两个函数 **appendTail** 和 **deleteHead** ，分别完成在队列尾部插入整数和在队列头部删除整数的功能。(若队列中没有元素，**deleteHead** 操作返回 **-1**)

示例：

输入：

["CQueue","appendTail","deleteHead","deleteHead"]

[[],[3],[],[]]

输出：**[null,null,3,-1]**

```
/*class CQueue {
    stack<int> stack1;
    stack<int> stack2;
public:
    CQueue() {
    }

    void appendTail(int value) {
        stack1.push(value);
    }

    void copy(stack<int> &a,stack<int> &b)
    {
        while(a.size()>0)
        {
            b.push(a.top());
            a.pop();
        }
    }

    int deleteHead() {
        copy(stack1,stack2);
        int res=stack2.top();
        stack2.pop();
        copy(stack2,stack1);//必须复原 维护 下一次尾部插入 不会乱序

        return res;
    }
};*/

class CQueue {
```

```
    stack<int> stack1,stack2;
public:
    CQueue() {
        while (!stack1.empty()) {
            stack1.pop();
        }
        while (!stack2.empty()) {
            stack2.pop();
        }
    }

    void appendTail(int value) {
        stack1.push(value);
    }

    int deleteHead() {
        // 如果第二个栈为空
        if (stack2.empty()) {
            while (!stack1.empty()) {
                stack2.push(stack1.top());
                stack1.pop();
            }
        }
        if (stack2.empty()) {
            return -1;
        } else {
            int deleteItem = stack2.top();
            stack2.pop();
            return deleteItem;
        }
    }
};



/**
 * Your CQueue object will be instantiated and called as such:
 * CQueue* obj = new CQueue();
 * obj->appendTail(value);
 * int param_2 = obj->deleteHead();
 */
```