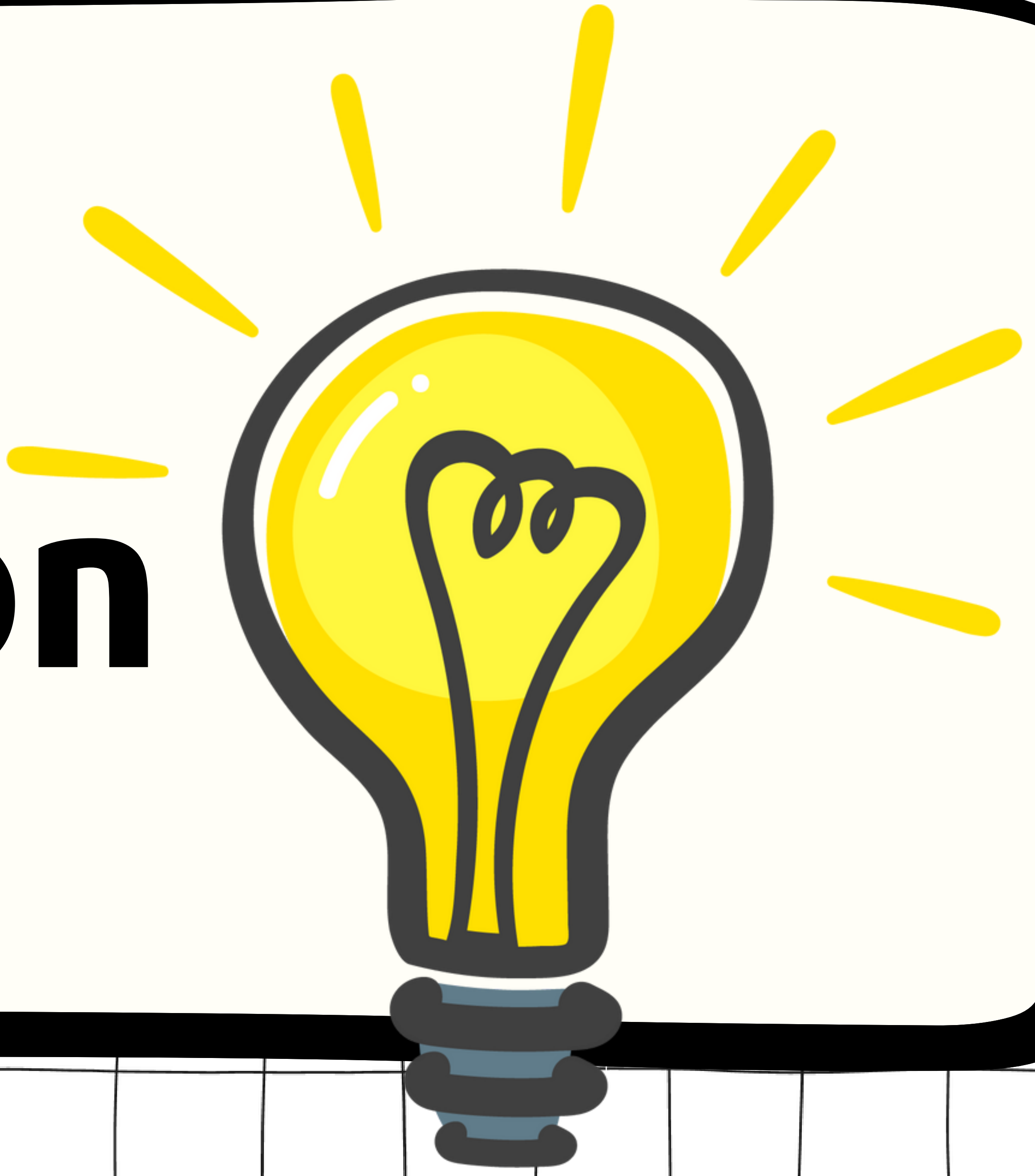
