II. CONTENTS

III. INTRODUCTION

"ROW TRANSPOSITION CIPHER IMPLEMENTATION"

Encryption is defined as a process of expressing data in the form of a code (encoding). This can also be mentioned as converting Plaintext to Ciphertext.

Decryption is the process of decoding the encoded data. Converting the ciphertext into plain text. This process requires a key that we used for encryption. We require a key for encryption. There are two main types of keys used for encryption and decryption. As part of this project, we are instructed to perform Encryption and Decryption on a Ciphertext in row transposition.

We chose the python programming language and implemented the tasks using two methods:

1. Encryption and Decryption data type using dictionaries and arrays.
2. Encryption and Decryption data type using matrix and arrays.

As a team, we plan our activities regarding any person's skills, and we decide to split our tasks to accomplish the 2nd assignment on time. During the process, we worked into two different approaches in code. In the beginning, we were thinking in handy out only one approach, but in the end, the decision was to have the two approaches in one code in order to show our effort and interest in the lecture. One code was using data type dictionaries and arrays, and another was done with arrays and matrix. We also create a menu that allows the user to choose what feature to use.

IV. TASKS

Question 1:

Using any programming language of your choice, implement the Row Transposition Cipher encryption algorithm. Encryption using Python programming language

Answer :

```
def Encrypted(combine_dict, msg_filled):
    #create an array and its length is as same as that of key
    Array = [""] * key_length
    for column_length in range (key_length): #Run how many times in column
        msg_length = column_length
        while msg_length < len(msg_filled):
            # Store message string value to array
            Array[column_length] += msg_filled[msg_length]
            # The message pointer adds the key length
            msg_length = msg_length + key_length

        # Create a dictionary that has pairs of key and value
        for x in range (key_length):
            dict_x = {key[column_length] : Array[column_length]}
            combine_dict.update(dict_x)
            x +=1
    return combine_dict
def Sorted(combine_dict, sorted_str):
    #Sort Key alphabetical order in dictionary, and this also sorted the pair of
value
    sorted_keys = sorted(combine_dict)
    #Store sorted key corresponding value to empty string "sorted_str"
    sorted_dict ={}
    for w in sorted_keys:
        sorted_dict[w] = combine_dict[w]
        sorted_str += sorted_dict[w]
    print(sorted_str)
    return sorted_str
def msg_extend(lowecase_msg,key_length):

    # calculate remainder empty spaces
    msg_list = list(lowecase_msg)
    row = len(lowecase_msg) % key_length
    # Padding empty spaces with X in the last row
    msg_list.extend('X' * int(key_length - row))
    msg_filled = ''.join(msg_list)
    return msg_filled
```

```
print("Encrypt The message M")
msg = input()
print("Enter the Key of w ") # The word w is a key
key = input()
key_length = len(key) # its length is k
#create an empty dictionary
combine_dict = {}

# msg.lower() means all input message covert to lowercase
Encrypted(combine_dict, msg_extend(msg.lower(),key_length))

print("Unencrypted Message is M =", msg)
print("Key is w = ", key)
print("Encrypted Message is C = ", Sorted(combine_dict, sorted_str = ""))
```

My whole picture of the Row Transposition Encrypt algorithm with Python programming language is that it creates an array whose length is the same as the key's length—padding the lowercase plaintext with the character 'X' in the last row.

Secondly, pair the key and the code's arrays to form a dictionary. Because dictionary has a property {key: Value}, this gives an approach to follow-up code working (Python w3schools).

After creating the dictionary, sort the dictionary of keys in alphabet order to sort the corresponding value simultaneously.

Eventually, convert the sorted dictionary and extract the dictionary value, which forms the encrypted message string and print out.

Take a simple example; if the key is NYIT, the plaintext message is How Are You. The given table shows the encrypted process.

Key = NYIT, the key length is four, so the code creates the exact size of the row.

Step 1:

Lowercase the plaintext

N	Y	I	T
h	o	w	
a	r	e	
y	o	u	X

Step 2:

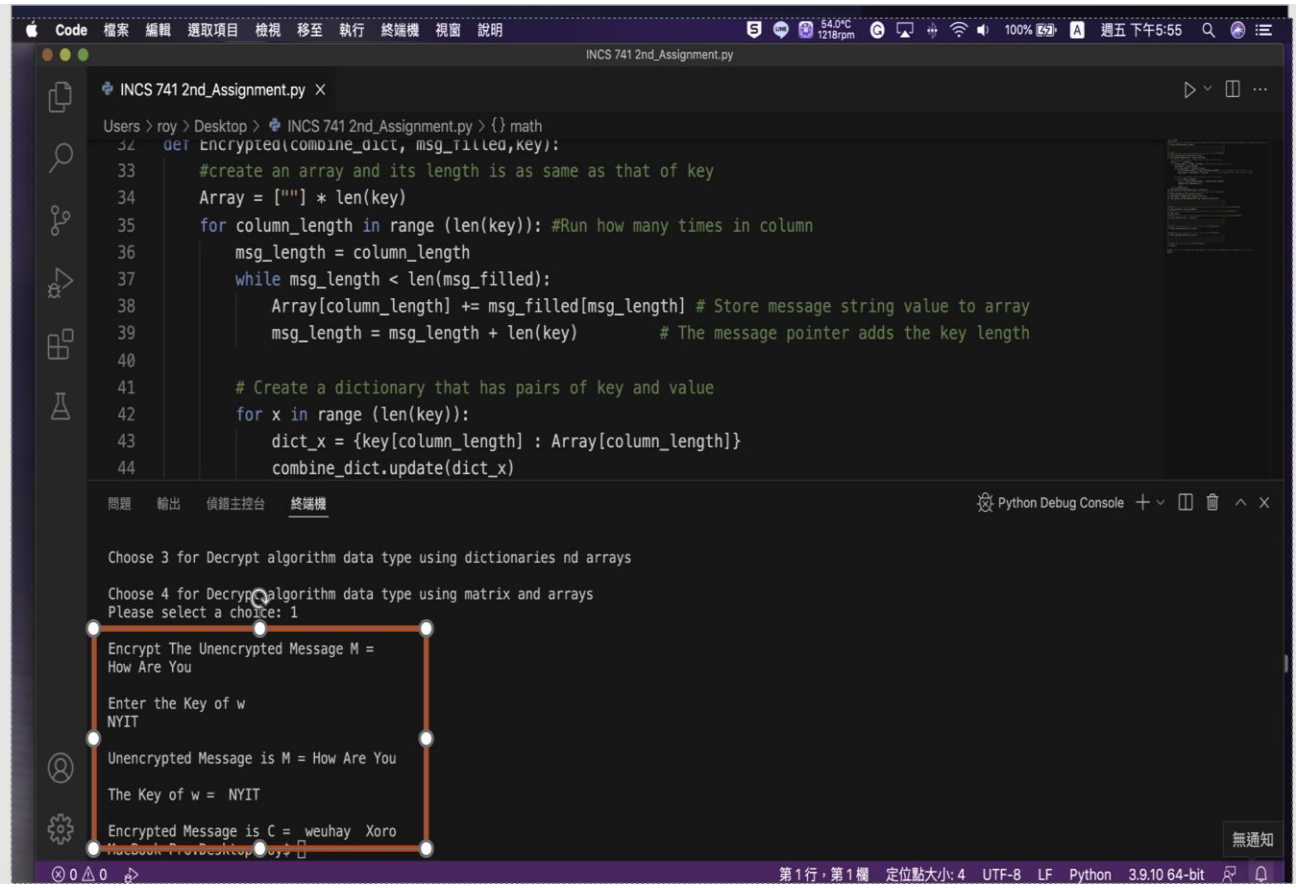
dictionary {‘N’: ‘hay, ‘‘Y’: ’oro,’ ‘I’: ‘weu’, ‘T’: ‘ X’}

