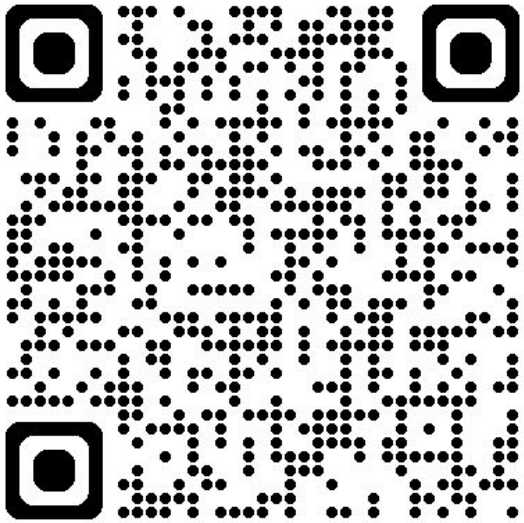
COMP1511/1911 Programming Fundamentals

Week 9 Lecture 2

A larger application continued...
Exam Information

Link to Week 9 Live Lecture Code

https://cgi.cse.unsw.edu.au/~cs1511/25T3/code/week_9/



My Experience Surveys

http://myexperience.unsw.edu.au/

Last Lecture

- Deleting Nodes
- Bigger Linked List Example

Today's Lecture

- Continue larger linked list example from last lecture
- Exam info
- Format
- Preparation
- Hints and tip
- Revision

Code Demo

Email Management System stage 2 and 3

Search and delete all

Email Management System

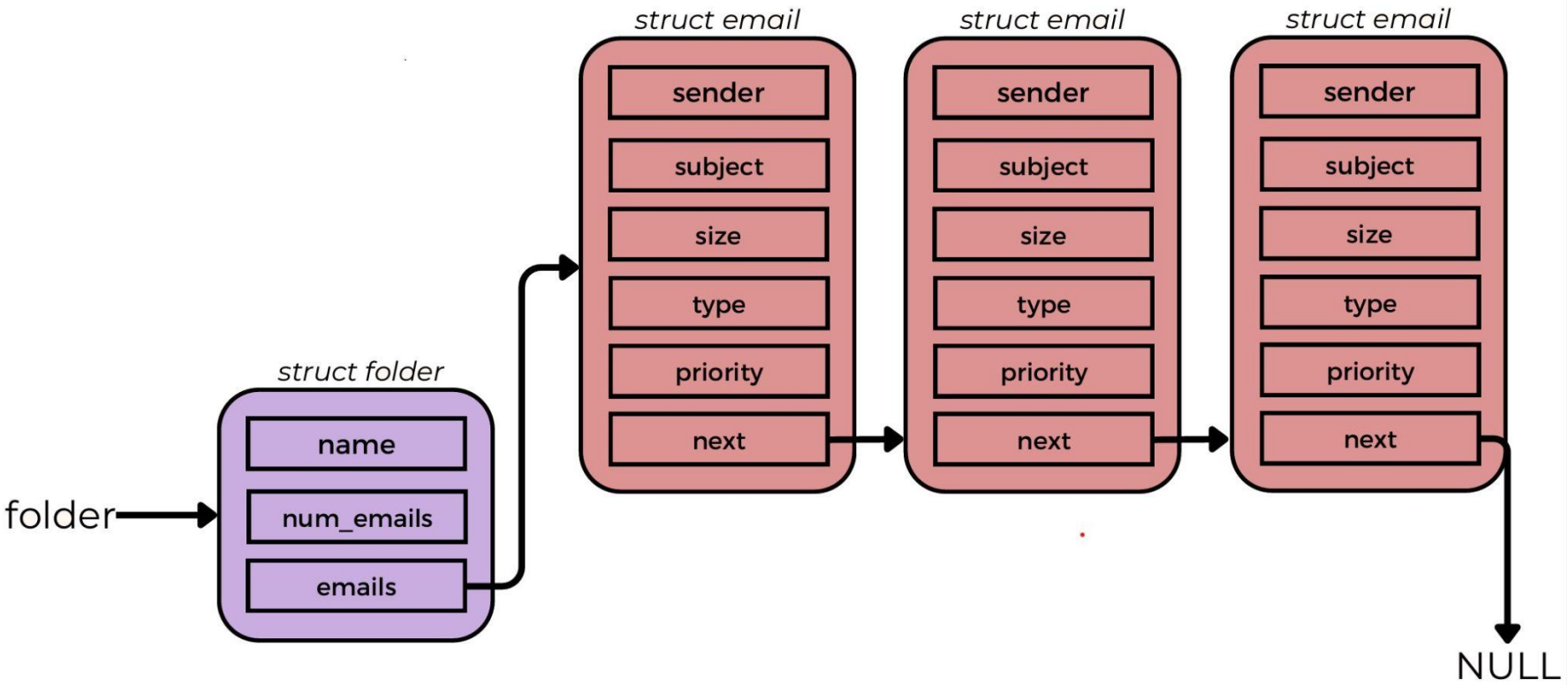
structs

```
struct folder {
    char name[MAX_LEN];
    //to use later :)
    //int num_emails;
    struct email *emails;
};
```

```
struct email {
    char sender[MAX_LEN];
    char subject[MAX_LEN];
    double size;
    enum email_type type;
    enum priority_type priority;
    struct email *next;
};
```

