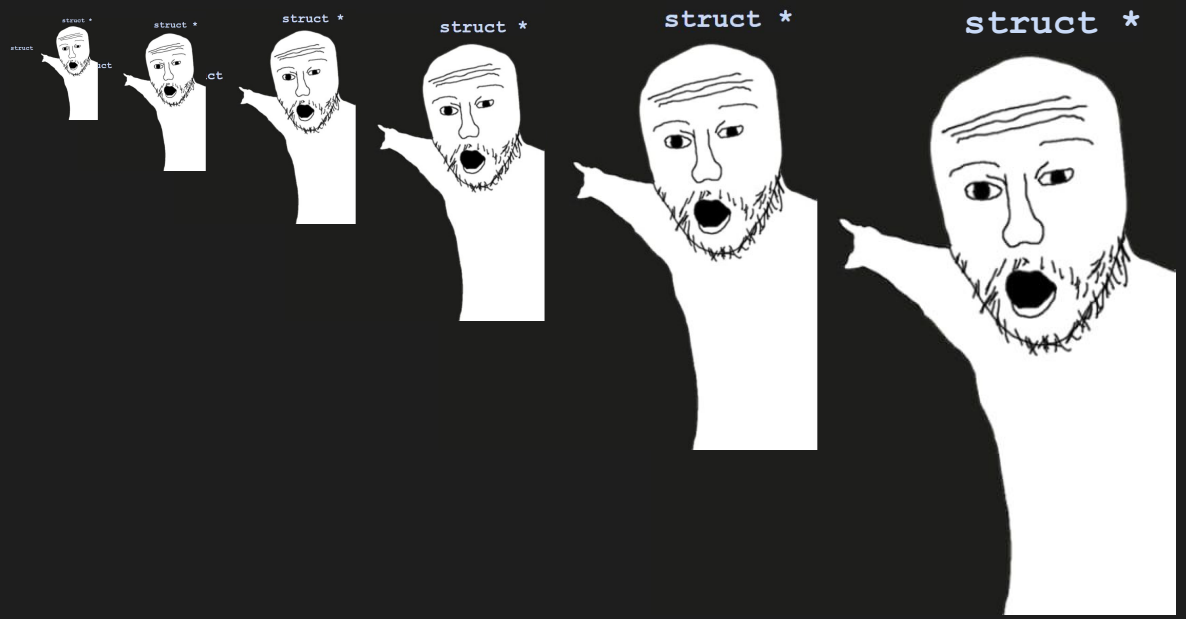


comp1511 week 8

starting ~ 11:07am

notices

- assignment 2 released :)
- assignment 1 marking in progress

today

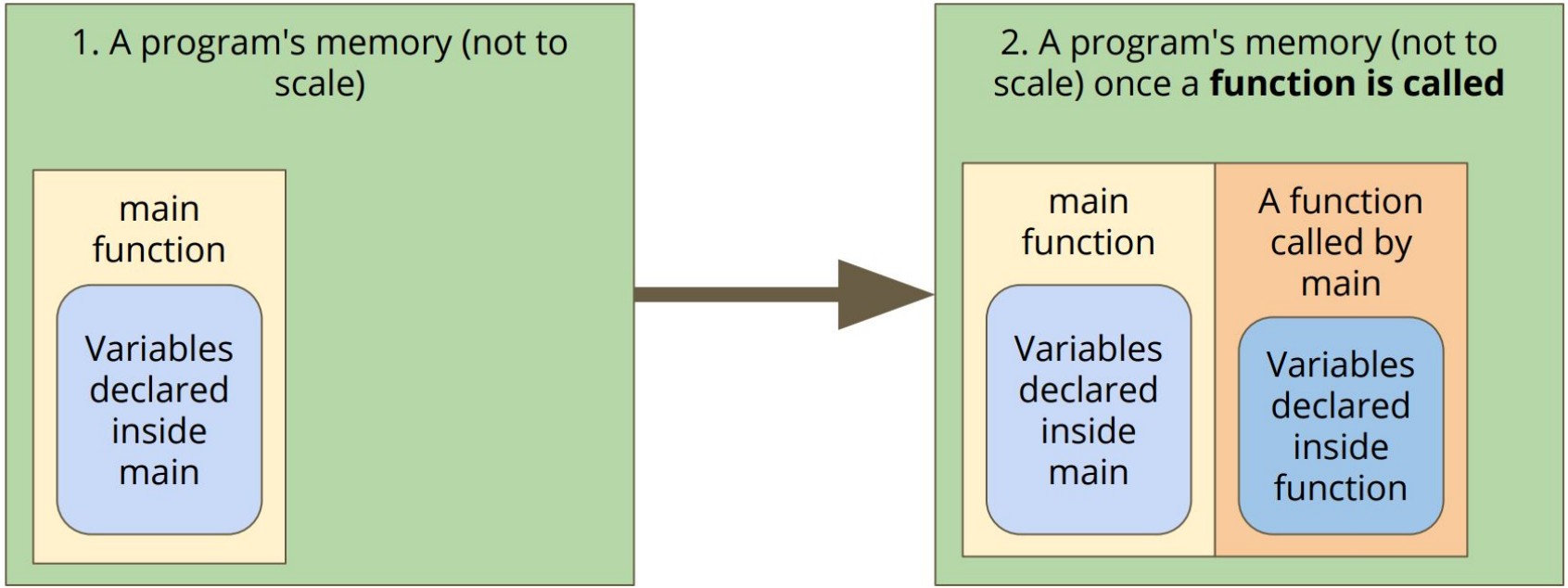
- malloc
- linked lists
  - Intro and diagramming
