COMP1511 PROGRAMMING FUNDAMENTALS

Lecture 17

Exam Information and Starting Revision

LAST WEEK...

- More linked lists
- Intro to Abstract Data Types: Stacks
- Brief look at recursion

TODAY...

- Exam info!
  - Format
  - Preparation
  - Hints and tip
- Starting our revision. Based on poll results:
  - Strings/Char (today)
  - Pointers (tomorrow)
  - Linked Lists (tomorrow)

WHERE IS THE CODE?

LIVE LECTURE CODE CAN BE FOUND HERE:

https://cgi.cse.unsw.edu.au/~cs1511/21T3/live/Week10/

HOUSEKEEPING

MY EXPERIENCE SURVEYS
HTTP://MYEXPERIENCE.UNSW.EDU.AU/

THE EXAM

WHAT IS IN IT?

- Everything that we have learnt so far
  - Simple IF statements and WHILE loops
  - Variables: int, double, char, structs
  - Strings
  - **Arrays**
  - Pointers
  - **Linked Lists**
  - ADTs (Stack)
  - Recursion

THE EXAM

TIME/DATE

- Date: 3rd December
- Time: 1-7pm
  - The exam is six hours long
- The lab this 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
  - Give is the same as in your labs/weekly tests
  - Autotests are run the same way etc
  - Submit as many times as you want - only last submission will be marked
- We will contact you via UNSW EMAIL before and during the exam if we need to.

THE EXAM

EXAM HURDLES

- There's an array hurdle, question 1 or 3
- There's a linked list hurdle, question 2 or 4
- You must earn a mark of 50% or more in at least one array hurdle question
- You must also earn a mark of 50% or more in at least one linked list hurdle question

THE EXAM

EXAM CONDITIONS

- "Open book"
- Exam conditions still apply!!!
  - Do not communicate with anyone about the exam within 24 hours of the exam start time – this is considered plagiarism
  - NO EXTERNAL HELP – you cannnot ask questions online or in discussion groups – we will be monitoring this
  - No discussion of the exam or sharing your code with anyone except for COMP1511 staff

THE EXAM

CONTACTING US

- If you experience any issues during the exam - you can contact us on

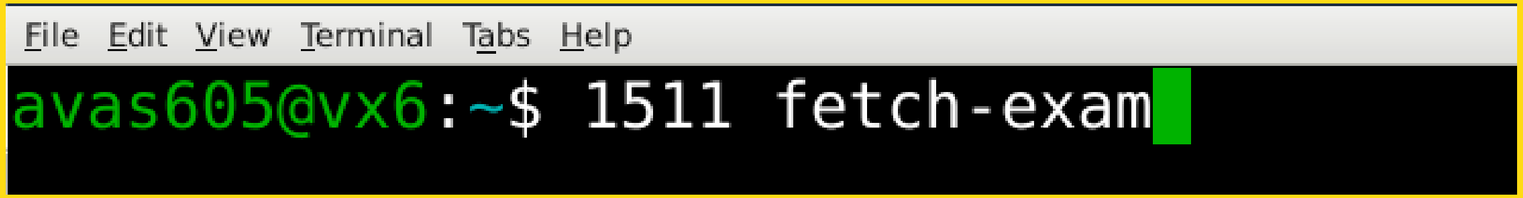
cs1511.exam@cse.unsw.edu.au

- Please use the above email address, we will have a team dedicated to monitoring the exam, and I will be there throughout the whole exam also.

THE EXAM

EMAIL WITH DETAILS

- You will receive an email on your UNSW email address a few days before the exam, which will contain:
  - A link to the exam website (it will not be available until the start time of the exam)
  - We personalise the papers, so your paper will become available at 1pm on the 3rd December
  - The command needed to fetch your exam. You will use a similar command in your practice exam to get you used to the process:



THE EXAM

FIT TO SIT

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.

