what category of thing would you rather be able to fully forget on command so you can enjoy it again?

clicker.cs.illinois.edu

Q1

~Code~
340

CS 340

Bitwise Operations 0b10

Updates

1. Exam 1 scores are released.
    a. More information on Campus Wire

2. HW 4 is out (due next wednesday)

3. MP 4 - UTF-8 (due next Tuesday)

4. Collaboration time today! 2-4pm

Bitwise Operations

Today's LGs - Build on your mental model of how data is stored and interpreted on a computer (1's and 0's).

Be able to converse about bits
Be able to shift bits
Be able to apply logical operations at a bit level
Be able to use bit operations to isolate the bits you need
Be able to convert between code points and UTF-8
Be able to convert between bits and negative numbers

Agenda

1. MP4 - UTF-8

2. Negative numbers

3. Bitwise practice

UTF-8

0-255

Char - 1 byte of info 0000 0000 - 1111 1111

0-127

ASCII - a mapping from a number to a symbol

0-10FFFF ?

Unicode - a mapping from a number to a symbol

codepoint

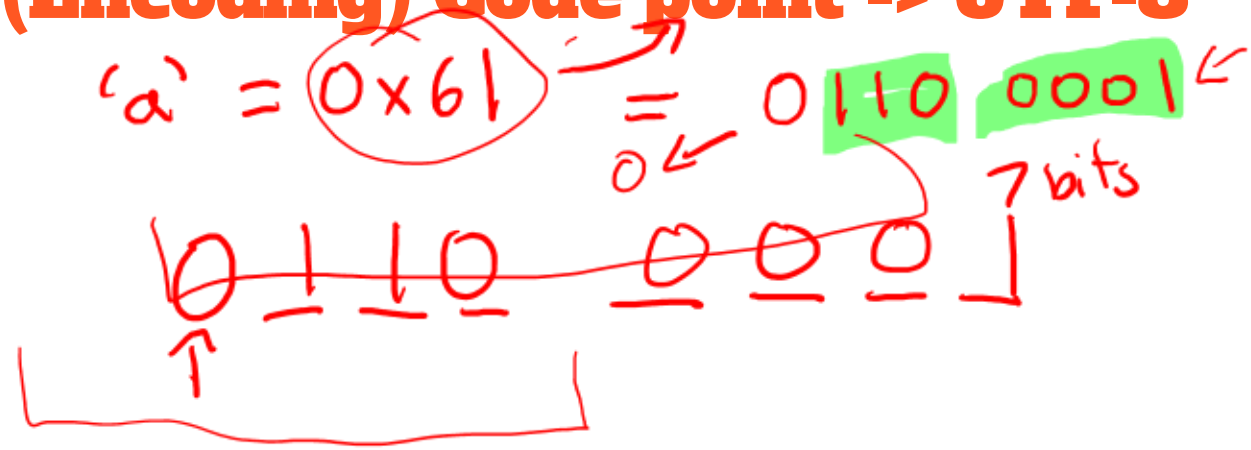
UTF-8 - encode a unicode code point into chars

UTF-8

Code Point -

UTF-8 -

(Encoding) Code point -> UTF-8



Byte	Meaning
0xxxxxxx	only byte of character
10xxxxxx	second, third, or fourth byte of a character
110xxxxx	first byte of a two-byte character
1110xxxx	first byte of a three-byte character
11110xxx	first byte of a four-byte character
11111xxx	invalid

Bits needed	Groups used
0–7	1
8–11	2
12–16	3
17–21	4

(Encoding) Code point -> UTF-8

= E2
9C
85

Byte	Meaning
0xxxxxxx	only byte of character
10xxxxxx	second, third, or fourth byte of a character
110xxxxx	first byte of a two-byte character
1110xxxx	first byte of a three-byte character
11110xxx	first byte of a four-byte character
11111xxx	invalid

Bits needed	Groups used
0–7	1
8–11	2
12–16	3
17–21	4

How many X slots are there in a 3-group UTF-8 character?

