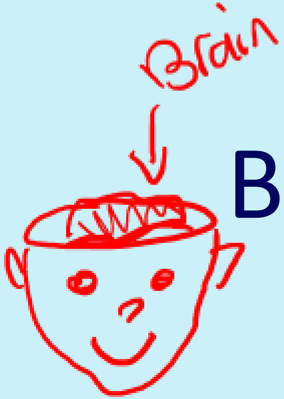
Paper up front!

$$\begin{array}{r} 101101 \\ +\ 101000 \\ \hline ? \end{array}$$

need paper?

CS 340



Building a Bit of Bit Intuition

Learning Goals

- Build intuition with bits, binary, and hexadecimal
  - A few “trick” questions coming up
    - Goal: point out misconceptions and nuances

- Learn some cool things we can do with bits to save space!

Plan for Today

- Warm Up/Review
- UTF-8
- Bit Operators
- Bit Masks
- Sets

8

To get the 0.1% extra credit, answer a majority of the clicker questions.

What is 1101 in hexadecimal?

clicker.cs.illinois.edu

Q1

~Code~
340

10	11	12	13	14	15
A	B	C	D	E	F

0b

0x

0x1101

0xD

0x44D

What is the answer in decimal?

clicker.cs.illinois.edu

Q2

~Code~
340

111

0b1101 → 0010 − 0011 = 3 → −3

+ 0b1111 → 0000 − 0001 = 1 → −1

~~1~~1100 → −4

1100 → 0011 → 0100 → 4

Given two signed chars, what is the answer in decimal?

clicker.cs.illinois.edu

Q3

~Code~
340

0x8B → 1000 1011 → -117

- 0x03 → 0000 0011 = 3

→ 1000 1000

-120

0111 0100 +1

0111 0101 = 117

0111 0111

0111 1000 = 120

Takeaways

- 0b ← means binary, 0x ← means hexadecimal
- Data is represented by 1’s and 0’s
  - The type determines how the 1’s and 0’s are interpreted
- Negative numbers are stored as the 2’s complement
  - Pros
    - No negative 0
    - No complex hardware needed for many operations

UTF-8

- A system for storing unicode characters in 1-4 bytes
  - Many c-string functions work with both ascii chars and UTF-8 codepoints

Decode the codepoint!

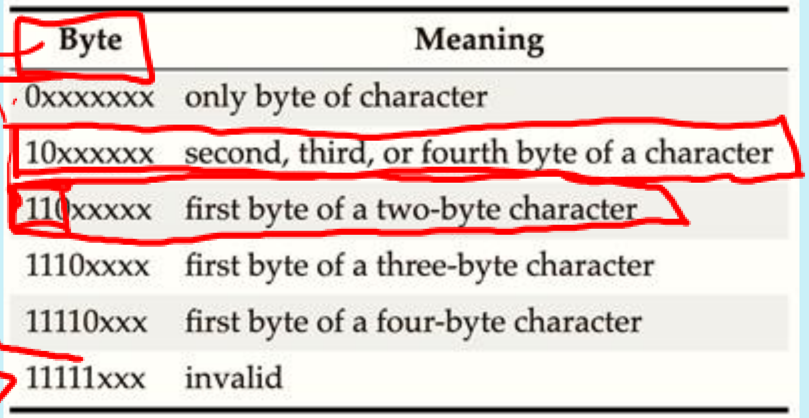
10 11 12 13
A B C D

0xD2 → 1101 0010

0xB0 → 10 11 0000

64 32 16 8 4 2 1
0 ← 10010110000

= 1,200



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

Decode the codepoint!

0x15 → 0101

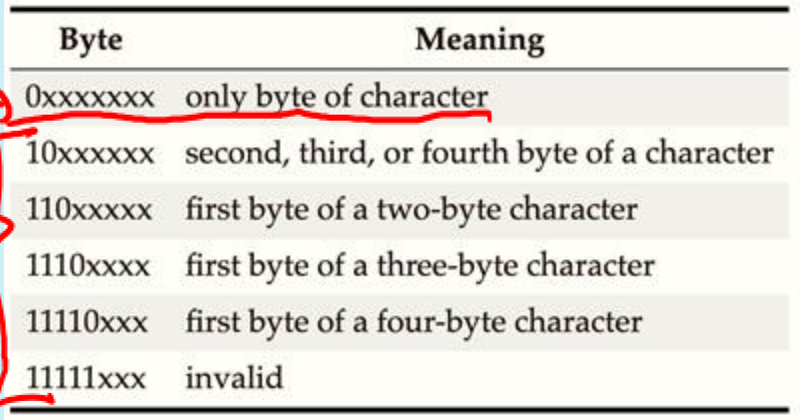
0001 0101 = 21

!

