15-213 Recitation: How to Succeed in x13 + Datalab

Your TAs

Friday, August 29th

Agenda

- Introduction
- Course Details
- Office hours
- x13 Advice from TAs
- Datalab

Introduction

- Welcome to 15-213/15-513/14-513!
- 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
  - Ed
  - *Definitely* consult the course textbook
  - **Carefully read the assignment writeups!**
- All labs are submitted on Autolab
- All labs should be worked on using our **shark machines**

Office hours

- Office Hours start Tuesday, September 2!
- Queue link: https://213ohq.com/ohq/
- Please locate the TA in the specified location!
- Semester's OH schedule (subject to change)
  - Will be added in a pinned post in Ed.

OH Etiquette

- Office hours are for getting ideas on how to debug or better approach your homework!
- Please try to narrow down your problem area as much as possible to help TAs help you!
- **Write a description!**
  - If you don't have a description, you may be frozen/removed from the queue. Make sure to use the tags!

OH Etiquette

- TAs will only spend 10 minutes per student and then you can rejoin the queue
- We will close the queue early so everyone can be helped, so please keep this in mind and come early!

Advice from your TAs :)

What is success in x13?

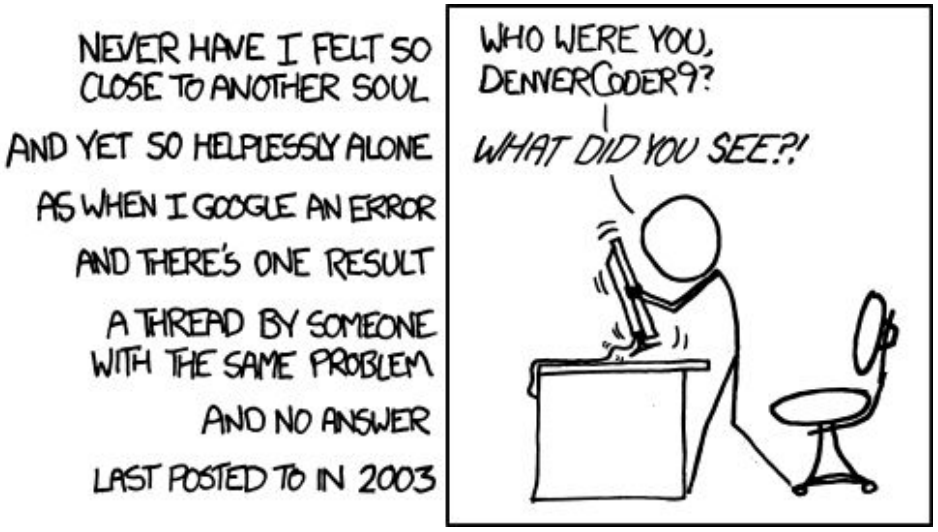
- Some of you (probably most) see success as getting an **A**
- … BUT you can still succeed without getting an **A** - in fact, true success in x13 is **learning the material!**
- This can be difficult because we cover a wide range of different topics, many of which will probably be new to you (and that's okay!!)

How do I learn the material?

- Engage with the topics in lecture
- Read the textbook
- Learn the material before having to apply it
- Ask questions!

What if I'm still confused?

- It's okay to be confused! These topics can be difficult and take time to truly understand.
- (Some) online resources are okay to use, but a general google search probably won't give you helpful results…



Source: https://xkcd.com/979/

I need help with a concept

- Read the textbook
- Come to OH and ask your TAs :)
- Come to Prof. OH (they don't bite, we promise)
- Ask on Ed
- Ask your recitation TAs to cover the topic again
  - *cough cough wink wink*

I need help with a problem/bug

- Step away and come back after a small break
- Try to solve on your own first! (debugging for an hour is not that long)
  - Generally, give yourself a day to mull over the problem (your brain will continue to think about it while you do other tasks!)

I need help with a problem/bug

- If it is a general bug, try some *reputable* sites to find similar problems (see next slide)
- Come to OH!
- Post on Ed!
- Rubber duck method

Good online resources

- https://itsfoss.com/linux-man-page-guide/
- https://man7.org/linux/man-pages/
- https://en.cppreference.com/w/c
  - Make sure to use the C (not C++) version!
- https://www.cs.virginia.edu/~evans/cs216/guides/x86.html
- https://beej.us/guide/
- http://www.stackgrowsdown.com/

Other helpful advice!

- Learn GDB early ***before*** you have to rely on it to debug
- Read the writeups (yes, there can be, and *will* be, relevant material on all 20 pages of a writeup)
- Don't start labs late
- Save some grace days for **`malloclab`!**
  - (~40 hours is average)

Other helpful advice!

- You don't have to pass every test case of every assignment
- Be comfortable with the command line (it's not that scary!)
- Be comfortable with different editors
  - I'm looking at you VScode 👀
  - But Vim is also cool :)
- If you need help, ask! We are here to help you!

Introduction to `Datalab`

Datalab: Getting Started

- Download the handout from autolab
- Method 1:
  - `scp <path to datalab.tar> <andrewid>@shark.ics.cs.cmu.edu:<my course directory>`
  - `ssh <andrewid>@shark.ics.cs.cmu.edu`
  - `cd to the datalab.tar file`
  - `tar -xf datalab.tar`

`Datalab`: Getting Started

- Download the handout from autolab
- Method 2:
  - `autolab download 15213-f25:datalab`
  - `cd into the datalab folder`
  - `tar -xf datalab-handout.tar`

`Datalab`: Getting Started

- Upload `bits.c` file to Autolab for submission
  - `make submit`

`Datalab`: Running your code

- **`dlc`**: a modified C compiler
- **`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 autolab as "**View writeup**"
  - **Read the writeup please!**

Datalab: 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`**

`Datalab` Activity

Form groups of 3 - 4

- We'll be working on a series of exercises
  - Operators and Puzzles
- There's a handout on the website - download this!

Divide and Conquer (Bit Count)

- **Objective**: Count how many bits are set in a number.
- For each challenge, you can use any operator allowed in the integer problems in datalab.
- Let's start simple: for a 1-bit number, we just return the value itself!

```
int bitCount1bit(int x) {return x;}
```

Divide and Conquer (Bit Count)

- What if there are two bits in the input? (4 ops max)

```
int bitCount2bit(int x) {
   int bit1 = _________ & _________;
   int bit2 = _________ & _________;
   return _________  + bit2;
}
```

Divide and Conquer (Bit Count)

- What if there are four bits in the input? (8 ops max)

```
int bitCount4bit(int x) {
   int mask = ________________;
   int halfSum = ____________________________;
   int mask2 = ________________;
   return ___________  + ___________;
}
```

Divide and Conquer (Bit Count)

- What was the partition strategy we employed in the previous example?
- How did we combine the results of each partition?

Let's apply these ideas to a 8-bit input now!

Divide and Conquer (Bit Count)

- What if there are eight bits in the input? (12 ops max)

```
int bitCount8bit(int x) {
   int mask = ________________;
   int quarterSum = ____________________________;
   int mask2 = ________________;
   int halfSum = ____________________________;
   int mask3 = ________________;
   return ________________  + ________________;
}
```

Reminders

- **`C-programming lab`** is due Sep 2 (Tuesday)
- **`datalab`** is due Sep 9
  - We recommend you start just a BIT early!
  - Read the lab writeup!