3 bytes = 24 bits

clicker.cs.illinois.edu

Q2

~Code~
340

Byte	Meaning
0xxxxxxx	only byte of character
10xxxxxx	second, third, or fourth byte of a character
110xxxxx	first byte of a two-byte character
1110xxxx	first byte of a three-byte character
11110xxx	first byte of a four-byte character
11111xxx	invalid

2+

1+

Bits needed	Groups used
0–7	1
8–11	2
12–16	3
17–21	4

(Decoding) Code point ⇆ UTF-8

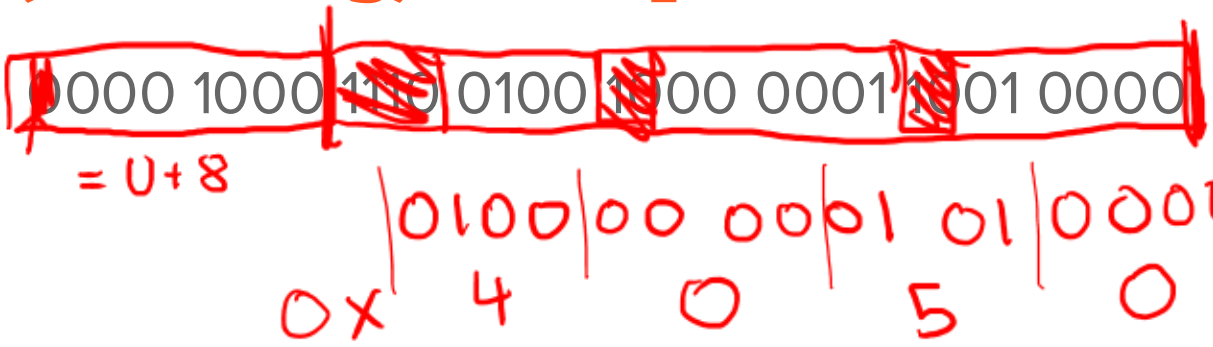
1. find start and end bytes
2. chop off the headers
3. put bits together

Byte	Meaning
0xxxxxxx	only byte of character
10xxxxxx	second, third, or fourth byte of a character
110xxxxx	first byte of a two-byte character
1110xxxx	first byte of a three-byte character
11110xxx	first byte of a four-byte character
11111xxx	invalid

Bits needed	Groups used
0–7	1
8–11	2
12–16	3
17–21	4

(Decoding) Code point -> UTF-8

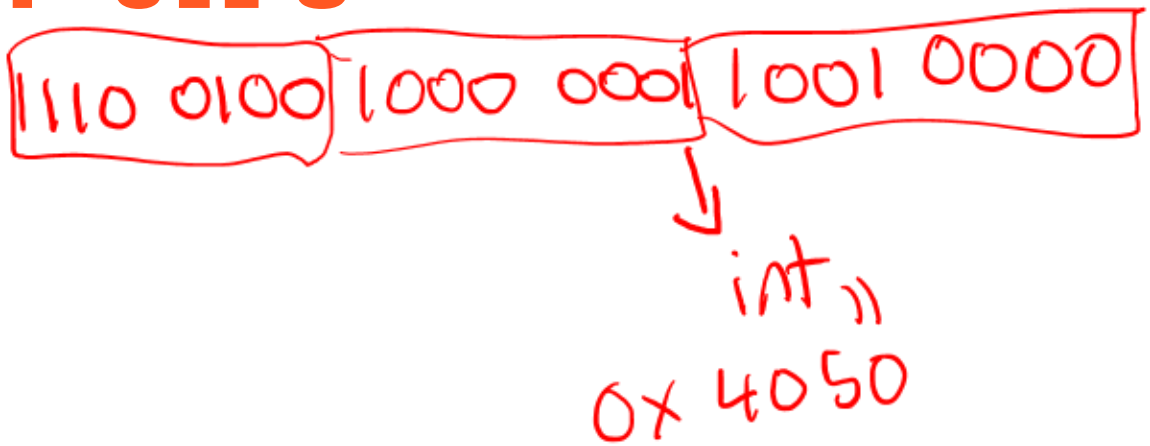
<Backspace> 映



Byte	Meaning
0xxxxxxx	only byte of character
10xxxxxx	second, third, or fourth byte of a character
110xxxxx	first byte of a two-byte character
1110xxxx	first byte of a three-byte character
11110xxx	first byte of a four-byte character
11111xxx	invalid

