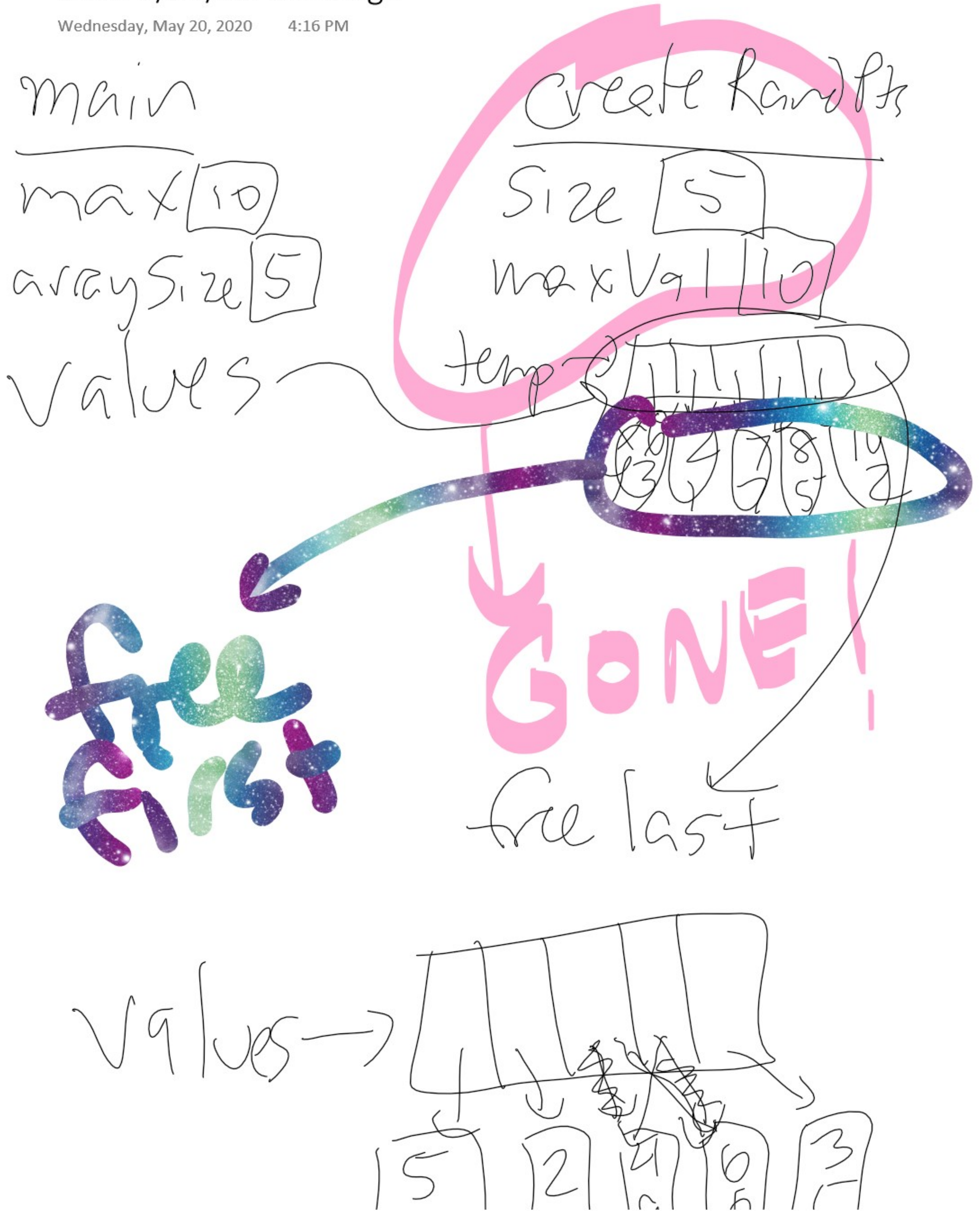
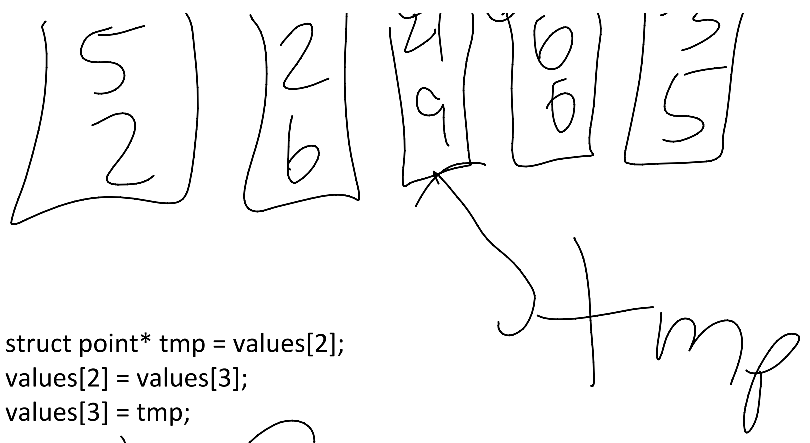
Class 5/20/20 Drawings

