Numeric types

Ben Langmead
ben.langmead@gmail.com
www.langmead-lab.org

Source markdown available at github.com/BenLangmead/c-cpp-notes
In computers, all data are stored in binary

Binary is the number system where each digit is a power of 2

We are used to powers of 10 (decimal)

https://biscitmx.com/category/unplugged/
If we used our fingers to count in binary, we could count to $2^{10} - 1 = 1023$
Integer is like an array of bits, but we can’t use [] for individual bits

- Binary: 0 0 1 1 0 1 0 1
- Place value: $2^7$ $2^6$ $2^5$ $2^4$ $2^3$ $2^2$ $2^1$ $2^0$

$2^5 + 2^4 + 2^2 + 2^0 = 32 + 16 + 4 + 1 = 53$

- Instead, we need *bitwise operators*, discussed later
C integers use “two’s complement” representation for signed integers. Illustration with 4 bits:


When a two’s complement number overflows, it wraps around to a negative number
#include <stdio.h>

int main() {
    int i = 2147483647;
    int i_plus_1 = i + 1;
    printf("i = %d, i+1 = %d\n", i, i_plus_1);
    return 0;
}

$ gcc -c overflow.c -std=c99 -pedantic -Wall -Wextra
$ gcc -o overflow overflow.o
$ ./overflow
i = 2147483647, i+1 = -2147483648
Floating point numbers use their bits to store a few different things:

- **Sign**: 1 bit, positive or negative
- **Exponent**
- **Mantissa**

\[
\text{sign} \ \text{exponent} \ \text{mantissa} \\
\begin{array}{c}
0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\[
\downarrow
\]

\[
+ \ (0.1011)_2 \times 10^{(110)_{3\text{bit excess-k}}}
\]

\[
= (0.1011)_2 \times 10^2
\]

\[
= (10.11)_2
\]

\[
= (1 \times 2)^1 + (1 \times 2)^0 + (0 \times 2)^{-1} + (1 \times 2)^{-2} = 2.75
\]

[https://aha.betterexplained.com/t/8bit-floating-point-representation/1121](https://aha.betterexplained.com/t/8bit-floating-point-representation/1121)
Integer and floating-point representations differ:

- Integers have limited range, but integers in the range can be represented precisely. Floating point have limited range and can only approximate most numbers in the range.
- Integers use all available bits for two’s-complement representation. Floating point have separate sets of bits for sign, exponent and mantissa.
float a = 1 or int i = 3.0, it’s not as simple as copying bits

When going from integer types to floating-point (or double), we are getting an approximation, not the exact integer