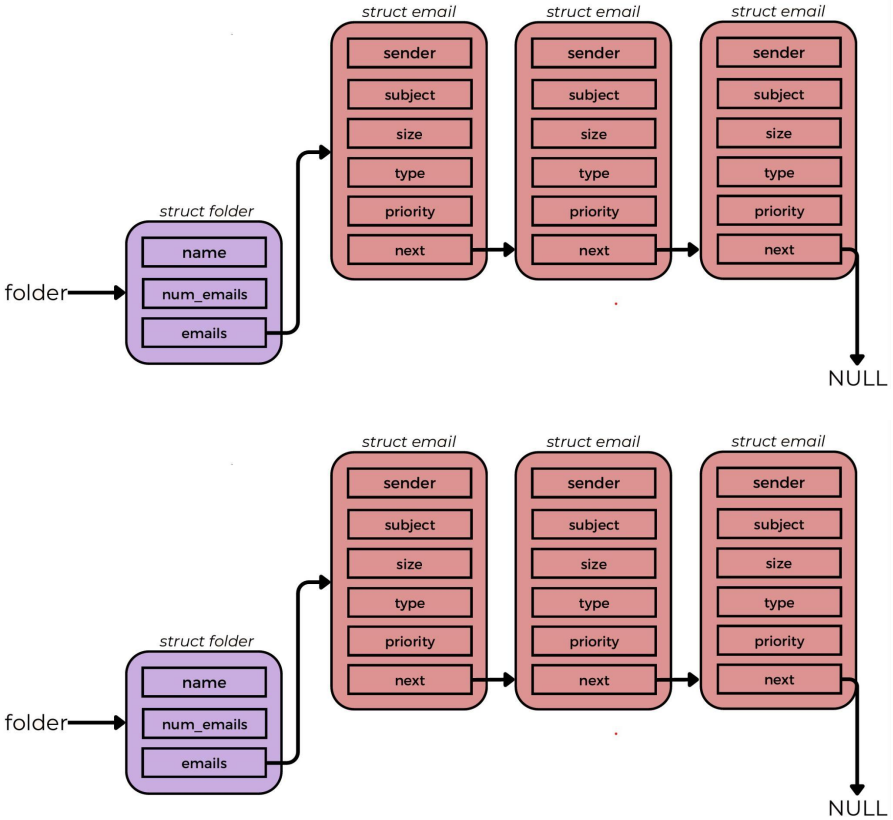
Where are our nodes? Where is the head or list?

Visualisation of the system



Visualisation of the system

We can create many folders, each containing linked lists of emails.



Functions to Write

- Stage 1
  - create_folder
  - insert_email_at_head
  - search_email
- Stage 2
  - clear_folders
  - delete_email_of_priority
- Stage 3
  - merge_folders
  - split_folder (to leave for you to try - I will provide solutions later)

Delete All Nodes the Correct Way

Let's test it and check it with dcc --leak-check

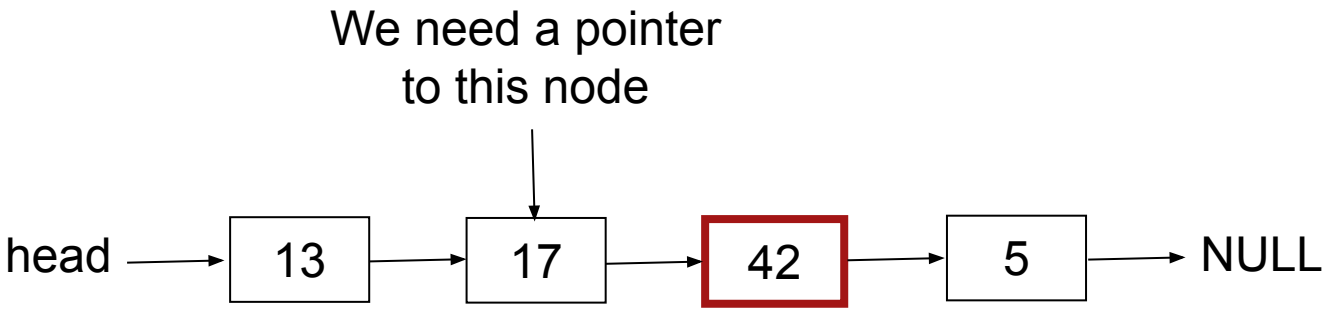
```
// Delete all nodes from a given list
void delete_all_nodes(struct node *head) {
    struct node *current = head;
    while (current != NULL) {
        head = head->next;
        free(current);
        current = head;
    }
}
```