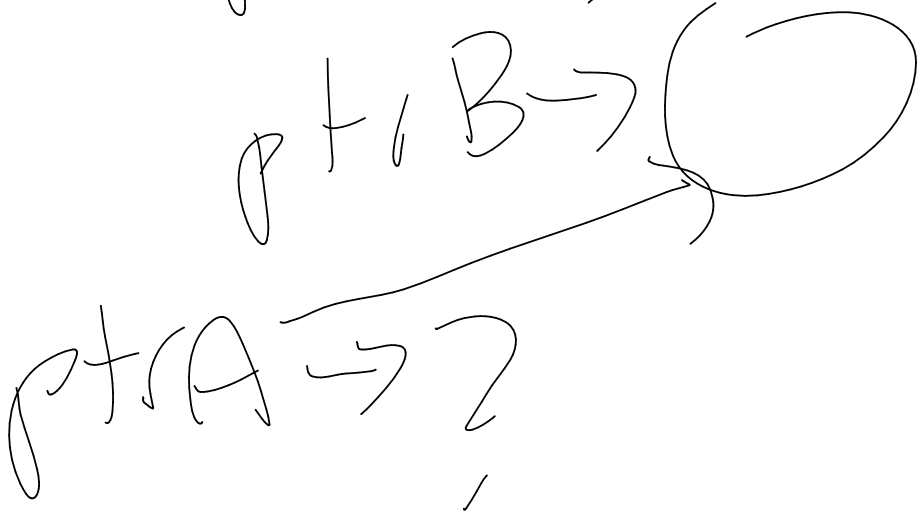
Wednesday, May 20, 2020 4:16 PM



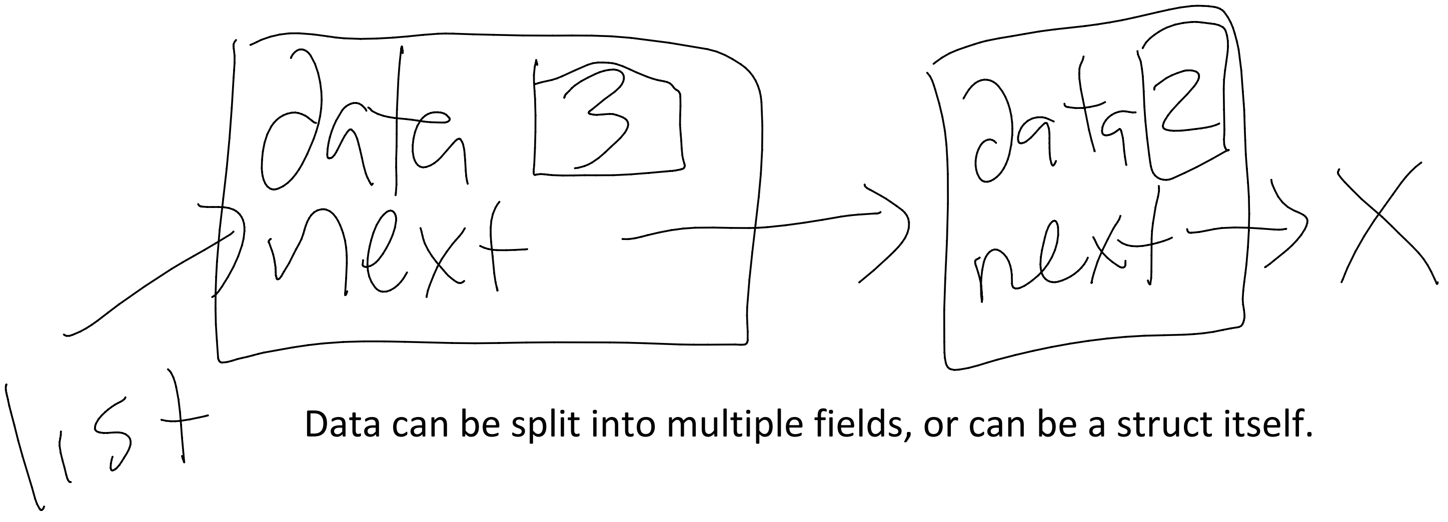


```
struct point* tmp = values[2];
values[2] = values[3];
values[3] = tmp;
```

ptrA = ptrB;



Linked List Idea



Data can be split into multiple fields, or can be a struct itself.

Operations I might need:

1. Insert an item (to front, to back, or somewhere in between)
2. Delete an item (from front, back or arbitrary element)

It is easier to process information about a linked list than to build it, but to test anything, we must build a list!!!