clicker.cs.illinois.edu

Q4

~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

Decode the codepoint!

clicker.cs.illinois.edu

Q5

~Code~
340

10 11 12 13 14 15
A B C D E F

0xC2F3

1100 0010

~~1111 0011~~

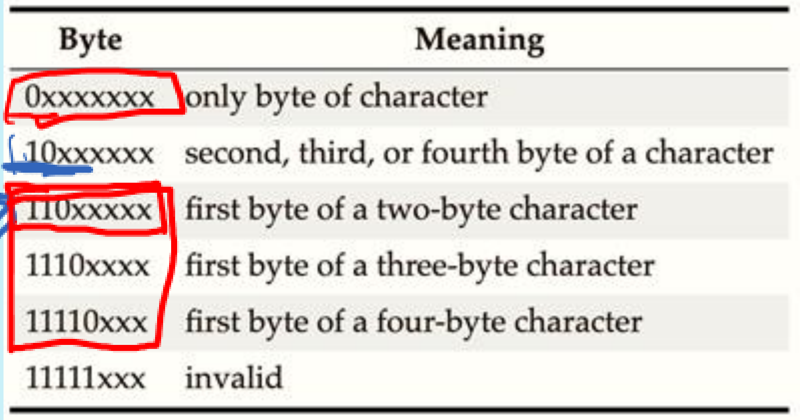
110 x x x x x

10 x x x x x x

560000

Invalid

011



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

Encode the codepoint in Bytes

clicker.cs.illinois.edu

Q6

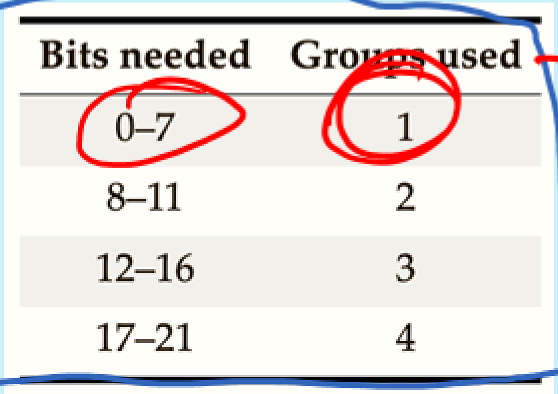
~Code~
340

25

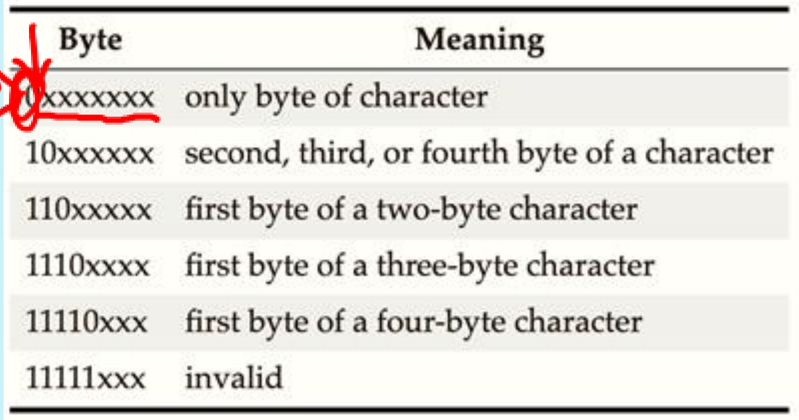
1 2 3 4 5
11001

0x19

$\boxed{0001\ 1001}$



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



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

Encode the codepoint in Bytes

clicker.cs.illinois.edu

Q7

~Code~

340

21

1111 0000 → F0

1001 1111 → 9F

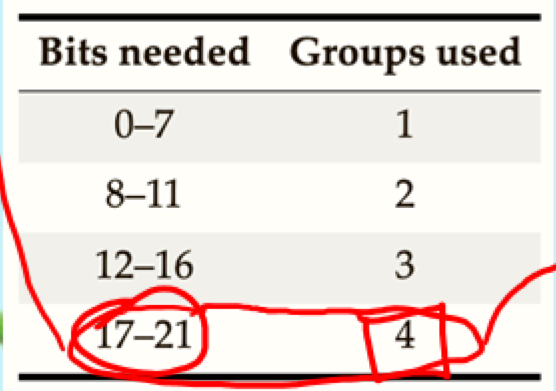
1000 1100 → 8C

1011 1101 → BD

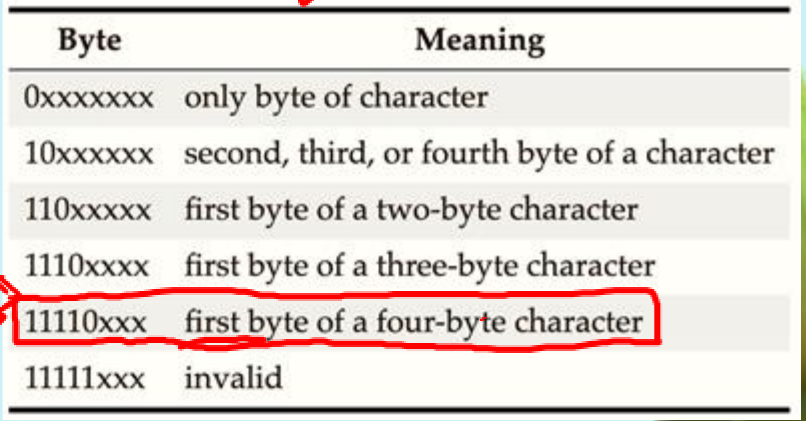
00151

U+1F33D

AKA - 0001 1111 0011 0011 1101



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



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

Bit Operators

~~to~~

A	B	or
1	1	1
1	0	1
0	1	1
0	0	0

$$\begin{array}{r} 10110 \leftarrow \text{or} \\ 01011 \\ \hline 11111 \end{array}$$

0xF6 | 0x01

Bit Operators

100101 →

& | ^ ~

AND, OR, XOR, NOT

>>

<<

~~1011~~

1000

7 << 3 = 8

0111 → 7

0001

7 >> 2 = 1

a b	
0 0	0
0 1	1
1 0	1
1 1	0

What 10 bit value does this produce?

