

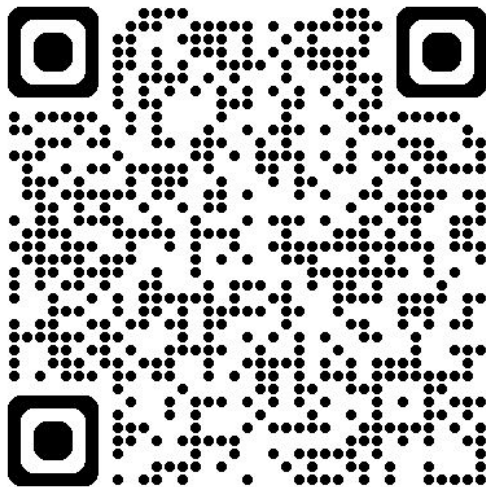
COMP1511/1911 Programming Fundamentals

Week 10 Lecture 2

The Final Lecture

Link to Week 10 Live Lecture Code

https://cgi.cse.unsw.edu.au/~cs1511/25T3/code/week_10/



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/>

Don't forget to bring your student ID card to the final exam!

- You must display your current UNSW student identification card on your desk for the duration of the exam.
- If you don't have your ID card, you will be required to provide other official photo identification such as your Drivers License or Passport. Electronic versions of identification will not be accepted.
- If your name has changed or your card has expired or is lost or damaged, book an appointment with The Nucleus: Student Hub immediately for a replacement ID card [Get your student ID card | Current Students - UNSW Sydney](#).
- Note from 26T1 UNSW will not accept any ID except for a valid UNSW student identification card.

Wellness for Exams

Important tips: the hours before an exam



Make sure you get a good night's sleep

If you are well rested, you will think more clearly, remember more and be able to solve problems creatively. Don't pull an all-nighter; being alert is your greatest asset.



Pack your bag the night before

Be sure to check what materials you're expected to bring and include extra stationery in case you need it. Neatly organise your materials, especially notes for open-book exams.



Be familiar with your exam location

Make sure you know how to get to the exam with minimum stress well ahead of time. You can visit the exam venue a few days before to establish how you'll get there.



Arrive ahead of time

Plan to get to there at least 30 minutes before the exam start time. This will ensure you aren't stressed and leave some buffer room for unexpected delays.



Avoid panic talk!

You don't want other people's nervousness or stress to rub off on you before an exam. Stand apart from the main scrum of students and keep calm.



Fuel yourself

Easily digestible, solid foods like rice, bread, savoury biscuits or bananas are the best. Keep coffee and sugar to a minimum as these will only give a short boost.

More information can be found here: [Exam preparation | Current Students – UNSW Sydney](#)

Study Resources

- Week 8 Lab Test (array hurdles and debug questions)
- Full Practice Exam Week 10 Labs
- More full practice papers to be released this weekend
- Practice Exam questions Week 9 and 10 Lectures
 - You can try the ones we did again for yourself.
 - Some we will not get through are left for self study
- Extra Challenges questions (some are 1 or 2 dots)
- Extra Revision questions
- Redo Problem Sets
- Try Lecture Coding problems yourself

Last Lecture

- Exam Revision
 - Array Hurdles
 - Linked List Hurdles
 - Array coding activity with Tom

Today's Lecture

- More exam revision questions
 - Debug
 - Hurdles
 - String, command line argument questions
- The final Kahoot
- Q&A with the tutors

There are many more questions than we will cover in the lecture.
The rest can be used for your own revision.

Debug List (●○○)

`debug_list.c`

The following code should find the minimum value in a list and return the value -1 if the list is empty:

```
$ ./debug_list 4 10 5 2 9
$ 2
$ ./debug_list
$ -1
```

Debug Example (●○○)

`debug_string.c`

Note: you may NOT use the `strcpy` or `strcat` function.