So, here is what we will do:

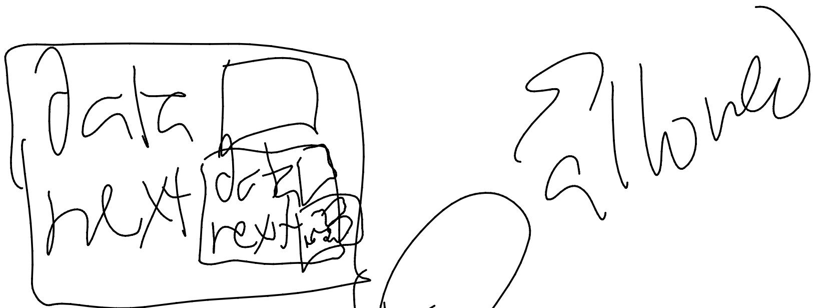
1. Easiest insert function is only inserting to the front, so we will write this first.
2. Then, with the ability to build a list, we will write a few functions that use the list as input, but don't manipulate the list.
3. After that, we will talk about inserting in the middle, back
4. Then delete, which is the "hardest"

First, let's define the struct we will use!!!

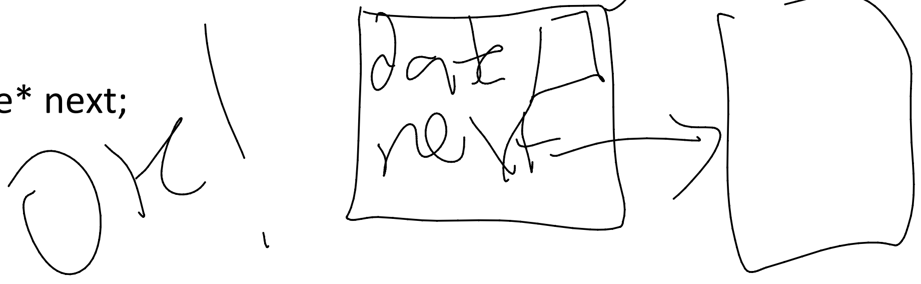
Let's try this:

```
struct node {
    int data;
    struct node next;
};
```

Try to draw this =)
WE would never finish drawing it, it would be infinitely nested!!!



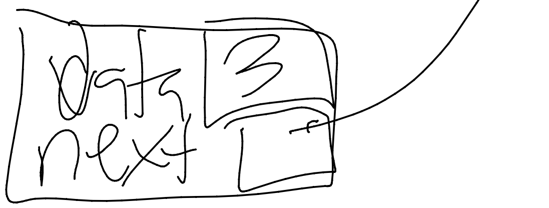
```
struct node {
   int data;
   struct node* next;
};
```



list → X

struct node* list = NULL;
(IMPORTANT!)