  - Iterating through a linked list
  - Inserting into a linked list

malloc

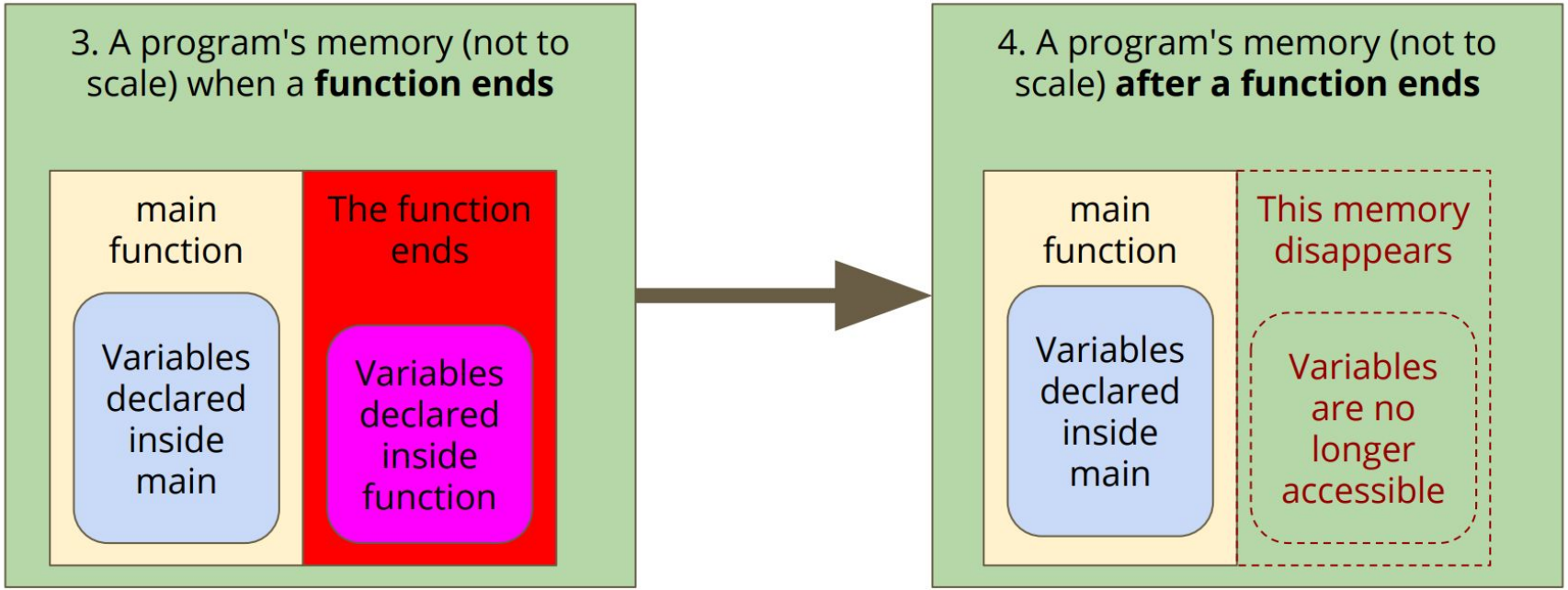
1. **malloc()** - what is it? what inputs? what outputs?
2. **sizeof()** - what does it give us?
3. **how would you malloc a struct?** malloc.c