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

```
$ ./debug_string
$ Enter the first string: Last
$ Enter the second string: Lecture
LastLecture
```

This is an exercise for you to try

Sample Hurdle Exam Questions

Linked List Hurdle Example (●○○)

`linked_list_difference.c`

Perform some computation or comparisons on 2 linked lists

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

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

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.

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

Linked List Hurdle Example (●○○)

`linked_list_largest.c`

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

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

This is an exercise for you to try

Linked List Hurdle Example (●○○)

`list_find_divisible.c`

Count all the elements in the linked list that are divisible by a given value and output the count.

```
int find_divisible(int value, struct node *head);
```

// To find elements in list 6->1->7->12->NULL that are divisible by 3 run like:

```
./list_find_divisible 3 6 1 7 12
```

2

This is an exercise for you to try

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

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

This is an exercise for you to try

Linked List Hurdle Example (●○○)

`list_even_difference.c`

Given two linked lists, count the number of even numbers in both linked lists and return the difference.

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

This is an exercise for you to try

Array Hurdle Example (●○○)

`array_count_odd.c`

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

```
int count_odd(int size1, int arr1[],  
              int size2, int arr2[]);
```

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

```
struct node *delete_duplicate(struct node *head);
```

Linked List Hurdle Example (●●○)

`list_insert_ordered.c`

Insert a new node into a sorted linked list, maintaining the sorted order.

```
struct node *insert_sorted(struct node *head,  
                           int number);
```

// To insert 3 in list 1->2->5->NULL

```
./list_insert_sorted 3 1 2 5
```

```
[1, 2, 3, 5]
```

This is an exercise for you to try

Linked List Hurdle Example (●●○)

`list_delete_div.c`

delete the first node that is divisible by the given value

```
struct node *delete_div(int value,  
                          struct node *head);
```

// To delete first value divisible by 3 in list 2->9->1->6->NULL

```
./list_insert_sorted 3 2 9 1 6
```

```
[2, 1, 6]
```

This is an exercise for you to try

Array Hurdle Example (●●○)

`array_multiplied.c`

Write a C program that reads integers into an array from terminal until a number is entered which when multiplied by another number previously entered results in 56. E.g.

```
./array_multiplied
```

2

3

28

28 * 2 = 56

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
```

This is an exercise for you to try

Array Hurdle Example (●●○)

`array_max_subarray_sum.c`

Write a C program to find the largest sum of contiguous subarray in an array.

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

This is an exercise for you to try.

Non-hurdle programming questions

Command Line Arguments (●●●)

`common_argument.c`

Write a program to print out the most common command line argument and print out the number of occurrences e.g

```
$ ./common_argument hooray no hooray hooray potato
```

```
Most common argument is hooray and it occurs 3  
times
```

```
$ ./common_argument
```

```
There are no arguments
```

Isogram (●●●)

`isogram.c`

An isogram is a word, in which no letter of the alphabet occurs more than once. Write a C program that reads in strings until Ctrl+D, and checks whether each string is an isogram.

For example:

```
$./isogram
```

```
Hello
```

```
This is not an isogram
```

This is an exercise for you to try

Array Example (●●●)

`substring.c`

Write a C program that reads in words from standard input until Ctrl+D, and checks whether a word read in as a command line argument is a substring in the word. If it appears, print it again. For example:

```
$./substring courage age bloom encourage blooming  
blooming
```

This is an exercise for you to try

Q&A with the tutors

Your chance to hear/ask about some or all of the following

- Exam tips
- What to enrol in next term
- Getting internships and jobs
- Being a tutor



Thanks and Goodluck!

A big thanks to all the amazing staff in the course...

the Admins, the Lecture Moderators, Assignment writers, Forum staff, Revision Session staff, Help Session staff and Tutors!

And of course a huge congratulations and thank you to all of the amazing students who have worked hard throughout the term and answered and asked so many great questions!!!

We hope you have learnt a lot!

Reach Out

Content Related Questions:

[Forum](#)

Admin related Questions email:

cs1511@unsw.edu.au

