

COP 3502 9/15/25

① Insert in order in a linked list ✓

② Deleting from a linked list ✓

③ Reversing a L.L.

④ Exercises

- front to back

- insert -1 after each even term

→ X empty to insert into
→ good to check if NULL to avoid NULL ptr error.

```
node* insertInOrder(node* list, int val) {
```

```
    if (list == NULL)
```

```
        return createNode(val);
```

```
    if (val <= list->data) {
```

```
        node* newfront = createNode(val);
```

```
        newfront->next = list;
```

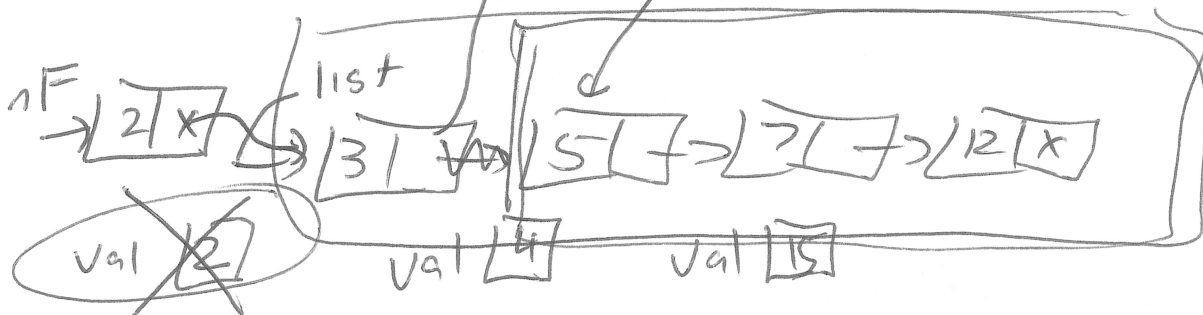
```
        return newfront;
```

```
    }
```

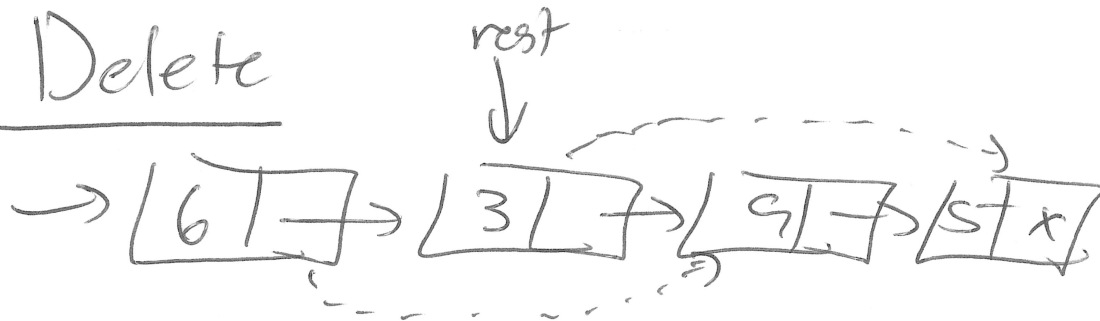
```
    node* restList = insertInOrder(list->next, val);
```

```
    list->next = restList;
```

```
    restList->next = list; return list;
```



Delete



delnum [6]
delnum [9]
delnum [2]

1) LIST NULL RET NULL

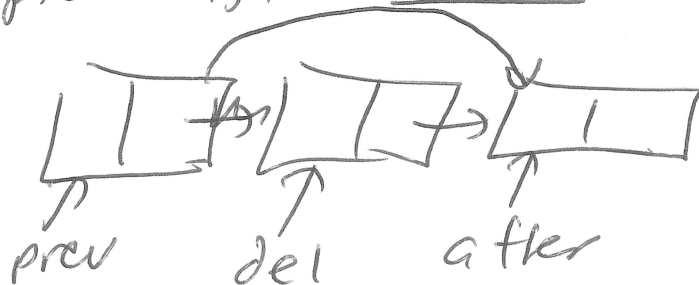
2) DEL 1st item

— Save ptr 2nd item ←

— free memory 1st node

— Return saved ptr

3) Save ptr right BEFORE deleted node

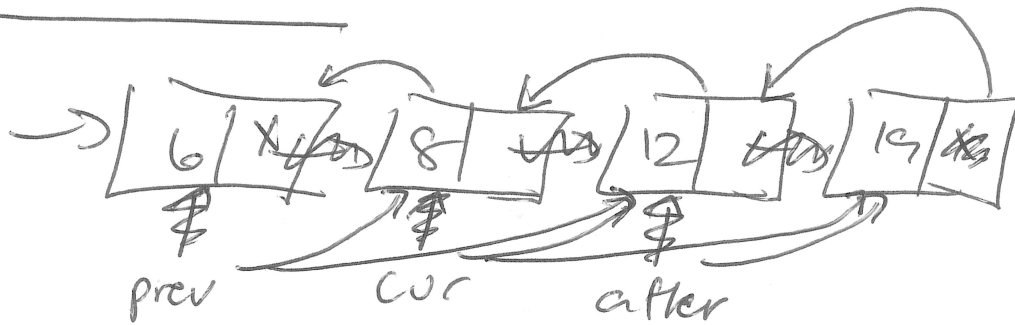


(1) patch prev to after

(2) free (del)

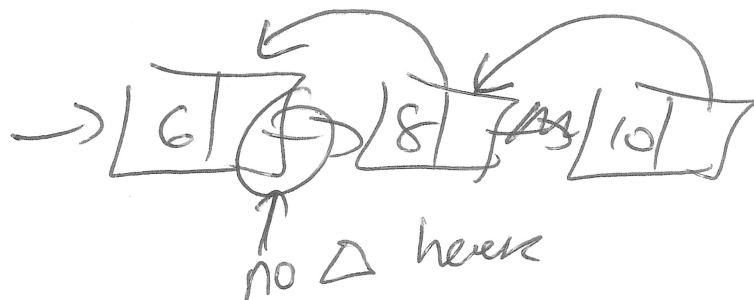
(3) return orig front

Reverse



$$\begin{array}{l} \text{prev} = \text{cur}; \\ \text{cur} = \text{after}; \end{array} \quad \left. \vphantom{\begin{array}{l} \text{prev} = \text{cur}; \\ \text{cur} = \text{after}; \end{array}} \right\} \text{advancing}$$

1 node Special (no work)



ERROR