(1)
ソースコードは付属のPDFとして添付した。
コードについての説明の詳細もそちらにコメント形式で記述した。

例として、数値20を代入した場合(sx.i = 20 , sx.u = 20, sx.f = 20, dx.d = 20)の出力結果（bit pattern）は以下の通り。表示されているビット順は、左に行くほど上位ビット、右に行くほど下位ビットとなっている。

```
unsigned integer:
00000000 00000000 00000000 00010100

integer:
00000000 00000000 00000000 00010100

float:
01000001 10100000 00000000 00000000

double:
01000000 00110100 00000000 00000000 00000000 00000000 00000000 00000000
```

(2)
使用したコンピュータ、コンパイラ情報はこちら：

- M1 Macbook Air (late 2020)
- ARM64 (customized ARMv8 by apple)
- OS Version : macOS 12.4
- Compiler : gcc (Apple clang version 13.0.0 (clang-1300.0.29.30))

unsigned int :
1, -1, UINT_MAXの値をそれぞれ出力した結果は以下の通り。

```
1:
00000000 00000000 00000000 00000001

-1:
11111111 11111111 11111111 11111111

4294967295:
11111111 11111111 11111111 11111111
```

unsigned intは32ビットであり、最上位ビットは符号ビットではなく通常のビットとして扱われる。-1を入れた場合、全体のビットが反転していることから、マイナスの値は補数表現として取り扱われるものの、unsigned intとして読み出した場合はUINT_MAXの値として扱われるとわかる。

int :
1, -1, INT_MAX, INT_MINの値をそれぞれ出力した結果は以下の通り。

```
1:
00000000 00000000 00000000 00000001

-1:
11111111 11111111 11111111 11111111

2147483647:
01111111 11111111 11111111 11111111

-2147483648:
10000000 00000000 00000000 00000000
```

intは32ビットであり、最上位1ビットは符号ビットとして扱われる。マイナスの値は補数表現として取り扱われるため、0と1が反転している。

float :
1, -1, FLOAT_MAX, FLOAT_MINの値をそれぞれ出力した結果は以下の通り。

```
0:
00000000 00000000 00000000 00000000

1:
00111111 10000000 00000000 00000000

2:
01000000 00000000 00000000 00000000

-1:
10111111 10000000 00000000 00000000

3.402823e+38:
01111111 01111111 11111111 11111101

-3.402823e+38:
00000000 00000000 00000000 00000001
```

floatは最上位1ビットが符号ビット、仮数部が8ビット、仮数部が21ビットの計32bitで構成されている。0を表す際は、仮数部、指数部ともに全てのビットが0になっている。
2の値の例を見ると、これは通常のIEEE 754で規定されたフォーマットに従っており、仮数部を127とすることで、 (-1)^0 * (1 + 0 + 0 + …) * 2^0 = 2 と表現されていることがわかる。

double :
1, -1, DOUBLE_MAX, DOUBLE_MINの値をそれぞれ出力した結果は以下の通り。

```
0:
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

1:
00111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000

2:
01000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

-1:
10111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000

1.797693e+308:
01111111 11101111 11111111 11111111 11010111 10111001 01100000 10011010

-1.797693e+308:
11111111 11101111 11111111 11111111 11010111 10111001 01100000 10011010
```

doubleは最上位1ビットが符号ビット、仮数部が11ビット、仮数部が52ビットの計64bitで構成されている。2の出力結果例から、float同様IEEE754準拠のフォーマットであることがわかり、floatを64bitに拡張したような形式であるとわかる。

(3)
それぞれの出力結果は以下の通り。exponent（指数部）で表現されているとわかる。
０徐算に関する値（無限大）はいずれも仮数部11bitが全て１となっており、符号を最上位１ビットで表している。
（上から順に 1.0/0.0, -1.0/0.0, 0.0/0.0, 2^1000です）

```
double:
1.0/0.0:
01111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000

-1.0/0.0:
11111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000

0.0/0.0:
01111111 11111000 00000000 00000000 00000000 00000000 00000000 00000000

2^1000
01111110 01110000 00000000 00000000 00000000 00000000 00000000 00000000
```