Welcome!

COMP1511 18s1

Programming Fundamentals

COMP1511 18s1

— Lecture 20 —

Illegal C + Memory

Andrew Bennett

`<andrew.bennett@unsw.edu.au>`

Overview

after this lecture, you should be able to…

understand the **memory layout** of a C program

understand some of the **security implications** of this

have more of an understanding about **illegal C**

identify and **prevent** basic vulnerabilities in C code

(… and more?)

(**note**: you shouldn't be able to do all of these immediately after watching this lecture. however, this lecture should (hopefully!) give you the foundations you need to develop these skills. remember: programming is like learning any other language, it takes consistent and regular practice.)

Admin

Don’t panic!

assignment 3 out now!

week 11’s tute/lab help you get started

week 11

lab due **tonight**

weekly test due **friday**

don’t forget about **help sessions**!

see course website for details

Questions?

https://echo360.org.au/

note: you may need to go via **Moodle**

https://moodle.telt.unsw.edu.au

(let me know if you can/can't access it!)

What topics are you confused about? What questions do you have?

What is your response?

let’s talk about: **memory in C**

Memory Layout in C

we’ve talked about this a bit already

everything is in memory

(including your code!)

(diagram: week 6 slides)

Some Terminology

stack: function memory

heap: dynamic memory (e.g. from malloc)

Stack Frames

every function has its own memory

we call this a **stack frame**

…

it stores all of the local variables, etc
but also other necessary information:

where the stack frame **starts** / **ends**

where to go **in the code** when this function **returns**
(the **return address**)

Implications

what happens if these are incorrect?

Illegal Array Access

(demo: *interactive array tool*)

But wait, it gets better…

(demo: *popping a calc*)

Smashing The Stack for Fun And Profit

Smashing The Stack For Fun And Profit

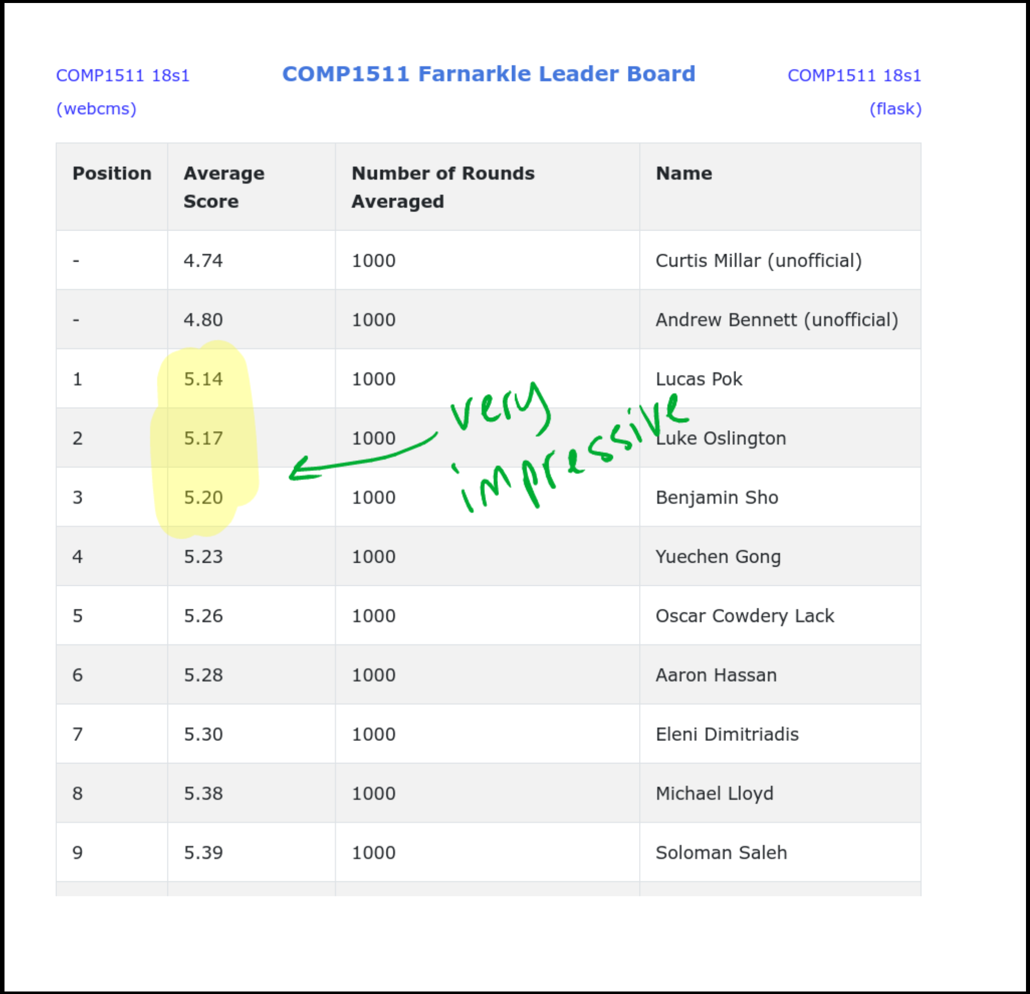
remember **farnarkling**?

