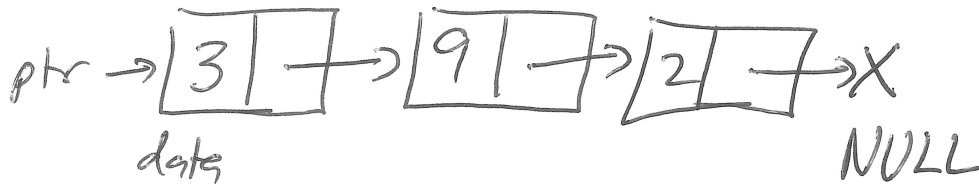


COP 3502 - 2/4/2022 ^{Lists} Linked ~~links~~

To store bunch of values, we have an array

[2|6|3|4|8]

Not great if we resize a lot.



struct node {

int date; // ptr to larger struct

struct node next;

};

BAD → ~~[int date;
struct node next;]~~ Infinite Nesting!

~~[date
[]]~~ oops!

- ① Go through a list (iterate)
- ② Insert an item
- ③ Delete an item
- ④ Search for an item

TODAY'S APPLICATION

BUILT MY OWN STRINGS!

head $\rightarrow$ ['c'] $\rightarrow$ ['a'] $\rightarrow$ ['t'] $\rightarrow$ X

✓ strlen

✓ strcpy

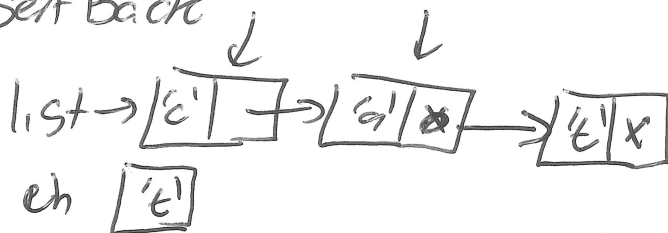
✓ strcmp

✓ (2) makestr

✓ (3) print

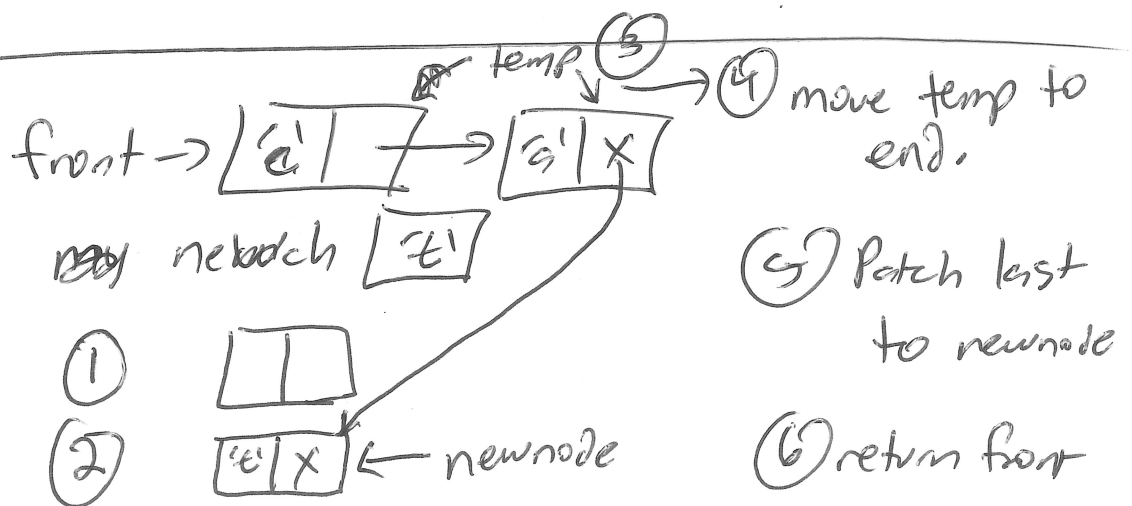
✓ (1) insertBack

✓ (0) makeNode



returns new front of list

node* insertBack(node* front, char item)



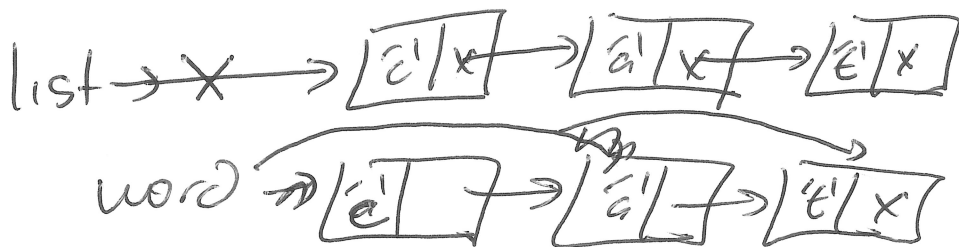
Print

Okay to lose front of the list.
Whoever is calling print still has it.



cat

Picture of strcpy



Insert Back 'c' to list
word = word → next;