Search and Delete

- We want to search for a node with a particular value in it and then delete it
- Where could the item be
  - Nowhere - if it is an empty list or the list does not contain the value
  - At the head (deleting the first node in the list)
  - Between any 2 nodes in the list
  - At the tail (deleting the last node in the list)
  - There could be multiple occurrences! For now let’s just consider the first occurrence

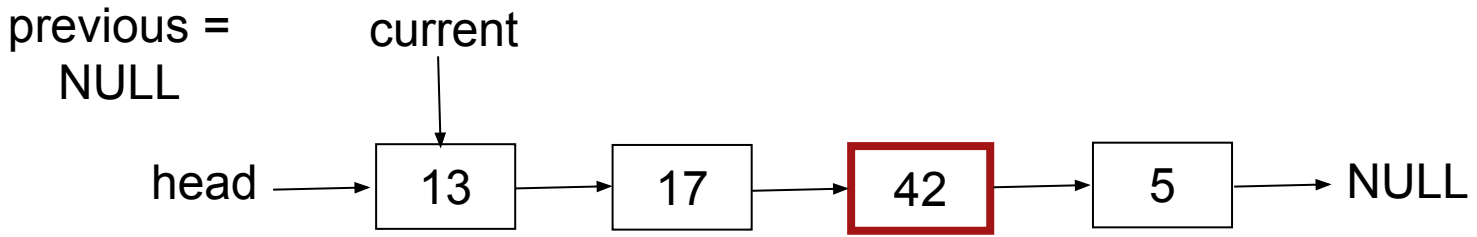
Search and delete: between 2 nodes

- To delete a node we need to link the previous node to the next node
  - If we want to delete the node with 42, we need to find the node before it



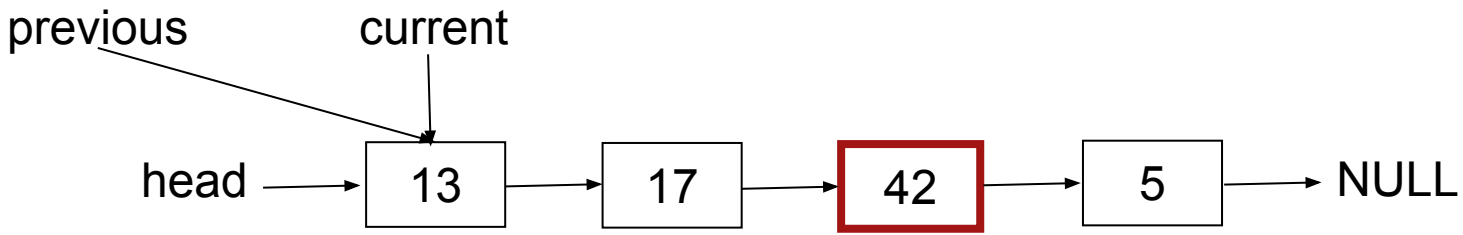
Search and delete: between 2 nodes

```
// Approach 1: Have a previous node pointer
struct node *previous = NULL;
struct node *current = head;
while (current != NULL && current->data != search_key) {
    previous = current;
    current = current->next;
}
```



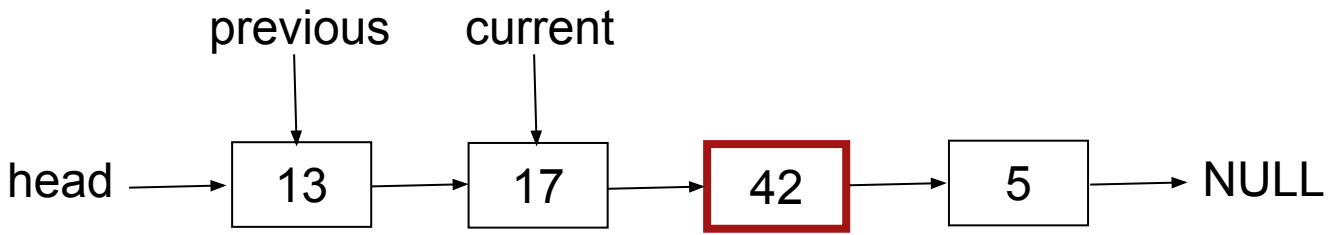
Search and delete: between 2 nodes

```
// Approach 1: Have a previous node pointer
struct node *previous = NULL;
struct node *current = head;
while (current != NULL && current->data != search_key) {
    previous = current;
    current = current->next;
}
```



