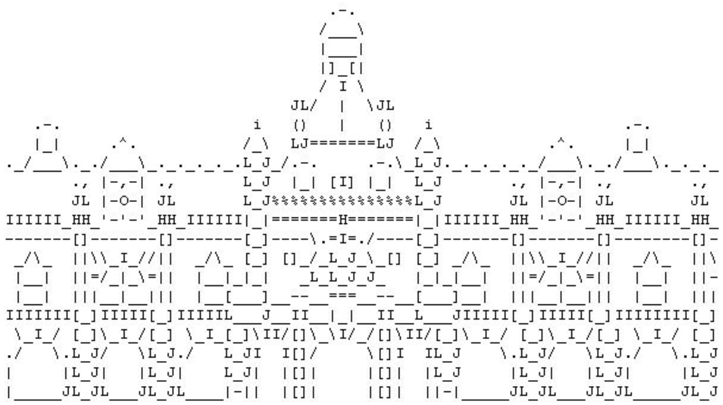
ASCII

History...

- Computers originally didn’t have a set standard of how characters were stored in memory
  - The character “A” in one computer could be stored differently than an “A” in another
- Prevented computer scientists from sharing code, among other things

History…

- As computers spread and became interconnected, a world-wide standard for character encoding as adapted - **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange (**ASCII**)



ASCII

- Uses every (positive) number in the *char* variable range to represent a character
    - Range: 0_{10} - 127_{10}
        - With extended codes going through 255
- ASCII Table: http://www.asciitable.com/

ASCII

- Can be accessed using the %c operator
  - %c is commonly associated with chars - but allows for conversions between base 10 numbers and the characters in the ASCII table
- Ex:

```
char letter_A = 65;

printf(“%c\n”, letter_A); /*Will print out “A”*/
```

ASCII Coding: Part I

- Make a program that scans in a number into a character (scanf code below), then prints out the character represented by the number
    - Restrictions: The number must be > 0 and <= 255; else , print “This character does not exist”

```
int num1;

scanf(“%d”, &num1);
```

ASCII Manipulation

- These numbers can be changed as well
    - What letter will be printed out if this code is run?

```
int letter_A = 65;

letter_A = letter_A + 10;

printf(“%c\n”, letter_A);
```

ASCII Coding: Part II, Basic Cipher

- Scan in 1 (one) character - store it as a *char* variable

- Print that variable out as a *char* and as a *int*.

A B C D E F G I L M N O P Q R S T V X Z 1 2 3 4
a c e g k l n p r t v z & x y s o m q i h f d b

ASCII Coding: Part II, Basic Cipher

- Scan in 1 (one) character - store it as a *char* variable

- ADD 10 to the value of the scanned character
- Print that variable out as a *char* and as a *int*.

ASCII Coding: Part II, Basic Cipher

- Scan in 1 (one) character - store it as a *char* variable

- ADD 10 to the value of the scanned character
- If the character is now out of the CAPITAL letter range, print “Not a capital letter” ➔ otherwise, print the *char* and *int* of the variable
  - Use the ASCII table for reference

ASCII Coding: Part II, Basic Cipher

- Scan in 2 (two) characters - store them as a *char* variable

- ADD 10 to the value of the scanned characters
- If the characters are now out of the CAPITAL letter range, print "Not a capital letter" ➜ otherwise, print the *char* and *int* of the variables
  - Use the ASCII table for reference

ASCII Coding: Part II, Basic Cipher

- Scan in an unspecified amount of characters - store them as *char* variables

- ADD 10 to the value of the scanned characters
- If the characters are now out of the CAPITAL letter range, print “Not a capital letter” ➔ otherwise, print the *char* and *int* of the variables
  - ○ Use the ASCII table for reference

A B C D E F G I L M N O P Q R S T V X Z 1 2 3 4
g k l n p r t v z & x y s o m q i h f d b a c e