If, during an online exam you feel unwell to the point that you cannot continue with the exam, you should take the following steps:

- Stop working on the exam and take note of the time
- You must contact us immediately via email at cs1511.exam@cse.unsw.edu.au
- Immediately submit a Special Consideration application saying that you felt ill during the exam and were unable to continue
- You *must* provide a medical certificate dated within 24 hours of the exam, along with screenshots of the conversation you have had with us

Fit to Sit Policy:
https://www.student.unsw.edu.au/exam-rules

THE EXAM

SUPPLEMENTARY EXAM

- If you are granted special consideration based on the Fit to Sit university policy, a supplementary exam will be held between the 10 Jan - 14 Jan 2022. If you think you will need to sit this exam, make sure you are available.
- Fit to Sit Policy: https://www.student.unsw.edu.au/exam-rules

THE EXAM

FORMAT

The format of the exam will be:

- 20 Short Answer Questions (1 mark each)
  - They require you to understand what pieces of code are doing, interpreting code, interpreting diagrams, etc.
  - Examples are in the Week 10 lab practice test
- 8 Practical Questions (10 marks each)
  - Programming questions
  - Similar in style to the questions you did in your labs, weekly tests, revision
  - These are also rated to give you an idea of how difficult each question is.
  - We hope that everyone can attempt and complete the first four questions

THE EXAM

SHORT ANSWER QUESTIONS

Short Answer Questions
Number of questions: 20
Marks: 1 mark each (total of 20 marks)

- These questions will be about whether you understand core coding concepts and the C programming language
- Your answers will either be multiple choice or short answers
- Some are: "What will this code do?"
- Some are: "How does this concept work?"
- Some examples are in the Week 10 Lab

THE EXAM

SHORT ANSWER QUESTIONS - DETAILS

- You will fetch a file called exam_mc.txt to answer them in
- This file is in a special format
- Type your answers within the {{{ triple curly brackets }}}
- Only answers in the {{{ triple curly brackets }}} will be accepted
- Some questions will have validation (so, you might only be able to answer with the letters ‘A’, ‘B’, ‘C’, ‘D’ for example)
- It will have the same structure as the file prac_mc.txt in the week 10 lab.

THE EXAM

PRACTICAL QUESTIONS

Practical Questions
Number of questions: 8
Marks: 10 mark each (total of 80 marks)

- Questions are similar to the Revision Exercises and Labs
- Stages of difficulty from basic to extreme challenge (will be marked in the same rating system as your labs)
- Some will have provided code as frameworks
- Each question will need to be written, compiled and tested
- You will have access to autotests (but they're just tests!)
- Harder questions will have less autotests
- There will be no specific style marking, so you don't need to explain your code in comments

THE EXAM

PRACTICAL QUESTIONS - DETAILS

- When you fetch your exam in the beginning, it will also copy over any starter files you may need for the practical questions.
- Read all the questions before starting
- Start with the easier questions
- Prepare! A couple of minutes thinking and drawing a diagram will clarify how you're going to approach a question
- Use your lab/test practice! Debugging and testing will be important here
- Less questions answered completely is better than more questions partially answered

THE EXAM

PRACTICAL QUESTIONS - QUESTIONS 1 AND 2

- These are: ●○○

- Similar to Practical test question 1 or 2
- Question 1 is an array hurdle
- Question 2 is a linked list hurdle

- Tests your ability to:
  - Create simple C programs
  - Use variables (int, double, char)
  - Use scanf and printf
  - Use IF statements and WHILE loops
  - Use of simple structs
  - Use of arrays of int/double/struct in Q1
  - Use of linked list of ints/doubles (no insertion or removal of nodes) in Q2

THE EXAM

EXAMPLE QUESTION 1

- Loop through an array of structs and gather some kind of information

Given an array of structs, where each struct is:

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

Print out the total of the number of steps taken in a specific direction. So for example, if direction is 'l', find all the structs with direction as 'l' and add them numbers in those structs up. Edit the function
int total(int size, struct direction array[MAX])

THE EXAM

EXAMPLE QUESTION 2

●○○

- Perform some computation on a linked list

Given a linked list, print the difference between the largest value of the list and the smallest value of the list

- Edit the function
  int range(struct node *head)

```
% ./biggest_diff 5 4 3 2 1
4
% ./biggest_diff 3 7 3 7 12
9
```

THE EXAM

PRACTICAL QUESTIONS - QUESTIONS 3 AND 4

- These are: ●●○

- Similar to Practical test question 4
- Question 3 is an array hurdle
- Question 4 is a linked list hurdle

If you have answered Q1 and Q2, this means that you have already passed the hurdles of the exam

- These are harder applications of the hurdles
- You will need to know everything from Q1 and Q2, in addition to:
  - Looping through more than once (maybe)
  - Some insertion/removal of nodes in Q4
  - Testing more difficult conditions and keeping track of more than one thing.

