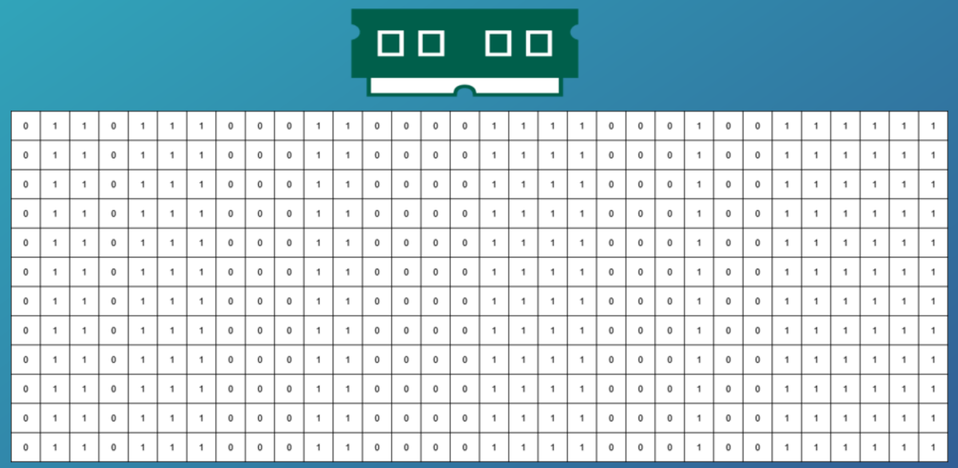
Integers and Number Systems

```
int number1 = 15;// Decimal
int number2 = 017;// Octal
int number3 = 0x0F;// Hexadecimal
int number4 = 0b00001111;// Binary

fmt::println("number1 : {}", number1);
fmt::println("number2 : {}", number2);
fmt::println("number3 : {}", number3);
fmt::println("number4 : {}", number4);
```

Number Systems



22

23.8

“Steve”

…

Number Systems

2371	$2 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 1 \times 10^0$
924	$9 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$
47	$4 \times 10^1 + 7 \times 10^0$

Number Systems

Base 2

100101	$1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
10010	$1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
111	$1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

Number Systems

3 Digits

Binary	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Number Systems

3 Digits
in memory

0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1

Number Systems

4 Digits

Binary	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Number Systems

4 Digits
in memory

0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1

Number Systems

5 Digits

Binary	Decimal
00000	0
00001	1
00010	2
00011	3
00100	4
00101	5
00110	6
00111	7
01000	8
01001	9
01010	10
01011	11
01100	12
01101	13
01110	14
01111	15

Binary	Decimal
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	32

Number Systems

5 Digits
in memory

0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1

Number Systems

Generalization

Digits	Data Range
1	0 ~1
2	0~3
3	0~7
4	0~15
5	0~31
…	…
n	$0\sim2^{n-1}$

Number Systems

In practice

Digits	Bytes	Data Range
8	1	0 ~255
16	2	0~65,535
32	4	0~34,359,738,367
64	8	0~18,446,744,073,709,551,615

