Week 4 - Monday

COMP 1800

Last time

- What did we talk about last time?
- Reading input with **`input()`**
- Conversion functions
  - **`int()`** to turn things into integers
  - **`float()`** to turn things into floating-point numbers
  - **`str()`** to turn things into strings
- Work time for assignments

Questions?

Cryptography

Cryptography

- "Secret writing"
- The art of encoding a message so that its meaning is hidden
- **Cryptanalysis** is breaking those codes

Encryption and decryption

- **Encryption** is the process of taking a message and encoding it
- **Decryption** is the process of decoding the code back into a message
- A **plaintext** is a message before encryption
- A **ciphertext** is the message in encrypted form
- A **key** is an extra piece of information used in the encryption process

Transposition cipher

- In a transposition cipher, the letters are reordered but their values are not changed
- Any transposition cipher is a permutation function of some kind

Example: Rail Fence Cipher

- In the rail fence cipher, a message is written vertically along a fixed number of "rails," wrapping back to the top when the bottom is reached
- To finish the encryption, the message is stored horizontally
- This is also known as a **columnar transposition**
- Encryption of "WE ARE DISCOVERED, FLEE AT ONCE" with three rails:

W	R	I	O	R	F	E	O	E
E	E	S	V	E	L	A	N	X
A	D	C	E	D	E	T	C	J

- **Ciphertext:** WRIORFEOEEESVELANXADCEDETCJ

Even-odd shuffle

- We can do a simple transposition cipher by:
  - Putting all characters with even locations in a string
  - Putting all characters with odd locations in a string
  - Concatenating the odd string with the even string
- Example:
  - Plaintext: **`'My hovercraft is full of eels'`**
  - Even characters: **`'M oecati ulo es'`**
  - Odd characters: **`'yhvrrf sfl fel'`**
  - Full ciphertext: **`'yhvrrf sfl felM oecati ulo es'`**

Iterating over a string

- So far, we've only talked about using a **`for`** loop with a **`range()`** function
- We can use that to iterate over all the characters in a string by using the length of the string and indexing into it

```
for i in range(len(text)):
  print(text[i])
```

- We can also iterate over all the characters in a string directly with a **`for`** loop

```
for letter in text: # equivalent to loop above
  print(letter)
```

Even-odd encryption in Python

- Algorithm:
  - Loop over all characters
    - Concatenate characters with even locations onto a special even string
    - Concatenate characters with odd locations onto a special odd string
  - Concatenate the odd string with the even string and return the result

```
def evenOddEncrypt(plaintext):
```

Even-odd decryption in Python

- We can reverse the process to decrypt the ciphertext
- Algorithm:
  - Find half of the length
  - Slice the ciphertext into the odd characters before the halfway point
  - Slice the ciphertext into the even characters after the halfway point
  - Loop up to half the length, concatenating from both the even and odd characters into the result
  - If there's an extra even character, concatenate it
  - Return the result

```
def evenOddDecrypt(ciphertext):
```

Operations with Characters

Characters?

- So far, the only type we've talked about that contains text is the string type
- However, we sometimes need to be able to do operations on individual characters
- Specifically, it's nice to know where a character falls numerically in the giant list of all the possible characters

Sometimes it's useful to know the number

- We can convert a string with a single character in it into an integer with the **`ord()`** function

```
number = ord('a') # number contains 97
```

- It can also be useful to get the offset from a starting point, such as the beginning of the alphabet

```
letter = 'r'
number = ord(letter) - ord('a') + 1
# number is 18
```

What if you have a number?

- If you know the numerical value of a character, you can convert that number back into a string using the **`chr()`** function

```
letter = chr(97)   # letter contains 'a'
```

- As before, you can use an offset point

```
letter = chr(ord('A') + 13 - 1)
# letter contains 'M', the 13th letter
```

ASCII table

- Everything in the computer is 1's and 0's
- Each character has a number associated with it
- These numbers are sometimes listed in tables
- The ASCII table only covers 7 bits of information (0-127)
- **NEVER EVER TYPE THESE NUMBERS IN CODE**
- What's important to know:
  - All the characters are numbered
  - The uppercase letters are contiguous
  - The lowercase letters are contiguous
  - The numerical digits are contiguous

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Shift Cipher

Definition

- A shift cipher encrypts a message by shifting all of the letters down in the alphabet
- Using the Latin alphabet, there are 26 (well, 25) possible shift ciphers
- We can model a shift cipher by thinking of the letters A, B, C, … Z as 0, 1, 2, … 25
- Then, we let the key ***k*** be the shift
- For a given letter with value ***x***:

encrypt (***x***) = (***x*** + ***k***) mod 26

Example: Caesar Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- ***E***("KILL EDWARD") = "NLOO HGZDUG"
- What is ***E***("I DRINK YOUR MILKSHAKE")?
- What is ***D***("EUHDNLWGRZQ")?
- This code was actually used by Julius Caesar who used it to send messages to his generals

Shift encryption in Python

- Algorithm:
  - Loop over all characters
    - Convert character to ASCII value
    - Convert ASCII value to a value from 0-25 by subtracting the value of `'A'`
    - Add the key to the result
    - Compute the result modulus 26 (which makes numbers bigger than 25 wrap around)
    - Add back the value of `'A'` to turn a value from 0-25 back into an ASCII value
    - Turn the ASCII value back into a character and concatenate it onto the ciphertext
  - Return the ciphertext

```
def shiftEncrypt(plaintext, key):
```

Shift decryption in Python

- Reversing the process to decrypt the ciphertext is simple
- All we need to do is "encrypt" the ciphertext with the negation of the key we used to encrypt
- For example, if we encrypted with a key of 7, we can decrypt by encrypting with a key of -7
- Our decrypt function should simply call the encrypt function with a negative key

```
def shiftDecryption(ciphertext, key):
```

Quick note

- Our implementation expects all input characters to be from `'A'` up to `'Z'`
- That's why subtracting `ord('A')` will make the values be between 0 and 25
- Inputting strings that contain characters other than uppercase letters (e.g. digits, lowercase letters, punctuation) will cause strange results

Upcoming

Next time…

- Substitution ciphers
- Generating a random key

Reminders

- Read Sections 3.5 and 3.6 of the textbook
- Work on Assignment 3