After sorting the dictionary = {‘I’: ‘weu’, ‘N’: ‘hay ‘, ‘T’: ‘ X’, ‘Y’: ’oro’ }

Step 3:

Get the dictionary value = weuhay Xoro, which is the ciphertext.

The screenshot below shows the result and meets the Q1 requirement.



Question 2:

What will be the decryption algorithm for a Row Transposition Cipher? Write down the pseudocode for the algorithm.

Answer :

Method 1->

Decryption pseudocode with data type dictionaries and arrays:

Main Decrypted algorithm with dictionaries arrays Function():

    Input an encrypted message

    Input a key

    Calculate a maximum number of messages in a raw

    Create an empty dictionary

    Call Decrypted function(empty dictionary, encrypted message, key, raw numbers):

    print Encrypted message

    print key)

    print Call Decrypted sorted function(dictionary, key, raw numbers, empty string)

Decrypted function(empty dictionary, encrypted message, key, raw numbers):

    Create a key length of empty array

    Create a sorted key

    for run x times of key length

        for run y times based on the calculation of maximum messages in a raw

            store accumulately message to array

            pair the array with the sorted key and update to the empty dictionary

    return dictionary

Decrypted sorted function(dictionary, key, raw numbers, empty string):

    Created a maximum message length of array

    for run x times in range of key length

Get the dictionary item of original key[x] and store accumulately to the empty string

for run y times based on the calculation of maximum messages in a raw

counter is equal to y

while counter < length of empty string:

store accumulately the string value to array

counter add accumulately the calculation of maximum messages in a raw

return array is called a plaintext

Explaination: The concepts and steps of decryption pseudocode with dictionaries and arrays are that the encrypted message wich pairs with a sorted key, and form into a dictionary.

Step 1: Sort key

Step 2: Use sorted pair with encrypted message and each key pair with the maximum length of messages in a raw in order to form into a dictionary

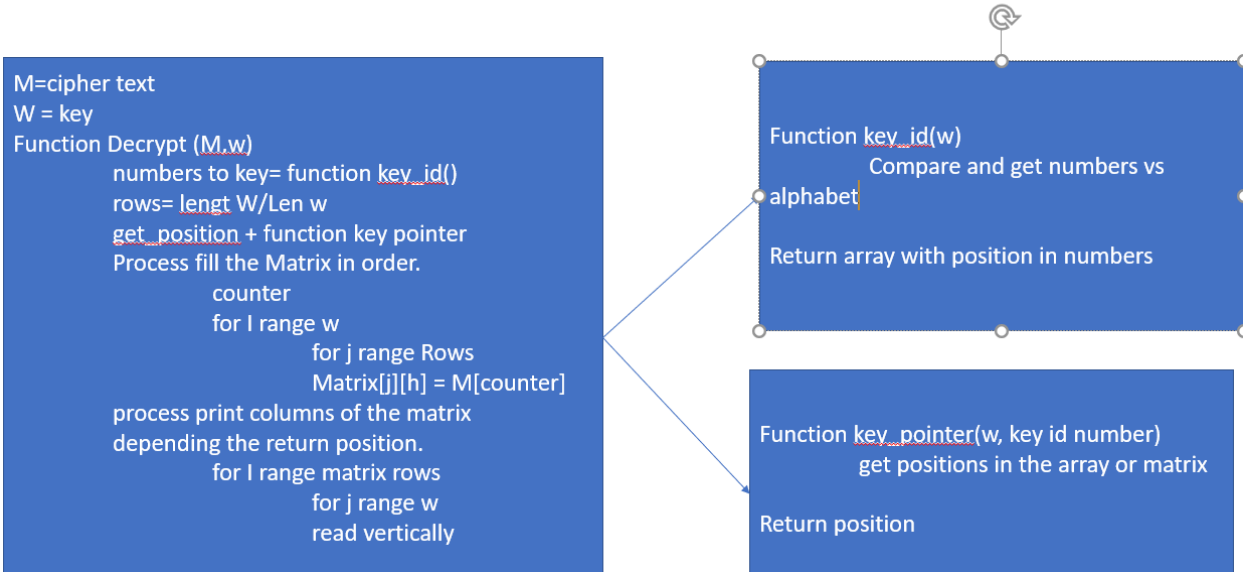
Step 3: Get the dictionary value of original key index and store its value to the string