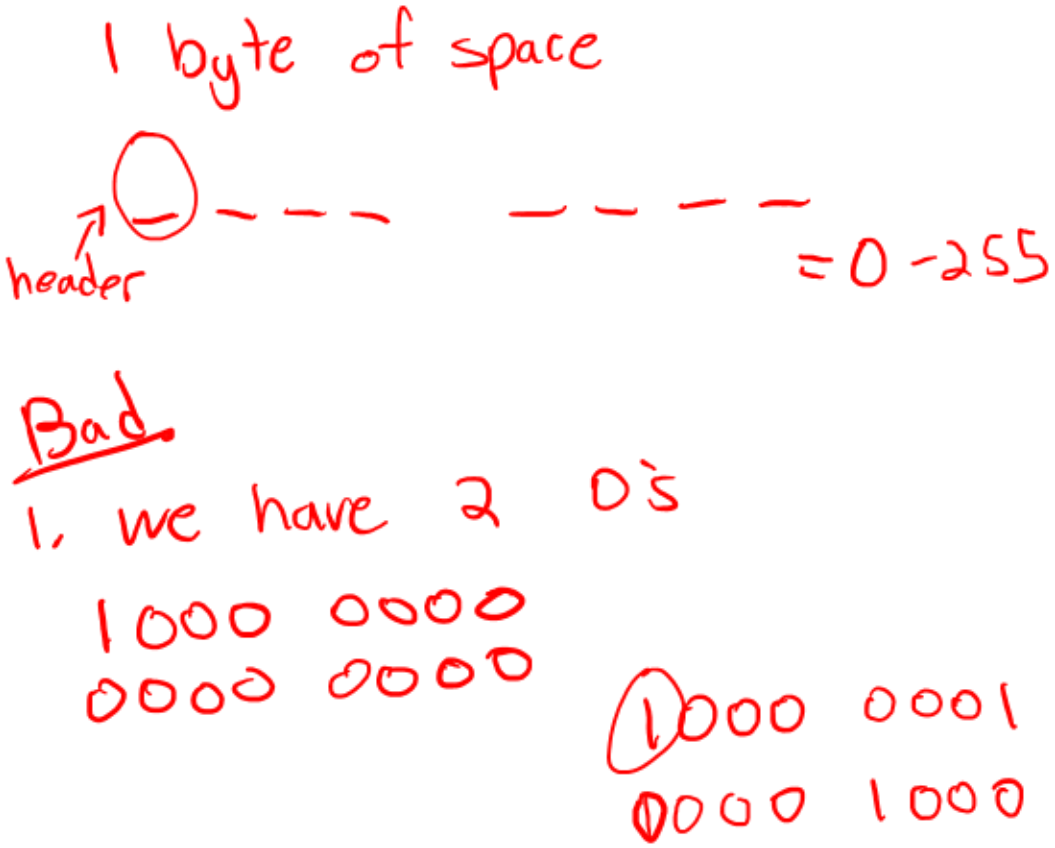
Bits needed	Groups used
0–7	1
8–11	2
12–16	3
17–21	4

