Program submissions: Up to 2 extra days.
Notes: Turn in notes before Monday of the next week using git. PDF, plaintext, or Word. These notes can be used on tests the following week.
Coding help: Add a comment in your code acknowledging who helped you.
Schedule: Programs are assigned Saturday night. Design documents are due Thursday night (LaTex with diagrams preferred, Word discouraged). Programs are due on Sunday night.
Late Policy: If you turn in programs late, you lose 25% each day afterwards.
MSI Tutoring: lss.ucsc.edu/tutor-trac/tutor-trac-20-21.html

Documentation:
DESIGN.pdf
- Describes the design of your program and answers pre-lab questions.
- Describes the algorithms and design decisions you have made.
- Describes the problem you are solving, inputs, and expected outputs.
- Includes pictures, diagrams, and pseudocode.

WRITEUP.pdf
- Contains your analysis of running your program.
- For example, this file would contain the results of your program, like a science experiment.

C ancestry
- Came from language called B
- Created in 1972, 2011 C11 was made, 2018 C18 was made

#include <stdio.h> → angled bracelets mean Standard I/O package
Return 0 = success
Int main(void) = no arguments passed
printf("%3.0f") = print 3 numbers with 0 after decimal point
printf("%6.1f") = print 6 numbers with 1 after decimal point
Scoping
- A scope is defined by curly braces (for loops, while loops)
- Internal variables have higher priority over higher scope variables.

String (*char)
- Should always end with a null byte ('\0')
-

./pig < pig.c - passes pig.c as a text file to ./pig, so functions such as getchar() read text from pig.c.

Compilation process of a C program
- 1. Preprocessor- processes all directives starting with a #, like include or DEFINE
  - Comments are removed
  - Creates an intermediary file, hello.i
  - #define is a copy-paste operation. Every instance of the defined phrase is copy-pasted.
- 2. The compiler- Converts hello.i into assembly code

        - Lexical phase- Groups key words, punctuation, variables together, throw away whitespace
        - Syntax analysis phase- Parses things, checks if things are syntactically correct.
            - Creates a tree of the tokens created from the lexical phase. Nodes of this tree might be “if”, “statement”, “>”, while roots might be “return”, “True”, “3”
- 3. The Assembler- Converts assembly language instructions into binary
    - Two pass- maps symbols first, then traverses through assembly to generate binary code.
- 4. The Linker- Links hello.c to other objects and libraries.
- 5. The Loader- Lives in the operating system. Allocates space in memory, resolves symbols between references, fix all dependent locations to point to allocated memory.
- 6. The Memory
    - Bottom layer- The text- your code
    - 4th layer- initialized data
    - 3rd layer- uninitialized data
    - 2nd layer- heap- grows upwards
    - 1st layer- stack- grows downwards

Compiler vs Interpreter

- Compiler translates a programming language, goes through a sequence of translations, then outputs an executable.
    - Translates entire program at once.
- Interpreter - Directly executes code without needed to compile.
    - Translates program one line at a time.
    - Hundreds of times slower.

GCC vs cc vs clang

- GCC- gnu C compiler, default on linux
- CC- Unix/Linux environment variable that points to default compiler
- Clang- Default on Mac

Makefile

- Automates building executables from source code, so you don’t have to compile every time.
- Makefile contains rules on how the Make should run.
-

```
Echo -n "Type something: " && cat -> /asgn1/hello.txt --- adds text to file
Cat <- hello.txt --- outputs text of file
Man 3 fprintf --- gives documentation for a function.
:set listchars=tab:\\*, --- in .vimrc
./cat 2> err.log --- sends any fprintf(stderror, ...) to err.log
./cat 1> out.log --- sends any fprintf(stdout, ...) to out.log
Char msg[] = {'h', 'e', 'l' 'l', 'o', 0}; --- OR --- char msg[] = "hello";
./a.out --- "." represents current directory. Something like ".."
```

```
represents above directory.
Clang-format: clang-format -i -style=file hello.c
```

Boolean Algebra

- George boole laid the foundation of boolean algebra.
    - AND, conjunction, && in C
    - OR, disjunction, || in C
    - NOT, negation, ! in C
- A xor B = (A or B) and !(A and B)
- A xor B = (A and !B) or (!A and B)
- A xor A = 0
- A xor 0 = A
- A xor 1 = !A

Functions

- All subordinate to main()
- Hide the implementation of abstractions
- return_type fuction_name (parameters) {
    - // declarations, assignment statements
- }
- return_type is either void, any object (except array; struct is discouraged), or pointer
- function_name - same rules as variable naming- use_snake_case for function names
- Declarations- locally known variables
- Assignment statements- sets the value of a variable

Call-by-name vs. call-by-reference

- Call-by-name: creates a local instance of the variable.
- Call-by-value: used by C, except for arrays (because they are pointers).

Formal parameters

- Name of the parameter as it is used inside the function.

Actual parameters

- Name of the value passed to the function
- The value can be copied to the formal parameter
- Call by reference is done by passing a pointer.
    - Put an ampersand in front of the variable.
    - int a = 3, b = 5;
    - swap(&a, &b);
    - void swap(int *a, int *b){
        - int temp = *a;
        - *a = *b;
        - *b = temp;
        - return;
    - }

Function prototypes

- Define the function before you use it.

- return_type function_name (parameters);

Preprocessor Directive

- Before compilation, C source files are processed by a preprocessor.
- Preprocessor is a macro processor to transform programs before compilation through text replacement.
- Used to include functions defined in other libraries.

Header files

- #ifndef __STACK_H_ // Prevents __STACK_H_ from ever being defined more than once. Equivalent to #pragma once
- #define __STACK_H_
- typedef uint32_t Item;
- typedef struct{
    - uint32_t size;
    - uint32_t top;
    - Item *entries;
- } Stack;
- Stack *stack_create();
- void stack_delete(Stack **s);
- bool stack_push(Stack *s, Item i);
- bool stack_pop(Stack *s, Item *i);
- #endif

- stdio.h for input/output
-

Static

- Can be declared inside or outside a function
- Declared inside a function means the value of the variable persists across function calls.
- static inline bool - A static function can be defined in a header file. Inline means to the compiler to not call a function, but just replace text.

Numbers

- Numbers exist independently of us; we just choose to represent them as 1, 2, 3, …
- For example, Arabics write numbers differently, and Babylonians used base 60.
- int8_t or uint8_t: 8 bits; int16_t or uint16_t: 16 bits; int32_t or uint32_t: 32 bits, int64_t or uint64_t: 64 bits

Decimals

- Floats represent numbers as multiples of negative powers of 2.
- Single precision float: 1 sign bit, 8 exponent bits, 23 fraction bits.
- Double precision float: 1 sign bit, 11 exponent bits, 52 fraction bits.
- Intel extended precision: 1 sign bit, 15 exponent bit, 1 integer bit, 63 fraction bits.
- Quad precision float: 1 sign bit, 18 exponent bit, 115 fraction bits.

Little endian: Low address holds the least significant byte.

Mixed bytes:

- When you have an expression of mixed types, C will promote the lower type to the higher type.

- Lower means that values of that type are a subset of the values of the higher type.
    - Integers are promoted to bigger integers, or floating point numbers.
- Word size- 32 bit computers have a native word size of 32, and 64 bit computers have 64 size words.

Numerical Computation

- Multiplication is shift and addition, division is shift and subtraction, addition is little more than XOR
- Taylor series- f(x) can be approximated by infinite differentials of f(x).
    - Often, the series converges too slowly to be useful.
    - It might be useful for e^x, sin(x), cos(x), but not useful for sqrt(x).