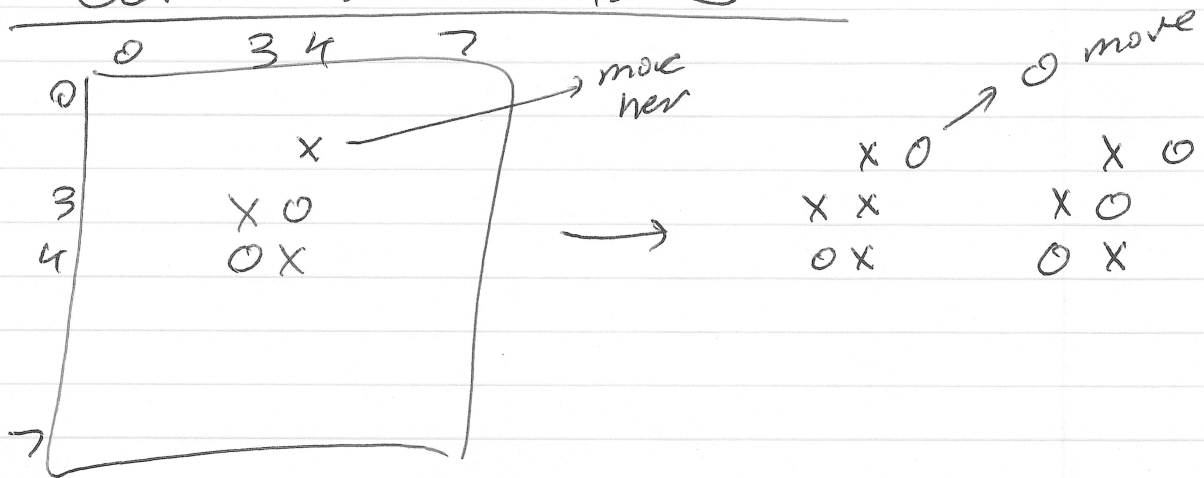


COP 3502 2/15/2023

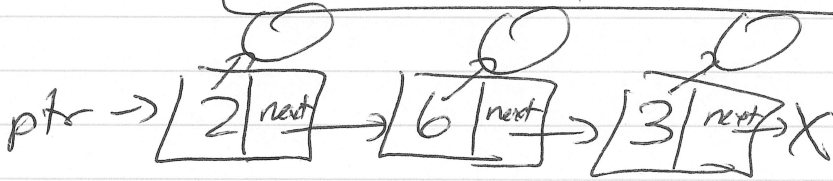


## Linked Lists

array → [ | | | | | ]

run out of room ⇒ realloc

array → [ | | | | | ] (double)



struct node {

// Data

int data; // struct student\* sPtr;

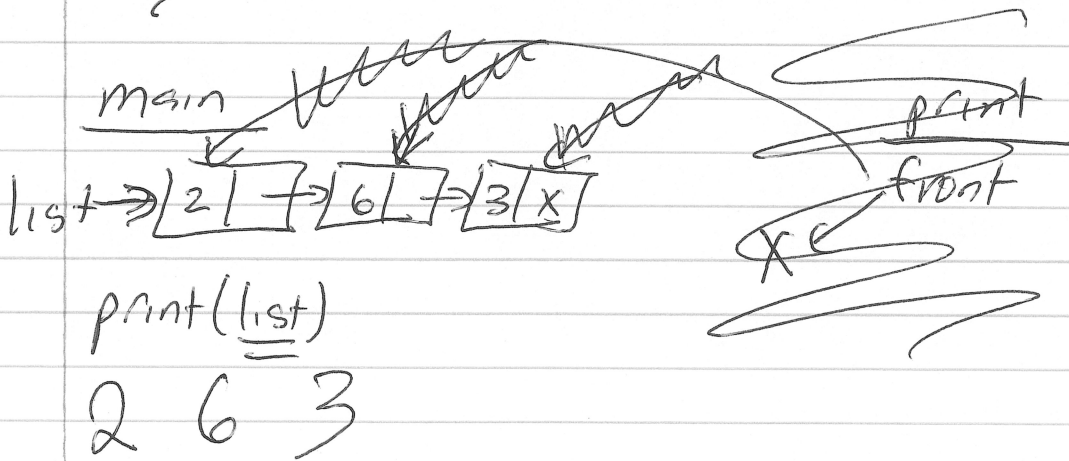
struct node\* next;

}

What we need to learn how to do

- ① Print out each item in a list that exists.
- ② Any sort of filtering of items in a list.
- ③ Add to Front
- ④ Add something "in order"
- ⑤ Deleting item
- ⑥ Other modifications

