CSE13S notes
Week of 1/16/23

Functions in C

- Not the same as math functions, which are statements of truth
- In math, functions f maps domain (a set) into the range (also a set)
- However, in C (and other languages) a function returns a value
    - Some languages might call a function a “subroutine” if it doesn’t return a value; this is done in C through returning void
- A function is a block of code that (hopefully) performs a certain, logically consistent task
- Functions are defined exactly once, and must be declared before use
    - Programs can declare and call a function as many times as desired
    - Defining a function: code within a function; declaring a function: declaring it exists
- main()
    - Special func in C
    - Runs when the program starts
    - All other functions are logically subordinate to main()
- Functions should:
    - Define abstractions that are consistent and make sense logically
    - Give names to those sequences of code
    - Hide the implementation of the code
- We can use them to
    - Refactor repeated sequences of code
    - Simplify the code to aid understanding
- Functions should never be:
    - Arbitrary sequences of statements

Function definition

```
Return_type function_name(parameters) {
     // declarations, assignment statements
}
```

- Function head:
    - return_type : defines the type of functions return value
        - Return type may be void or any objects type (except array type)
    - function_name() : functions name
    - parameters : contained in
    - Check slides
- In C, functions return a value

- Most return types are scalar (except structs, but long advises not to return structs)
- QUIZ: which of the following can't be returned in C? Answer is arrays!

Naming a function: functions have the same naming rules as variables. They can't start with a number of punctuation other than _ underscore or $ dollar sign. Furthermore, they can't use the same name as another function (C does not have nested functions).

- For this class, we will be using snake case: my_function_name

Parameters (arguments)

- In math:
  - Function f(x) = 2x+x. If we write f(2), then we substitute 2 for x : 2(2) + 2
  - This is call-by-name
- Most programming languages (in the modern era) do not use call-by-name
- Most programming languages use call-by-value, call-by-reference or both
  - C uses call-by-value, except for arrays, and only because of their relation to pointers
- Formal parameters
  - This is the name of the parameter as it is used inside of the function body
  - Check slides
- Call-by-reference
  - The references of the arguments are passed in meaning any changes to variable within function actually changes the reference
- Not on pointers:
  - C does not have true call-by-reference, so we use pointers
  - Addresses, instead of values, are passed as arguments to point to
  - Read the pointer chapter in the C book

C preprocessor : #include, #define

- Before compilation, C sources files are processed by a preprocessor
- #define is a preprocessor defines a macro for the program
  - The C preprocessor performs all text replacement for defined macros prior to compilation
- Condition directives:
  - A set of preprocessor directives that uses conditional statements to include code selectively

Header files

- Should only contain things that are shared between source files:
  - Function declarations
  - Macro definitions

    - Data structure and enumeration definitions
    - Global variables
- Standard header files
    - stdio.h for i/o
    - inttypes.h for fixed width integer types
    - time.h for date/time utilities
    - stdbool.h for boolean types
    - ctype.h for functions to determine type contained in character data
    - math.h common mathematical functions
- Extern
    - Extends visibility of variables and functions such that they can called by any program file, provided the declaration is known
    - Functions
- Static
    - Talked about next time
- Recursion
    - Function can call themselves and other functions that call itself

1/20/23
On the nature of numbers

- Numbers are not material things
    - 1,2,3,4,5,... are all representation of numbers
    - Numbers can be represented differently (base 2, 10, 16)
    - There are many ways to write the same number
- Our method for writing numbers comes from the Hindu-Arabic numeral system
- We write in base 10 since most of us have 10 fingers.
- The ancience babylonains used base 60 number system, which is why we have 60 seconds to a minute, 60 minutes to an hour
    - 60 is a very convenient number, can be divided by almost anything

Kinds of numbers

- N (natural number) = {1,2,3,4,...}
- Z (integers) = {...,-3,-2,-1,0,1,2,3,...}
- Q (rational numbers) = {a/b : a, b is an element of Z}
- R (all real numbers) = Q + all irrational numbers
    - Irrational numbers cannot be written as fractions
    - More irrational numbers than rational numbers
- C (complex numbers) = R + all imaginary numbers
    - i = sqrt(-1)
    - Probably all numbers that exist

Integers and Natural numbers (and computers)

- Computers do arithmetic with fixed width integers
    - Its like saying you can only do arithmetic up to 10 digits
- A digit for a computer is called a bit
    - Bits are either 0 or 1

Specifying integers in C

- Unsigned integers :
    - Shorts : maybe 16 bits?
    - Long : maybe 32 bits?
    - Long long: at least 32 bits, maybe 64 bits
- Instead use #include <stdint.h>
    - int8_t, uint8_t : 8 bits (signed and unsigned, respectively)

Binary arithmetic

- Binary arithmetic is just like normal arithmetic, except you have only two digits (bits): 0 and 1
    - 0 + 0 = 0
    - 0 + 1 = 1
    - 1 + 1 = 10

Real Numbers

- R = real numbers
    - Continuous
    - Uncountably infinite
- There are just as many numbers between *any* x and y as there in all of R
- R includes all of:
    - Integers (Z)
    - Rational numbers (Q)
    - irrational numbers (R-Q)

Float Point Numbers

- Are a proper subset of real numbers
    - F is a subset of R
- Are a proper subset of rational numbers
    - F is a subset of Q
- Are a proper subset of integers
    - F is a subset of Z
- It's a mistake to think (CHECK SLIDES)

Decimal and binary fractions

- There is no power of 10 that divides evenly by 3

- The fundamental theorem of arithmetic states every n which is an element of N has a unique prime factorization

Big endian, little endian

- When we have an integer

Random Numbers

- True random numbers cannot be created using computers
- Why?
    - Programs are inherently deterministic
- It has the advantage of repeatability, but
    - It has the disadvantage of predictability
-