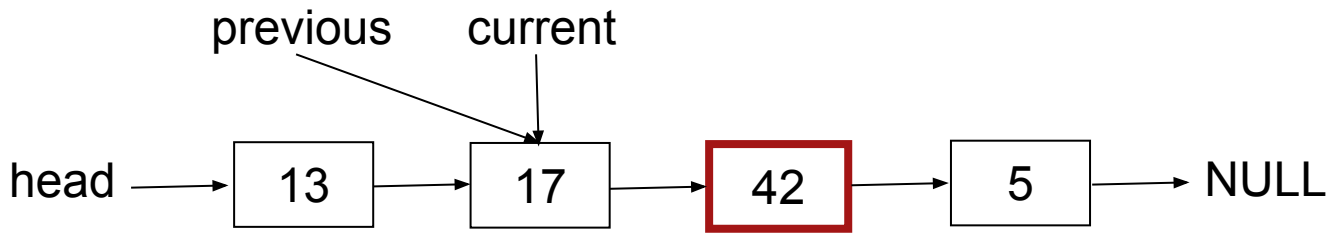
Search and delete: between 2 nodes

```
// Approach 1: Have a previous node pointer
struct node *previous = NULL;
struct node *current = head;
while (current != NULL && current->data != search_key) {
    previous = current;
    current = current->next;
}
```



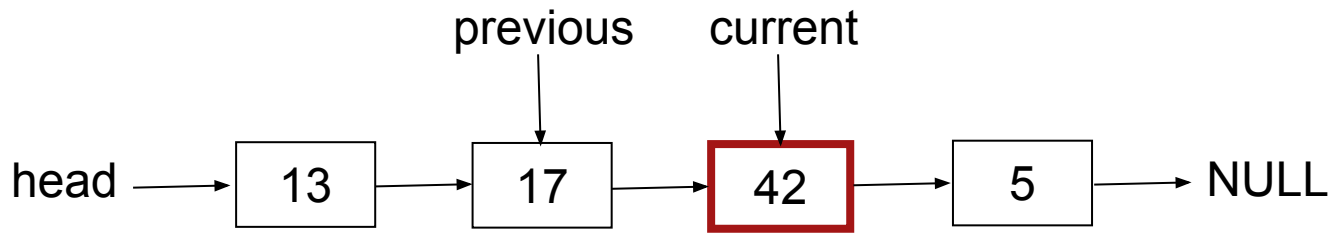
Search and delete: between 2 nodes

```
// Approach 1: Have a previous node pointer
struct node *previous = NULL;
struct node *current = head;
while (current != NULL && current->data != search_key) {
    previous = current;
    current = current->next;
}
```



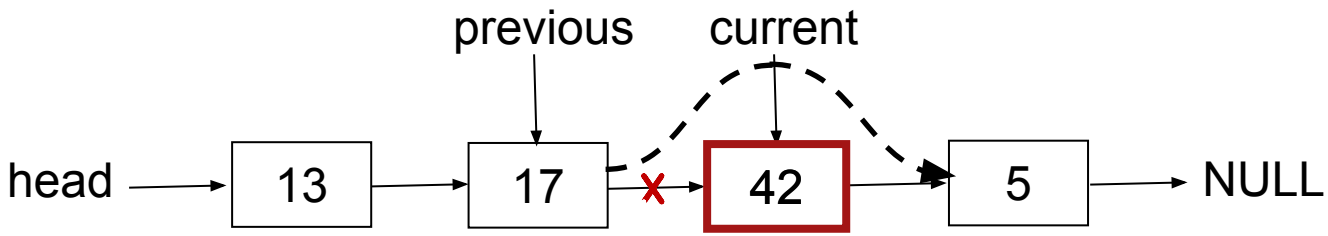
Search and delete: between 2 nodes

```
// Approach 1: Have a previous node pointer
struct node *previous = NULL;
struct node *current = head;
while (current != NULL && current->data != search_key) {
    previous = current;
    current = current->next;
}
```



Search and delete: Approach 1

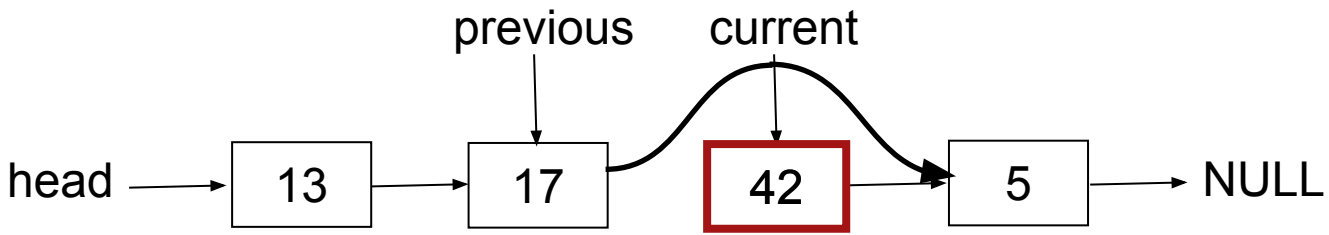
Then we need to connect current node to the one after the one we are deleting.



Search and delete: Approach 1

Then we need to connect current node to the one after the one we are deleting.

