Class **LinkedList** is used to represent single linked list, described as the following:

class LinkedList {  
    public:   
        class Node;  
    private:  
        Node\* head;  
        Node\* tail;  
        int size;  
    public:   
        class Node {  
            private:   
                int value;  
                Node\* next;  
                friend class LinkedList;  
            public:  
                Node() {  
                    this->next = NULL;  
                }  
                Node(Node\* node) {  
                    this->value = node->value;  
                    this->next = node->next;  
                }  
                Node(int value, Node\* next = NULL) {  
                    this->value = value;  
                    this->next = next;  
                }  
        };  
        LinkedList(): head(NULL), tail(NULL), size(0) {};  
 void partition(int k);  
};

In this class; head, tail and size are the pointers of the first element, the last element and size of linked list.

Request: Implement function

void LinkedList::partition(int k);

To rearrange elements in this linked list by this way: split all elements of this linked list to 3 groups include: group I (the value  of each element is smaller than k), group II (the value of each element is equal to k) and group III (the value of each element is larger than k); after that, move group I to the head  of linked list, group II is after group I and group III is after group II. Note that the orders of the elements of each group are unchanged.

Example:

- The given linked list: 10->55->45->42->50

- k=45

In this case, group I includes elements with value 10 and 42, group II includes elements with 45 and group III includes elements with value 55 and 50. Rearrange by above way,  this linked list will become 10->42->45->55->50 (the orders of th elements in each group is unchanged: 10 before 42 and 55 before 50).

*Note: In this exercise, libraries iostream, sstream and using namespace std; have been used. You can add other functions for your answer, but you are not allowed to add other libraries.*

**For example:**

| **Test** | **Result** |
| --- | --- |
| LinkedList\* l1 = new LinkedList();  l1->add(20); l1->add(30); l1->add(10); l1->add(60); l1->add(40); l1->add(45); l1->add(55);  l1->partition(45);  cout << l1->toString() << "\n"; | [20,30,10,40,45,60,55] |