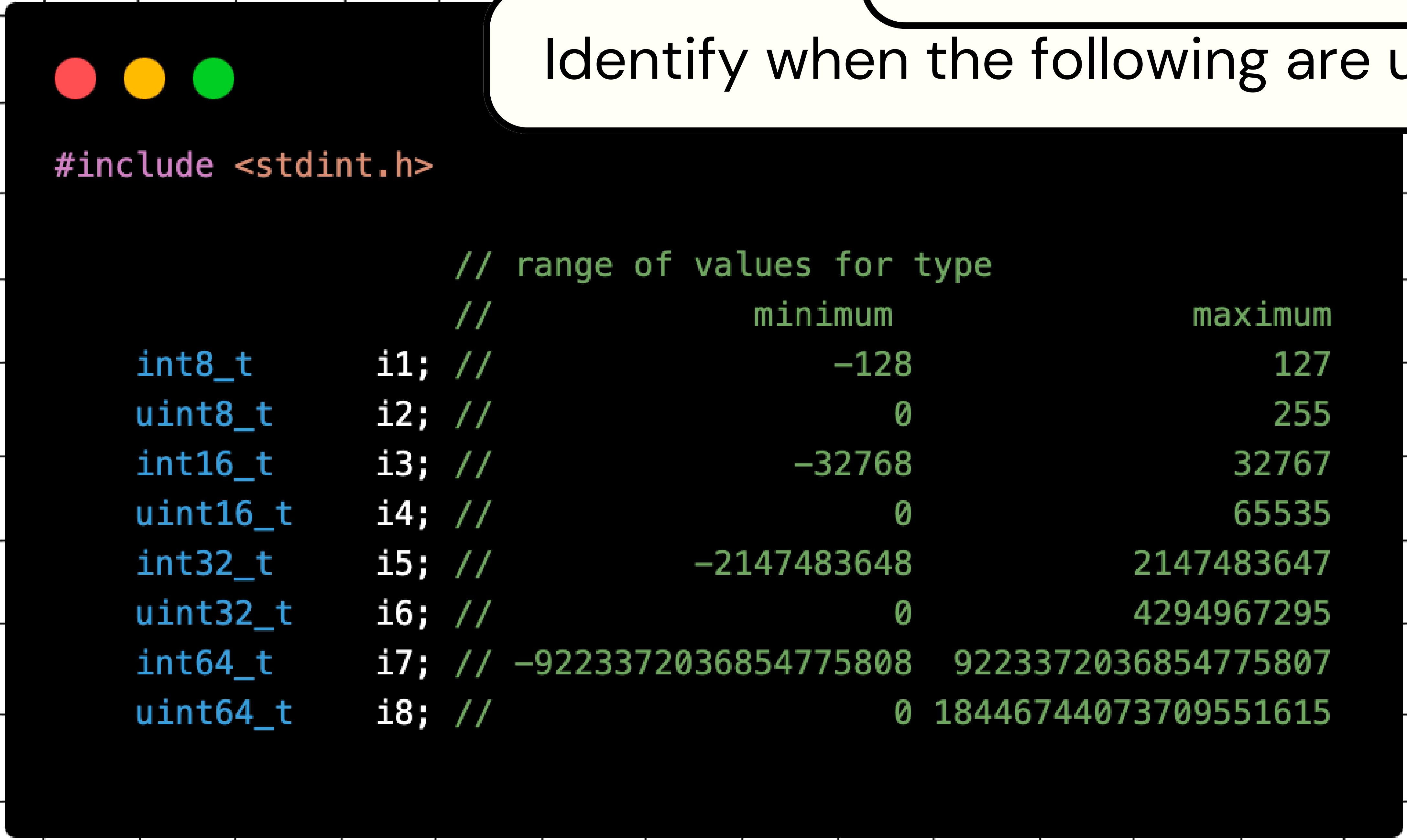
Numerical Representation