Step 4: Print the string which is a plaintext

Method 2->

Decryption pseudocode with data type matrix and arrays:

Decryption code



Function decrypt.

```
for i in range(len(M)):
    h = 0
    if decrypt_counter == len(w):
        decrypt_counter = 0
    else:
        h: int = int(position_number[decrypt_counter])
    for j in range(matrix_rows):
        matrix[j][h] = M[decrypt_counter2]
        decrypt_counter2 += 1
    if decrypt_counter2 == len(M):
        break
    decrypt_counter += 1
print()
for i in range(matrix_rows):
    for j in range(len(w)):
        plain_text += str(matrix[i][j])
```

Function key_id ()

```
for i in range(len(alphabet)):
    for j in range(len(w)):
        if alphabet[i] == w[j]:
            total += 1
            numbers_to_key[j] = total
```

Function key_pointer()

```
def key_pointer(w, key_id_number):
    position_number = ""
    for i in range(len(w) + 1):
        for j in range(len(w)):
            if key_id_number[j] == i:
                position_number += str(j)
```

Question 3:

Using any programming language of your choice, implement the Row Transposition Cipher decryption algorithm. Decryption using Python programming language

Answer :

Method 1->

Decryption algorithm data type using dictionaries and arrays:

```
################ Decrypt using dictionaries and arrays Function#########

def Main_Decrypted_dictionaries_arrays():
    print("\nEnter the Encrypted Message C = ")
    msg = input().lower()
    print("\nEnter The Key of W  = ")
    key = input()
    # calculate how many raws
    num_row = math.ceil(len(msg)/len(key))
    #create an empty dictionary
    combine_dict = {}
```

```
    Decrypted(combine_dict,msg_extend(msg,len(key)),key,num_row)
    print("\nEncrypted Message is C = ",msg)
    print("\nThe key of w = ",key)
    print("\nDecrypted Message is M = ", Decrypted_sorted(combine_dict,key,num_ro
w,sorted_str = ""))

def Decrypted (combine_dict,msg,key,num_row):
    #create an array and its length is as same as that of key
    array = [""] * len(key)
    sorted_key = sorted(key)
    for i in range (len(key)):
        for msg_pointer in range(num_row):
            # Store message string value to array
            array [i] += msg[ msg_pointer + i * num_row]

        for column_counter in range(num_row):
            dictionary_x ={sorted_key[i] : array[i]}
            combine_dict.update(dictionary_x)
    return combine_dict
def Decrypted_sorted(combine_dict,key, num_row,sorted_str):
    #create an array to print the plaintext from row to row
    plaintext = [""] * num_row
    for i in range (len(key)):
        sorted_str += combine_dict.get(key[i])

    for column_length in range (num_row): #Run how many times in column
        msg_length = column_length
        while msg_length < len(sorted_str):
            # Store message string value to array
            plaintext[column_length] += sorted_str[msg_length]
            # The message pointer adds the key length
            msg_length = msg_length + num_row
    return "".join(plaintext)
```

Method 2->

Decryption algorithm data type using matrixs and arrays:

```
######## Encrypt & Decrypted using matrix and arrays key pointer Function########
def key_pointer(w, key_id_number):
    position_number = ""
    for i in range(len(w) + 1):
        for j in range(len(w)):
            if key_id_number[j] == i:
                position_number += str(j)
    return position_number
######## Encrypt & Decrypted using matrix and arrays key id Function#########
```

```
def key_id(w):
    alphabet = "abcdefghijklmnopqrstuvwxyz"
    numbers_to_key = list(range(len(w)))

    total = 0
    for i in range(len(alphabet)):
        for j in range(len(w)):
            if alphabet[i] == w[j]:
                total += 1
                numbers_to_key[j] = total

######## Decrypted algorithm using matrix and arrays main Function########

def Main_Decrypted_matrix_arrays():
    M = input("\nEnter the Encrypted Message C = ").lower()
    w = input("\nEnter the Key of w = ").lower()
    # assigning numbers to keywords
    numbers_to_key = key_id(w)
################ identify number of rows of the matrix#################
    matrix_rows = int(len(M) / len(w))
    position_number = key_pointer(w, numbers_to_key)
    matrix = [[0] * len(w) for i in range(matrix_rows)]
    plain_text = ""
    decrypt_counter = 0
    decrypt_counter2 = 0


################ Fill up the matrix ################################
    for i in range(len(M)):
        h = 0
        if decrypt_counter == len(w):
            decrypt_counter = 0
        else:
            h: int = int(position_number[decrypt_counter])
        for j in range(matrix_rows):
            matrix[j][h] = M[decrypt_counter2]
            decrypt_counter2 += 1
        if decrypt_counter2 == len(M):
            break
        decrypt_counter += 1
    print()
    for i in range(matrix_rows):
        for j in range(len(w)):
            plain_text += str(matrix[i][j])
    print("\nEncrypted Message is C = " + plain_text)
```

Question 4: (Encryption Test)

- Using a **w** value of ***NYITV***, use your code to encrypt the following text: “CRYPTOLOGY IS THE PRACTICE AND STUDY OF TECHNIQUES FOR SECURE COMMUNICATION IN THE PRESENCE OF THIRD PARTIES CALLED ADVERSARIES.”
- Output your results.

