15-213 Recitation: Data Lab

Your TAs
Sep 9, 2019

Agenda

- Introduction
- Course Details
- Data Lab
  - Getting started
  - Running your code
  - ANSI C
  - Reminders
- Floating Point

Introduction

- Welcome to 15-213/18-213/15-513!
- Recitations are for…
  - Reviewing lectures
  - Discussing homework problems
  - Interactively exploring concepts
  - Previewing future lecture material
- Please, **please** ask questions!

Course Details

- How do I get help?
  - Course website: http://cs.cmu.edu/~213
  - Office hours: **5:30-9:30PM** from Sun-Fri in GHC 5207(Monday: 6:30-9:30)
  - Piazza
  - *Definitely* consult the course textbook
  - **Carefully read the assignment writeups!**
- All labs are submitted on Autolab.
- All labs should be worked on using the **shark machines.**

Data Lab: Getting Started

- Clone the lab (`github.com/cmu15213f19/datalab-f19-<id>`)
  - `cd <my course directory>`
  - `git clone`
- Upload `bits.c` file to Autolab for submission

Data Lab: Running your code

- `dlc`: a modified C compiler that interprets *ANSI C* **only**
- `btest`: runs your solutions on random values
- `bddcheck`: exhaustively tests your solutions
  - Checks all values, formally verifying the solution
- `driver.pl`: Runs both dlc and bddcheck
  - Exactly matches Autolab's grading script
  - You will likely only need to submit once
- For more information, **read the writeup**
  - Available under assignment page as "**View writeup**"
  - **Read it. Read the writeup... please.**

Data Lab: What is ANSI C?

Within two braces, all *declarations* must go before any *expressions*.

This is *not* ANSI C.

```
unsigned int foo(unsigned int x)
{
    x = x * 2;
    int y = 5;

    if (x > 5) {
        x = x * 3;
        int z = 4;
        x = x * z;
    }

    return x * y;
}
```

Data Lab: What is ANSI C?

This is ANSI C.

```
unsigned int foo(unsigned int x)
{
    int y = 5;
    x = x * 2;

    if (x > 5) {
        int z = 4;
        x = x * 3;
        x = x * z;
    }

    return x * y;
}
```

This is *not* ANSI C.

```
unsigned int foo(unsigned int x)
{
    x = x * 2;
    int y = 5;

    if (x > 5) {
        x = x * 3;
        int z = 4;
        x = x * z;
    }

    return x * y;
}
```

Data Lab: Reminders

- Casting between **int** and **long** is ok, in either direction
- Be aware of operations and their types!
  - ! returns an **int** *even if the argument is a long*
- Good idea to append “L” suffix to every integer constant
  - `(1L << 63)` is not the same as `1 << 63`
  - `(!x << 63)` is not the same as `((long)!x) << 63`

Form Groups of 3 - 4

- Series of exercises
  - Operators
  - Floating point
  - Puzzle

Floating Point: Rounding

1.BGRXXX

In the below examples, imagine the underlined part as a fraction.

- Guard Bit: the least significant bit of the resulting number
- Round Bit: the first bit removed from rounding
- Sticky Bits: all bits after the round bit, OR'd together

Examples of rounding cases, including rounding to nearest even number

- 1.10 ¦11: More than ½, round up: 1.11
- 1.10 ¦10: Equal to ½, round down *to even*: 1.10
- 1.01 ¦01: Less than ½, round down: 1.01
- 1.01 ¦10: Equal to ½, round up *to even:* 1.10
- 1.01 ¦00: Equal to 0, do nothing: 1.01
- 1.00 00: Equal to 0, do nothing: 1.00

All other cases involve either rounding up or down - *try them*!

Questions?

- Remember, data lab is due this Thursday!
  - You really should have started already!
- Read the lab writeup.
  - **Read the lab writeup.**
    - ***Read the lab writeup.***
      - ***<u>Read the lab writeup.</u>***
        - **<u>Please. :)</u>**