Numerical Representation of Values



Question 1

Identify when the following are useful



```c
#include <stdint.h>

                    // range of values for type
                    //             minimum               maximum
    int8_t      i1; //                -128                   127
    uint8_t     i2; //                   0                   255
    int16_t     i3; //              -32768                 32767
    uint16_t    i4; //                   0                 65535
    int32_t     i5; //         -2147483648            2147483647
    uint32_t    i6; //                   0            4294967295
    int64_t     i7; // -9223372036854775808  9223372036854775807
    uint64_t    i8; //                   0 18446744073709551615
```

Question 2

Base Systems

How can you tell if an integer is in decimal (base 10), hexadecimal (base 16), octal (base 8) or binary (base 2)

Is this good language design?

How could I write a C program to answer this question?

1	8	10	15
16	100	127	200

Question 3

Operations

What are the values of the following expressions

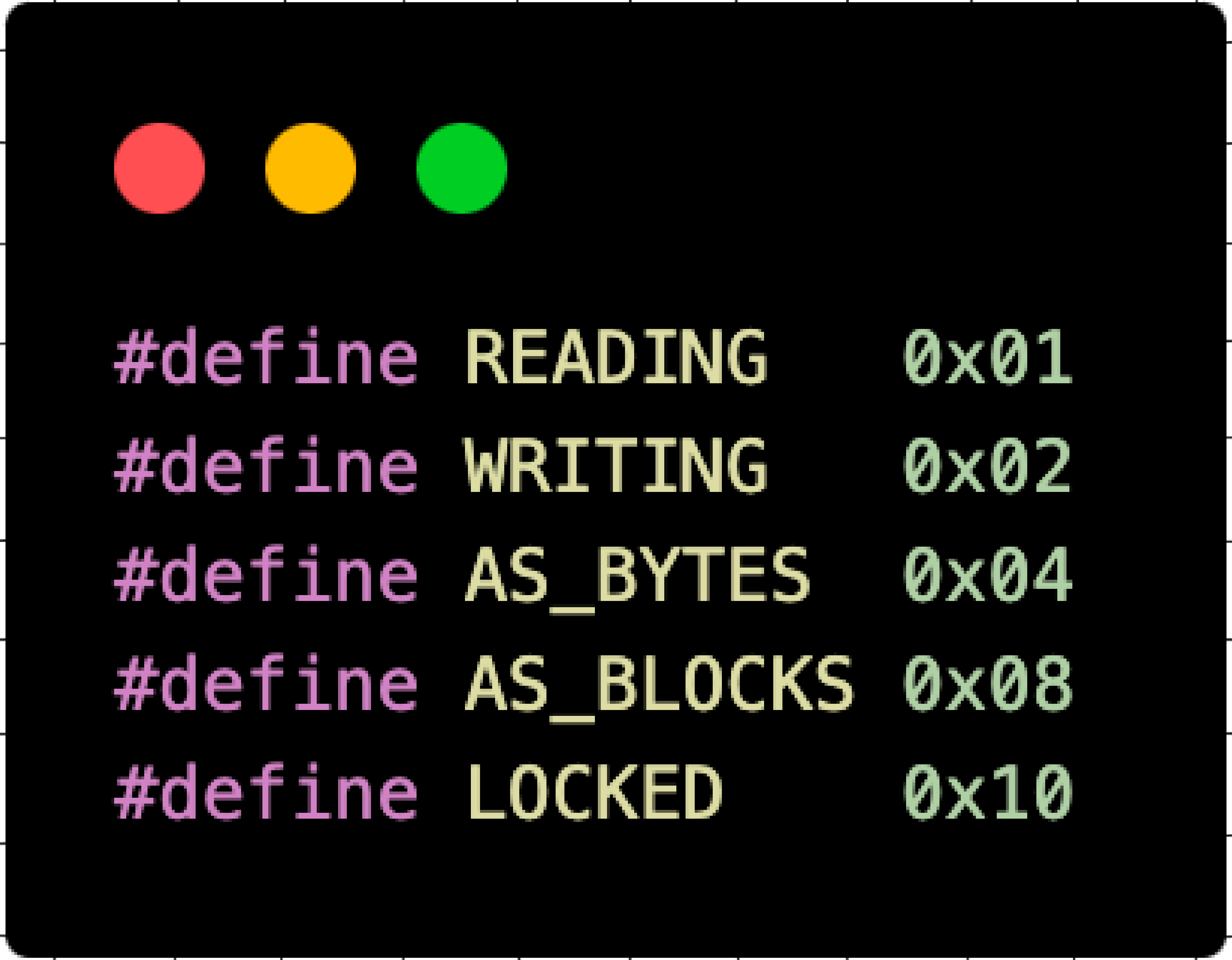
```
uint16_t a = 0x5555, b = 0xAAAA, c = 0x0001;
```

a | b	a & b	a ^ b	a & ~b
c << 6	a >> 4	a & (b << 1)	b | c

Question 4

Flags

- Mark device as locked for reading bytes
- Mark device as locked for writting bytes
- Set the device as locked, leaving other flags unchanged
- Remove the lock on a device leaving flags unchanged
- Switch a decive from reading to writting leaving other flags unchanged
- Switch a device between reading and writing leacing other flags unchanged



```
#define READING   0x01
#define WRITING   0x02
#define AS_BYTES  0x04
#define AS_BLOCKS 0x08
#define LOCKED    0x10
```

Question 5

Flags

- that prints (to terminal) whether the printer is out of ink.
- that tells the printer the ink has been replaced.
- to use colour and select scan mode. Assume no mode has been selected yet.
- that toggles between print and scan mode. Assume 1 mode is already selected.

```
printerControl = 0 0 0 0 0 0 0 0
                       ^ ^ ^ ^ ^
                       | | | | |
                       | | | | L [NO_INK]
                       | | | L [COLOUR]
                       | | L [SELECT_PRINT]
                       | L [SELECT_SCAN]
                       L [START]
```

