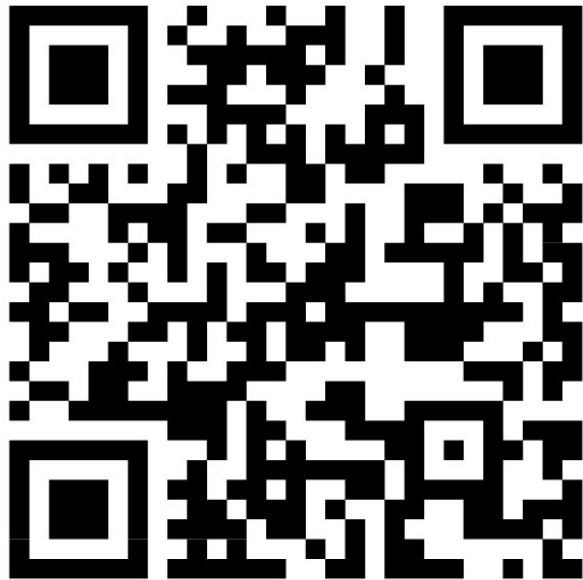


COMP1511/1911 Programming Fundamentals

Week 10 Lecture 1

Revision

My Experience Surveys



Tell us about your experience and shape the future of education at UNSW.

Click the link in Moodle

Please be mindful of the [UNSW Student Code of Conduct](#) as you provide feedback. At UNSW we aim to provide a respectful community and ask you to be careful to avoid any language that is sexist, racist or likely to be hurtful. You should feel confident that you can provide both positive and negative feedback but please be considerate in how you communicate.



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

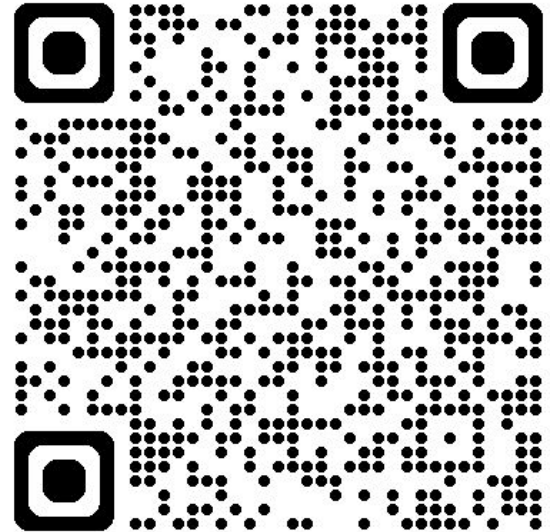
Revision Sessions

In Week 11, we will be running some exam revision classes.

Registration and details will be announced soon .

Week 10 Practice Exams

- If you are in an online tut-lab
 - you can sign up for an in-person lab for week 10
 - sign up details also on Ed forum.
 - Access code: COMP1511
- Don't miss this chance to see what the exam environment is like



<https://buytickets.at/comp1511unsw/1447098>

Week 10 Practice Exams

During the practice exams:

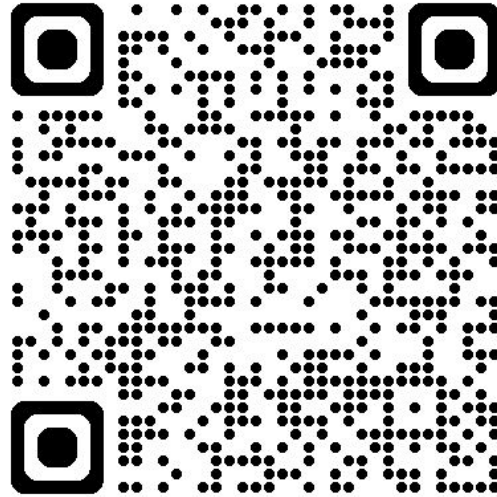
- Any autotests or submissions will not be stored on student history
- Any code written in the exam environment will not be accessible to you once you log out of the exam environment.
- Practice exam questions will also be available at home via the week 10 lab on the course website,
 - This is useful for study
 - However please come to the labs to see the exam environment!