Number Systems

Hexadecimal System

0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1

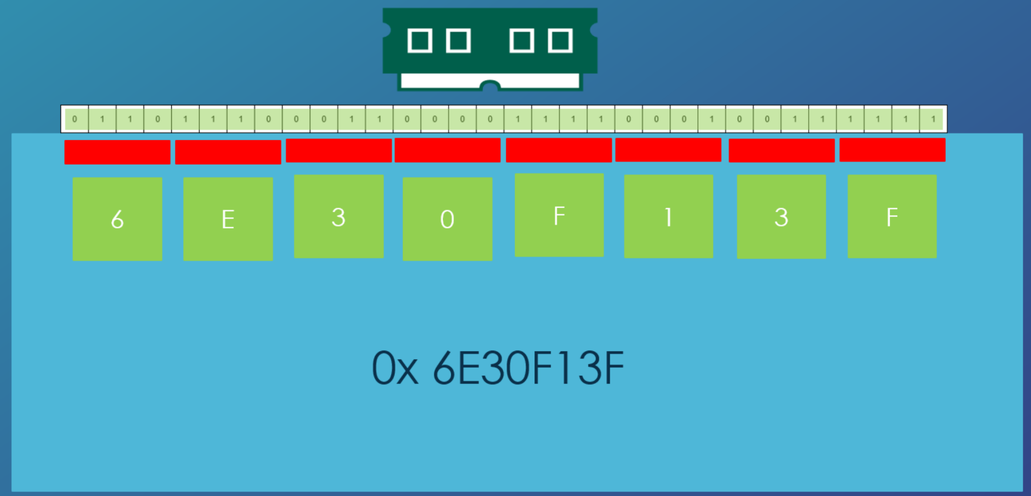
Number Systems

Hexadecimal System

Binary	Decimal	Hex
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

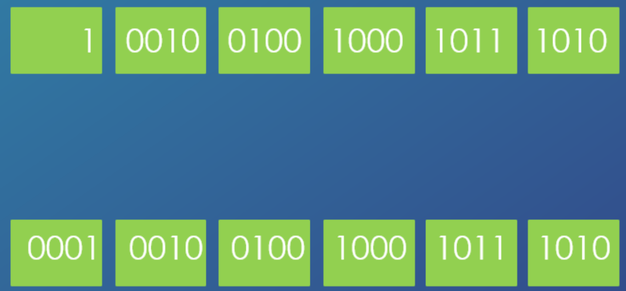
Number Systems

Hexadecimal System



Number Systems

Padding



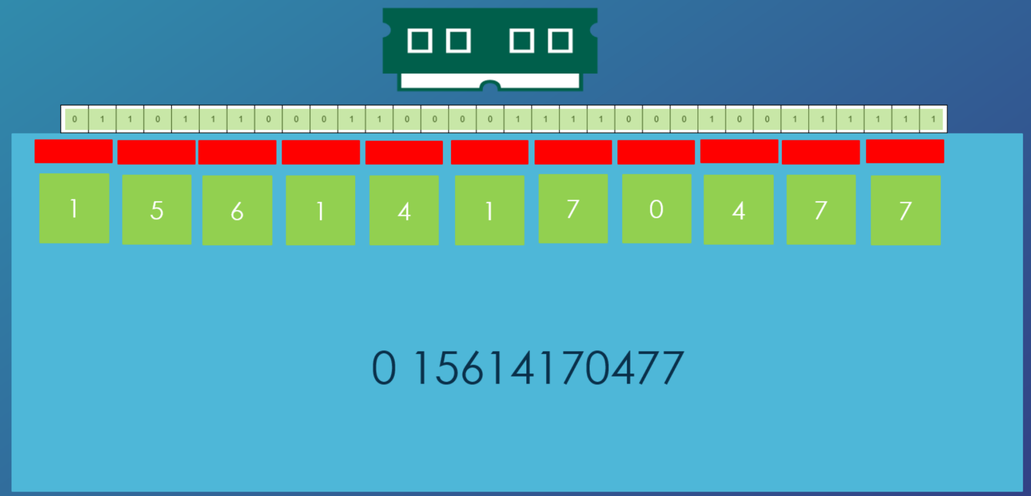
Number Systems

Octal System

Binary	Decimal	Oct
000	0	0
001	1	1
010	2	2
011	3	3
100	4	4
101	5	5
110	6	6
111	7	7