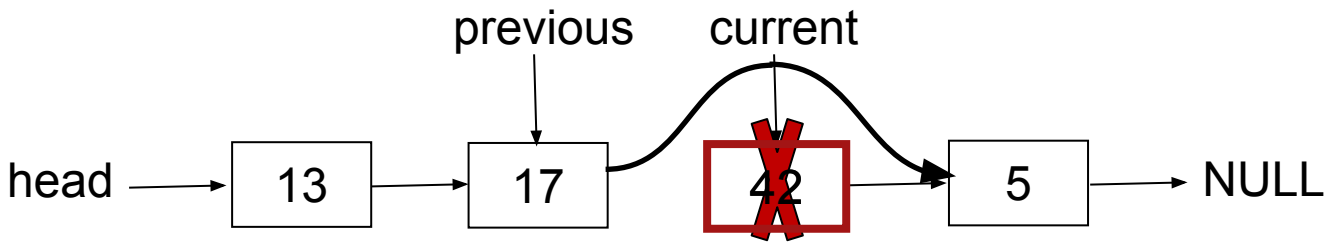
```
previous->next = current->next;
```



Search and delete: Approach 1

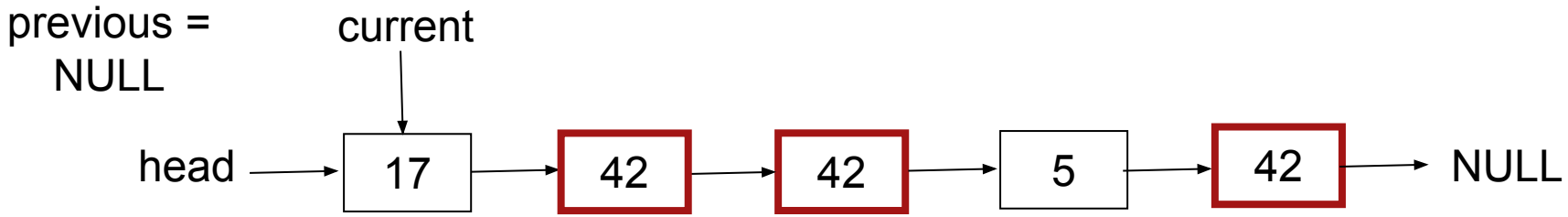
Now we can free the node we want to delete

```
free(current);
```



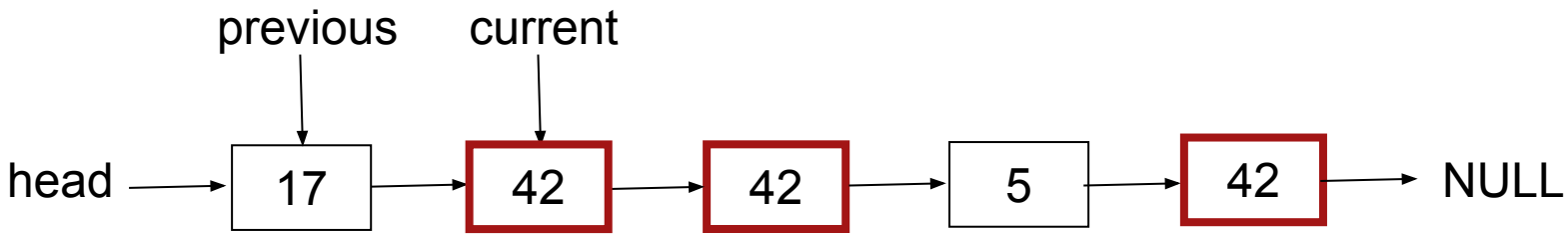
Exercise:

Modify our search and delete to delete all occurrences.



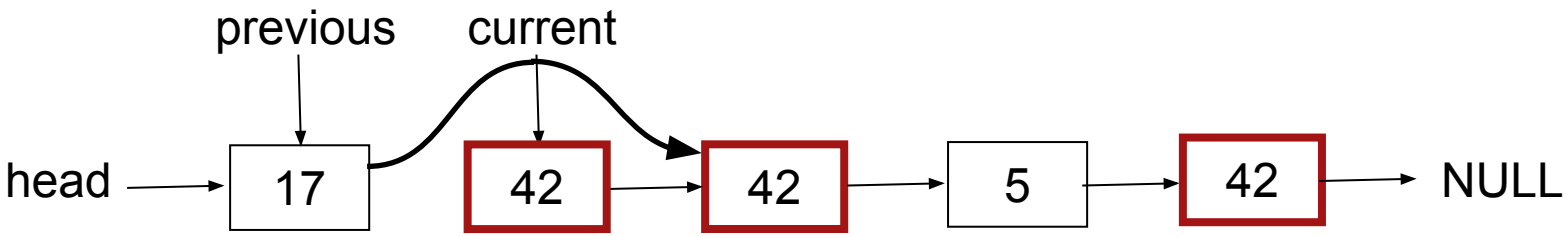
Exercise:

Modify our search and delete to delete all occurrences.