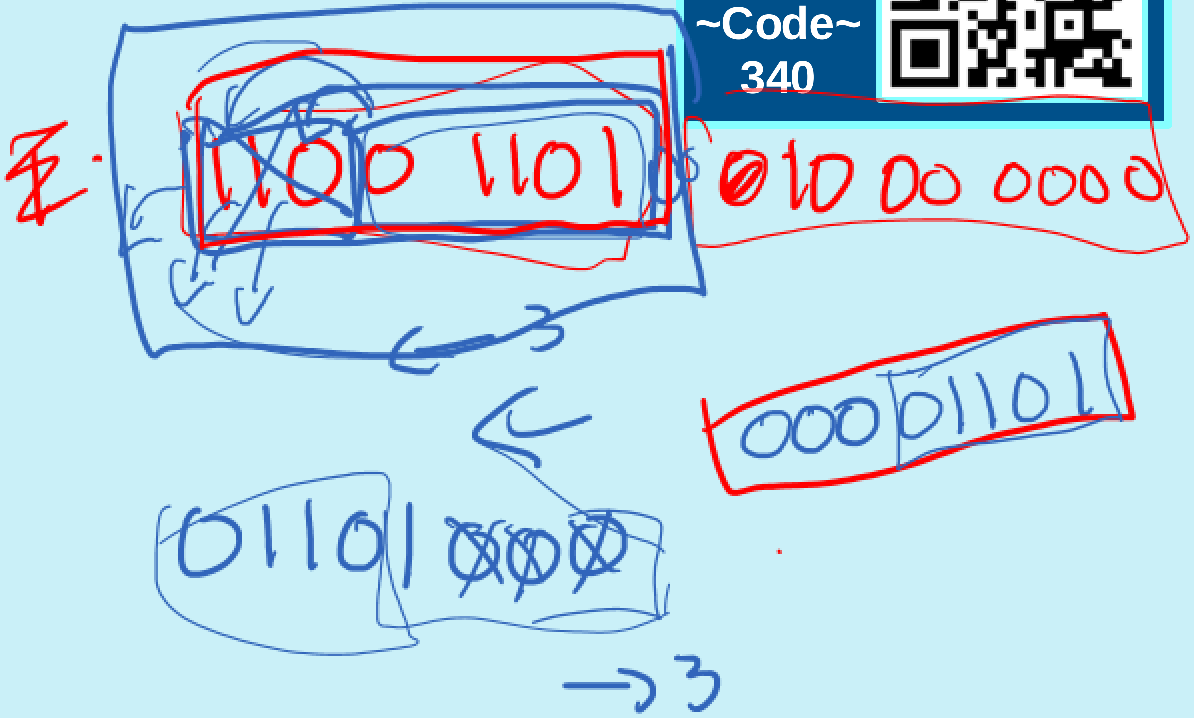
(~0) << 3

clicker.cs.illinois.edu

Q8

~Code~

340



What 10 bit value does this produce?

(((~0) << 3) ^ ((~0) << 6))

clicker.cs.illinois.edu

Q9

~Code~
340

int x = 0x16;

00010110

16 + 6 = 32

int x = 6;

0 ← 0110

int y = x << 2;

↓8

011000

16

34

clicker.cs.illinois.edu

Q10

~Code~
340

What 10 bit value does this produce?

y = 0b1001110100

not signed

1111111000

1111111000

1111000000

xor

0x030

z = (((~0) << 3) ^ ((~0) << 6)) 6))

y & z

y & z = ?

z = 0000111000

& 1001110100

0000110000

Bit Mask

0x000$\boxed{1111}$00

- Is a value used to select bits from another value.
- The 1’s in the mask indicate which bits to keep and the 0’s which to remove
- The target value is then & with the mask to get just those bits from the target value

Sets

Definition - a collection of things with no repeats and no inherit order necessary.

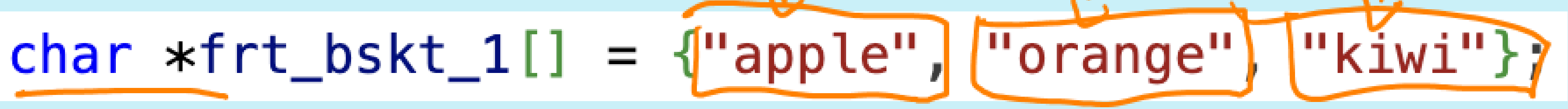
How many bytes are needed to store all the information in this fruit_bskt_1