THE EXAM

PRACTICAL QUESTIONS - QUESTIONS 5 AND 6

- These are: ●●●

- Likely using strings (Q5)
- Possibly fgets, fputs, command line arguments etc
- Manipulate linked lists (adding and removing items etc) (q6)
- Potentially use malloc() and free() with structs and pointers
- Might use an Abstract Data Type (Q6)
- Again, more complex combinations, and some questions requiring interesting problem solving

THE EXAM

PRACTICAL QUESTIONS - QUESTIONS 7 AND 8

- These are a combination of ☠ and ●●●

- For those aiming for a HD mark
- Everything taught in the course might be in these questions
- Think " Exercises", even some of the hard ones!
- Will also test your ability to break a problem down into its parts
- This week's lab has a past Question 8 so you can see the difficulty level
- Partial completion of this question will award some marks

THE EXAM

WHAT SHOULD YOU STUDY?

- The basics are important!
- Know how to use both **arrays** and **linked lists**
- Go back and do the Weekly Revision Tests if you haven't already
- The revision exercises on the course webpage are also very useful (this section will be added to the website this week)

- Variables, Structs, 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 successfuly with char arrays, and linked lists
- Stretch Goals: Abstract Data Types and Multi-File Projects / Recursion

THE EXAM

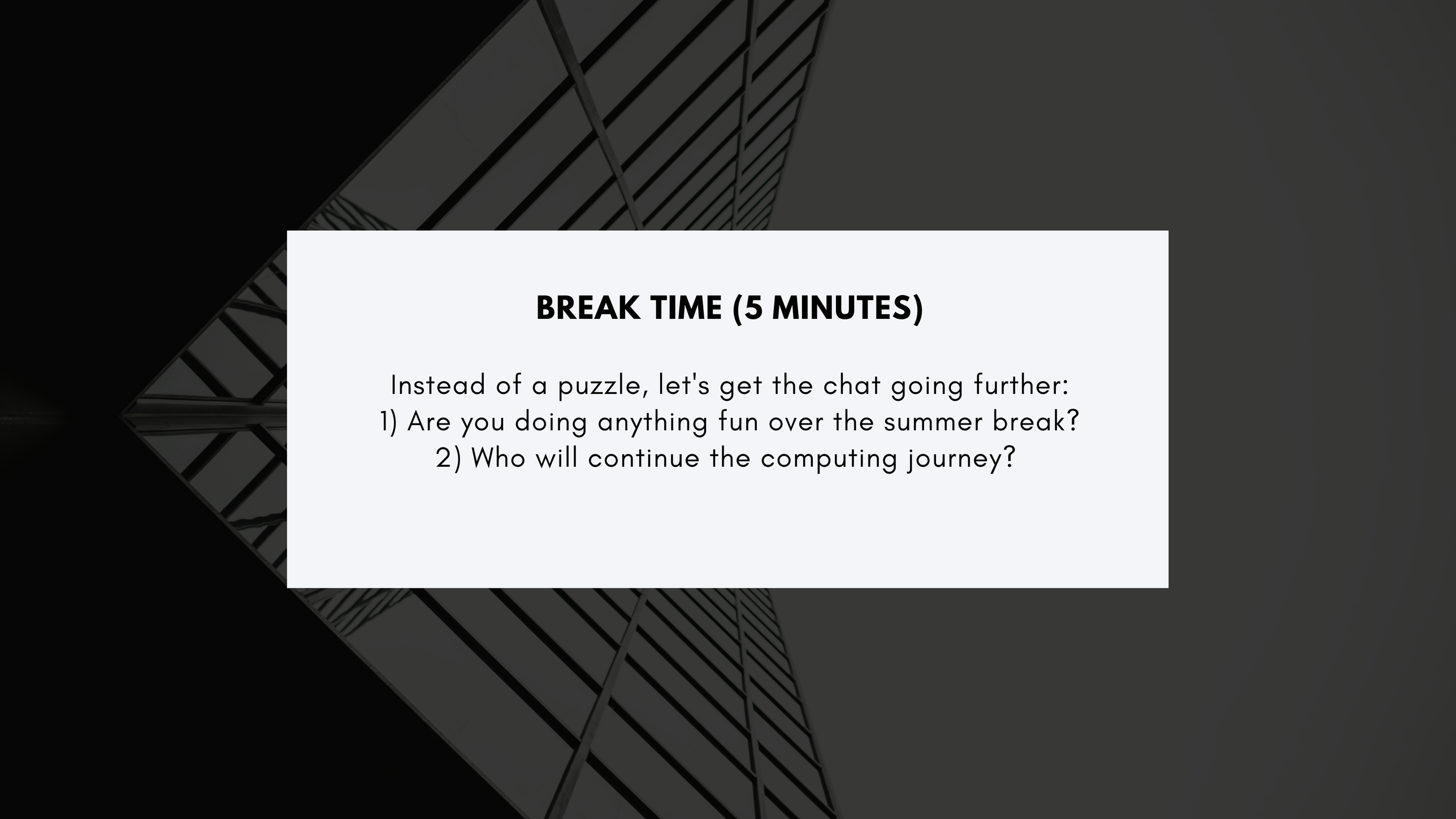
MARKING

- Most of the marking will be automated
- Make sure your input/output format matches the specification
- Answers for hurdles will also be checked by hand
- Marks will be earnt for correct code, not for passing autotests
- Minor errors, like a typo in an otherwise correct solution, will only result in a small loss of marks

THE EXAM

YOU'VE GOT THIS

- Whilst some parts of the exam (the later questions) have been designed to be very challenging, you do not need to complete them to be successful in getting a great mark in this subject
- Make sure that you take regular breaks during the six hours or your brain will turn to mush.
- You do not need to work the whole of the six hours, and you will reach a point where the work you are doing may not be getting you any extra marks - this is OK and it is a good time to finish up, take a good breathe, and celebrate finishing the exam
- Take drink breaks and snack breaks as needed so you can keep yourself fuelled :)