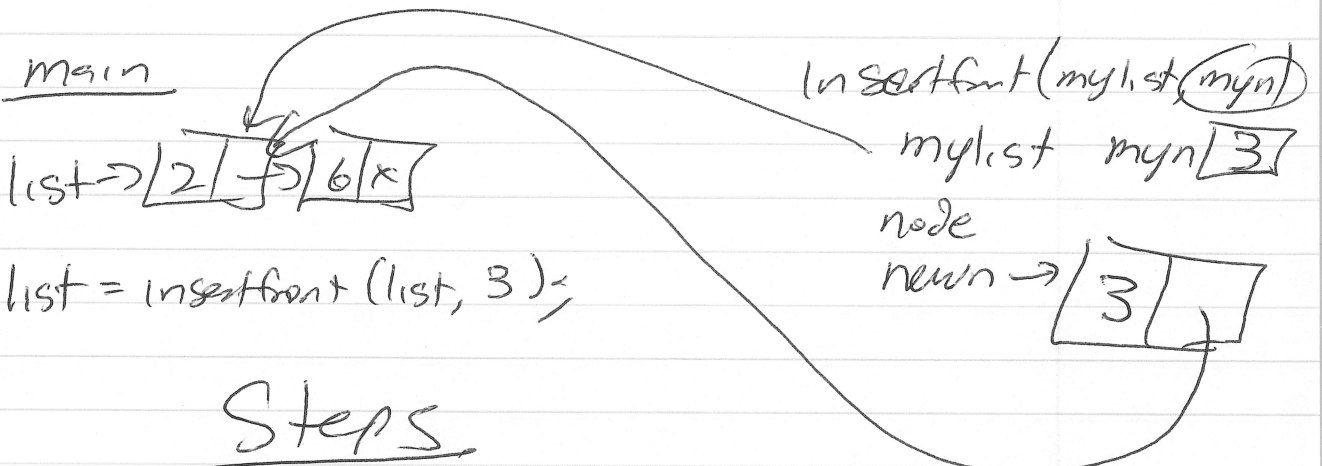
```
void print(node* front) {  
    while (front != NULL) {  
        printf("%d", front->data);  
        front = front->next; // i++  
    }  
}
```



```
int inorder(node* front) {  
    if (front == NULL) return 1;  
    while (front->next != NULL) {  
        if (front->data > front->next->data)  
            return 0;  
    }  
    return 1;  
}
```

# Insert Front

insertfront

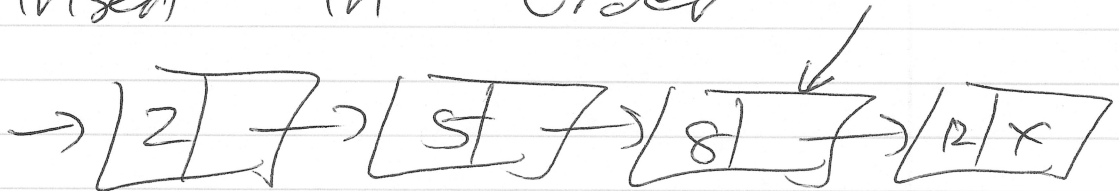


## Steps

- ① made newnode return newnode
- ② list newnode to existing list
- ③ return ptr newnode

---

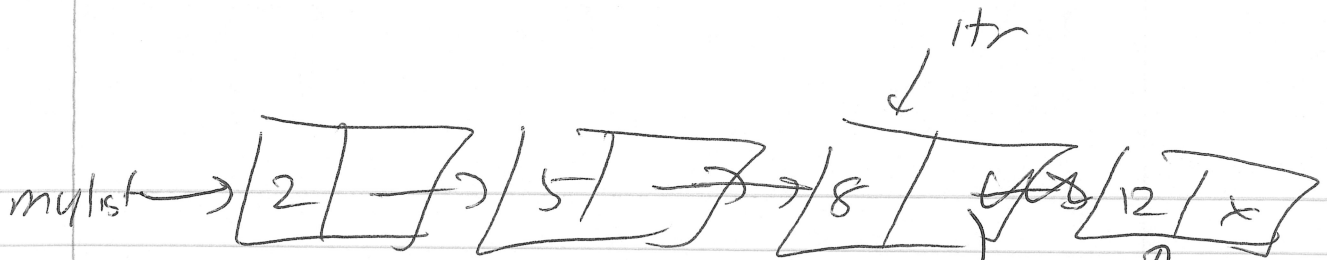
## Insert In Order



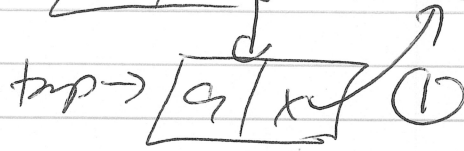
Cases: 1  
9  
15

if list is NULL OR my # is less 1st # ⇒ insert to front

Iterate to one node BEFORE you want to insert



(1) tmp → next = itr → next;

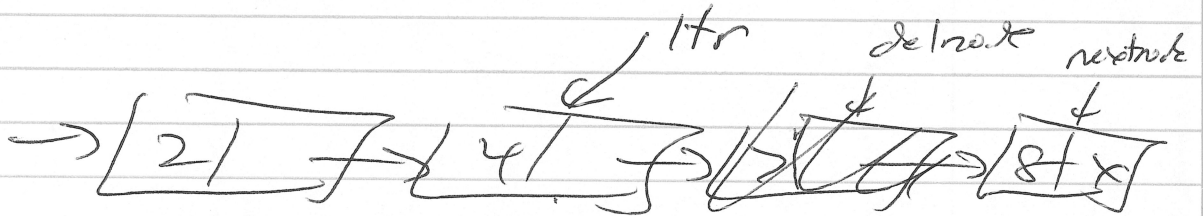


(2) itr → next = tmp;

## Delete

return ptr new front

iterate to node before deleted node



del 7 itr → next = delnode → next; free(delnode)	delnode = itr → next; nextnode = delnode → next; free(delnode) itr → next = nextnode;
--	--