MP4 - UTF-8



Agenda

1. MP4 - UTF-8
2. **Negative numbers**
3. Bitwise practice



Negative Numbers with Bits

4 bytes 32 bits

3 bits

0 1 2

000 001 010

- flip the bits
- add 1

1. One 0!

2. (-2) + 3 = 1

100
011
+ 001
100

1
110
+ 011
1001 ← 1

3 4 5 6 7

011 100 101 110 111

-4 -3 -2 -1

101 = -3

010

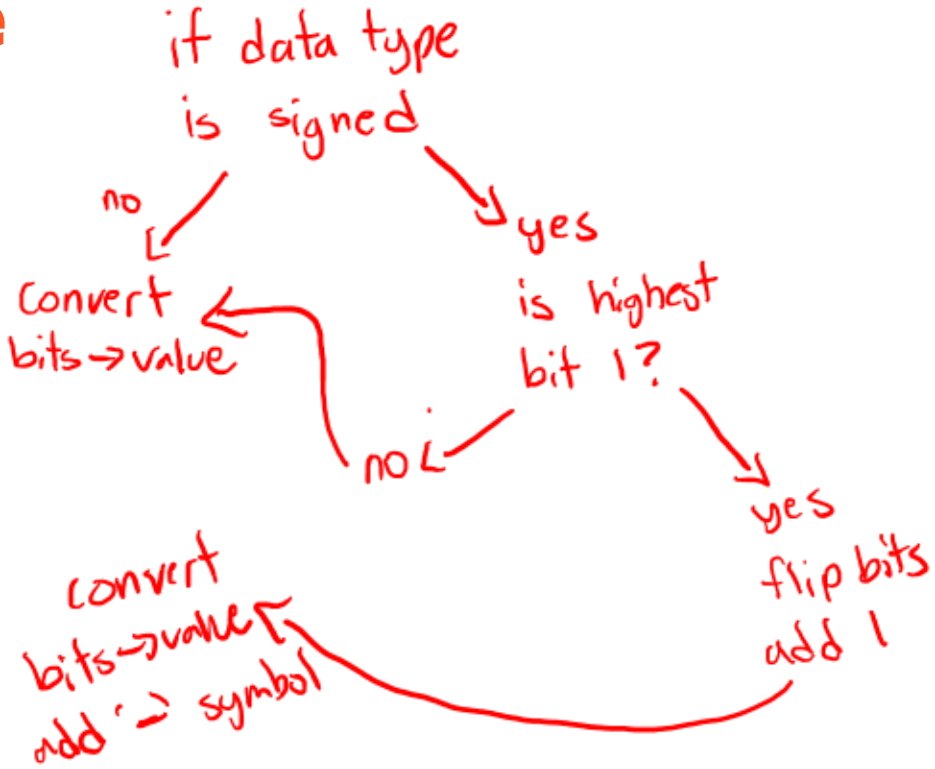
011 ← -3

-1 + -2 = -3

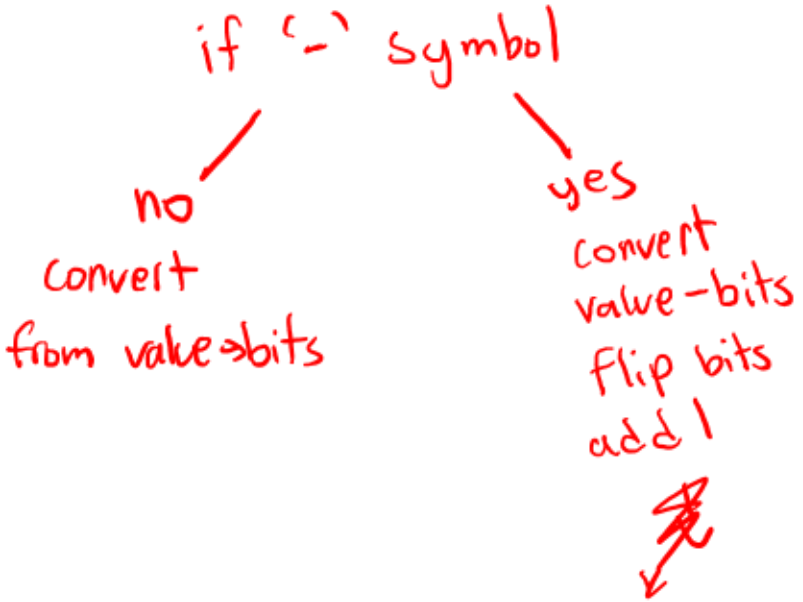
-4 - 3

1
111
+ 110
101

Bits to Value



Value to Bits



-5

X

0101

0000 0101 -> 5

1111 1010 = 1111 1011

-5

What value do these 8 bits represents? Assume it is a signed data type and write the answer in decimal.