The Exam: Time and Date and Location

- Date: 25th November
- 3 hours + 10 minutes reading time
- There will be 2 sessions of the exam.
- The students sitting the exam in the afternoon will be corralled for a period of twenty-thirty minutes as we conduct the changeover.
- Students in the morning exam, will not be able to leave early.
- **Mon 25/11/2024 Morning (09:45-13:00)**
- **Mon 25/11/2024 Afternoon (12:50-16:40)**

The Exam: Time and Date and Location

<https://cgi.cse.unsw.edu.au/~exam/24T3/seating/register.cgi>



Last Lecture

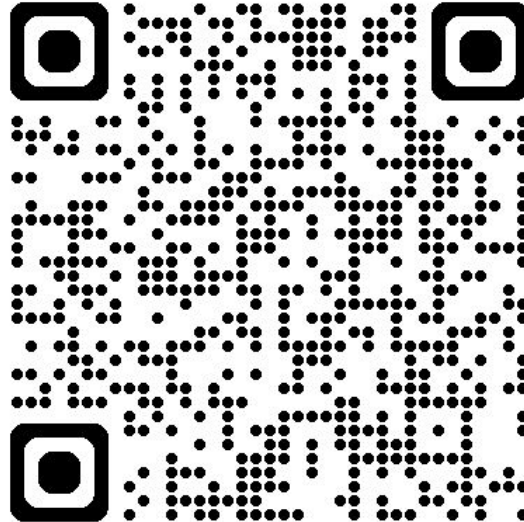
- Exam Details
- Exam Revision
 - 1 dot hurdle questions for linked lists and arrays

Today's Lecture

- String and Command Line Argument Revision
 - Includes important concepts from Lab 9 question 1
- More difficult Linked List Questions
 - (2 dot level) search and delete approach 2 with curr->next
 - (3 dot level) split lists (Email Management System)
- Quick look at the exam account
- Sample exam questions (to be continued on thursday)
 - 1 dot Linked List and array hurdles
 - 2 dot Linked List and array hurdles
 - Debug string question

Link to Week 10 Live Lecture Code

https://cgi.cse.unsw.edu.au/~cs1511/24T3/live/week_10/



Coding with Strings and Command Line Args

`basic_strings.c`

`string_pointer.c`

`basic_command_line_args.c`

`command_line_args_list.c`

Strings

Consider the variables `s0`, `s1`

- What are their sizes in bytes?
- What are they storing right now?
- What is their `strlen`?

`MAX` is defined as 128.

```
char s0[] = "abcd";  
char s1[MAX];
```

Strings

Can we make a copy of the variable s0 by doing the following?

```
char s0[] = "abcd";  
char s1[MAX];  
s1 = s0;
```

Strings

What if we had another variable s2. Can s2 store a copy of s0?

```
char s0[] = "abcd";  
char s1[MAX];  
char *s2;
```

Command Line Arguments

Write a program to print out each command line arguments (not including the program_name) along with their string lengths.

Command Line Arguments, Nodes, Strings

Assume we have the following linked list struct definition.

Write a program to allow the user to enter the data on the command line, for example:

```
struct person {  
    char *name;  
    struct person *next;  
};
```

```
./prog Sofia Nicole Daniel Ibbby Grace
```

This is similar to Question 1 in Lab 9

Command Line Arguments, Nodes, Strings

Assume we have the following linked list struct definition.