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

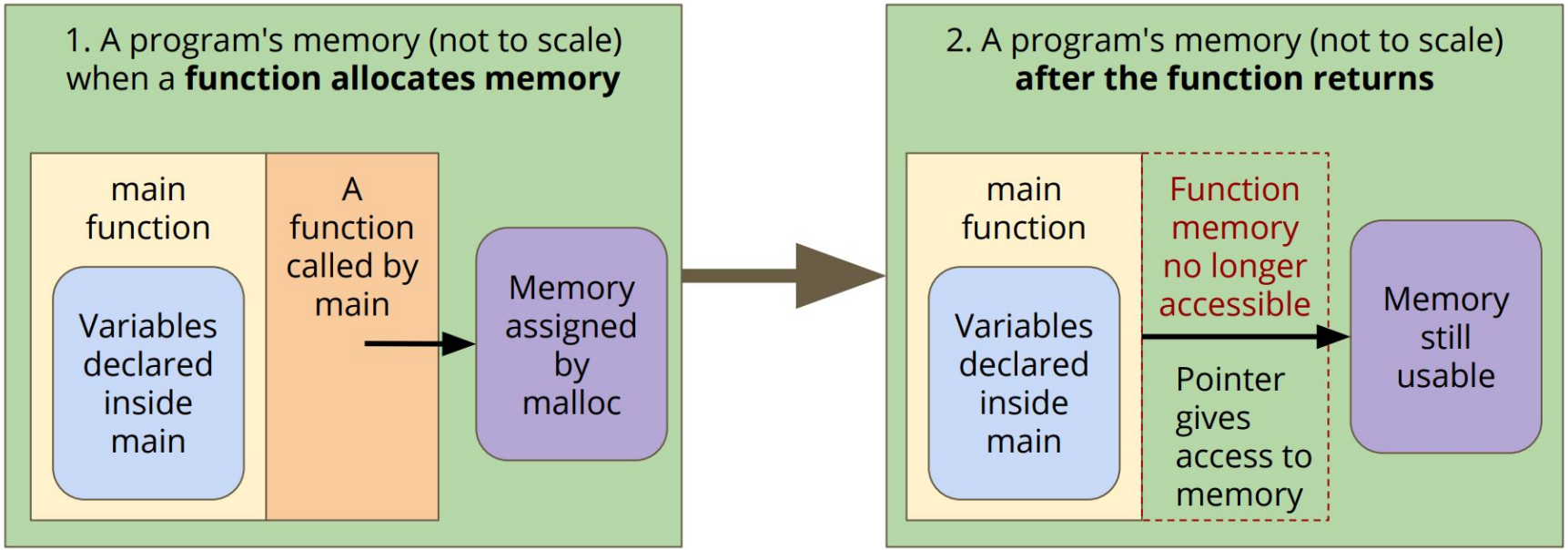
what happens to memory we create inside functions?



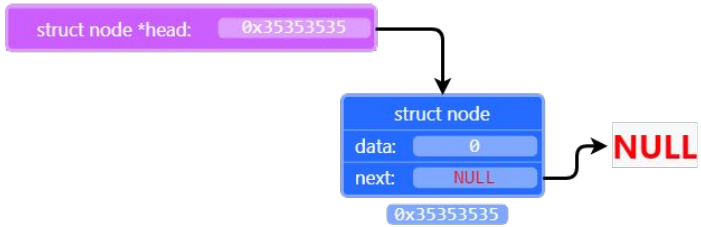
what happens to memory we create inside functions?