clicker.cs.illinois.edu

Q3

~Code~
340

negative

0b1111 0100 = -12

0000 1011

+ 0000 1100 = 12

What value do these 8 bits represents? Assume it is a signed data type and write the answer in decimal.

clicker.cs.illinois.edu

Q4

~Code~
340

0b0000 1100 = 12

What value do these 8 bits represents? Assume it is a unsigned data type and write the answer in decimal.

clicker.cs.illinois.edu

Q5

~Code~
340

0b1000 0100 = 132

128 64 32 16 8 4 2 1

Agenda

1. MP4 - UTF-8

2. Negative numbers

3. Bitwise practice

HW Example 0b01

Making Masks with Bitwise Operations

View

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
short x = (-1) << 4;
```

Which of the following causes `x == y` to be true?

(a)
```
short y = 0xf;
```

(b)
```
short y = - 0xf;
```

(c)
```
short y = 0xfff1;
```

(d)
```
short y = - 0x10;
```

What is -1 in bits? Assume a 2-byte size and write the answer in binary.

clicker.cs.illinois.edu

Q6

~Code~
340

0000 0000 0000 0001
1111 1111 1111 1110 +1

Making Masks with Bitwise Operations

View

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
short x = (-1) << 4;
```

x = (-1) << 4

-1 = 1111 1111 1111 1111

== ~0

What is x in bits? Assume a 2-byte size and write the answer in binary.

clicker.cs.illinois.edu

Q7

~Code~
340

Making Masks with Bitwise Operations

View

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
short x = (-1) << 4;
```

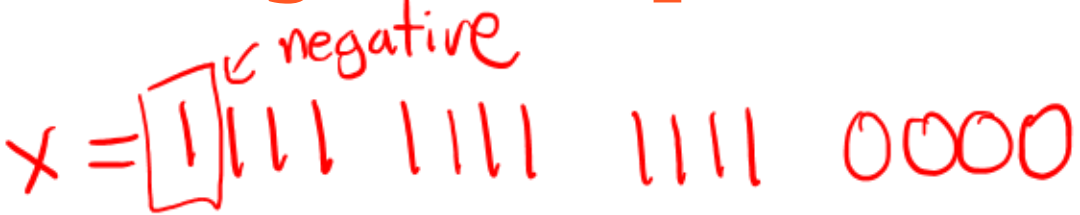
x = (-1) << 4; 0b1111 1111 1111 0000

Does x represent a positive or negative number assuming 2's complement?

clicker.cs.illinois.edu

Q8

~Code~
340



Making Masks with Bitwise Operations

View

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
short x = (-1) << 4;
```

What is x as a negative value (not the bits but the value)? Assume a 2-byte size and write the answer in hex.

clicker.cs.illinois.edu

Q9

~Code~
340

1111 1111 1111 0000 - flip

0000 0000 0000 1111 add 1

0000 0000 0001 0000 = 0x0010

-0x10

Making Masks with Bitwise Operations

View

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
short x = (-1) << 4;
```

Making Masks with Bitwise Operations

View

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
short x = (-1) << 4;
```

Which of the following causes `x == y` to be true?

- (a) `short y = 0xf;`
- (b) `short y = - 0xf;`
- (c) `short y = 0xfff1;`
- (d) `short y = - 0x10;`

HW Example 0b02

Making Masks with Bitwise Operations

View...

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
int x = ~((-1) << 24);
```

Which of the following causes `x == y` to be true?

- (a) `int y = 0x1000000;`
- (b) `int y = (1 << 24) - 1;`
- (c) `int y = -0x1000000;`
- (d) `int y = -0xffffff;`

What is -1 in bits? Assume a 4-byte size and write the answer in hex.

0x FF FF FF FF

clicker.cs.illinois.edu

Q10

~Code~
340

Making Masks with Bitwise Operations

View...

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
int x = ~((-1) << 24);
```

x = ~((-1)<<24)

What is -1 << 24 in bits? Assume a 4-byte size and write the answer in hex.

