

COP 3502 9/20/21

- ① Reading @ scanf o/o2, o/os, o/of or d/f
- ② OH today → 3-4pm Online (9/20m)
- ③ P2 - $n=5$ Permutation of 10 things 70-75%
 $n=6$ Don't do all $12!$ perm (skip some) 90%
 $n=8$ full credit.
- ④ Posted - P1 test date, strategy

⑤ LINKED LISTS

```
typedef struct node {  
    int data; // PTR struct storing data  
    struct node* next;  
} node;
```



Strategy

- ① malloc 1 node at a time
 - ② place node in an existing list
- USUALLY VIA FUNCTIONS!

Create Node

```
node* createNode(int mydata data) {
```

```
    node* tmp = malloc(sizeof(node));
```

```
    tmp->data = data mydata;
```

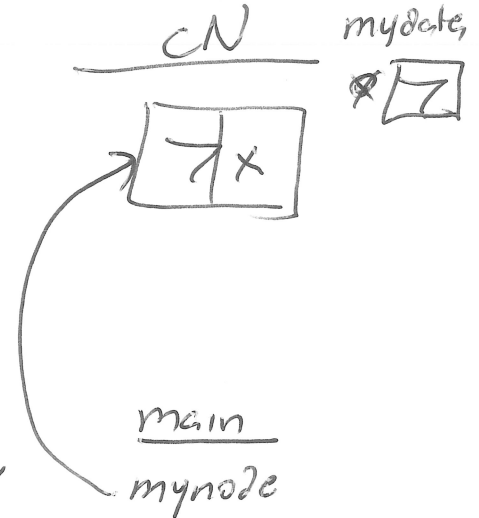
```
    tmp->next = NULL;
```

```
    return tmp;
```

```
}
```

main

```
node* mynode = createNode(7);
```



InsertFront

(1) Create a new node

(2) Link to rest list

(3) Return ptr to new front list

```
node* insertFront(int mydata) {
```

```
    node* mylist, int mydata)
```

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

```
    newfront->next = mylist;
```

```
    return newfront;
```

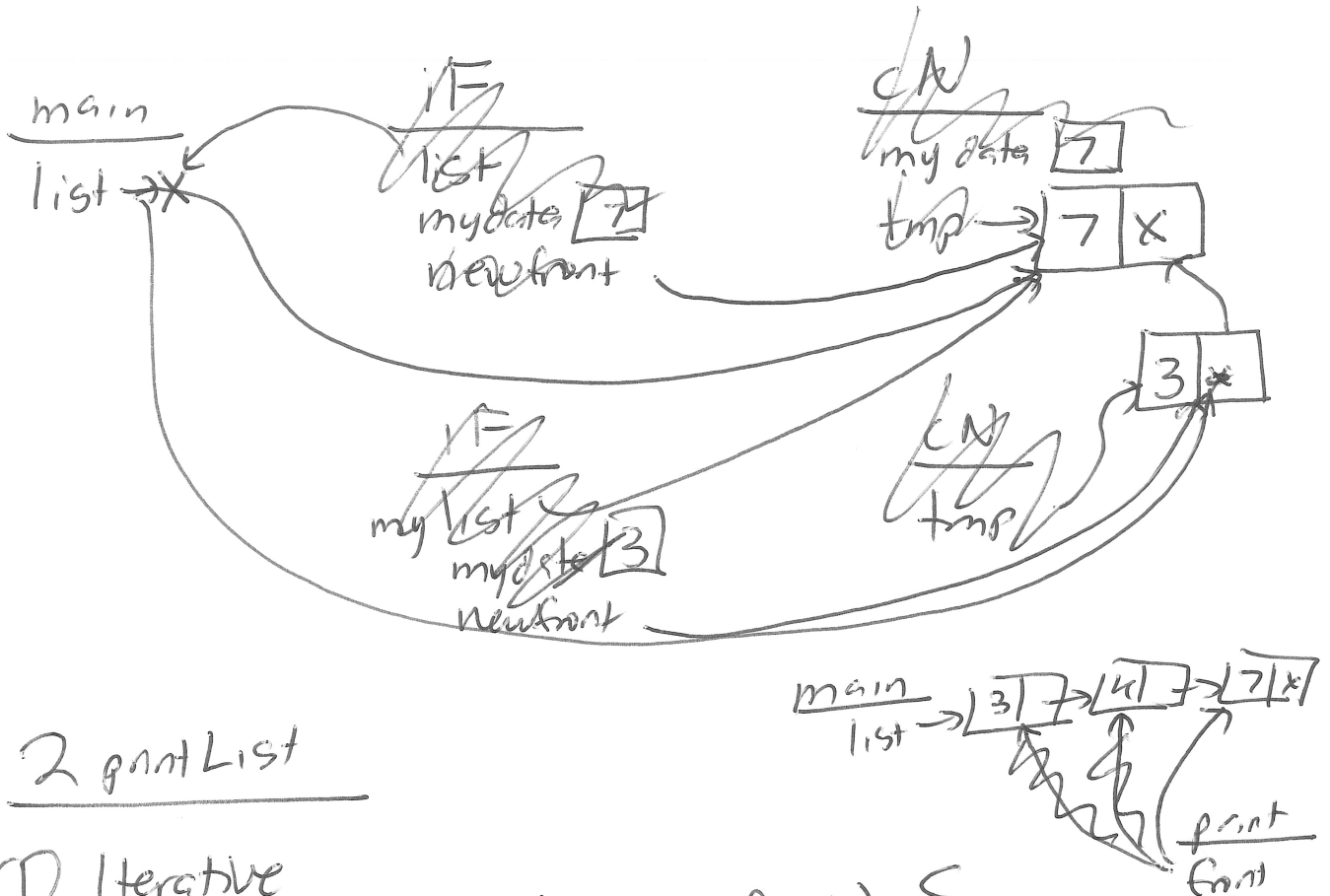
```
}
```

main

node* list = NULL;

list = insertFront(list, 7); // insertFront(list, 7);

list = insertFront(list, 3);



2 printList

① Iterative

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

② Recursive

```
void printRec(node* front) {  
    if (front == NULL) return;  
    printf("%d ", front->data);  
    printRec(front->next);  
}
```