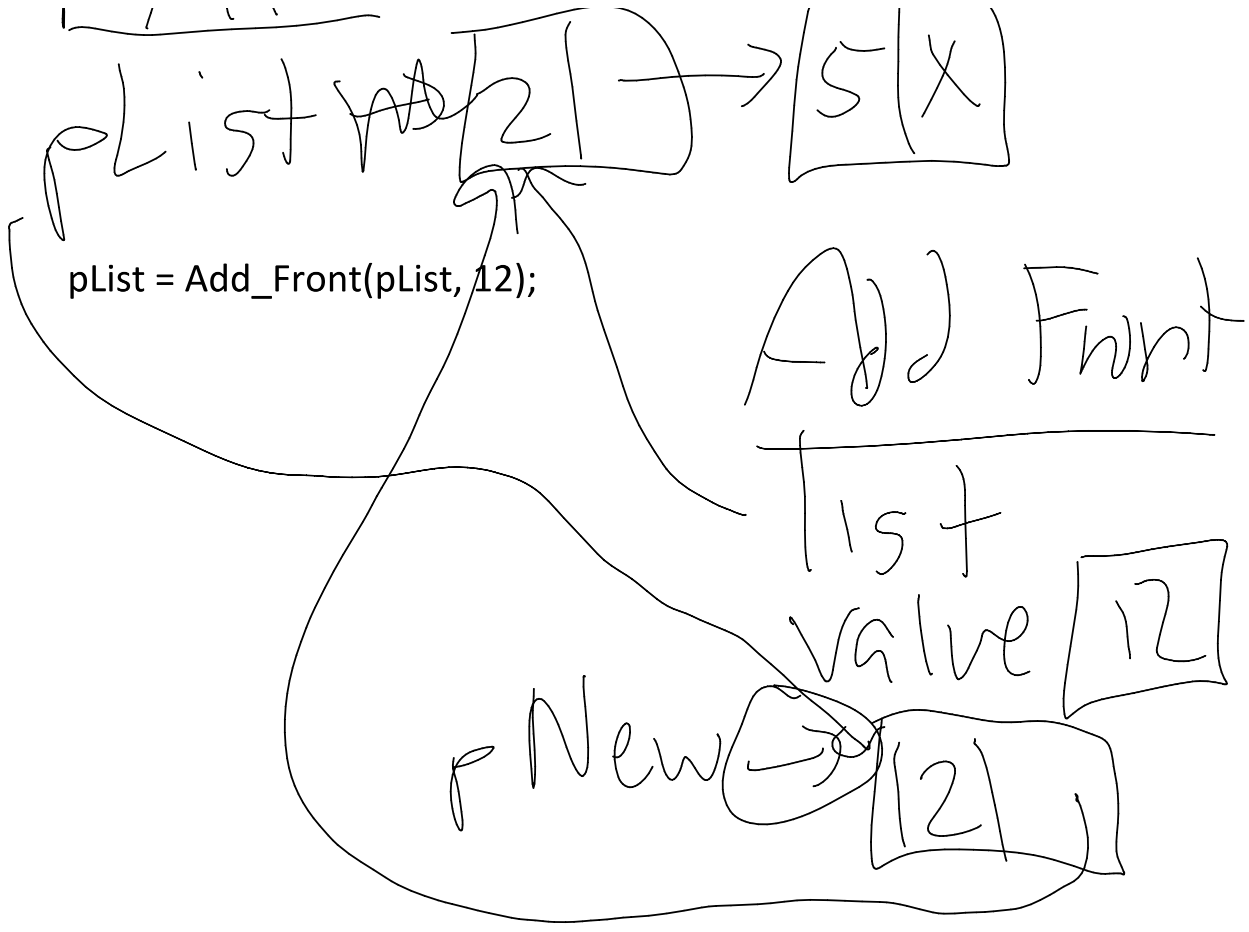
AddToFront
front
value $\boxed{3}$

temp → 

temp is the new front of the list
return temp;

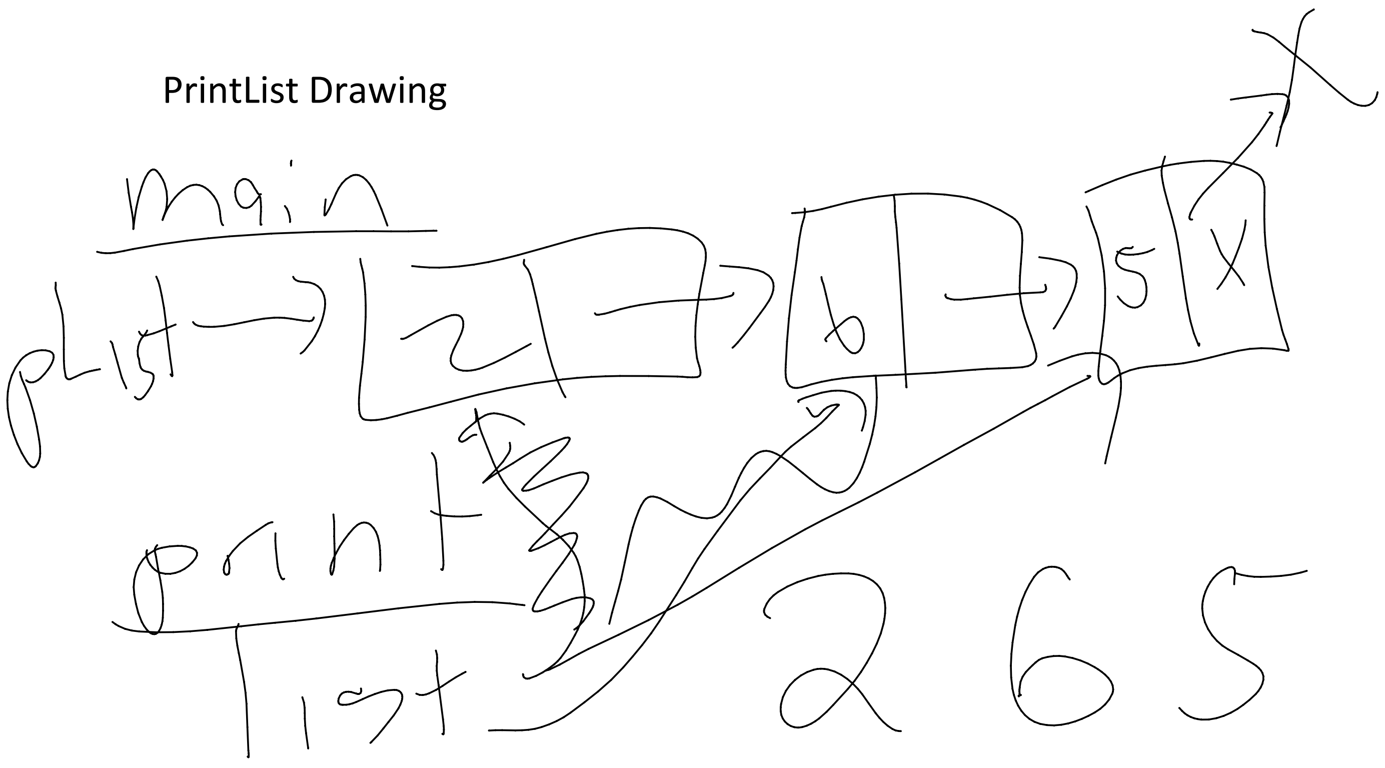
main
data is on the left, next on the right.

