5. Python built-in string-related functions (not string methods)

a. `ord()` / `chr()` – Show ASCII chart (see link on class website)
http://web.alfredstate.edu/faculty/weimandn/miscellaneous/ascii/ascii_index.html

b. Look at the codes in Bin, Oct, Dec, Hex

```
ord('x')                   →   120

bin(ord('x'))              →   '0b1111000'
oct(ord('x'))              →   '0o170'
str(ord('x'))              →   '120'
hex(ord('x'))              →   '0x78
```

c. Uppercase Letters

```
for code in range(65, 91):
    print(chr(code), end='')
  →    ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Converting an int to a string - including a different Radix

```
bin(42)                    →   '0b101010'
oct(42)                    →   '0o52'
str(42)                    →   '42'
hex(42)                    →   '0x2a'
```

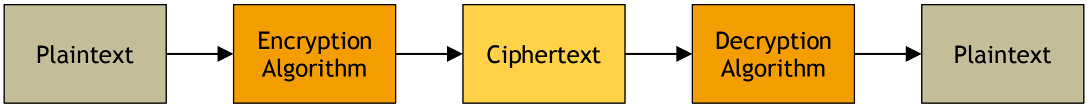
Converting a string to an int: int(string, radix)

```
fromBinStr = int('101010', 2)
fromOctStr = int('52', 8)
fromDecStr = int('42')
fromHexStr = int('2a',16)

print(fromBinStr, fromOctStr, fromDecStr, fromHexStr)
  →    42 42 42 42
```

Four ways to enter integer literals

```
binLiteral = 0b101010
octLiteral = 0o52
decLiteral = 42
hexLiteral = 0x2a

print(binLiteral, octLiteral, decLiteral, hexLiteral)
  →    42 42 42 42
```

Section 3.3, Miller 3rd ed. Encoding and Decoding Messages



Section 3.4, Miller 3rd ed. Transposition Cipher

- Rearrange the letters of the plaintext
- String of length n → n! permutations

Rail-Fence Cipher - *Encryption*

1. Group all odd-characters together
2. Group all even-characters together
3. Concatenate the two

Example

`'connecticut shoreline'` → length is 21 (odd length)

c	o	n	n	e	c	t	i	c	u	t		s	h	o	r	e	l	i	n	e
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

oddChars: `'onciu hrln'`

evenChars: `'cnetctsoeie'`

ciphertext = oddhars + evenChars = `'onciu hrlncnetctsoeie'`

Example - Even Length Strings
`'abababab'` → length is 8

a	b	a	b	a	b	a	b
0	1	2	3	4	5	6	7

oddChars: `'bbbb'`
evenChars: `'aaaa'`
Ciphertext = oddChars + evenChars = `'bbbbaaaa'`

Example - Odd Length Strings
`'ababababc'` → length is 9

a	b	a	b	a	b	a	b	c
0	1	2	3	4	5	6	7	8

oddChars: `'bbbb'`
evenChars: `'aaaac'`
Ciphertext = oddChars + evenChars = `'bbbbaaaac'`

→ evenChars might be 1 longer that oddChars

See 03-03-transpositionCipher.py - 3 versions of `scramble2Encrypt()`

Rail-Fence Cipher - *Decryption* starting with ciphertext

1. Separate odd and even characters

```
ciphertext = 'bbbbaaaac'
```

- **oddChars:** the first `len(ciphertext) // 2` characters
- **evenChars:** the rest of the characters

```
halfLength = len(ciphertext) // 2
oddChars = ciphertext[ : halfLength ]
evenChars = ciphertext[ halfLength : ]
```

2. Go back and forth between evenChars and oddChars taking next character

3. If len(evenChars) > len(oddChars), take one more evenChar

See modified 03-03-transpositionCipher.py - `scramble2Decrypt()`

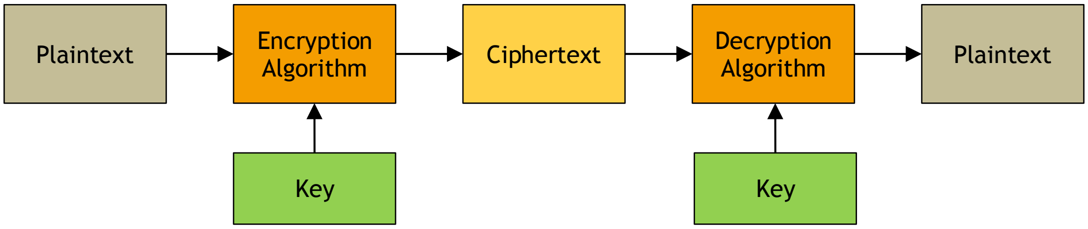
Section 3.5, Miller 3rd ed. Substitution Cipher

Encryption: Substitute each character with a different character

Decryption: Reverse the substitution

Use a **key** to tell which letter is substituted for which.

Alphabet:	All possible chars the plaintext can have
Key:	Must be a ***permutation*** of the alphabet



Small Toy Example

Alphabet:	`'abcdef'`
Key:	`'dabfce'`
Plaintext:	`'bed'`

Alphabet	a	b	c	d	e	f
key	d	a	b	f	c	e
	0	1	2	3	4	5

Encryption:

Plaintext	Index i in alphabet	Ciphertext: key[i]
b	1	a
e	4	c
d	3	f

Ciphertext: `'acf'`

Decryption:

Ciphertext	Index i in key	Decrypted: Alphabet[i]
a	1	b
c	4	e
f	3	d

Decrypted: `'bed'`