Intro. Computing with the C Programming Language

Variables and Data Types

- PIC: Ch. 3
- GCM: Ch. 1, Ch. 2

Shin Hong

6 November 2023

Data Types in C

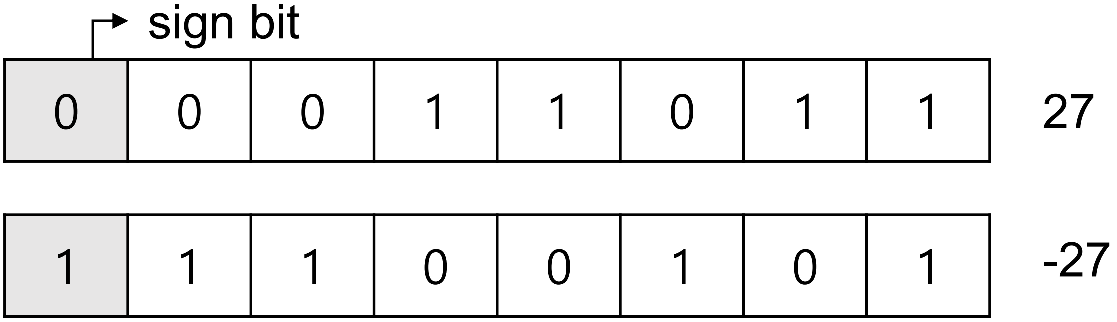
- Basic data types
  - `int`
  - `float`
  - `double`
  - `char`
  - `_Bool` (ISO C 99)
  - pointer

- Variants
  - `short, long, long long`
  - `signed, unsinged`

- Every data type has a range of values which determines the number of bytes to store a value
  - the number of bytes for a data type depends on both computer architecture and compiler

Number Representation

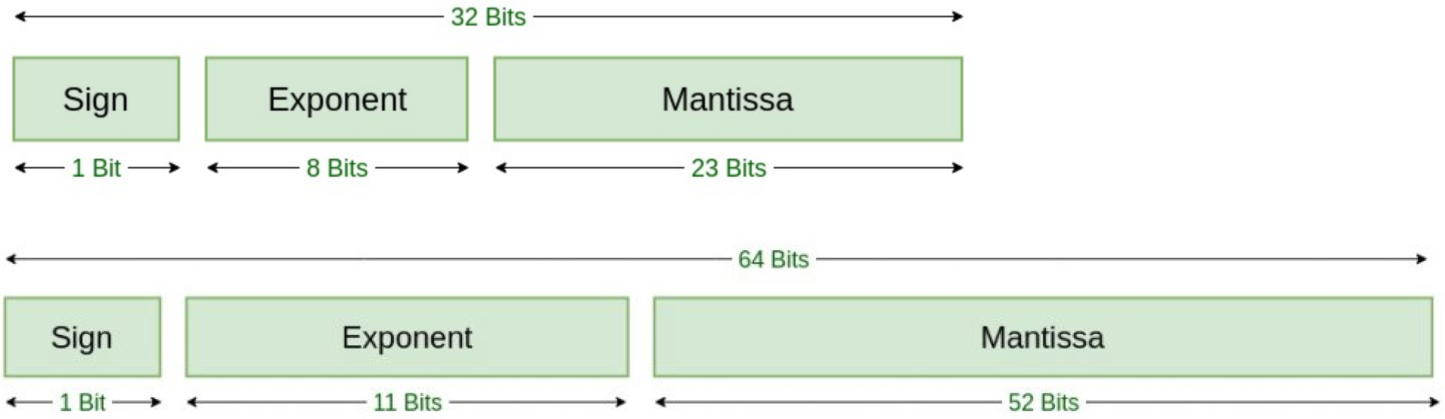
- Signed integer



sign bit

0	0	0	1	1	0	1	1	27
1	1	1	0	0	1	0	1	-27

- Floating number (IEEE 754 standard)



* images are taken from https://www.geeksforgeeks.org/ieee-standard-754-floating-point-numbers/

Constant Notation (1/2)

- Integers
  - decimal (base 10)
  - octal (base 8): starting with `0`
  - hexadecimal (base 16): starting with `0x`
  - e.g., `40 == 050 && 050 == 0x28`

Constant Notation (2/2)

- Real number
  - *digits* and *decimal point* and *digits*
    - e.g., `125.8`, `3.`, `-.0001`
  - scientific notation: *mantissa* and *e* and exponent
    - e.g., `1.7e4`, `2.25e-3`
  - by default, a real number is considered as `double`
  - to represent a `float` value, a real number must be ended with ‘`f`’ or ‘`F`’
    - e.g., `12.5f`

Predefined Constants

- `limits.h` defines the maximum and minimum values of a data type as symbol (macro)
  - `INT_MAX, INIT_MIN`
  - `UINT_MAX`
  - `DBL_MAX, DBL_MIN`

- `stdbool.h` defines two macros, `true` and `false` for the `_Bool` type

Variable

- Variable allows a programmer to call memory locations storing a value by a symbolic name

- A sequence of letters, decimal digits and '_' can be used as a variable name except the first character is a digit or it is a reserved word
  - https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html#Identifiers
  - identifiers are case-sensitive
  - the maximum length of a variable name is undefined, but the first 31 characters are used for identification
  - the same rule for a function name and other identifiers