list → 5|X

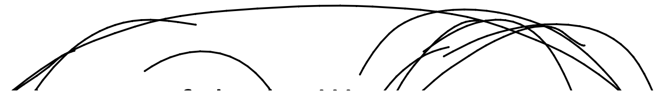


pList = Add_Front(pList, 12);

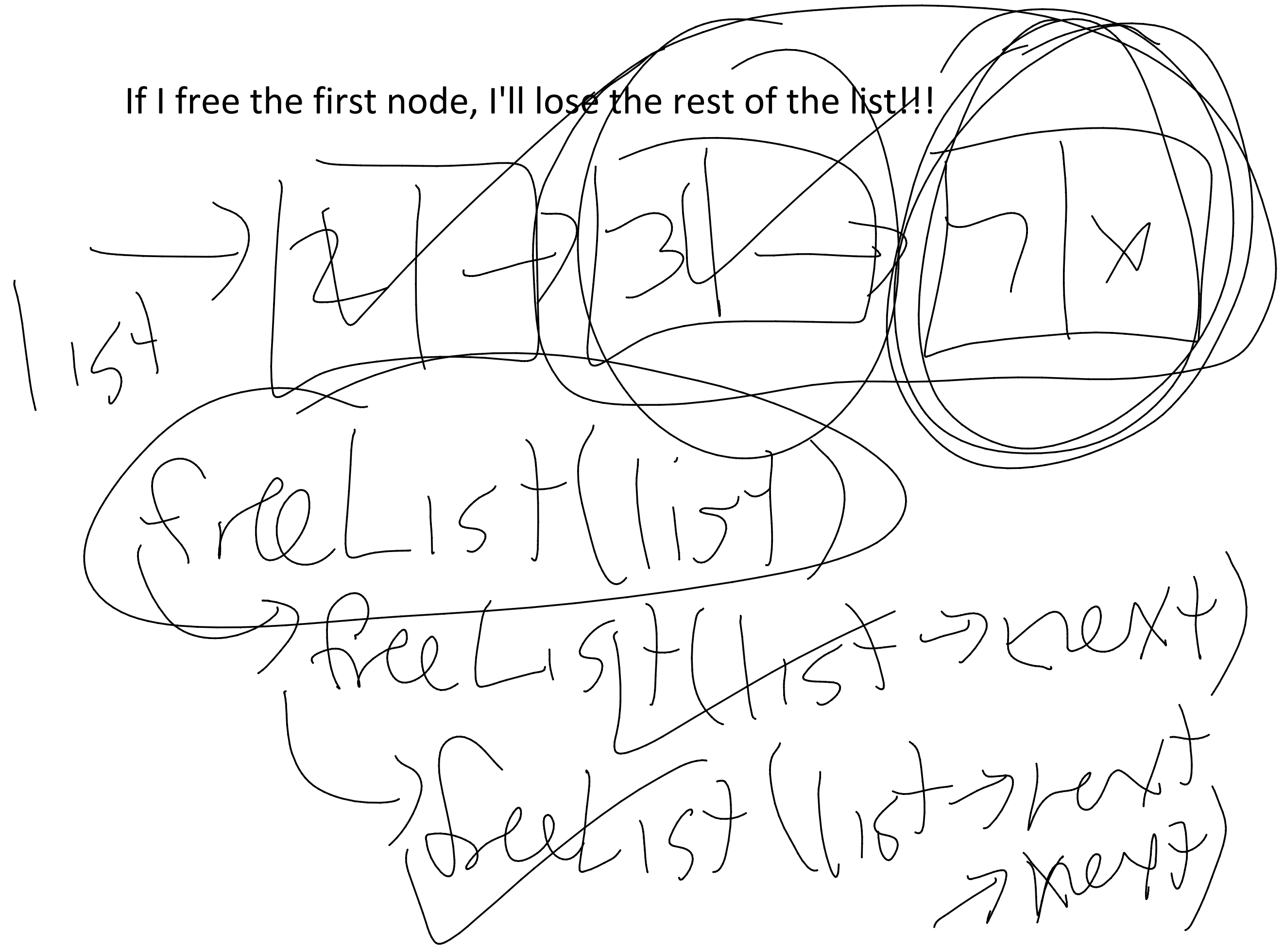
PrintList Drawing



Basically, list = list->next is like i++ for arrays, but for linked lists.

