Algorithms and Data Structures for Data Science
lab_cipher

CS 277
Brad Solomon

February 9, 2024

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Department of Computer Science

Learning Objectives

Practice manipulating items and list indices

Write open-ended code with multiple valid algorithmic approaches

Learn fun trivia about cryptography

Substitution Ciphers

Plaintext:	A	B	C	D	E	U
Ciphertext:	C	I	P	H	E	R

BADDUDE

Caesar Cipher

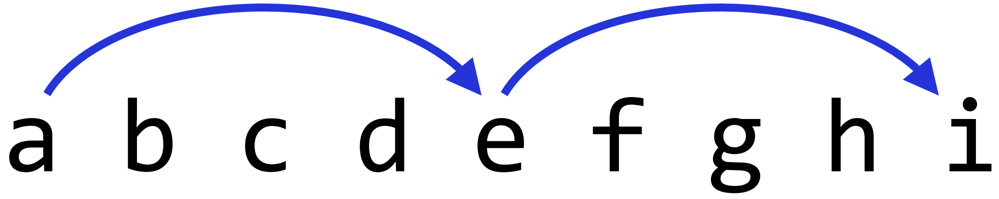
Plaintext:	A	B	C	D	E	U
Ciphertext:	E	F	G	H	I	Y

BADDUDE

Caesar Cipher

Plaintext:	A	B	C	D	E	U
Ciphertext:	E	F	G	H	I	Y

BADDUDE



a b c d e f g h i

Caesar Cipher Encode

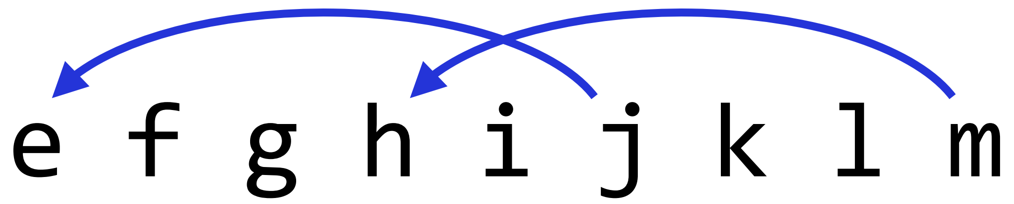
The Caesar cipher takes as input a text and an offset integer

```
Text: ABCD, Offset: 2
```

Caesar Cipher Decode

Given an encrypted string and an offset, we can decode the message

```
Text: mjqqtz, Offset: 5
```



Caesar Cipher (Encode and Decode)

1. CaesarEncode / CaesarDecode must work for positive and negative

2. Make sure you are using the appropriate alphabet!

Vigenere Cipher

The Vigenere Cipher takes as input two strings, a text and a key.

```
Text: ’dddbbd’, Key: ‘ba’
```

Vigenere Cipher

The Vigenere Cipher takes as input two strings, a text and a key.

Each letter in the key is associated with a number

```
Text: ’dddbbd’, Key: ‘ba’
```

Offset:

A	0
B	1
C	2
D	3
…	…

Vigenere Cipher

The Vigenere Cipher takes as input two strings, a text and a key.

Each letter in the key is associated with a number

To encode the message, increment text and key indices

(Loop back through the key when necessary)

Offset:

A	0
B	1
C	2
D	3
…	…

`Text: ’dddbbd’, Key: ‘ba’`

Text:	d	d	d	b	b	d
Key:	b	a	b	a	b	a

Code:

Vigenere Cipher

While not necessary, you may find it easier to handle using a matrix!