The Farnarkle AI Leaderboard

link

backup link

But that's not possible…

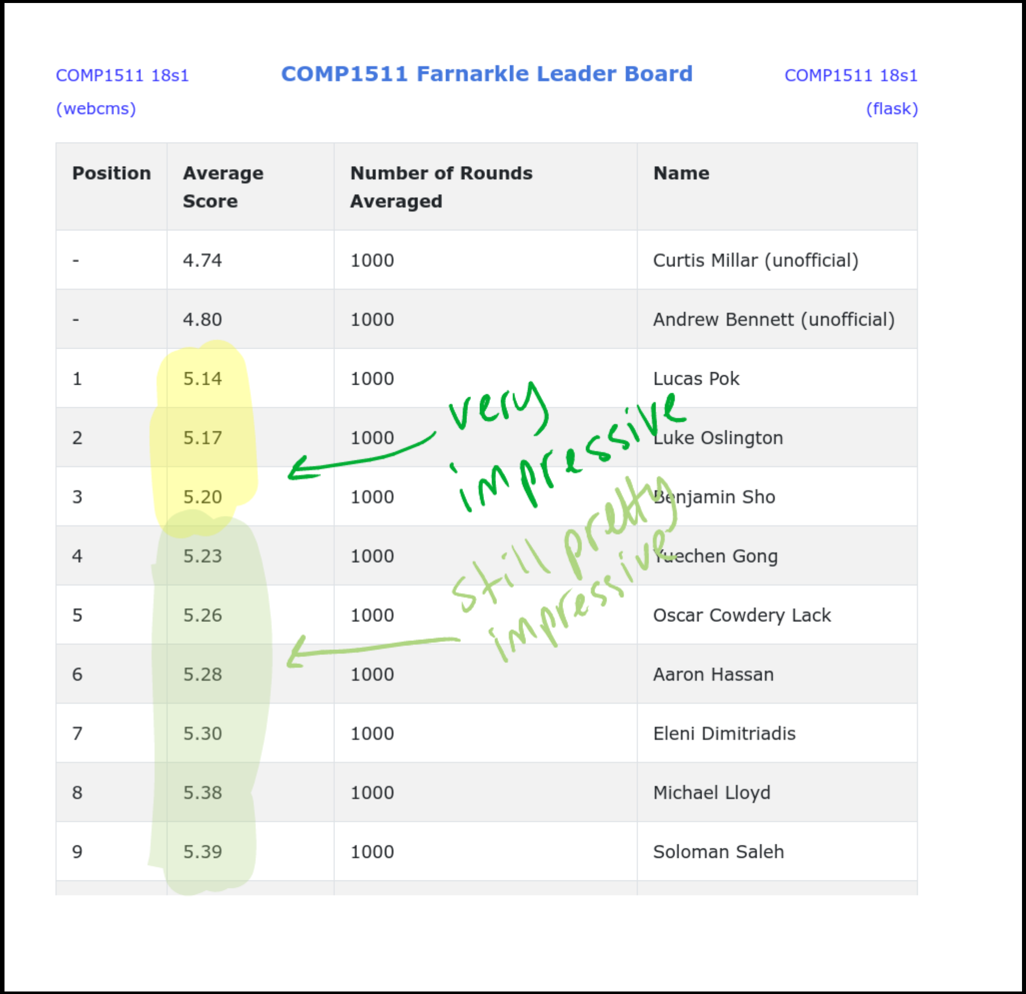


COMP1511 18s1 (webcms) | **COMP1511 Farnarkle Leader Board** | COMP1511 18s1 (flask)

Position	Average Score	Number of Rounds Averaged	Name
-	4.74	1000	Curtis Millar (unofficial)
-	4.80	1000	Andrew Bennett (unofficial)
1	5.14	1000	Lucas Pok
2	5.17	1000	Luke Oslington
3	5.20	1000	Benjamin Sho
4	5.23	1000	Yuechen Gong
5	5.26	1000	Oscar Cowdery Lack
6	5.28	1000	Aaron Hassan
7	5.30	1000	Eleni Dimitriadis
8	5.38	1000	Michael Lloyd
9	5.39	1000	Soloman Saleh