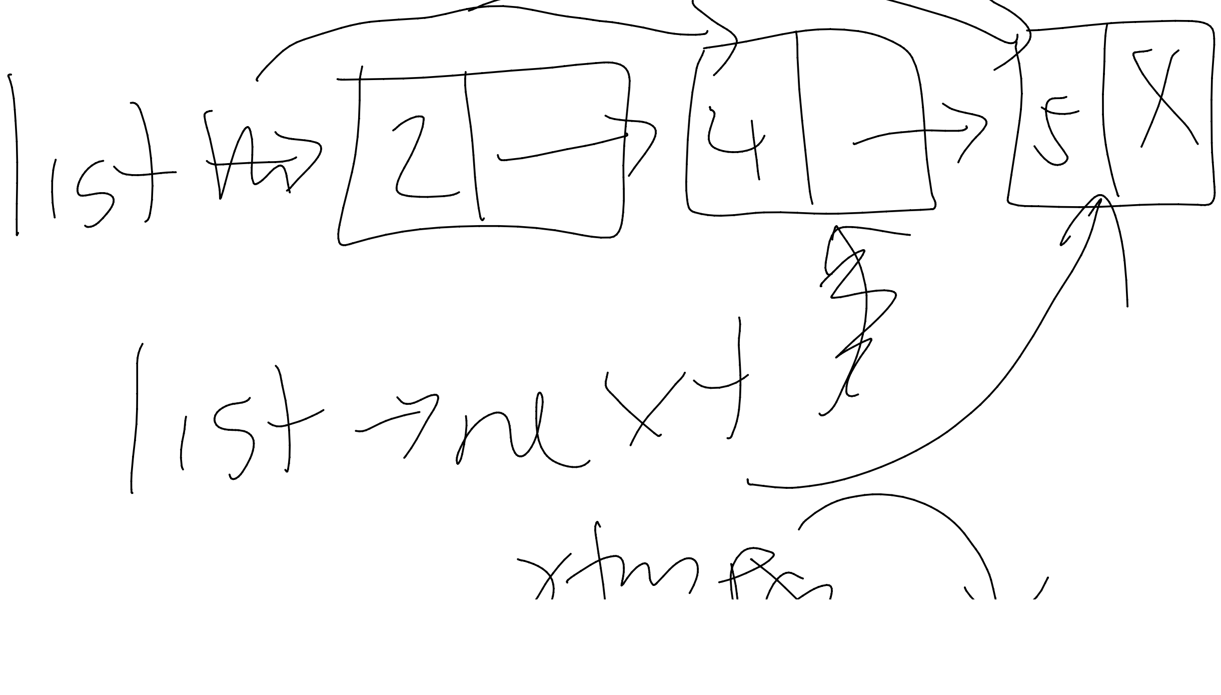


If I free the first node, I'll lose the rest of the list!!!

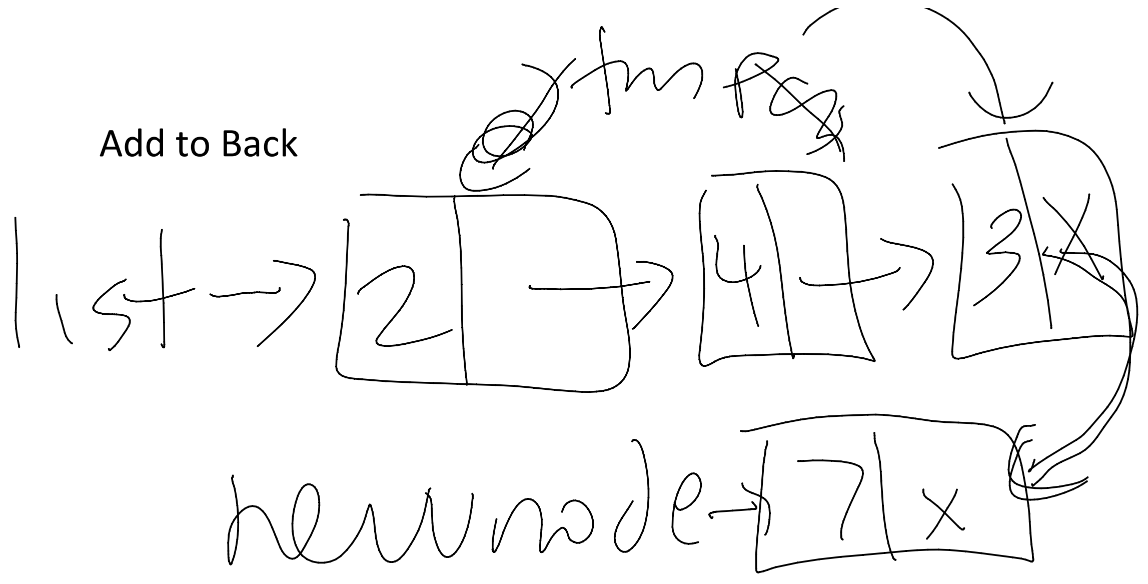


Some Practice Functions:

1. Find length of a linked list.
2. Count the number of times a value is in the list.
3. See if the numbers in the list are already in sorted order.