Compare this struct to the structs in the assignment that you are working with that contain strings e.g

Are there any benefits from either approach?

```
struct person {  
    char *name;  
    struct person *next;  
};
```

```
struct dungeon {  
    // name of the dungeon  
    char name[MAX_STR_LEN];  
    // etc
```

Linked List More difficult Examples

Search and delete with approach 2

Email Management System, split_list

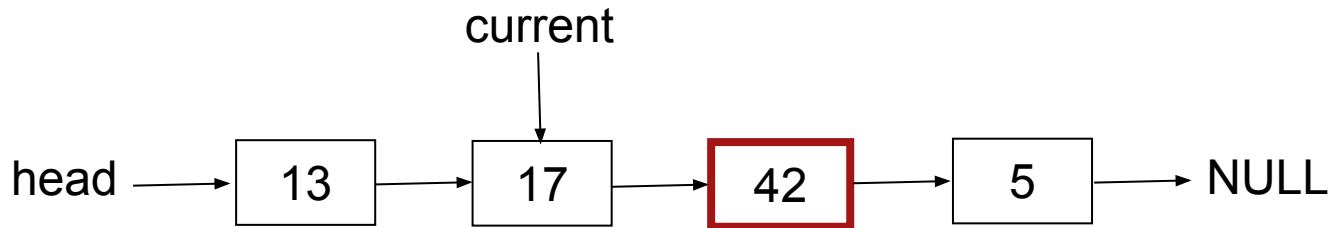
Linked List Coding Reminder

Some special boundary conditions that you need to consider when you manipulate lists:

- Empty list
- List with 1 element
- Something happening at the beginning of the list
- Something happening at the end of the list
- Something will not occur, the item is not in the list (inserting after a number that doesn't exist etc)

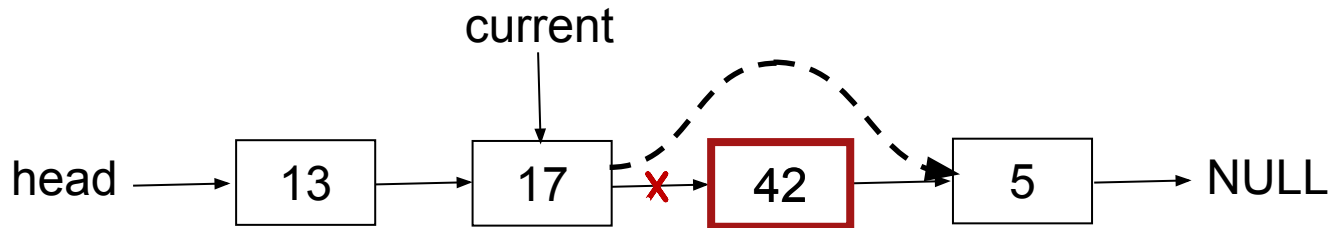
Search and delete Approach 2: general case

```
// Approach 2: Just use 1 pointer to traverse
// but check the next node
struct node *current = head;
while (current->next != NULL &&
       current->next->data != search_key) {
    current = current->next;
}
```



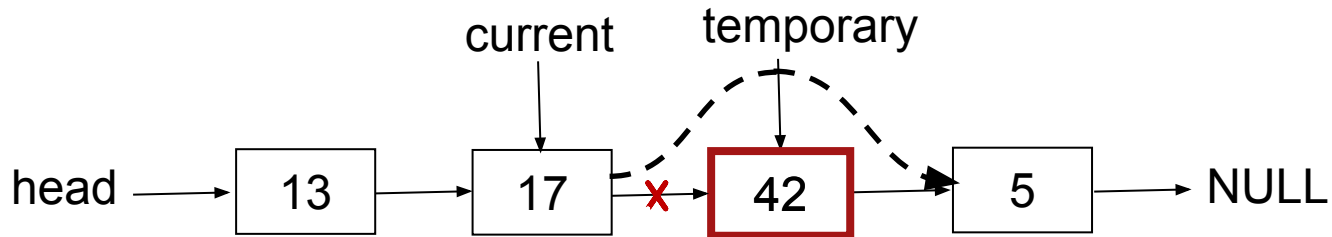
Search and delete: Approach 2

Then we need to connect current node to the one after the one we are deleting. But we still need a pointer to the node we want to free. How can we do that?