clicker.cs.illinois.edu

Q11

~Code~ 340

8 8 8



```
char *frt_bskt_1[] = {"apple", "orange", "kiwi"};
```

orange, banana, pineapple...

How could I use less bytes to represent the same information?

A set with any combination of: apple, orange, banana, blueberry, pineapple, kiwi, and dragon

```
→ char frt[] = {'a', 'o', 'k'}; = 3 bytes
  char frt = 3;
```

dr k pin bl ba or ap

0000 0000

→ 0010 0011 1 Byte

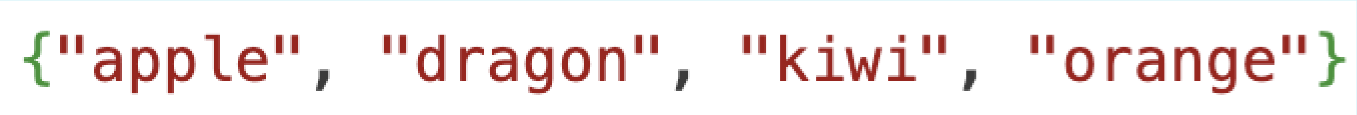
0x23

What hexadecimal number represents this combo?

clicker.cs.illinois.edu

Q12

~Code~
340



```
{"apple", "dragon", "kiwi", "orange"}
```

Bit Vector Sets

Using each bit as a flag to indicate something being present

How would I combine two sets? AKA Union

clicker.cs.illinois.edu

Q13

~Code~
340

char set1 = 0x05

char set2 = 0x17

??? set1 | set2

0000 0101

0001 0111

0001 0111

How would I see what sets have in common? AKA Intersection

clicker.cs.illinois.edu

Q14

~Code~
340

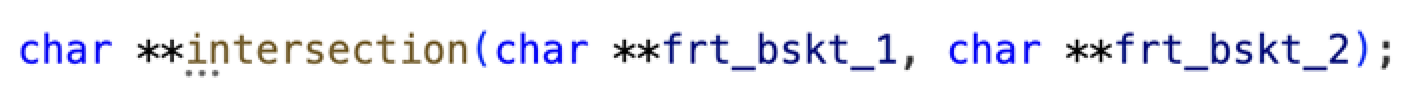
char set1 = 0x24

&

char set2 = 0x1E

???