‘badace’, ‘cabe’

Plaintext

	A	**B**	**C**	**D**	**E**
A	A	B	C	D	E
B	B	C	D	E	A
C	C	D	E	A	B
D	D	E	A	B	C
E	E	A	B	C	D

Key

Vigenere Cipher Decode

To decode given the encryption and key, trace backwards in matrix

Find encoded character in key row and identify column letter

```
‘cceda’, ‘bae’
```

Plaintext

	A	**B**	**C**	**D**	**E**
A	A	B	C	D	E
B	B	C	D	E	A
C	C	D	E	A	B
D	D	E	A	B	C
E	E	A	B	C	D

Key

Coding the lab

1) Create one or more lists of all allowed characters

2) Consider how you can swap all characters using a single integer

3) Consider how you can swap all characters using a single character

4) Extend single character solution to a full Vigenere encoding

5) Consider how to reverse both ciphers

Python Strings

Python strings have built-in lists for sets of characters

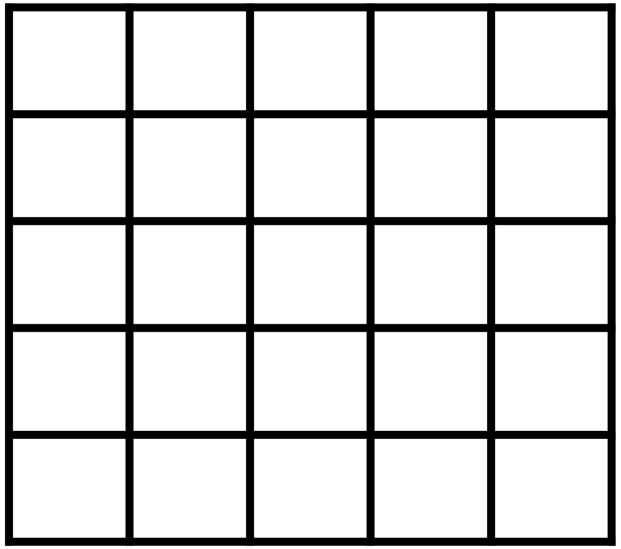
Well supported languages often can make your life easier!

```
alpha = list(string.ascii_lowercase)
print(alpha)
# ['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']

whitespace = ' \t\n\r\v\f'
ascii_lowercase = 'abcdefghijklmnopqrstuvwxyz'
ascii_uppercase = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
ascii_letters = ascii_lowercase + ascii_uppercase
digits = '0123456789'
hexdigits = digits + 'abcdef' + 'ABCDEF'
octdigits = '01234567'
punctuation = !"#$%&'()*+, -./:;<=>?@[\]^_`{|}~
printable = digits + ascii_letters + punctuation
+ whitespace
```

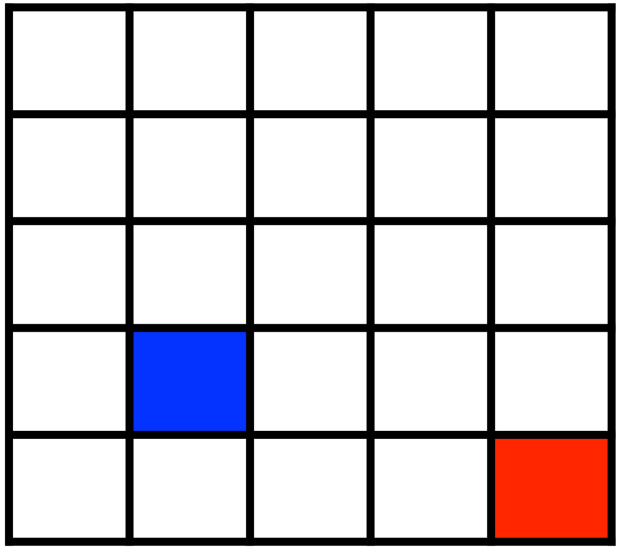
Programming Toolbox: Multidimensional Lists

How can we make a matrix in Python?



Programming Toolbox: Multidimensional Lists

How is a matrix in Python indexed?



Programming Toolbox: Multidimensional Lists


```
outerList = []

for i in range(5):
    innerList = []

    for j in range(5):
        innerList.append(i+j)

    outerList.append(innerList)

print(outerList)

print(outerList[3][1])
```

Programming Toolbox: Multidimensional Lists

0	1	2	3	4
1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8

```
outerList = []

for i in range(5):
    innerList = []

    for j in range(5):
        innerList.append(i+j)

    outerList.append(innerList)

print(outerList)

print(outerList[3][1])
```