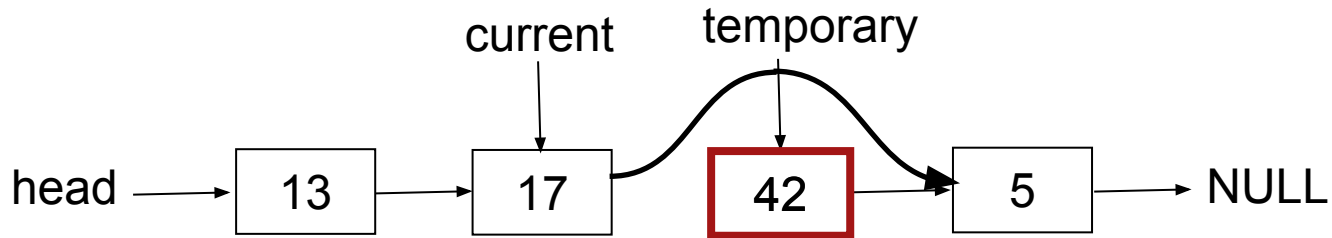
Search and delete: Approach 2

```
struct node *temporary = current->next;
```



Search and delete: Approach 2

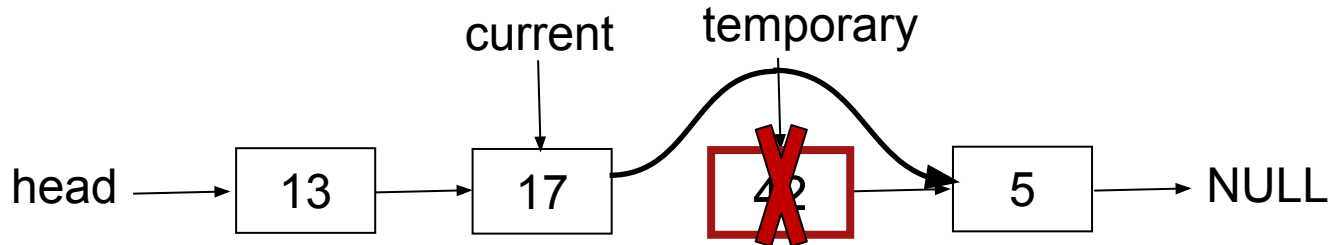
```
struct node *temporary = current->next;  
current->next = temporary->next;
```



Search and delete: Approach 2

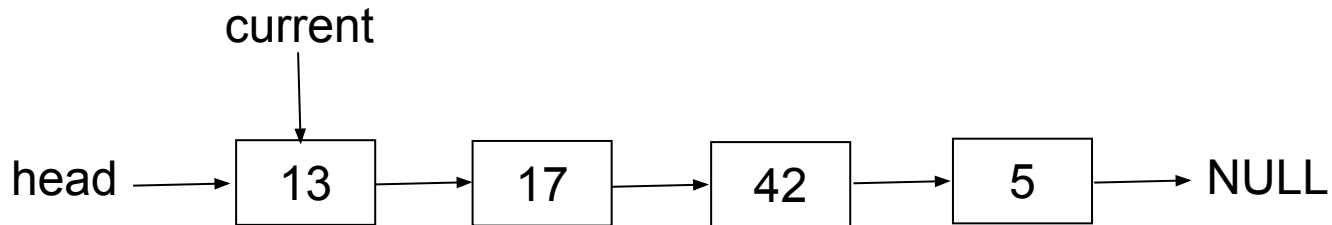
Now we can free the node we want to delete

```
free (temporary) ;
```



Search and delete Approach 2: Edge Cases

```
// What if the list is empty?  
// What if we are trying to delete the first node?  
// Or we don't find the value we are searching for?  
struct node *current = head;  
while (current->next != NULL &&  
       current->next->data != search_key) {  
    current = current->next;  
}
```