1's ... 24 0's = 8 = 0's

FF00 00 00

clicker.cs.illinois.edu

Q11

~Code~
340

Making Masks with Bitwise Operations

View...

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
int x = ~((-1) << 24);
```

What is x in bits? Assume a 4-byte size and write the answer in hex.

clicker.cs.illinois.edu

Q12

~Code~
340

prev = FF 00 00 00

Making Masks with Bitwise Operations

View...

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
int x = ~((-1) << 24);
```

x = ~((-1)<<24); 0x00 FF FF FF

Does x represent a positive or negative number assuming 2's complement?

clicker.cs.illinois.edu

Q13

~Code~
340

x = 0x00 FF FF FF positive

Making Masks with Bitwise Operations

View...

Assume that `char`, `short`, and `int` are signed integer types with 8, 16, and 32 bits respectively; and that we've declared and initialized `x` as

```
int x = ~((-1) << 24);
```

What is the answer?

clicker.cs.illinois.edu

Q14

~Code~
340

```
int x = ~((-1) << 24);
```

Which of the following causes `x == y` to be true?

0x00 FF FF FF

(a)
```
int y = 0x1000000;
```

~~0x000~~

0x00 00 00 01

0x01 00 00 00

-1

0x00 FF FF FF

(b)
```
int y = (1 << 24) - 1;
```

(c)
```
int y = -0x1000000;
```

(d)
```
int y = -0xffffff;
```

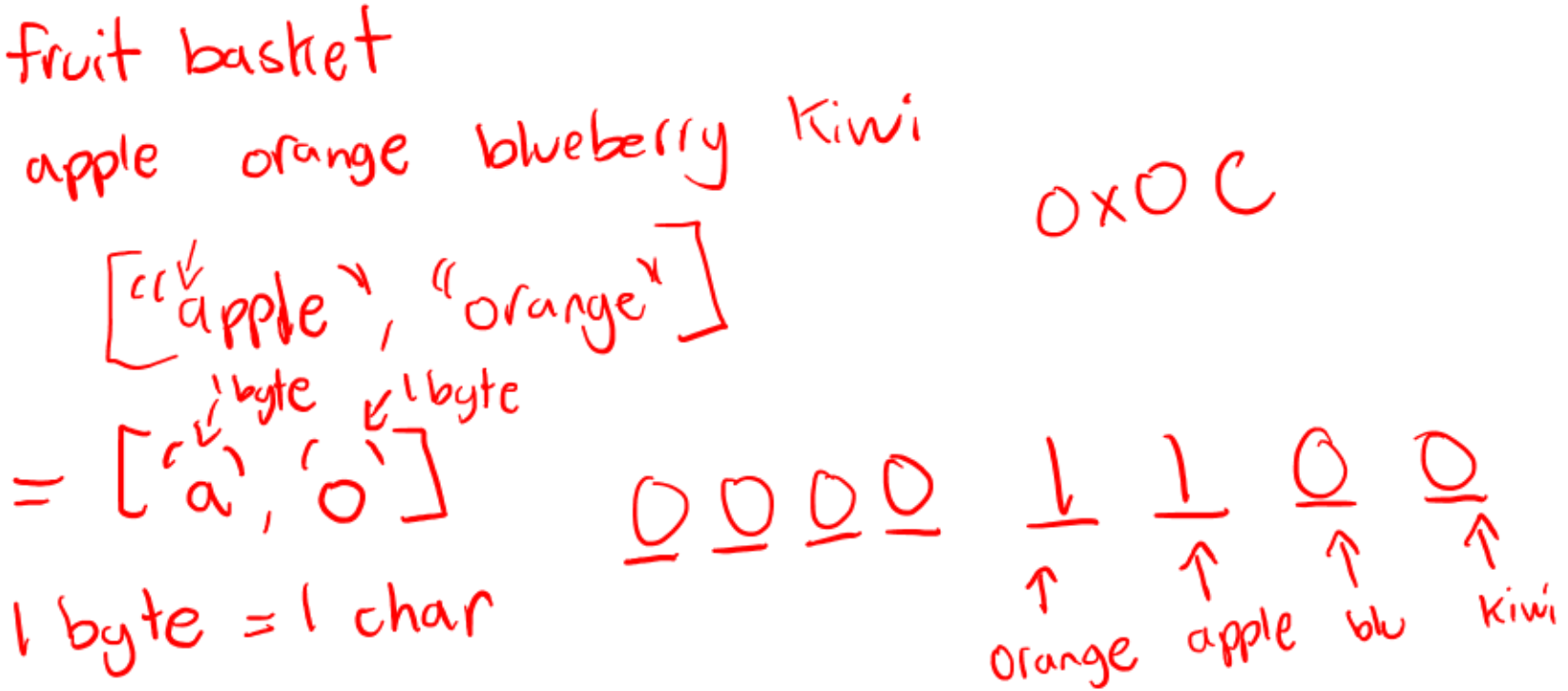
Bitwise Operations

Today's LGs - Build on your mental model of how data is stored and interpreted on a computer (1's and 0's).

