template<class T>
class LN {
    public:
        LN (const LN<T>& ln) : value(ln->value), next(ln->next) {}  
        LN (T v, LN<T>* n = nullptr) : value(v), next(n) {}  
        T   value;  
        LN<T>* next;  
};

LN<int>* x

[diagram of linked list with values 2, 4, 3, 5, 3, 8]
int sum = 0;
for (LN<int>* c = x; c!=nullptr; c=c->next)
    sum += c->value;
std::cout << "Sum = " << sum << std::endl;
To **remove** the node whose value is 7, we must change next in the node preceding it (whose value is 2). We will discuss look-ahead (shown below) and ghost pointers.