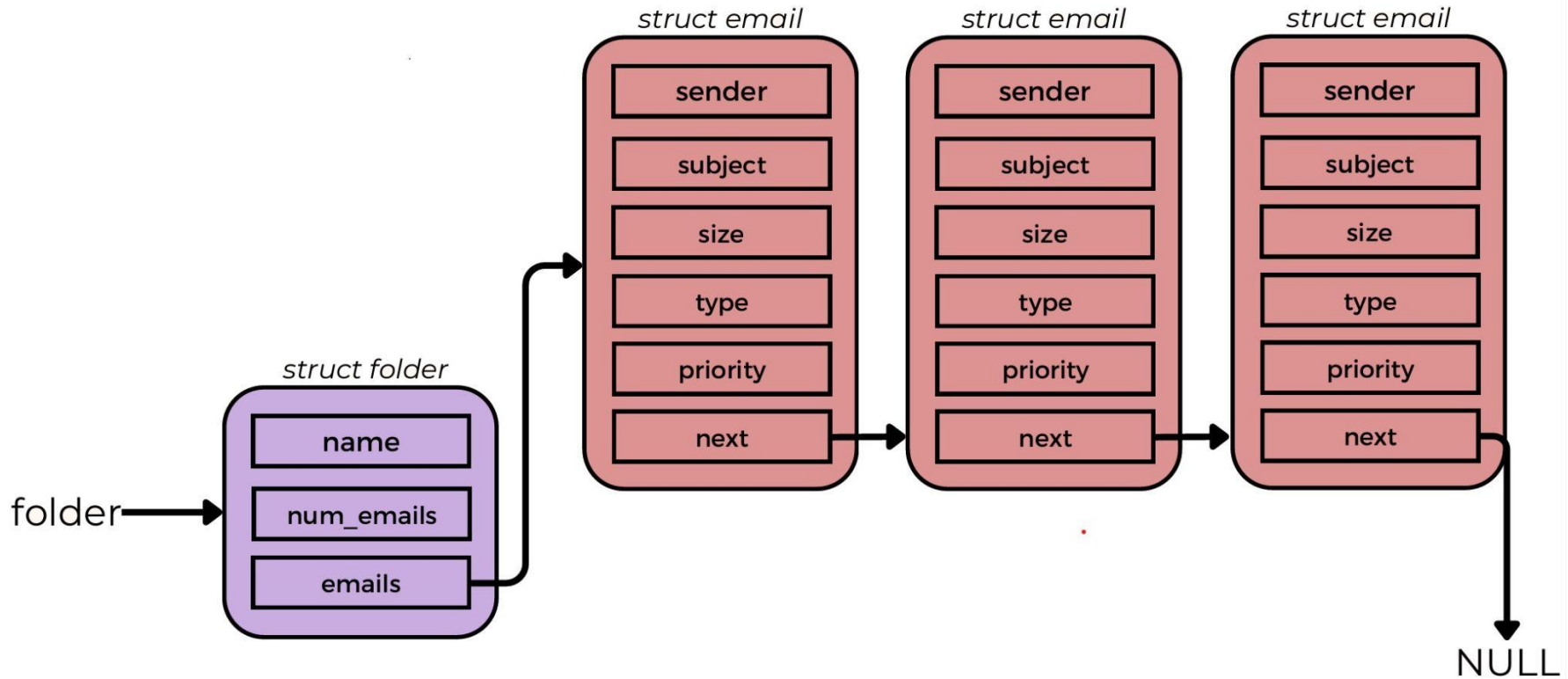


Email Management System (EMS): structs

```
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;  
};
```

EMS: Visualisation of the system



EMS: Linked List (●●●)

```
// Split emails in a given folder into 3 folders based on
// email type keeping original ordering
// assumption: inbox, sent and draft are folders
// are already malloc-ed and initialised
void split_folder(struct folder *given_folder,
                 struct folder *inbox, struct folder *sent,
                 struct folder *draft) {
    // TODO: implement this function
    printf("split_folder not yet implemented.\n");
    exit(1);
}
```

A Quick Look at the Exam Account

Sample Hurdle Exam Questions

Linked List Hurdle Example (●○○)

`linked_list_range.c`

Find the range (the difference between the biggest term and the smallest term) of a linked list

This is an exercise for you to try

Linked List Hurdle Example (●○○)

`linked_list_equal.c`

Given two linked lists, return the number of values in the first linked list that are equal to the corresponding values in the second linked list.