Be able to converse about bits
Be able to shift bits
Be able to apply logical operations at a bit level
Be able to use bit operations to isolate the bits you need
Be able to convert between code points and UTF-8
Be able to convert between bits and negative numbers

Bit Sets

Set - Collection of things with no repeats or enforced order



How many bytes do I need to represent a set of 16 items?

clicker.cs.illinois.edu

Q15

~Code~
340

What operation would I need to find the union of two fruit basket sets?

clicker.cs.illinois.edu

Q16

~Code~
340

0x0C
0x09

0000 1100
0000 1001

OR

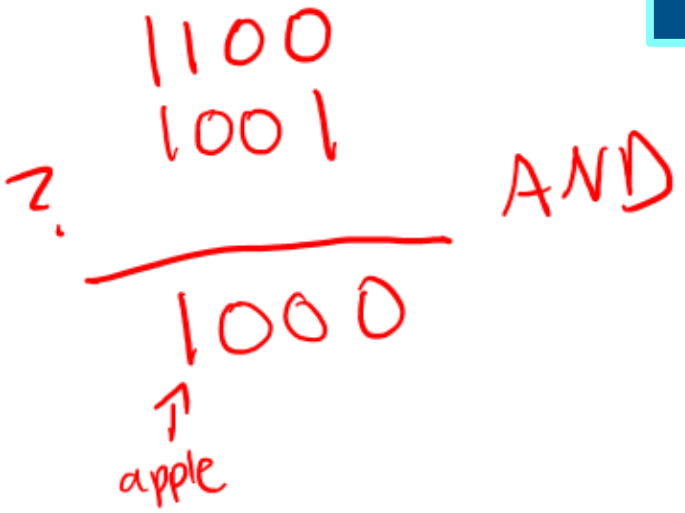
0000 1101

What operation would I need to find the intersection of two fruit basket sets?

clicker.cs.illinois.edu

Q17

~Code~
340



Bit Sets - Used as flags

```
NAME
       open, openat, creat - open and possibly create a file

LIBRARY
       Standard C library (libc, -lc)

SYNOPSIS
       #include <fcntl.h>

       int open(const char *pathname, int flags);
       int open(const char *pathname, int flags, mode_t mode);

       int creat(const char *pathname, mode_t mode);

       int openat(int dirfd, const char *pathname, int flags);
       int openat(int dirfd, const char *pathname, int flags, mode_t mode);
```

int

```
char path[PATH_MAX];
fd = open("/path/to/dir", O_TMPFILE | O_RDWR,
S_IRUSR | S_IWUSR);
```

0001 0100 = 1111

0010 1000