Data types

- Basic data types
  - `char`
  - `int`
  - `float`
  - `double`
- Qualifiers
  - `short,long`
  - `signed,unsigned`

`16≤short/int ≤ 32 ≤long`

`datatype1.c`

Data types

- **Basic types**
- Void: type of a function that returns nothing (in other words, a procedure). The type (void *) is used as a generic pointer type in ANSI standard C.
- char The character type (usually a single byte).
  - Write the character enclosed in single quotes, for example 'a'.
  - '\n' represents a linefeed character, '\t' a horizontal tab, '\v' a vertical tab, '\b' a backspace, '\r' a carriage return, '\f' a form feed, '\\' a backslash, '\'' a single quote, '\"' a double quote.
  - Characters can also be represented in octal: '\040' represents a character.
- int The integer type.
  - write it as a digit string in decimal (without a leading zero digit).
  - 0 followed by an octal digit string
  - 0x followed by a hexadecimal digit string.
- float The floating point (real) type.
  - Representation of a floating point value is similar to that in most other algorithmic languages.
  - Most programmers use a “long float”, namely “double”.

Binary Normalization

101.0111×2^{13}

$1.0101\,11 \times 2^{15}$ normalize

$1.010111 \times 2^{00001111}$

exponents are binary !

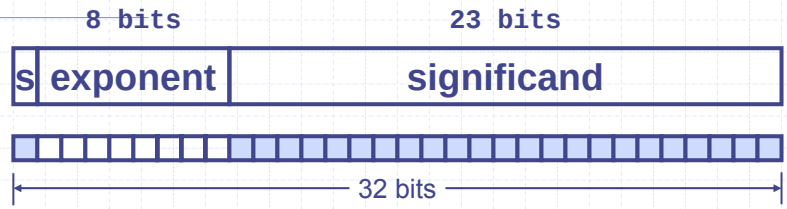
Normalized: one *digit* to the left of the binary point. It must be a 1!

We still use the term *digit*, although we mean "0" or "1".

Representation and Choices

- For each binary floating point number we need:
  - sign
  - significand (mantissa).
  - exponent
    - need a signed exponent!
- Suppose we want to store floating point numbers in 32 bits.
  - we need to decide how many bits should be used for the significand and how many for the exponent.
  - There is a tradeoff between *range* and *accuracy*.
- Large Range – large and small exponents
- High Accuracy – make the most out of the significand.
- We want it to be *easy* to compare two numbers.

32 bit IEEE 754 format



- Sign Bit:
  - 0 means positive, 1 means negative

Value of a number is:

$$(-1)^s \times F \times 2^E$$

exponent

significand

Format and precision

- Single precision memory format
  - Sign bit: 1
  - Exponent width: 8
  - Significand precision: 23 (24 implicit)
- Examples:
  - 3f80 0000 = 1
  - 3eaa aaab ~ 1/3
  - c000 0000 = -2
- Double precision memory format
  - Sign bit: 1
  - Exponent width: 11
  - Significand precision: 52 (53 implicit)
  - Stored as two contiguous 4 bytes datum in memory
- single, double may be defined to be an integer, fixed point, or floating point.

Boolean

- No boolean type in C: represented by integers
- 0 : FALSE, non-zero (preferably 1): TRUE
- can be defined by macros
  - Define TRUE as 1 and FALSE as 0

`bool1.c`

String

- represented by an array of characters, terminated by a null character '\0'
  e.g. `"hello"` is represented by

- string functions defined in `<string.h>`
- common string operations
  - `strlen, strcpy, strcmp, strcat,` ...

`string1.c`