```
#include <iostream>
using namespace std;

class Node
{
public:
	Node()
	{
		info = -1;
		next = NULL;
	}

	Node(int f)
	{
		info = f;
		next = NULL;
	}

	int getInfo() { return info; }
	Node * getNext() { return next; }
	void setInfo(int f) { info = f; }
	void setNext(Node * n) { next = n; }
	void print() { cout << info << "->"; }

private:
	int info;
	Node * next;
};

class List
{
public:
	List() { first = NULL; }

	Node * getFirst()
	{
		return first;
	}
	//Check if this list is empty
	bool isEmpty()
	{
		return first == NULL;
	}

	//Print the info of all nodes in this list starting Node * p
	void print(Node * p)
	{
		//TO BE COMPLETED
		if (p)
		{
			print(p->getNext());
			p->print();
		}
	}

	void deleteNode(Node *& nd, int s)
	{
		if (s == nd->getInfo())
		{
			Node * p = nd;
			nd = nd->getNext();
			delete p;
		}
		else
		{
			Node * p  = nd->getNext();
			deleteNode(p, s);
			nd->setNext(p);
		}
	}

	void insert(Node * & start, Node * nd)
	{
		if (start == NULL)
		{
			start = nd;
		}
		else
		{
			Node * p = start->getNext();
			insert(p, nd);
			start->setNext(p);
		}
	}

private:
	Node * first;   // The pointer to the first node in the list
};

int main()
{
	List lst;
	Node * startNode = lst.getFirst();

	//TO BE COMPLETED
	//Create 10 nodes and add them to the list lst using a loop

	for (int i = 0; i < 10; i++)
	{
		Node * nd = new Node(i+1);
		lst.insert(startNode, nd);
	}
	lst.print(startNode);
	lst.deleteNode(startNode, 1);
	lst.deleteNode(startNode, 10);
	lst.deleteNode(startNode, 5);
	lst.print(startNode);

	return 0;
}
```