very impressive

But that's not possible…



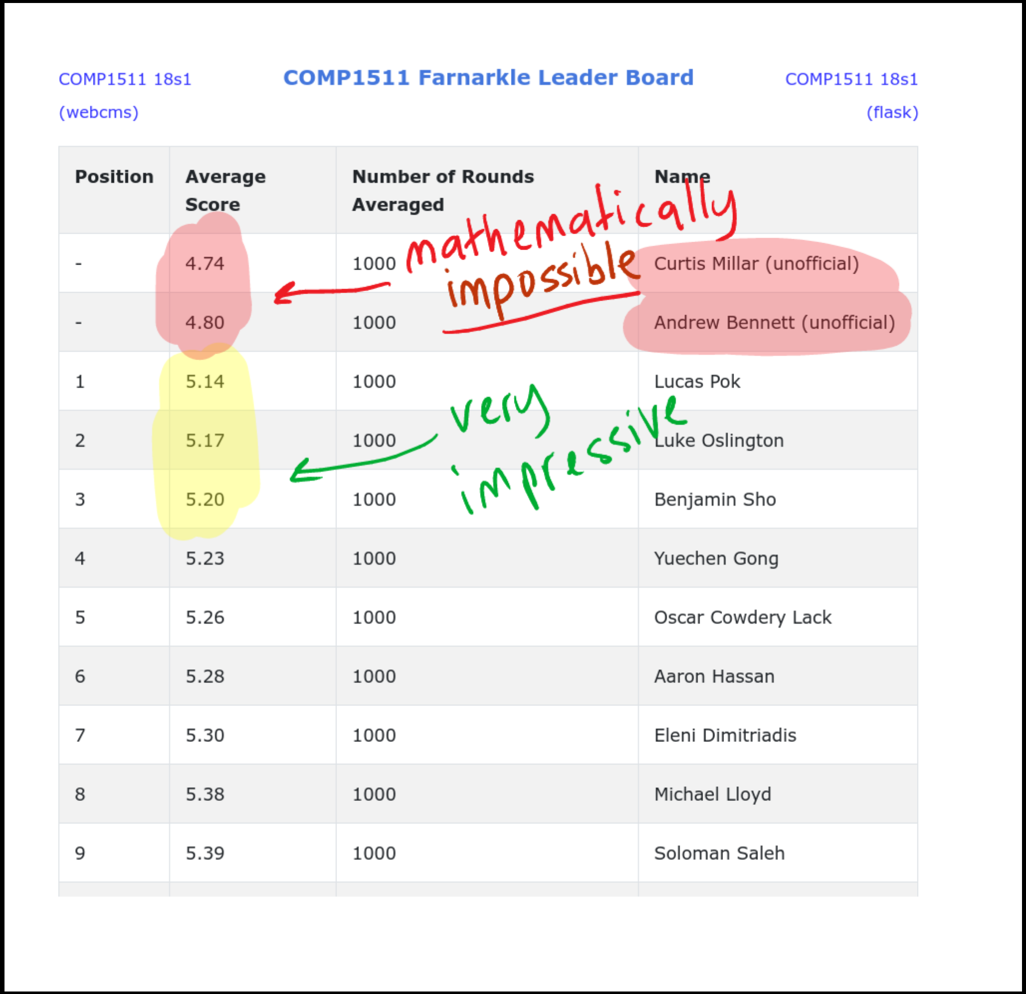
COMP1511 18s1 (webcms) | **COMP1511 Farnarkle Leader Board** | COMP1511 18s1 (flask)

Position	Average Score	Number of Rounds Averaged	Name
-	4.74	1000	Curtis Millar (unofficial)
-	4.80	1000	Andrew Bennett (unofficial)
1	5.14	1000	Lucas Pok
2	5.17	1000	Luke Oslington
3	5.20	1000	Benjamin Sho
4	5.23	1000	Yuechen Gong
5	5.26	1000	Oscar Cowdery Lack
6	5.28	1000	Aaron Hassan
7	5.30	1000	Eleni Dimitriadis
8	5.38	1000	Michael Lloyd
9	5.39	1000	Soloman Saleh

very impressive

still pretty impressive

But that’s not possible…



COMP1511 18s1 (webcms)

COMP1511 Farnarkle Leader Board

COMP1511 18s1 (flask)

Position	Average Score	Number of Rounds Averaged	Name
-	4.74	1000	Curtis Millar (unofficial)
-	4.80	1000	Andrew Bennett (unofficial)
1	5.14	1000	Lucas Pok
2	5.17	1000	Luke Oslington
3	5.20	1000	Benjamin Sho
4	5.23	1000	Yuechen Gong
5	5.26	1000	Oscar Cowdery Lack
6	5.28	1000	Aaron Hassan
7	5.30	1000	Eleni Dimitriadis
8	5.38	1000	Michael Lloyd
9	5.39	1000	Soloman Saleh

mathematically impossible

very impressive

What??

(demo: *andrewb's farnarkle AI*)

What??

links:

terminal output

explanation

(demo: *andrewb's farnarkle AI*)

questions?