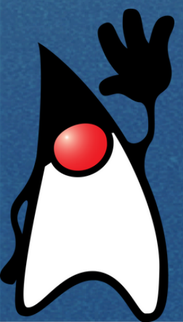
# Week 8!

COMP1511 24T2



Please fill in the feedback form for last week if you haven't gotten a chance yet!

# Overview

Assignment 2

---

Malloc

---

Diagramming Linked Lists


# Assignment 2 Released

Assignment 2 is out

Any questions?

Assignment 2 livestream is the best place to go to get an easy overview on how to get started

# malloc/free

# When to choose malloc()

Using **malloc**:

- dynamically choosing the size of our array (remember C doesn't like: int nums[size]; where size is scanned in at runtime)
- eventually we can decide to change the size of our array if that's required
- we can safely return a pointer towards the variable from a function

Not using **malloc**:

- simpler to use
- we don't need to worry about free()
- no memory fragmentation

# Diagramming Linked Lists

- **Nodes linked together via next fields.**
- **Head pointer stores the address of the first node**
- **Need a current pointer too**

malloc memory for a new node called node1

node1 data = 3
node1 next = NULL

make the head pointer points to node1

malloc memory for a new node called node2

node2 data = 9
node2 next = NULL

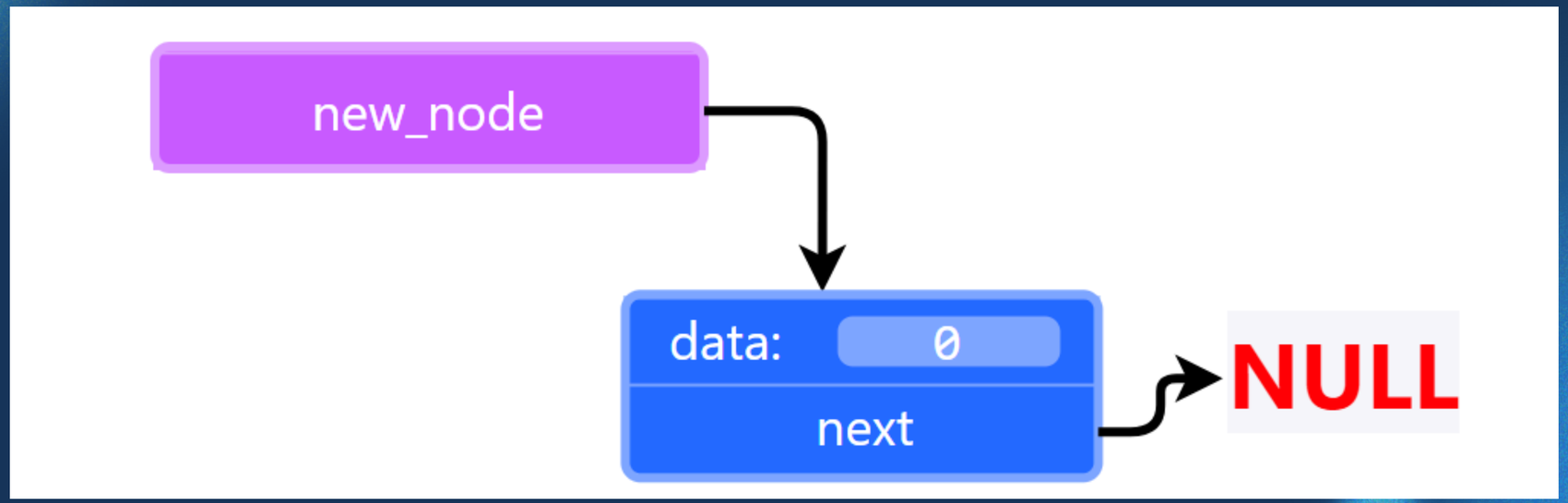
add node2 to the tail of the list, making node1 next point to node2

malloc memory for a new node called node3

node3 data = 5
node3 next = NULL

add node3 to the head of the list, making node3 next point to the current head of the list
make the head point to node3

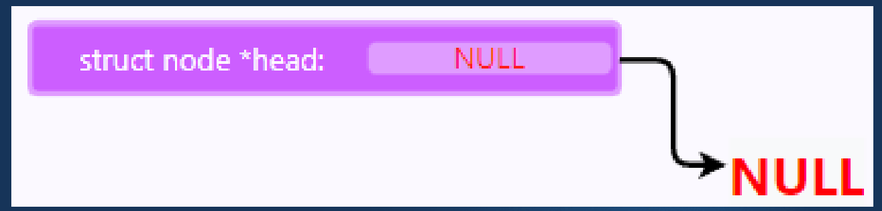
# Struct Pointers

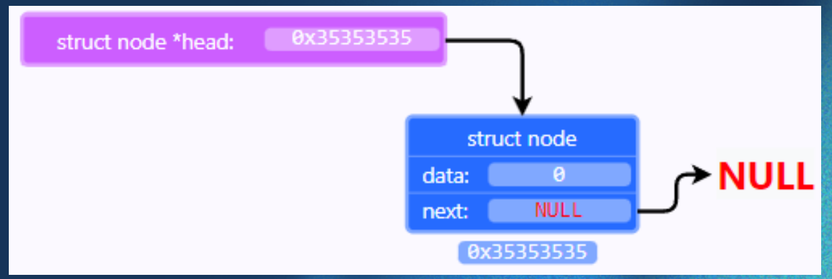


# Task (Inserting into LL)

Every group gets a different edge case:

1. n = 0, and the list is any length (Diagram 1, 2 or 3).
2. n is greater than the length of the linked list.
3. n is any value, and the list is empty (Diagram 1).
4. n is less than the length of the list, and the list is not empty (Diagram 2, 3)




For your task write pseudocode for inserting before the nth node (handle your edge case and others if you have time)

Bonus: what order do we want to check our edge cases and why?