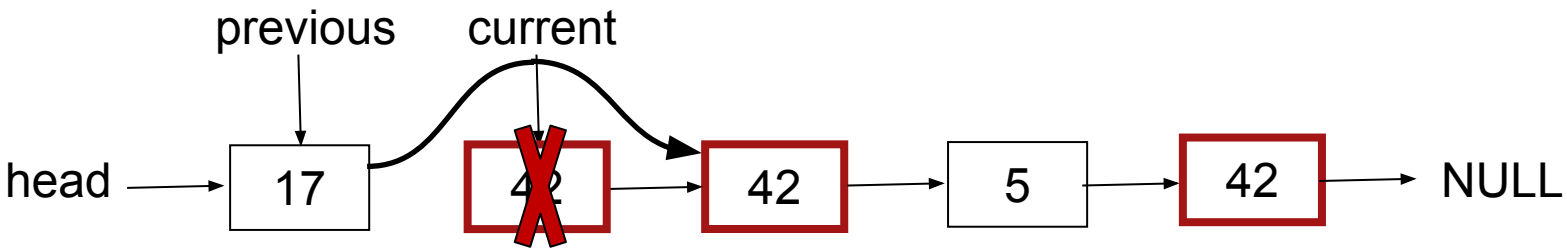
Exercise:

Modify our search and delete to delete all occurrences.



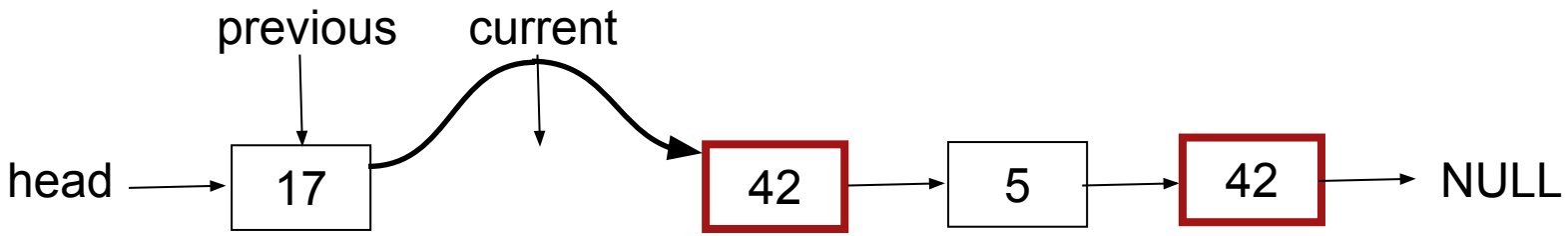
Exercise:

What do we do with previous and current once we delete the first node?



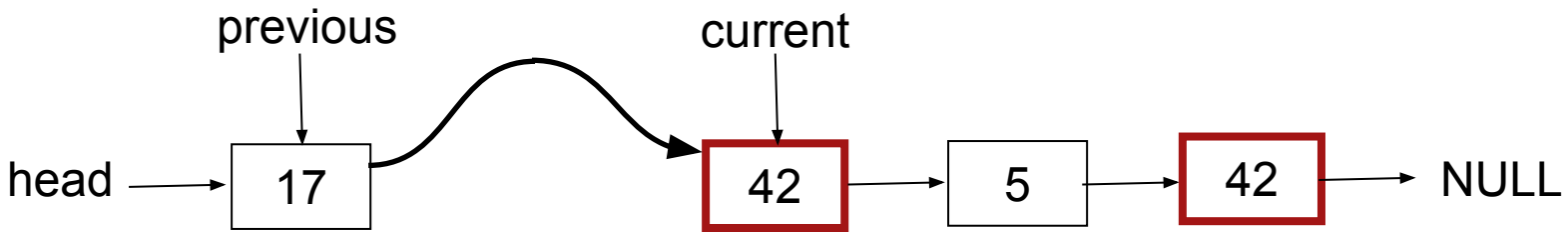
Exercise:

What do we do with previous and current once we delete the first node?



Exercise:

What do we do with previous and current once we delete the first node?




```
current = previous->next;
```

Exam: What is in it?

- Everything that we have learnt so far
- Lots of focus on:
  - Variables: int, double, char, structs
  - Simple IF statements and WHILE loops
  - **Arrays, 2D Arrays**
  - Strings
  - Pointers
  - **Linked Lists**

The Exam: Time and Date and Location

- Date: Wednesday 3rd December
- 3 hours + 10 minutes reading time
- There will be 2 sessions of the exam.
  - **Wed 03/12/2025 Morning** (09:45-13:00)
  - **Wed 03/12/2025 Afternoon** (12:50 corralling 13:25 - 16:40)

Morning vs Afternoon Sessions

- Students in a **Morning** session
  - **are not permitted to leave the exam early**.
  - **who arrive late *may* be allowed to enter the exam** (up to 30 minutes).
- Students in an **Afternoon** session
  - **must be corralled** in a tutorial/lecture room before their exam.
  - **who arrive late are not permitted to enter the exam**.
  - **who arrive late to the corralling room are late for the exam**.
  - **may leave the exam early** but may not re-enter after doing so.