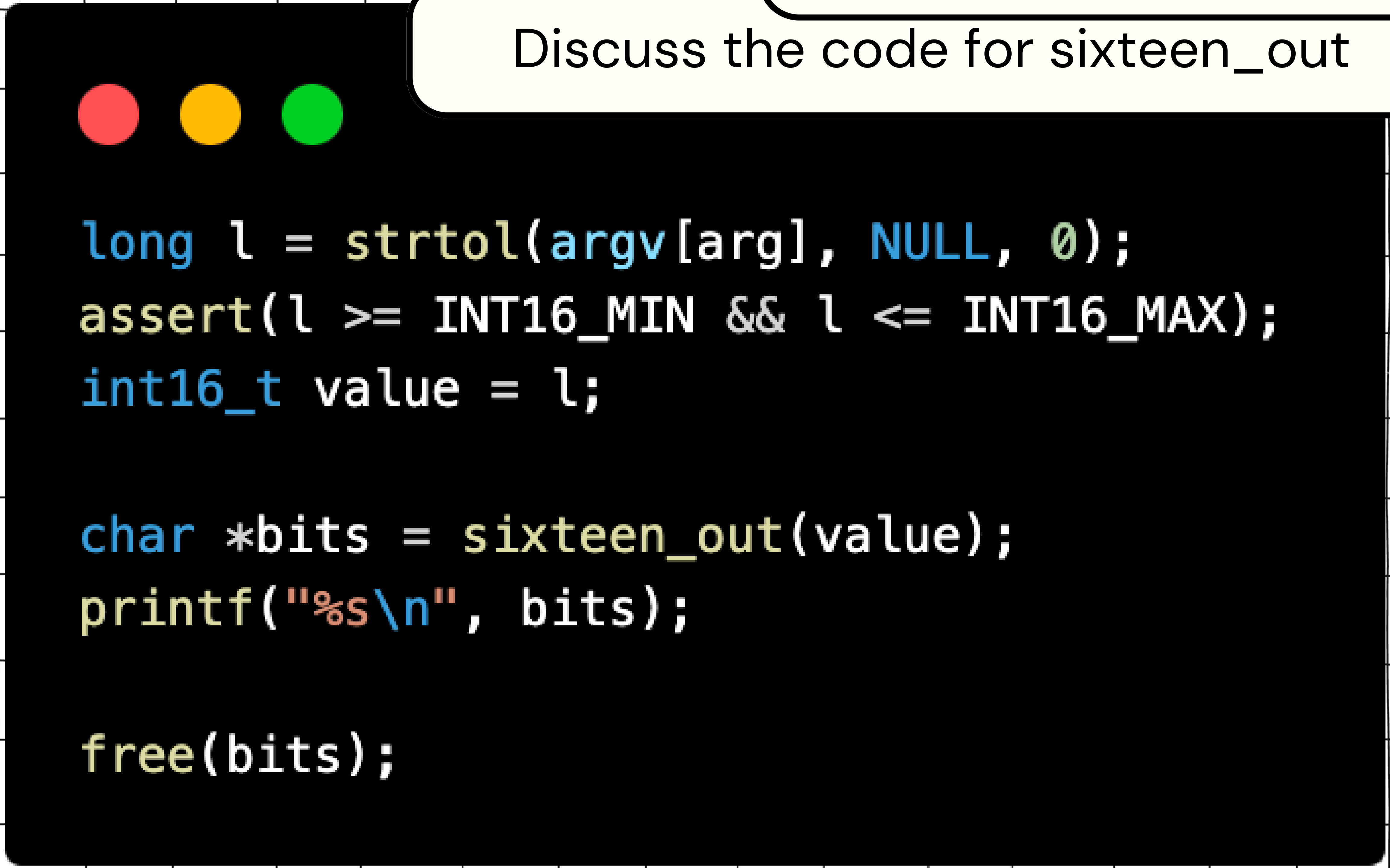
```c
#include <stdint.h>

// Whether the printer is out of ink
#define NO_INK (0x1)        // 0b 0000 0001
// Whether to print/scan in colour
#define COLOUR (0x2)        // 0b 0000 0010
// Select print mode
#define SELECT_PRINT (0x4)  // 0b 0000 0100
// Select scan mode
#define SELECT_SCAN (0x8)   // 0b 0000 1000
// Start print/scan
#define START (0x10)        // 0b 0001 0000

uint8_t printerControl = 0; // 0b 0000 0000
```

Question 6

Discuss the code for sixteen_out



```
long l = strtol(argv[arg], NULL, 0);
assert(l >= INT16_MIN && l <= INT16_MAX);
int16_t value = l;

char *bits = sixteen_out(value);
printf("%s\n", bits);

free(bits);
```

Question 7

Reverse Bits Function