Week 7 - Debugging

Daniel Alyoshin

CSCA48 - TUT002

February 27, 2026

Content Covered so Far

- C programming
- Memory model
- Arrays and strings
- Pointers
- CDTs & ADTs
- Linked lists
- Testing
- BSTs

Recall - Testing Steps

1. Test your functions **before** using them (i.e. verify the function works as expected before calling it elsewhere in the code)

2. By the time your functions are all implemented most of your testing should be complete and the program is likely mostly correct

3. Finally test the program as a whole by supplying a variety of **valid inputs** to ensure the program behaves as expected and **invalid inputs** to ensure the program handles them without breaking

Our Two Flavours of Debugging

1. `printf()` statements
    - Easy to use and quick to set up
    - Does not require running any other programs

Our Two Flavours of Debugging

1. `printf()` statements
   - Easy to use and quick to set up
   - Does not require running any other programs
2. GDB - GNU Debugger
   - Very powerful *if* you learn how to use it
   - Works in situations where `printf()` is unreliable

Debugging Strategy

1. Identify the problem
   - Test is providing an unexpected result
   - Program is crashing
   - Infinite loops
   - etc.

Debugging Strategy

1. Identify the problem
   - Test is providing an unexpected result
   - Program is crashing
   - Infinite loops
   - etc.
2. Pick a debugging strategy
   - GDB
   - `printf()`

Your Task

csca48.alyoshin.dev

Reminders

- Each student to submit their own `.c` file which can be the same as that submitted by members of their group.
- Ensure your file is named **exactly** as instructed on Quercus.
- You are **not** supposed to keep working on this, you will not be marked on correctness so as long as you submit work that shows you were engaged during this tutorial you will get full marks.