Answer :

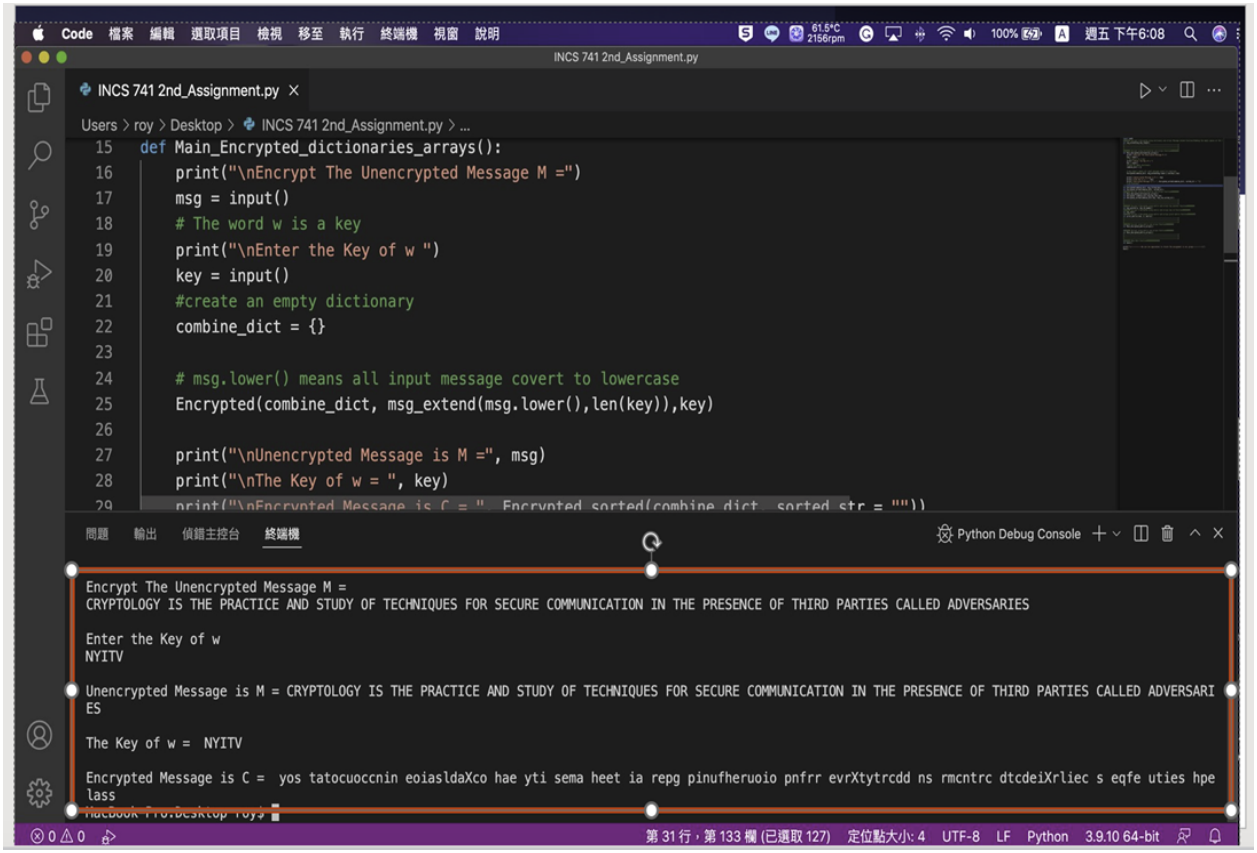
Method 1->

Encrypting algorithm using data type with dictionary and arrays:

Plaintext:
CRYPTOLOGY IS THE PRACTICE AND STUDY OF TECHNIQUES FOR SECURE COMMUNICATION IN THE PRESENCE OF THIRD PARTIES CALLED ADVERSARIES

Key: NYITV

Ciphertext: yos tatocuoccnin eoiasldaXco hae yti sema heet ia repg pinufheruoio pnfrr evrXtytrcdd ns rmcntrc dtcdeiXrliec s eqfe uties hpelass



(Reverse Test-1):

Reverse the encryption algorithm data type with dictionaries and arrays, using decryption with same data type to verify the right ciphertext.

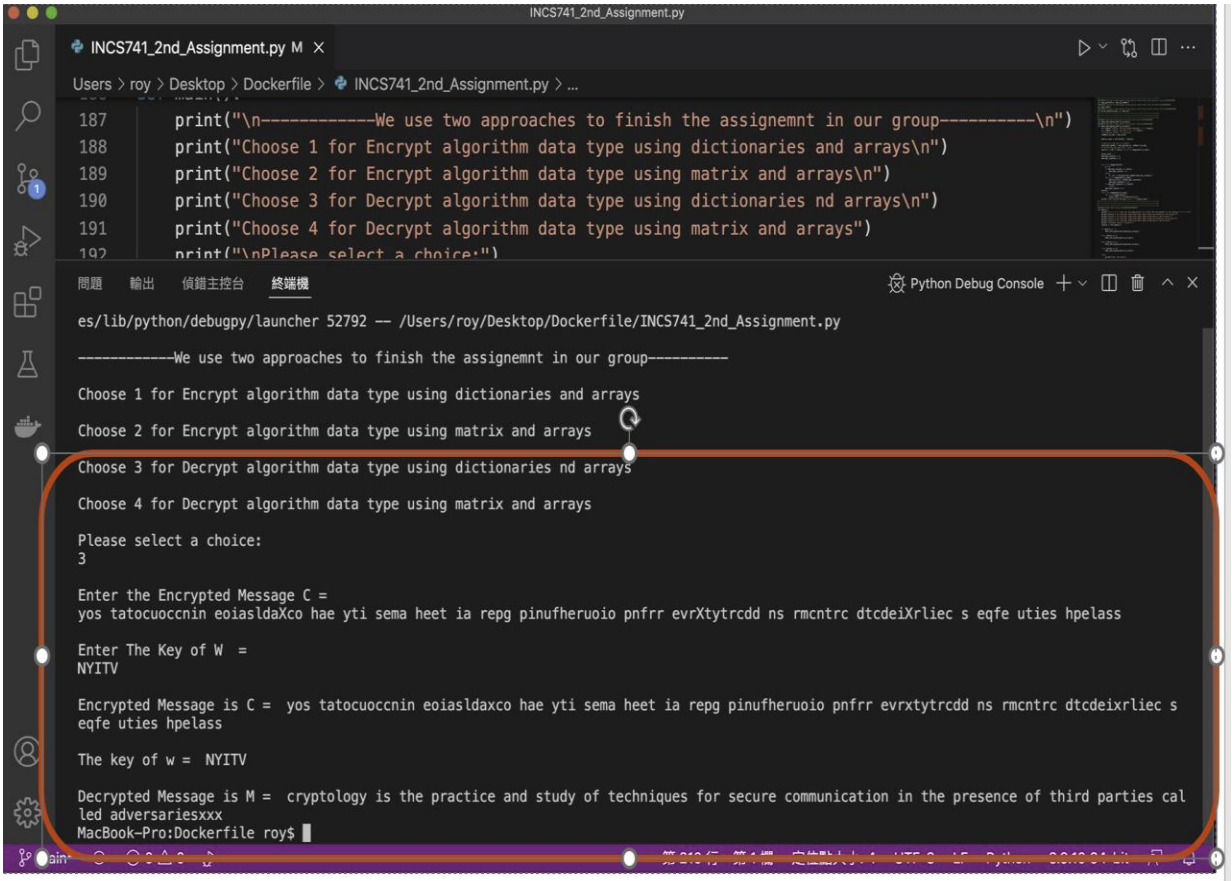
Ciphertext: yos tatocuoccnin eoiasldaXco hae yti sema heet ia repg pinufheruoio pnfrr evrXtytrcdd ns rmcntrc dtcdeiXrliec s eqfe uties hpelass