- When you are struggling to understand a question (particularly linked lists) = DRAW DIAGRAMS!
- Go over your lab and weekly test questions for extra practice. There are also revision questions available.



BREAK TIME (5 MINUTES)

Instead of a puzzle, let's get the chat going further:
1) Are you doing anything fun over the summer break?
2) Who will continue the computing journey?

LET'S START OUR REVISION

STRINGS AND CHAR

- Basic String Revision
- Strings in C are arrays of char that end in a NULL terminating character '\0'
- The standard library which has a lot of useful string functions is <string.h>
- You can read a string from terminal by using fgets() function

```
// Eg. string[] = "hello" is:
// string[0] = 'h'
// string[1] = 'e'
// string[2] = 'l'
// string[3] = 'l'
// string[4] = 'o'
// string[5] = '\0'
```

LET'S START OUR REVISION

STRINGS AND CHAR

- Playing with strings

```c
#include <stdio.h>

#define MAX 100

int main (void) {

    char string_array[MAX];
    printf("Enter a word: ");
    //read the string from terminal into string_array[] using fgets()
    fgets(string_array, MAX, stdin);

    // Let's say the word is: hello
    // I can print out the first character:
    printf("The first character of the word is: %c\n", string_array[0]);
    //I can change it up:
    string_array[4] = 'g';
    //This means the word is now "hellg"
    printf("%s\n", string_array);

    //I can truncate it to my heart's content:
    string_array[4] = '\0';
    //This means the word is now "hell"
    printf("%s\n", string_array);

    return 0;
}
```

LET'S START OUR REVISION

STRINGS AND CHAR PROBLEMS

Problem 1: Read in a string from terminal and reverse the characters in the string

Can you now read in from command line and reverse the characters?

Problem 2: Remove all characters in a string except for letters of the alphabet (read the string from standard input)

Can you now read in from command line and do this?

Problem 3: Read in words from the terminal and arrange the words in alphabetical order

Can you now read in from command line and do this?

WHAT DID WE LEARN TODAY?

EXAM

Details

REVISION

Strings/Chars

string.c

reverse_string.c

remove_char.c

alphabetical.c

ANY QUESTIONS?

DON'T FORGET YOU CAN ALWAYS EMAIL US ON CS1511@CSE.UNSW.EDU.AU FOR ANY ADMIN QUESTIONS

PLEASE ASK IN THE FORUM FOR CONTENT RELATED QUESTIONS