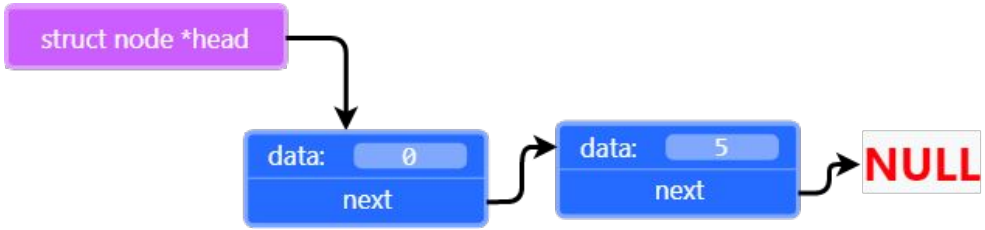


using malloc, we can assign some memory that's not tied to a function (placed on "the heap")



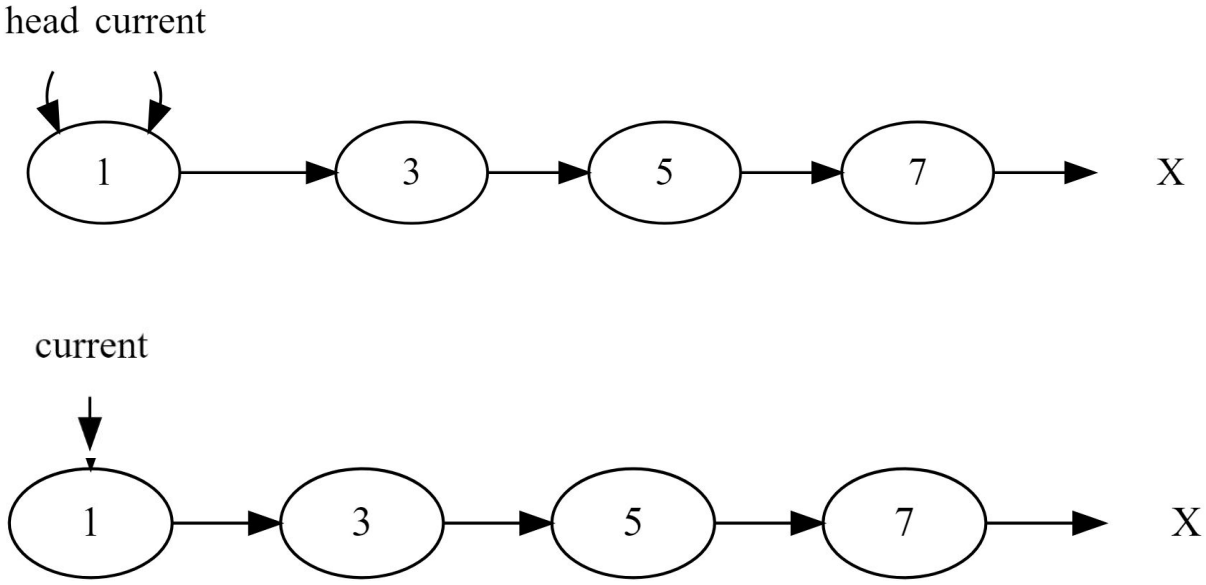
what is a linked list?
and why do we care?





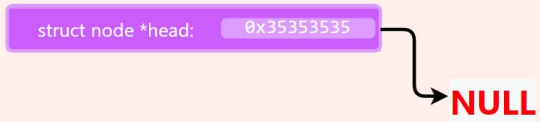
how would you add another node after the first node??

