RECITATION 6

INFO

- Jon Rutkauskas
- Recitation: Tue 12-12:50
- Office Hours: Tue 11-11:50
  Thur 11-12:50
  SENSQ 5806
  (additional hours by appointment if needed)
- On discord: @jrutkauskas
- By email: jsr68@pitt.edu
- Github:
  https://github.com/jrutkauskas/spring2019-449-rec
- Ask me any questions you have!!!

EXAM 1

- Exam 1 is next week! I won't see you again before it (outside of OH), so we'll spend some time today covering questions you may have, and I can also answer questions about Project 2

WARMUP POLLS

QUICK FREEBIE FOR PROJECT 2

- When your code crashes in the delightfully colorful `bigdriver`, it can sometimes leave your terminal printing in strange colors
- I have a quick C program that will reset your terminal to normal colors if this is a problem for you.
- Posted on Github under Recitation 6 as `resetcolors.c`

CODE

```c
#include <stdio.h>
int main()
{
 printf("\e[0m");
 return 0;
}
```

As a (completely unnecessary) side note, these ANSI Escape Sequences can be pretty cool and **fun**

SO PROJECT 2…

- Due by midnight Wednesday. Please make sure to get working on it.
- This is a debugging-heavy project and it's hard to judge how much time you need.
- Please don't wait until the last minute and start panicking at 11:20pm (though we've all been there).
- I won't have office hours until Thursday but I'm available on Discord or by email!

PROJECT NOTES AND SUGGESTIONS

- Make sure you are using the Makefile and that it compiles your code correctly (it's part of your grade)
- Jarrett suggests this but I want to **double** suggest it: Please make small functions to help out and help organize your code, *even if you only call them in one place.*
  - E.g., functions for removing a block from the linked list, splitting a block, expanding the heap, coalescing blocks, coalescing with the block to the left, to the right… etc. You get the picture
  - Makes it so much easier to test, debug, and keeps your code organized
- If you're really stuck with where something is going wrong, consider drawing some test cases out on paper and compare with what's happening in gdb
- Actually write your own tests in `mydriver.c`

LINKED CHAIN/LIST HELP

- By the way, if you still have any confusion about how to properly deal with linked chains/lists, here's a link to some lecture notes detailing the basic operations and some (java) code for implementation
- https://www.cs.cmu.edu/~adamchik/15-121/lectures/Linked%20Lists/linked%20lists.html

QUICK REVIEW QUESTION: WHICH OF THESE SHOULD **NOT** GO IN A HEADER FILE

public function prototypes

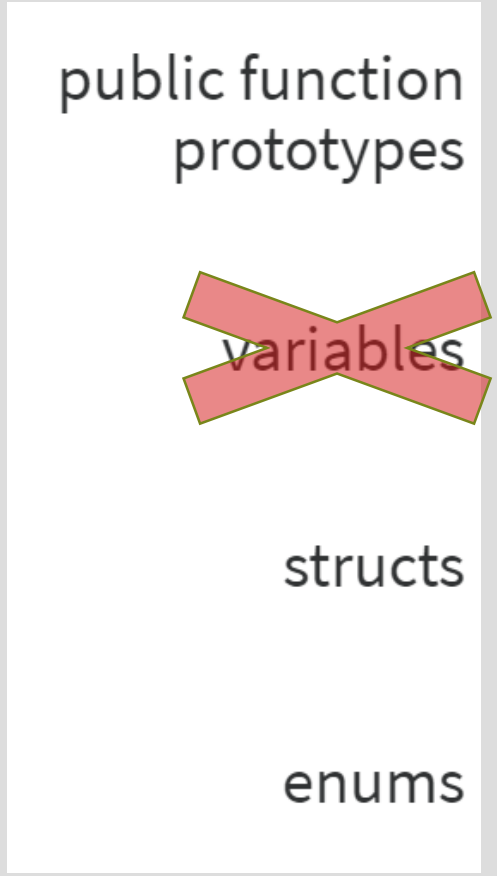
variables

structs

enums

QUICK REVIEW QUESTION: WHICH OF THESE SHOULD **NOT** GO IN A HEADER FILE

- We shouldn't put any variables, function definitions, or anything we want to keep private in headers
- Headers are kind of similar (in form, not purpose) to interfaces in Java (Remember 445?)
  - No code, no data, just definitions Though, you can have `structs`, `enums`, and `#defines` in them



QUICK REVIEW QUESTION: WHAT DOES THE PREPROCESSOR DO?

Checks your data types align

Links external library code

Produces machine code

Substitutes the text of your .c and .h files

QUICK REVIEW QUESTION: WHAT DOES THE PREPROCESSOR DO?

Checks your data types align

Links external library code

Produces machine code

Substitutes the text of your .c and .h files

QUICK REVIEW QUESTION: WHAT DOES THE PREPROCESSOR DO?

- The preprocessor only does text substitution
- Data type checking is done by the compiler
- The linker links the external library CODE
  - Though the preprocessor will #include the header files for external libraries
- The compiler produces machine code

Checks your data types align

Links external library code

Produces machine code

Substitutes the text of your .c and .h files

EXAM REVIEW TIME

- Ask me literally anything about course material – Old-fashioned style
- Remember I was sitting in these very seats 2 years ago.
- If you don't care about the exam (*gasp!*) then you can work on Project 2. I'll be around after to answer questions with that.
- Other good practice: I post all the questions from this class on Github; you can go back through and review them.