The Exam: Time and Date and Location

- There will be a form to pick your preference for your exam time later this week.
- Get in early to for the best chance of getting your preference.
- Your final confirmed times and locations will be sent to you by the exams team

The Exam Environment

- The practice exam in the lab NEXT week will provide you with a test environment that will be similar to your exam - this will allow you to familiarise yourself with the setup
- dcc, autotests, submit (like give) are the same as in your labs
  - Submit as many times as you want - only last submission will be marked
- **There will not be**
  - `dcc-help`
  - `dcc-sidekick`
  - `autotest-help`

The Exam: Exam Conditions

- Closed book, however:
  - Our course website will be open, which means you will have access and are allowed to refer to:
    - Any lecture material or code
    - Any tutorial material or code
    - Any laboratory material or code

The Exam: Exam Conditions

- Exam conditions apply!!!
- If you experience any issues during the exam, please raise your hand and wait for an invigilator.
- No discussion of the exam or sharing your code with anyone except for COMP1511/1911 staff
- Do not communicate with anyone about the exam within 24 hours of the exam start time - this is considered plagiarism

The Exam: Exam Conditions

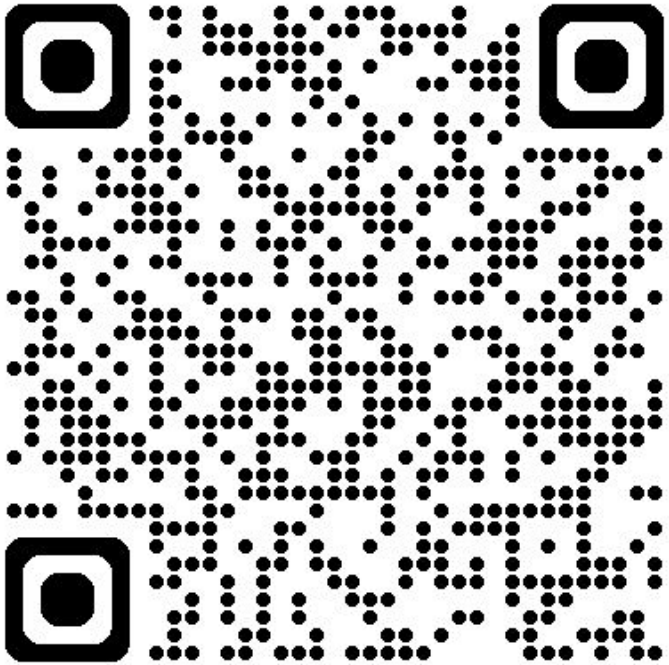
- You can bring a clear bottle of water
- Pens
- NO devices/phones etc
- No paper - we will give you a sheet of paper to do working on
  - This will be collected but not marked.
  - You can ask for more paper if you need it

The Exam: Exam Conditions

- When you come into the room and seat yourself, there will be instructions provided to you on how to start the exam
- You will have all the code in the home directory when you log into the exam.
  - This will be the same as in the practice exam
  - You do not need to fetch the exam.
  - You can re-fetch particular question files if needed.

The Exam: Official Exam Rules: Fit to Sit

- Fit to Sit Policy:
- By sitting the exam on the scheduled assessment date, you are declaring that you are fit to do so and cannot later apply for Special Consideration.



https://www.student.unsw.edu.au/exam-rules

The Exam: Feeling unwell during your exam

- If, during the exam you feel unwell to the point that you cannot continue with the exam
  - Stop working on the exam and immediately raise your hand.
  - Let the invigilator know you are unwell.
  - Submit your work and be escorted from the exam room to discuss with the course convenor/lecturer/admin
  - Obtain a medical certificate within 24 hours of the exam date
  - Submit a Special Consideration application within 3 working days saying that you felt ill during the exam and were unable to continue.

The Supplementary exam

- Tentative Date: 05/01/2026 - 09/01/2026
- Only for students granted special consideration
  - Applied for special consideration due to illness or misadventure
- If you think you will need to sit this exam, make sure you are available
- https://www.student.unsw.edu.au/exam-rules

The Exam: Hurdles

- There's an array hurdle, question 2 and 4
  - You must earn a mark of 50% or more in at least one array hurdle question
- There's a linked list hurdle, question 1 and 3
  - COMP1511 students must also earn a mark of 50% or more in at least one linked list hurdle question
  - COMP1911 students will still do these questions but they will not be hurdles
- These questions will be clearly marked on the exam paper as hurdles

The Exam Format

- 11 Practical Programming Questions
  - rated (with 1 dot, 2 dot and 3 dots) to give you an idea of how difficult each question is
  - we hope that everyone can attempt and complete the first eight questions
  - we hope many students can attempt and complete q9 too.