looping through a linked list

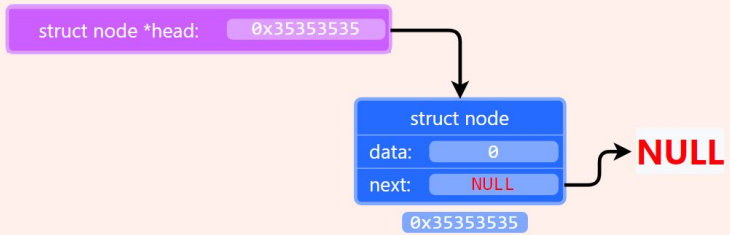


inserting into a linked list

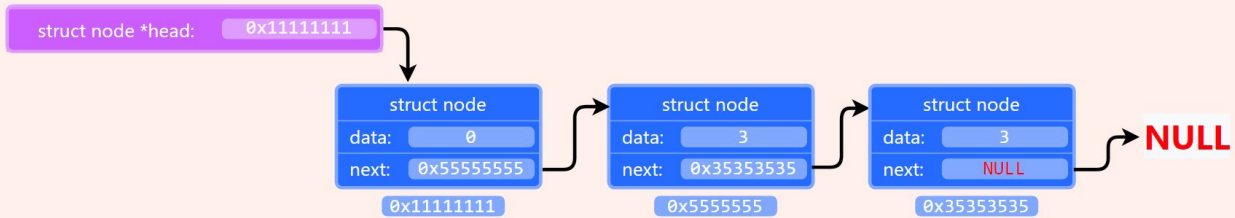
1. An empty list:



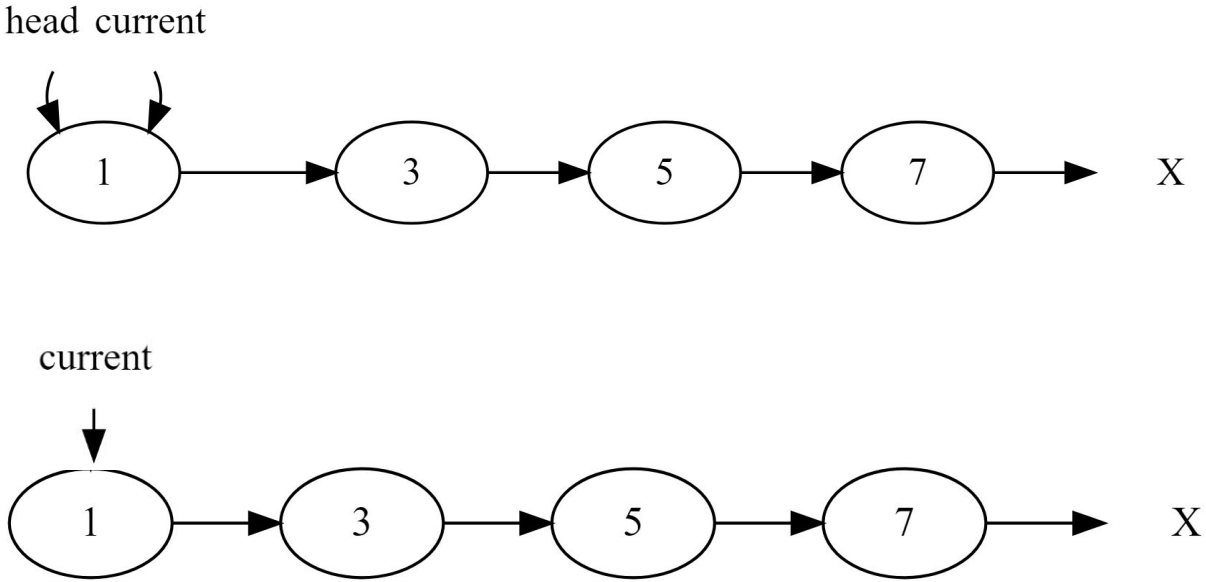
2. A list of length 1:



3. A longer list:



inserting into a linked list



style tips

- **nice work overall from everyone :D**
- **constants**: #define command letters
  - #define ADD_LOGS ‘l’
- **variables**: declare them closer to where they are used rather than all at the beginning (especially for loop counters) – why?
- **functions**: short and sweet, should have one task
  - really, really important for writing good code
- **function comments**: they go above the function and explain what the function does
  - input (including assumptions eg. “takes in a valid coordinate”), output (what it returns), any additional effects
  - comments that make sense on their own rather than “stage 1.2”
- ***any others you want to share?***