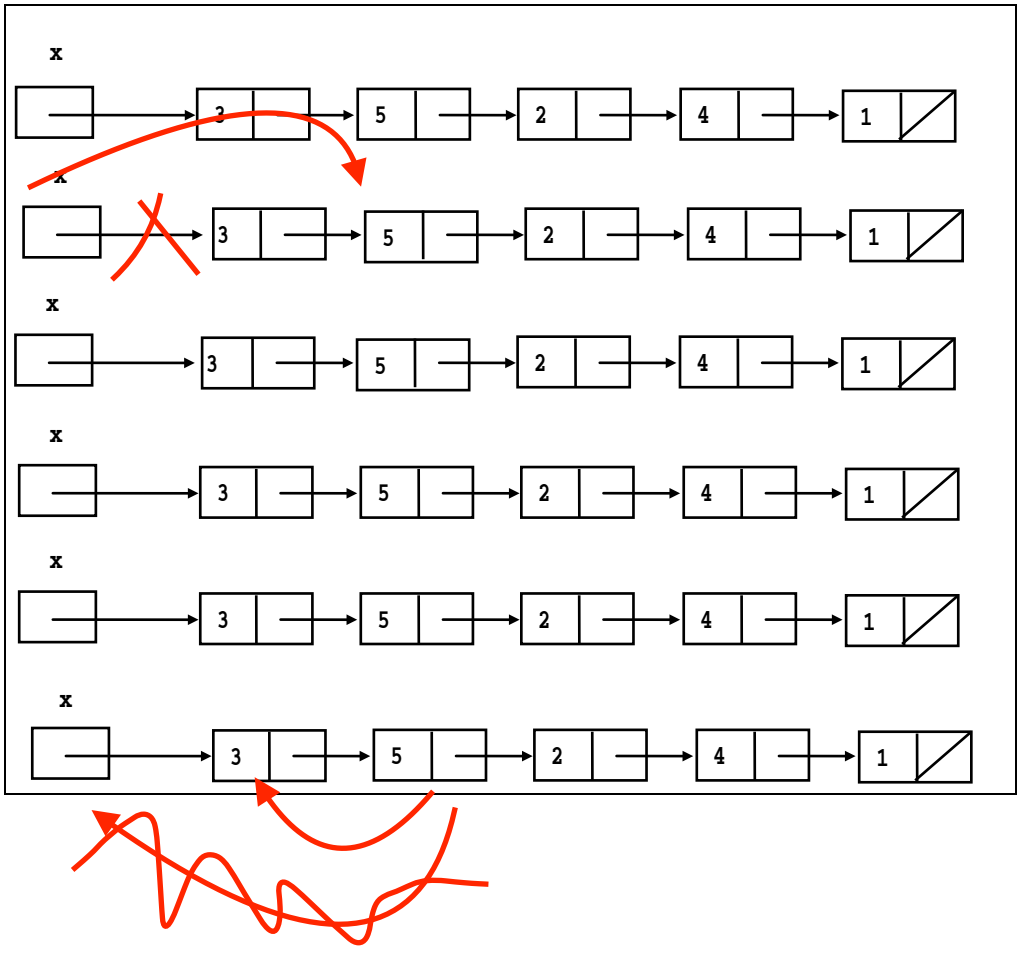
```
template<class T>
class LN {
  public:
    LN ()                                  : next(nullptr){}
    LN (const LN<T>& ln)                   : value(ln.value), next(ln.next){}
    LN (const T& v, LN<T>* n = nullptr)    : value(v), next(n){}

    T      value;
    LN<T>* next;
};
```

Each linked lists is created from the class **LN<int>**. Starting with each list as shown, indicate what state(s) change(s) when the statement to its left is executed. **Cross out** any values that are replaced and **Write in** new boxes or values (text or arrows).

```
X = new LN<int>
     (7,x->next->next);

x = x->next;

x->next->value = x->value;

x->next = x->next->next;

x->next->next = x->next;

x->next->next = x;
```



Assume **LN<int>* x** and **x** points to a linked list with 4 or 5 values. What changes are made when executing the following code?

```
LN<int>* answer = nullptr;
while (x != nullptr) {
    LN<int>* to_move  = x;
    x                 = x->next;
    to_move->next     = answer;
    answer            = to_move;
}

x = answer;
```