**Level 1: Red Pill, Blue Pill**

**Decode the following message:**

Ymnx nx dtzw qfxy hmfshj. Fkyjw ymnx, ymjwj nx st yzwsnsl gfhp.

Dtz yfpj ymj gqzj unqq—ymj xytwd jsix, dtz bfpj zu ns dtzw gji fsi gjqnjaj bmfyjajw dtz bfsy yt gjqnjaj.

Dtz yfpj ymj wji unqq—dtz xyfd ns Btsijwqfsi, fsi N xmtb dtz mtb ijju ymj wfggny mtqj ltjx.

Wjrjrgjw: fqq N'r tkkjwnsl nx ymj ywzym. Stymnsl rtwj.

Hint #1: The code is a simple letter shift cypher

Hint #2: <http://practicalcryptography.com/ciphers/simple-substitution-cipher/>

* What single letter is always capitalized?
* Identify the 2 letter words and use the “frequency analysis” tables
* Identify the most common letters and use the “frequency analysis” tables

Final Hint / Solution:

* <https://www.youtube.com/watch?v=ytftrd6rxps>

Final Question:

* What is the shift amount? How many letters is the coded message shifted by?

5

**Write the Decoded Message below:**

This is your last chance. After this, there is no turning back. You take the blue pill—the story ends, you wake up in your bed and believe whatever you want to believe. You take the red pill—you stay in Wonderland, and I show you how deep the rabbit hole goes. Remember: all I'm offering is the truth. Nothing more.

**Level 2: ASCII Code**

1. What does the ASCII acronym stand for?

American Standard Code for Information Interchange.

1. What is ASCII code used for?

It is a code for representing 128 English characters as numbers, with each letter assigned a number from 0 to 127. For example, the ASCII code for uppercase M is 77.

1. Why can’t computers just use the letters we type or the letters used by humans when we write things down?  
   Computers only understand numbers.
2. How do computers communicate with people who speak different languages and use different alphabets? What is used instead of the ASCII code table?

They use Unicode instead, which can create characters for other languages.

1. Write a Python program that uses the ord() function to convert some text letters into ASCII code numbers. Print the numbers to the console screen. Provide your program listing below.

ord('a')

97

ord('b')

98

ord('c')

99

ord('d')

100

ord('e')

101

ord('f')

102

ord('h')

103

ord('i')

104

ord('j')

105

1. Write a Python program that uses the chr() function to convert some ASCII code numbers into text letters. Print the letters to the console screen. Provide your program listing below.

chr(1)

'\x01'

chr(2)

'\x02'

chr(3)

'\x03'

chr(4)

'\x04'

chr(5)

'\x05'

chr(6)

'\x06'

chr(7)

'\x07'

chr(8)

'\x08'

chr(9)

'\t'

chr(10)

'\n'

chr(11)

'\x0b'

chr(12)

'\x0c'

chr(13)

'\r'

chr(14)

'\x0e'

chr(15)

'\x0f'

chr(16)

'\x10'

chr(17)

'\x11'

chr(18)

'\x12'

chr(19)

'\x13'

chr(20)

'\x14'

chr(21)

'\x15'

chr(22)

'\x16'

chr(23)

'\x17'

chr(24)

'\x18'

**Level 3: Coding Program**

Work with a partner to create a program to encrypt and decrypt messages using a numerical code.

1. To encrypt the message your program should do the following:

a.    Read a message contained in a text file

b.    Convert the text characters (letters in the words) in the text file to ASCII numbers using the ord() function

c.    Modify the ASCII numbers using a mathematical function (e.g. addition or multiplication)

d.    Write the modified ASCII numbers to a binary file

e.    The binary file is your encrypted message

Original Code:

file = open("file.txt", "w")

file.write(chr(115))

file.write(chr(97))

file.write(chr(109))

file.write(chr(112))

file.write(chr(108))

file.write(chr(101))

file.write(chr(32))

file.write(chr(116))

file.write(chr(101))

file.write(chr(120))

file.write(chr(116))

file.close

file = open("file.txt", "r")

print(file.read())

file.close

Modified Code:

inFH = open("resources/plainMessage.txt","r")

while True :

inChar = inFH.read(1)

if inChar == "" :

  break

asciiNum = ord(inChar)

print(asciiNum)

outFH = open("resources/encodedMessage.bin","wb")

while True :

inChar = inFH.read(1)

if inChar == "" :

  break

asciiNum = ord(inChar)

codedNum = asciiNum \* 12

print(codedNum)

outFH.write(bytes(codedNum))

outFH.close()

base = 5

def shift(char) :

val = ord(char)

if ((val >= ord('a')) and (val <= ord('z'))) :

  val = val + base

  if (val > ord('z')) :

    val = val - ord('z') + ord('a') - 1

if ((val >= ord('A')) and (val <= ord('Z'))) :

  val = val + base

  if (val > ord('Z')) :

    val = val - ord('Z') + ord('A') - 1

return chr(val)

while True :

inChar = inFH.read(1)

if inChar == "" :

  break

outFH.write(shift(inChar))

outFH.close()

2. To decrypt your message your program should do the following:

a.    Read the encrypted message contained in the binary file

b.    Convert the modified numbers to ASCII numbers using a reverse mathematical function (e.g. subtraction or division)

c.    Convert the ASCII numbers to text characters using the chr() functions

d.    Write the text characters to a text file

e.    The text file is your decrypted file.

while True :

inChar = inFH.read(1)

if inChar == "" :

  break

asciiNum = ord(inChar)

codeNum = asciiNum \* 5

byteNum = (codeNum).to\_bytes(2, byteorder='big')

outFH.write(byteNum)

newCode = int.from\_bytes(byteNum, byteorder='big')

newAscii = int(newCode / 5)

newChar = chr(newAscii)

print("Char: ",inChar," ASCII: ", asciiNum,"Coded: ",codeNum," Bytes: ",byteNum, " ASCII: ",newAscii," Char: ", newChar)

outFH.close()

3. Demo your program to Mr. Nestor

4. Provide a listing of the following:

a.    Your encryption program

b.    Your decryption program

c.    The contents of your original message text file

d.    A dump of the contents of your binary encrypted file

e.    The contents of your decrypted message text file

Original message: “hi”

Encrypted message: “Ȉȍ'