Array Hurdle Example (●○○)

`count_odd.c`

Given two arrays of the same size, return the number of items at corresponding indexes that are odd in both arrays.

This is an exercise for you to try

Array Hurdle Example (●○○)

`array_struct_capitals.c`

An array of structs of type:

```
struct initials {  
    char first_initial;  
    char last_initial;  
};
```

return the count of all structs that have don't have capital letters for both of their initials.

Linked List Hurdle Example (●●○)

`linked_list_duplicate.c`

Duplicate every node in the list by inserting the same node after the original node.

Linked List Hurdle Example (●●○)

`linked_list_delete_duplicate.c`

Delete the first instance of a duplicate in the given list e.g.

1->3->5->3->1->7->X

would give

1->3->5->3->7->X

Array Hurdle Example (●●○)

`array_odd_even.c`

Write a C program that reads integers from standard input until it reads a negative integer.

It should then print the odd numbers on one line and then print the even numbers on the next line.

You can assume a maximum of 1000 integers are read before a negative integer is read

Array Hurdle Example (●●○)

```
$ ./array_odd_even
```

```
1
```

```
2
```

```
3
```

```
2
```

```
-42
```

```
Odd numbers were: 1 3
```

```
Even numbers were: 2 2
```

Array Hurdle Example (●●○)

`array2d_min_row_sum.c`

Write a C program that finds the sum of the minimum numbers in each row in a 2D array.

Debug Example (●○○)

`debug.c`

The following code is meant to join two strings together and form one string. For example:

```
$ ./debug
$ Enter the first string: Pass
$ Enter the second string: Word
PassWord
```


What did we learn today?

- Exam Details!!!
- Revision
 - Search and delete approach 2
 - Finishing up the Email Management System code
 - example exam q1 and q2 hurdle exercises
 - example exam string debug question

The Final Lecture on Thursday

- The final Kahoot
- Continue with revision
- Q&A with some course tutors
 - Exam tips
 - What to enrol in next term
 - Getting internships and jobs
 - Being a tutor

Reach Out

Content Related Questions:
Forum

Admin related Questions email:
cs1511@unsw.edu.au

Don't forget to attend Help Sessions
if you need one on one help



Struggling with non-course specific issues?

My Feelings and Mental Health

Managing Low Mood, Unusual Feelings & Depression



Mental Health Connect

student.unsw.edu.au/counselling
Telehealth



**In Australia Call Afterhours
UNSW Mental Health Support Line**

1300 787 026
5pm-9am



Mind HUB

student.unsw.edu.au/mind-hub
Online Self-Help Resources



**Outside Australia Afterhours
24-hour Medibank Hotline**

+61 (2) 8905 0307

Uni and Life Pressures

Stress, Financial, Visas, Accommodation & More



**Student Support
Indigenous Student Support**

- student.unsw.edu.au/advisors
- nura-qili-centre-indigenous-programs

Reporting Sexual Assault/Harassment



Equity Diversity and Inclusion (EDI)

- edi.unsw.edu.au/sexual-misconduct

Educational Adjustments

To Manage my Studies and Disability / Health Condition



Equitable Learning Services (ELS)

- student.unsw.edu.au/els

Academic and Study Skills



Academic Skills

- student.unsw.edu.au/skills

Special Consideration

Because Life Impacts our Studies and Exams



Special Consideration

- student.unsw.edu.au/special-consideration