key: NYITV

Plaintext: cryptology is the practice and study of techniques for secure communication in the presence of third parties called adversariesxxx

Overall: The encrypted message in encryption dictionaries and arrays result is correct.

Ciphertext: yos tatocuoccnin eoiasldaXco hae yti sema heet ia repg pinufheruoio pnfrr evrXtytrcdd ns rmcntrc dtcdeiXrliec s eqfe uties hpelass



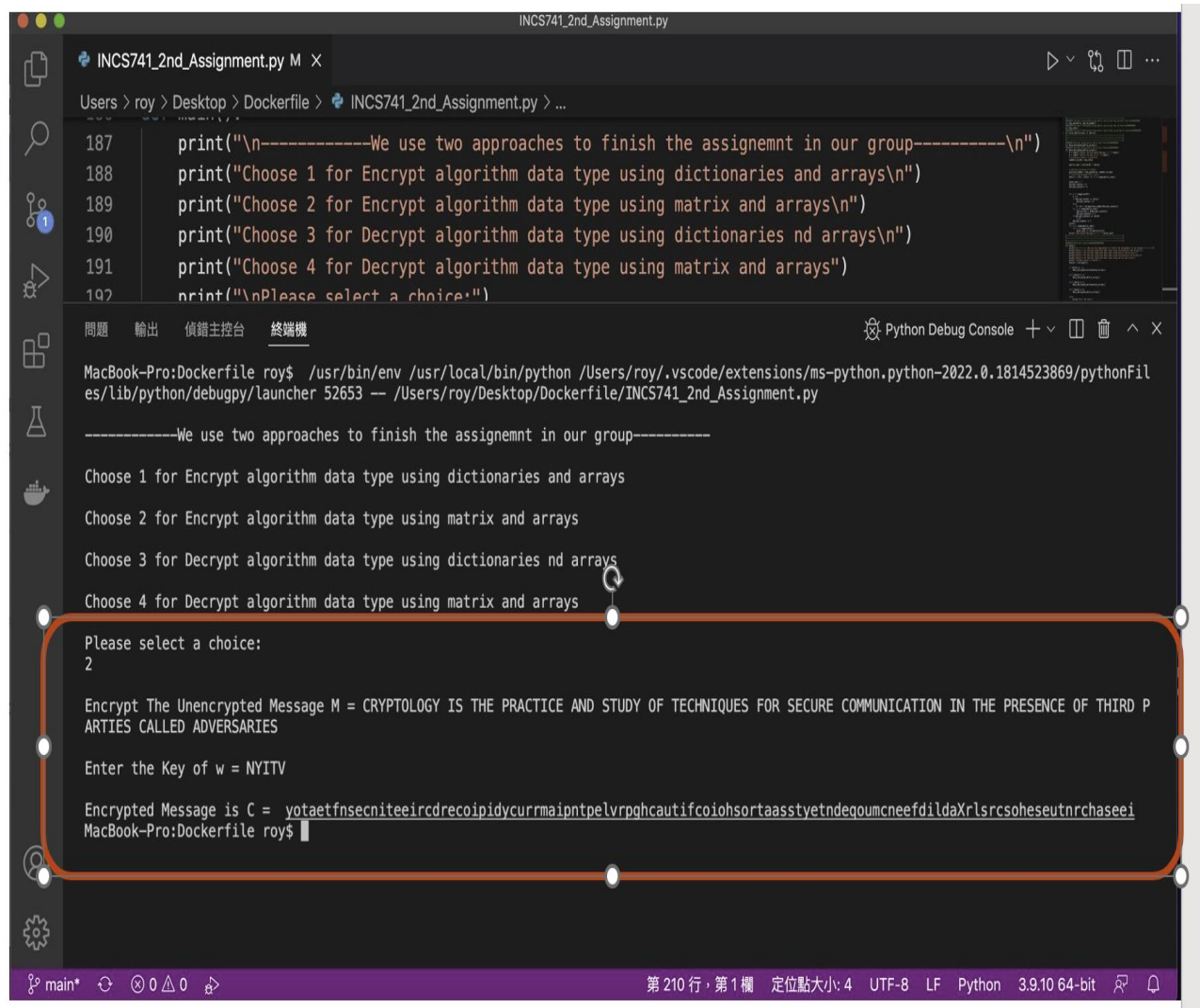
Method 2->

Encrypting algorithm using data type with matrix and arrays:

Plaintext:
CRYPTOLOGY IS THE PRACTICE AND STUDY OF TECHNIQUES FOR SECURE COMMUNICATION IN THE PRESENCE OF THIRD PARTIES CALLED ADVERSARIES

Key: NYITV

Ciphertext:
yotaetfnsecniteeircdrecoipidycurrmaipntpelvrpghcautifcoiohsortaasstyetndeqoumcneefdildaXrlsrcsoheseutnrchaseei



(Reverse Test-2):

Reverse the encryption algorithm data type with matrix and arrays, using decryption with same data type to verify the right ciphertext.