Arithmetic Expression

- Arithmetic operators
  - Binary operators: +, -, *, /
  - Unary operator: -
- Precedence
  - notion that one operator can have a higher priority over another operator in evaluation
    - determine how an expression with more than operators is evaluated
  - Expressions containing operators of the same precedence are evaluated from left to right (or right to left depending on the operators)

Precedence Order (Table A.5)

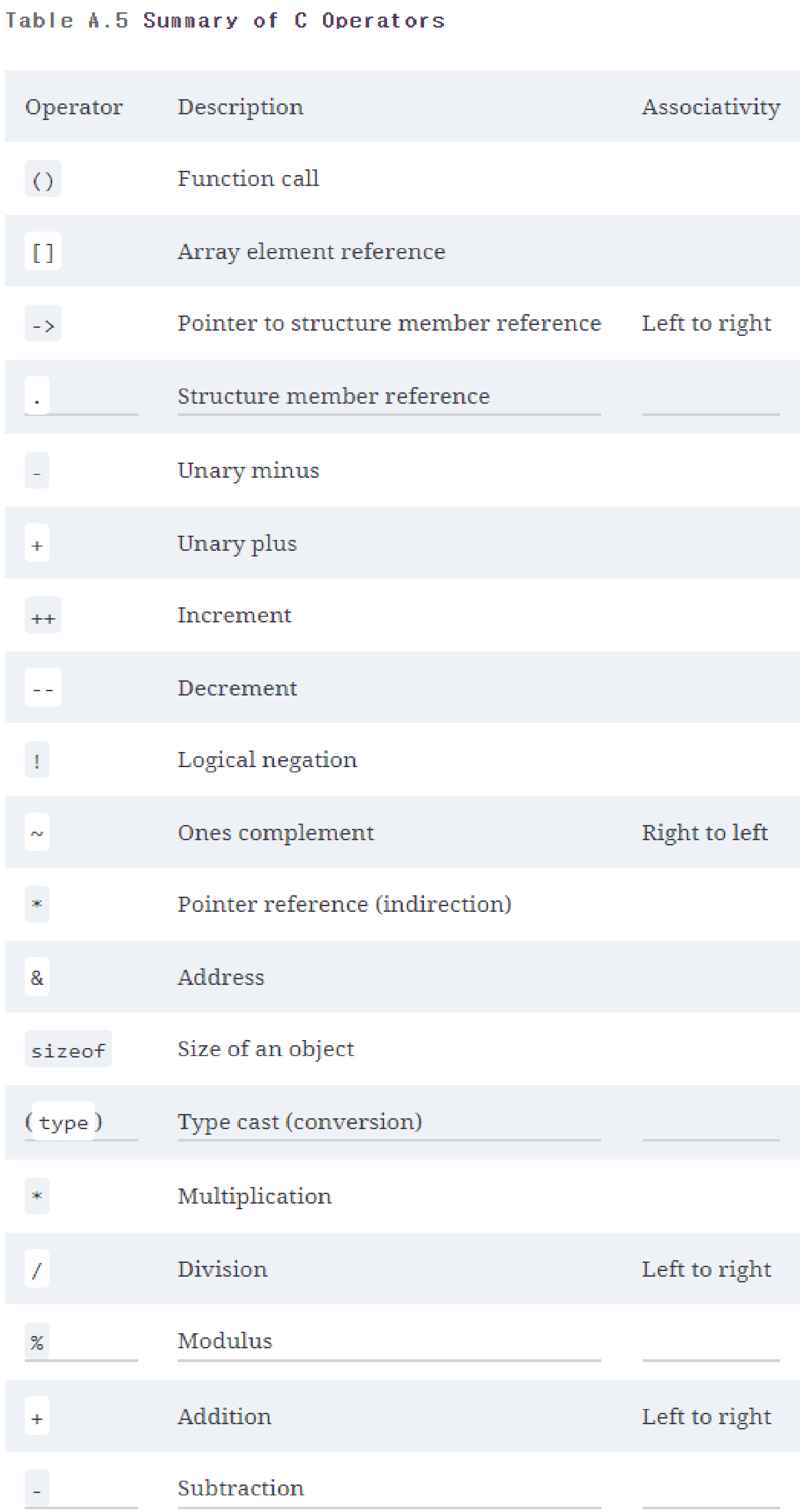


Table A.5 Summary of C Operators

Operator	Description	Associativity
()	Function call	Left to right
[]	Array element reference	
->	Pointer to structure member reference	
.	Structure member reference	
-	Unary minus	Right to left
+	Unary plus	
++	Increment	
--	Decrement	
!	Logical negation	
~	Ones complement	
*	Pointer reference (indirection)	
&	Address	
sizeof	Size of an object	
(type)	Type cast (conversion)	
*	Multiplication	Left to right
/	Division	
%	Modulus	
+	Addition	Left to right
-	Subtraction	

Implicit Conversion

- When a floating-point value is assigned to an integer variable, the decimal portion gets truncated

- Whenever two operands of an arithmetic operation are integers, the operation is carried out under the rules of integer arithmetic.

- If one operand is an integer and the other is a floating-point, the integer is first converted to the same floating-point type before the operation

- When a value of a *m*-bytes variable is assigned to a *n*-bytes variable for *m* > *n*, the *n*-least significant bytes are assigned