Bit vector sets have space and time savings



```
char **intersection(char **frt_bskt_1, char **frt_bskt_2);
```

How would I see what is in set1 but not set2?

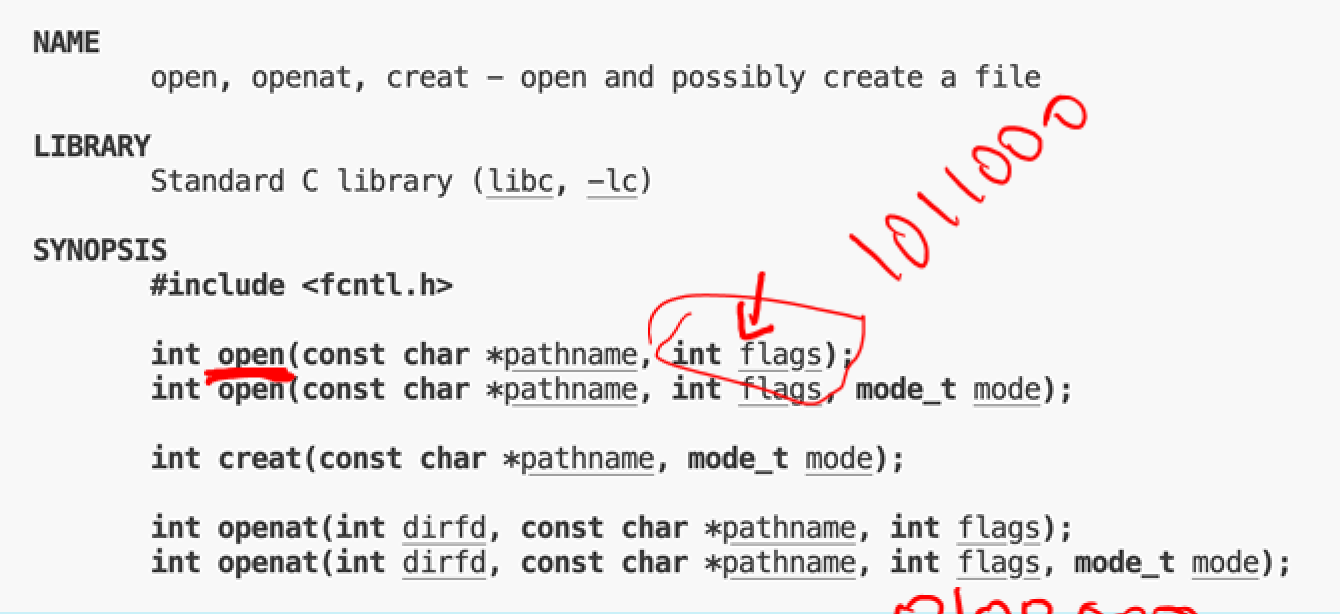
clicker.cs.illinois.edu

Q1

~Code~
340

```
char set1 = 0x1F

char set2 = 0x3C
```

Example IRL

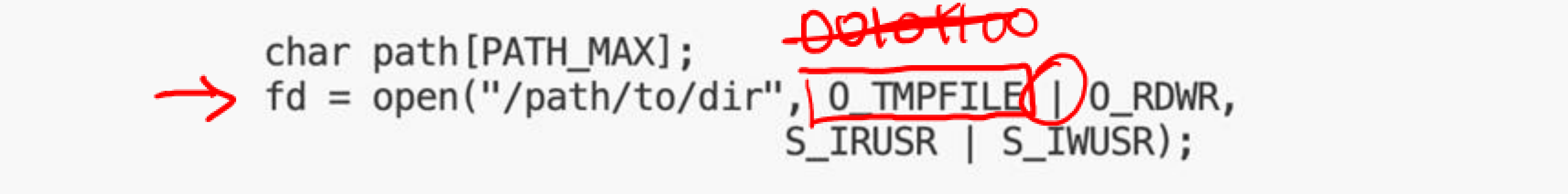


```
NAME
       open, openat, creat - open and possibly create a file

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);
```

1011000

0100 0000

~~00101100~~



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

What is coming up

- Finish HW (due before class next Thursday (12:30pm))
- Work on MP 2 (due 11:59pm Tuesday)
- Read website text for more details and information
- Tuesday’s Topic: Memory