**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?

**Write the Decoded Message below:**

**Level 2: ASCII Code**

1. What does the ASCII acronym stand for?

American stand code for information interchange

1. What is ASCII code used for?

The code is used for representing 128 English character as numbers

1. Why can’t computers just use the letters we type or the letters used by humans when we write things down?

Computer only understand numbers because it sees everything as a number

1. How do computers communicate with people who speak different languages and use different alphabets? What is used instead of the ASCII code table?

Unicode is used instead of ASCII

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.

Python 3.6.1 (default, Dec 2015, 13:05:11)

[GCC 4.8.2] on linux

  ord('h')

104



 chr(104)

'h'

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.

Python 3.6.1 (default, Dec 2015, 13:05:11)

[GCC 4.8.2] on linux

 chr(104)

'h'



  ord('h')

104

**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:
   1. Read a message contained in a text file
   2. Convert the text characters (letters in the words) in the text file to ASCII numbers using the ord() function
   3. Modify the ASCII numbers using a mathematical function (e.g. addition or multiplication)
   4. Write the modified ASCII numbers to a binary file
   5. The binary file is your encrypted message
2. To decrypt your message your program should do the following:
   1. Read the encrypted message contained in the binary file
   2. Convert the modified numbers to ASCII numbers using a reverse mathematical function (e.g. subtraction or division)
   3. Convert the ASCII numbers to text characters using the chr() functions
   4. Write the text characters to a text file
   5. The text file is your decrypted file.
3. Demo your program to Mr. Nestor
4. Provide a listing of the following:
   1. Your encryption program

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

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

while True :

inChar = inFH.read(1)

if inChar == "" :

break

asciiNum = ord (inChar)

##

## Simple Encoding - divide by 12

##

codedNum = asciiNum \* 5

##

print(codedNum)

outFH.write(bytes(codedNum))

outFH.close()

* 1. Your decryption program

1. outFH = open("resources/plainMessage.txt","r")
2. inFH = open("resources/encodedMessage.bin","wb")
3. while True :
4. inOrd = inFH.read(1)
5. if inOrd == "" :
6. break
7. asciiNum = char (inOrd)
8. ##
9. ## Simple Encoding - divide by 12
10. ##
11. codedNum = asciiNum / 5
12. ##
13. print(decodedNum)
14. outFH.write(bytes(decodedNum))
15. outFH.close()
    1. The contents of your original message text file

hi

* 1. A dump of the contents of your binary encrypted file

520

525

* 1. The contents of your decrypted message text file

hi