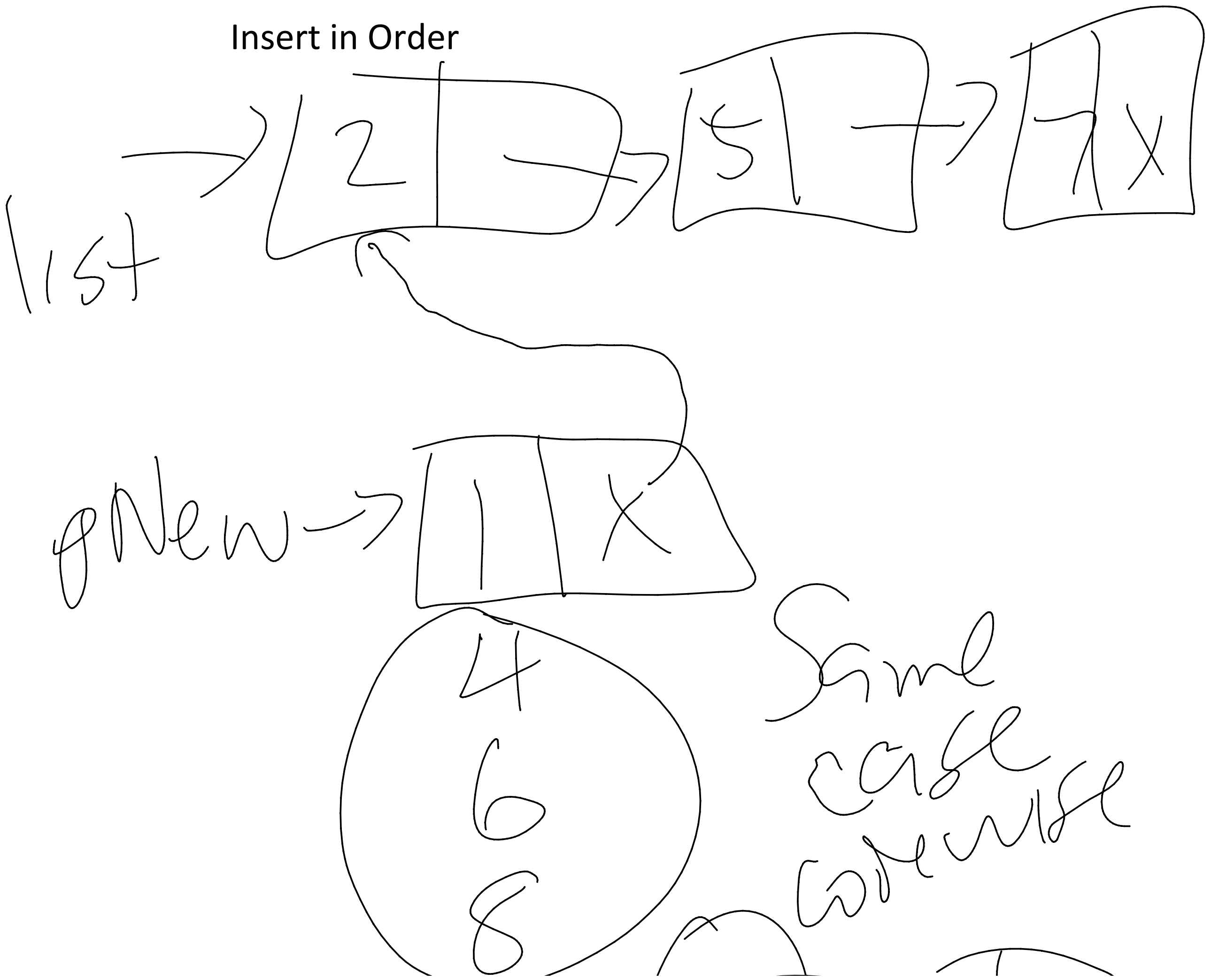


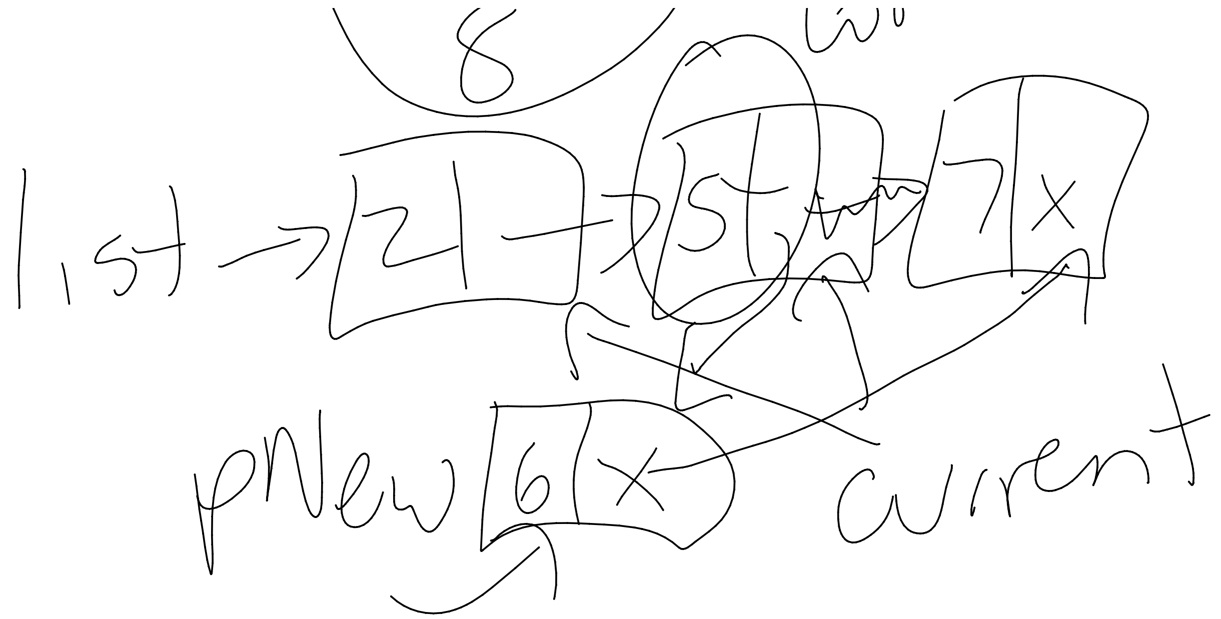
Add to Back



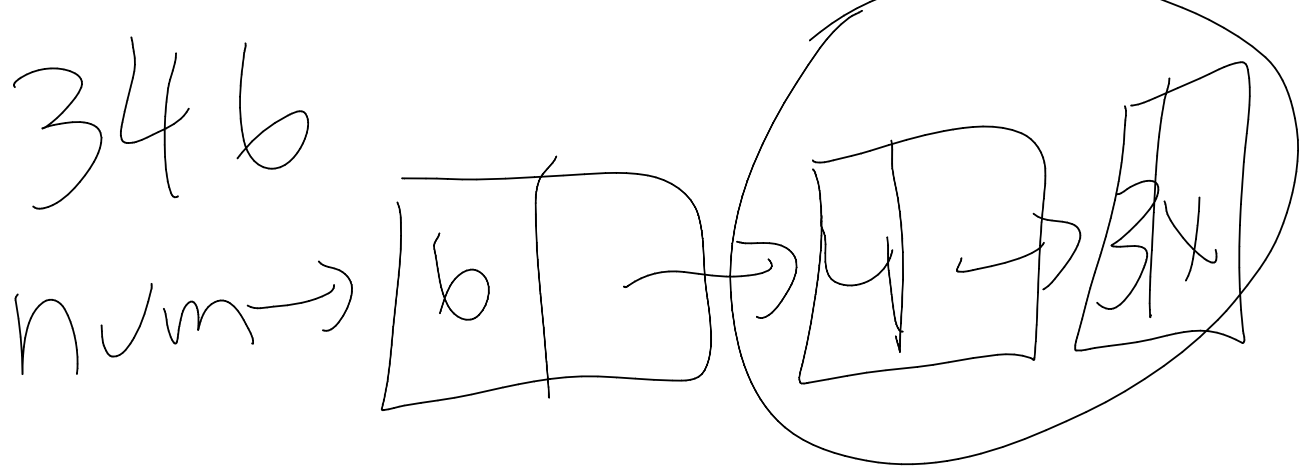
1. Create a tmp ptr.
2. Move it to the last node in list, don't fall off the list.
3. Then connect the next of temp to newnode (tmp->next = newnode;)

Insert in Order





Last Function:
Pretend the linked list is storing a large number, digit by digit.
Write a function that takes in a list and converts it to a number.
First item is one's digit, the second item is the ~~ten's digit~~ and so forth.



curPowTen	1	10	100

TILES: RTEWDTA RTEWDTA

WORD: WATER WORD: DAY

WORD: SODA WORD: ADA

RTEWDTA

WORD: SODA

WORD: ADA

RTEWDTA

RTEWDTA