Number Systems

Octal System



Number Systems

Binary Table	https://kb.iu.edu/d/afdl

Floating point types

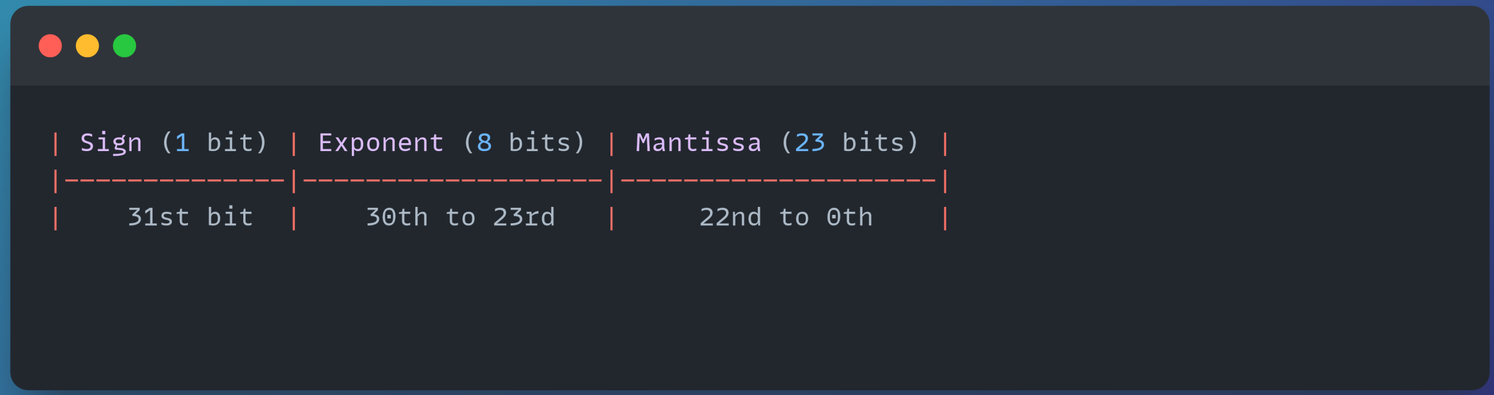
```
// Floating point types
// floats offer 7 digits of precision while double offers 15 digits of precision.
// The range of float is 1.17549e-38 to 3.40282e+38
//The range of double is 2.22507e-308 to 1.79769e+308
//The range of long double is 3.3621e-4932 to 1.18973e+4932
// Using https://ciechanow.ski/exposing-floating-point/ as a reference

float num1 = -343.53125f;
double num2 = -343.53125;
long double num3 = -343.53125l;
```

Floating point types: Memory representation

A single-precision float (float) in C++ is stored as a 32-bit binary value, and it is divided into three components:

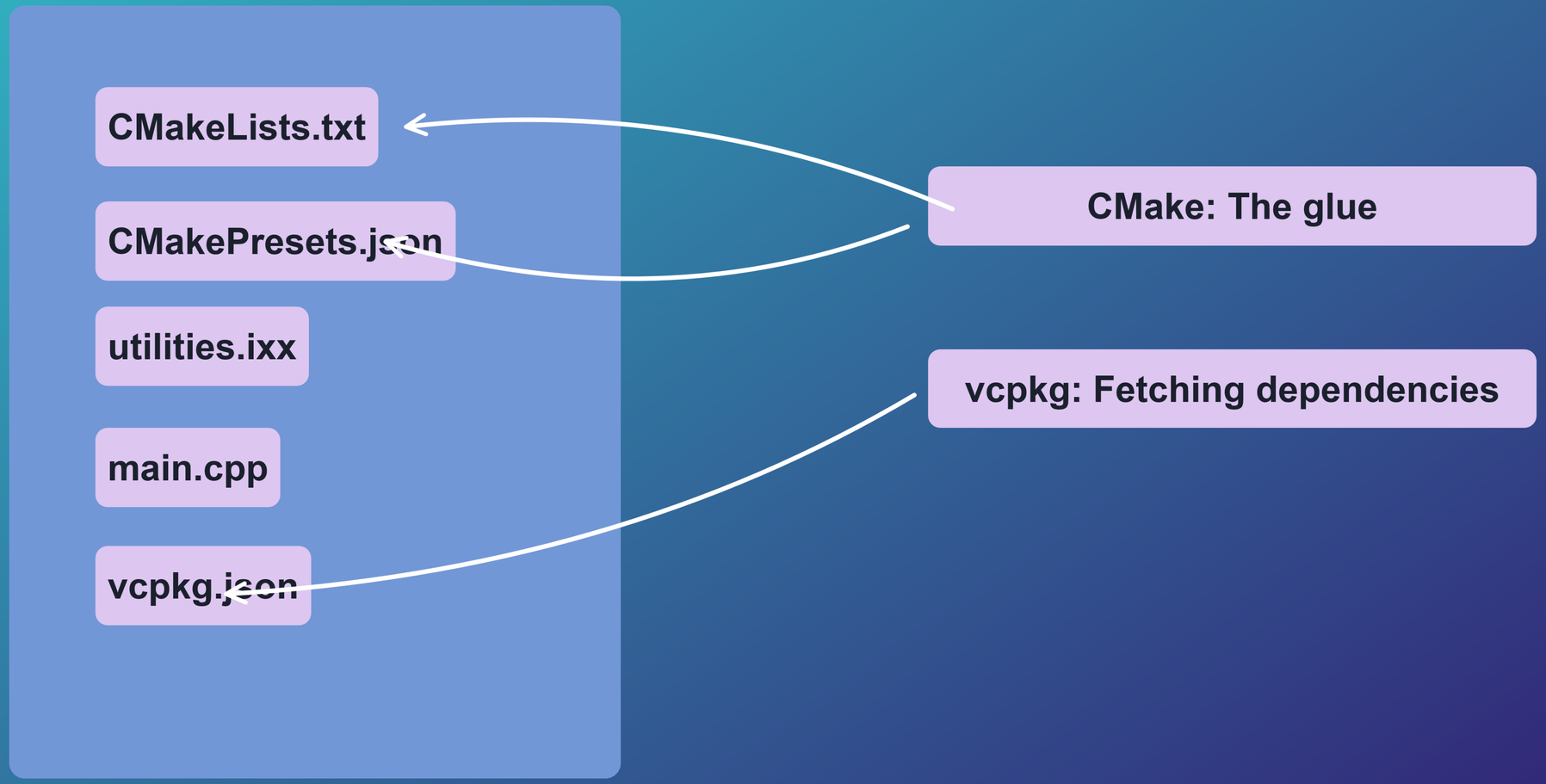
1. **Sign bit: 1 bit (the most significant bit).**
2. **Exponent: 8 bits.**
3. **Mantissa (fraction): 23 bits.**



*Going from a binary representation to the float representation like -2.23 requires to follow the procedure stipulated by IEEE. This is out of scope for this course.

Floating point types: Memory representation

```
export void print_type_ranges()
{
  // Ranges
  fmt::println(
    "The range for short is from {} to {}", std::numeric_limits<short>::min(), std::numeric_limits<short>::max());
  fmt::println("The range for unsigned short is from {} to {}",
    std::numeric_limits<unsigned short>::min(),
    std::numeric_limits<unsigned short>::max());
  fmt::println("The range for int is from {} to {}", std::numeric_limits<int>::min(), std::numeric_limits<int>::max());
  fmt::println("The range for unsigned int is from {} to {}",
    std::numeric_limits<unsigned int>::min(),
    std::numeric_limits<unsigned int>::max());
  fmt::println(
    "The range for long is from {} to {}", std::numeric_limits<long>::min(), std::numeric_limits<long>::max());
  fmt::println(
    "The range for float is from {} to {}", std::numeric_limits<float>::min(), std::numeric_limits<float>::max());
  fmt::println("The range(with lowest) for float is from {} to {}",
    std::numeric_limits<float>::lowest(),
    std::numeric_limits<float>::max());
  fmt::println("The range(with lowest) for double is from {} to {}",
    std::numeric_limits<double>::lowest(),
    std::numeric_limits<double>::max());
  fmt::println("The range(with lowest) for long double is from {} to {}",
    std::numeric_limits<long double>::lowest(),
    std::numeric_limits<long double>::max());
  fmt::println("int is signed: {}", std::numeric_limits<int>::is_signed);
  fmt::println("int digits: {}", std::numeric_limits<int>::digits);
}
```


CMakeLists.txt
CMakePresets.json
utilities.ixx
main.cpp
vcpkg.json
CMake: The glue
vcpkg: Fetching dependencies