The Exam Format

- Q1 - Q4 (12 marks each) HURDLES
- Q5 - Q8 (5 marks each) Debugging Questions
  - These questions will be about whether you understand core coding concepts and the C programming language
  - In the style: "Debug the following code to make sure it works to produce the desired output."
- Q9 - Q10 (11 marks each)
- Q11 (10 marks)

Questions

- Questions are similar in style to the revision exercises and problem sets
- Rated (with 1 dot, 2 dot and 3 dots) to give you an idea of how difficult each question is.
- Some will have provided code as frameworks
- Each question will need to be written, compiled and tested
- You will have access to autotests
    - Harder questions will have less autotests

Questions 1 and 2: These are (●○○)

- Similar in style to Practice test questions 1 or 2
- Question 1 is a linked list hurdle (not a hurdle for COMP1911)
  - Use of linked list of ints/chars/doubles
  - no insertion or removal of nodes
- Question 2 is an array hurdle
  - Use of arrays of int/double/struct
- Tests your ability to:
  - Create simple C programs, use variables (int, double, char, structs) , use scanf and printf
  - Use **if** statements and **while** loops

Example Question 1 (●○○)

Perform some computation or comparisons on a linked list

Given a linked list, return the largest value in that list.
If the list is empty, return -1
Edit the function

```
int largest(struct node *head);
```

Example Question 1 (●○○)

Perform some computation or comparisons on 2 linked lists

Given 2 linked lists return the difference in the number of nodes in each list.

Edit the function

```
int difference(struct node *head1,
               struct node *head2);
```

Example Question 2 (●○○)

Loop through an array and gather some kind of information

Given an array of structs, where each struct is:

```
struct direction {
    int number;
    char dir;
};
```

Print out the total of the number of steps taken in a specific direction.

Example Question 2 (●○○)

So for example, if direction is 'l', find all the structs with direction as 'l' and add the numbers in those structs up.

Edit the function:

```
int direction_count(int size,
                    struct direction array[]);
```

Example Question 2 (●○○)

Loop through an array and gather some kind of information

Given an array:
Count the number of multiples of 3 in the array.

```
int array_count_multiples (int size,
                           int array[]);
```

Questions 3 and 4: These are (●●◌)

- Similar in style to Practice test questions 3 and 4
- Question 3 is a linked list hurdle (not a hurdle for COMP1911)
  - You need to pass this OR Question 1 to pass the linked list hurdle
- Question 4 is an array hurdle
  - You need to pass this OR Question 2 to pass the array hurdle
- These are harder applications of the hurdles
- You may need to loop through more than once and/or insert/remove nodes, working with 2d arrays, test more difficult conditions and keep track of more than one thing.

Questions 9 and 10: These are (●●●)

- Question 9
  - Harder manipulation of arrays
  - Possibly fgets or string manipulation
- Question 10
  - Potentially use malloc() and free() with structs and pointers and/or command line arguments

More complex combinations, and some questions requiring interesting problem solving

Question 11: (●●●) and 

- For those aiming for a HD+ mark
- Everything taught in the course might be in these questions
- Will also test your ability to break a problem down into its parts
- The Prac Exam has an example of past Question 11 so you can see the difficulty level

Exam Marking

- Q1-4 and 9-11 are manually marked to award marks for correct code
- Q5-8 (debugging questions) are automarked only.
- Passing a certain number of autotests does not guarantee any specific mark
- There are extra tests used in automarking in addition to the ones you get given as autotests during the exam.

Exam Marking and Style

- There are no marks for style
  - so you don't need to explain your code in comments
  - but it needs to be readable so I can mark it
  - and so you can debug it and not get confused!!

The Exam: What should I study?

- The basics are important!
- Know how to use both arrays and linked lists
- Go back and do the problem sets if you haven't already or redo them for revision
- The revision exercises on the course webpage are also very useful to do and/or redo
- Try to do coding exercises from lectures without looking at the answers
- Try the Week 8 lab exam and week 10 practice exam!
- Try extra practice exams (coming out end of week 10)

The Exam: What should I study?

- Variables, Structs, enums, IF, Looping, Functions, Arrays, Linked Lists are very important to understand!
- You will need to have some understanding of Strings, Pointers, and Memory Allocation to be able to work successfully with char arrays, and linked lists
- Advice: Stop using chatgpt as a crutch if you are doing so
    - You will not have that in the exam

Log in Details

- Make sure you know you zid and zpass to log into the exam environment.
- You will not have access to phones or internet or paper notes to look up these log-in details!
- Please make sure you have them memorised

The Exam: Hints and Tips

- Read all questions before starting
- Start with the easier questions
- Make sure you read the question before jumping in and coding up something the question is not asking or using libraries it says not to use.
- Prepare! A couple of minutes thinking and drawing a diagram will clarify how you're going to approach a question
- When you are struggling with a question (particularly linked lists) DRAW DIAGRAMS!

The Exam: Hints and Tips

- Less questions answered completely is better than more questions partially answered
  - But don’t get stuck on a question you can’t make progress on for the whole 3 hours!!!
- The later questions have been designed to be very challenging
  - You do not need to complete them to get a great mark in this subject

The Exam: Advice

- Try to get a good night's sleep
- Try to eat properly before so you can concentrate.
- Make sure you breathe!
- You can do it!

What did we learn today?

- Email Management System Code
- Exam Details!!!

Next Week

- Revision
  - Forum poll to see what topics to focus on
- Q&A with tutors

Reach Out

Content Related Questions:

Forum

Admin related Questions email:

cs1511@unsw.edu.au