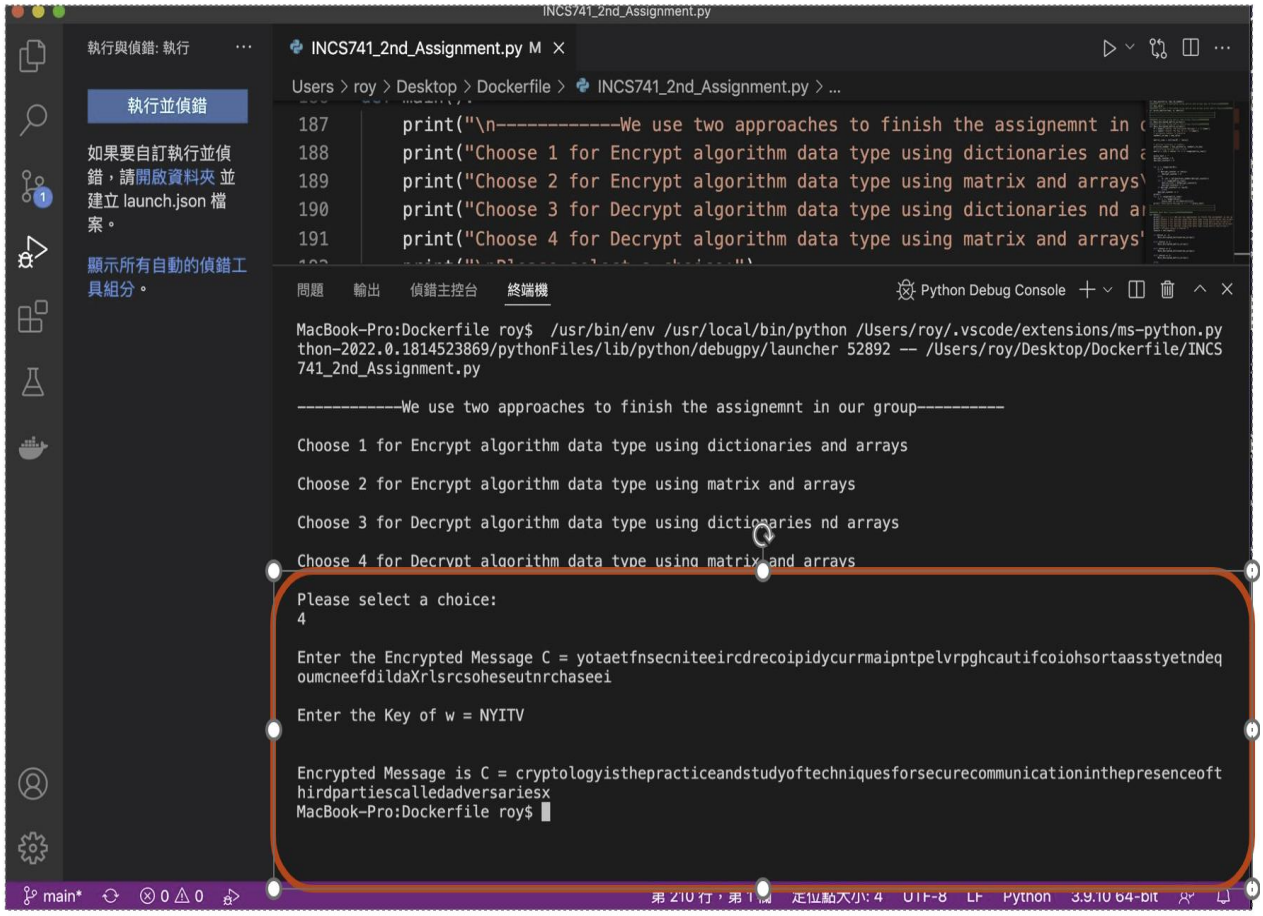
Ciphertext:
yotaetfnsecniteeircdrecoipidycurrmaipntpelvrpghcautifcoiohsortaasstyetndeqoumcneefdildaXrlsr
csoheseutnrchaseei

Key: NYITV

Plaintext:
cryptologyisthepracticeandstudyoftechniquesforsecurecommunicationinthepresenceofthirdparties
calledadversariesx

Overall: The encrypted message in encryption with matrix and arrays result is correct.

Ciphertext:
yotaetfnsecniteeircdrecoipidycurrmaipntpelvrpghcautifcoiohsortaasstyetndeqoumcneefdildaXrlsr
csoheseutnrchaseei



Question 5: (Decryption Test)

- Using a w value of NYITV use your code to decrypt the following text:
- “eroohalpsmeptroohalsefxphtnlefhhxtwstiiiieoecrastitosplmgeasentmitrasnefylypnhiasnetoiroitaetaxoeetonicrasetltesnicrfwmurnhrrhitrcrxhtpipsrmaimiitpiphlaleiucciptotpe”
- Output your results

Answer:

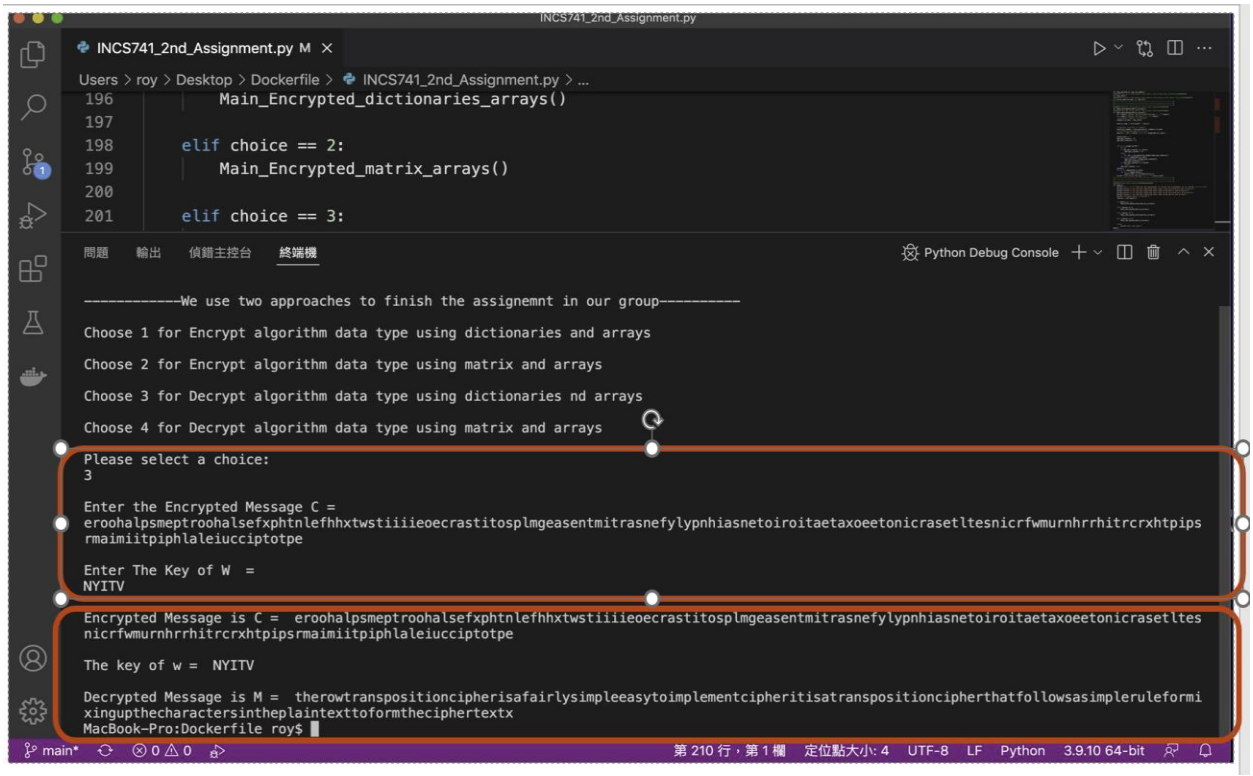
Method 1->

Decrypting algorithm using data type with dictionary and arrays:

Plaintext:
eroohalpsmeptroohalsefxphtnlefhhxtwstiiiieoecrastitosplmgeasentmitrasnefylypnhiasnetoiroitaet
axoeetonicrasetltesnicrfwmurnhrrhitrcrxhtpipsrmaimiitpiphlaleiucciptotpe

Key: NYITV

Ciphertext:
therowtranspositioncipherisafairlysimpleeasytoimplementcipheritisatranspositioncipherthatfollo
wsasimpleruleformixingupthecharactersintheplaintexttoformtheciphertextx



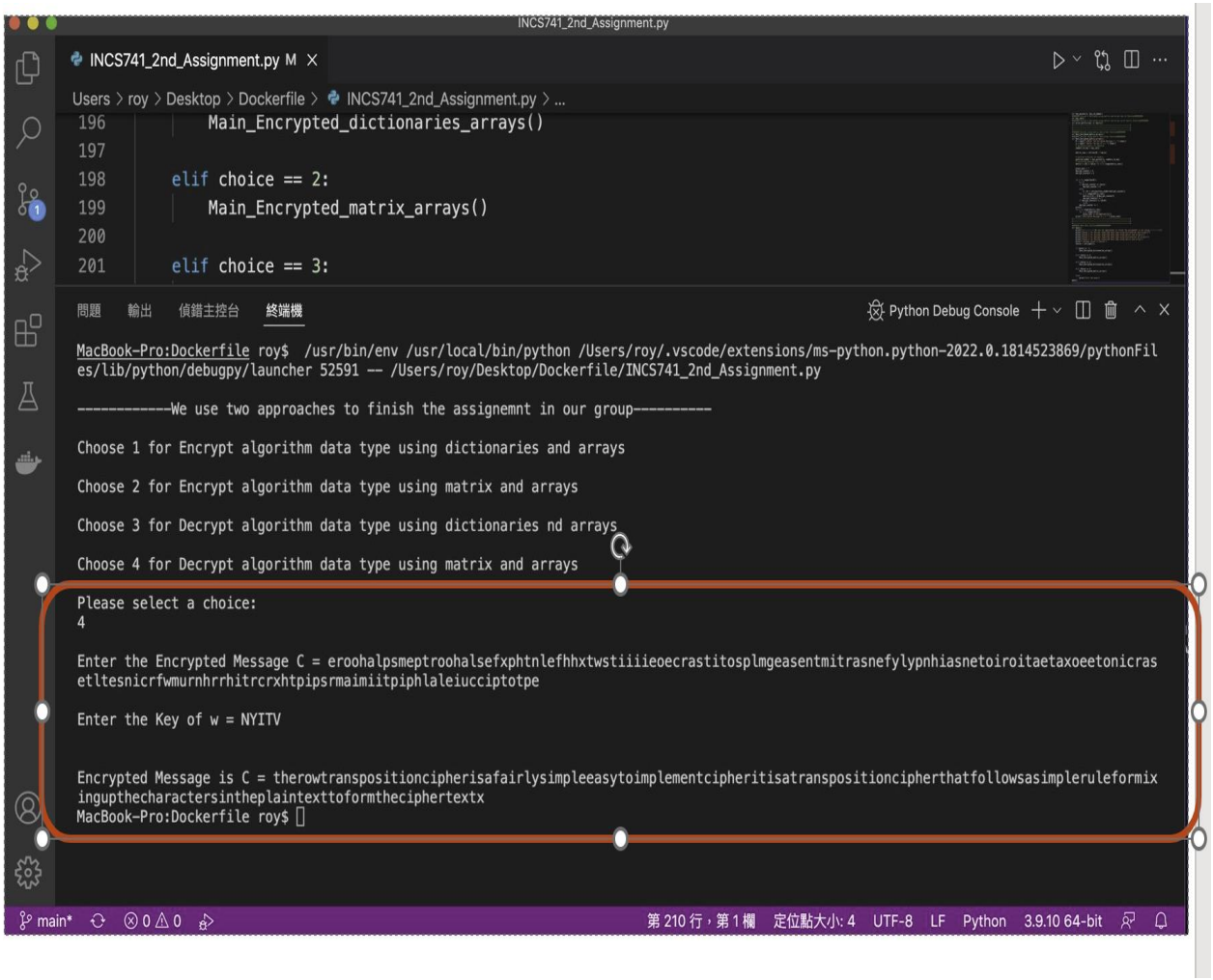
Method 2 ->

Decrypting algorithm using data type with matrix and arrays:

Plaintext:
eroohalpsmeptroohalsefxphtnlefhhxtwstiiiieoecrastitosplmgeasentmitrasnefylypnhiasnetoiroitaet
axoeetonicrasetltesnicrfwmurnhrrhitrcrxhtpipsrmaimiitpiphlaleiucciptotpe

Key: NYITV

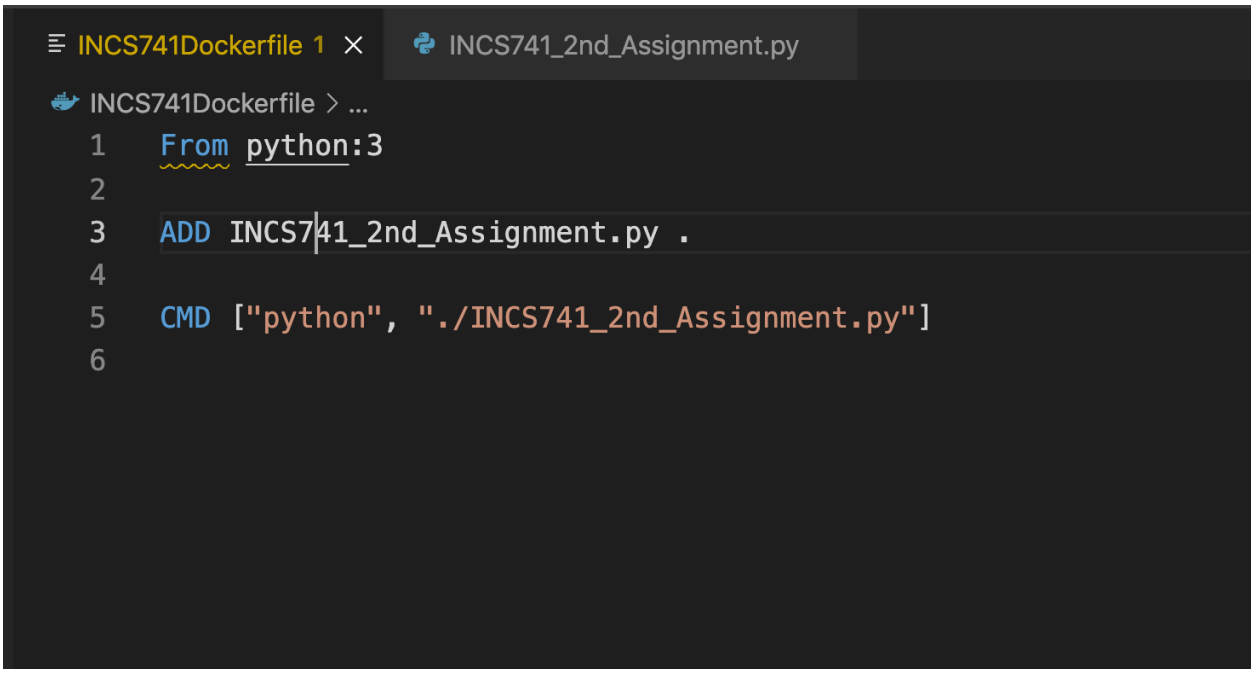
Ciphertext:
therowtranspositioncipherisafairlysimpleeasytoimplementcipheritisatranspositioncipherthatfollowsasimpleruleformixingupthecharactersintheplaintexttoformtheciphertextx



DOCKER FILE:

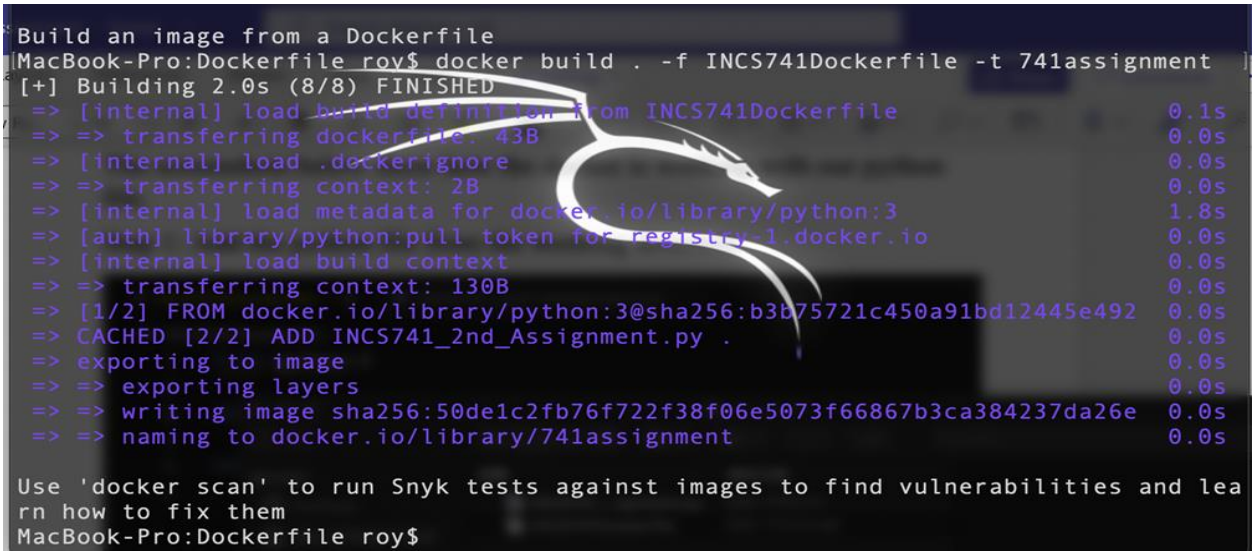
The screenshots below show how the docker is working with our python file.

Step 1: Add the python file from the working directory.



Step 2: Run the below command:

"docker build . -f INCS741Dockerfile –t 741assignemnt"



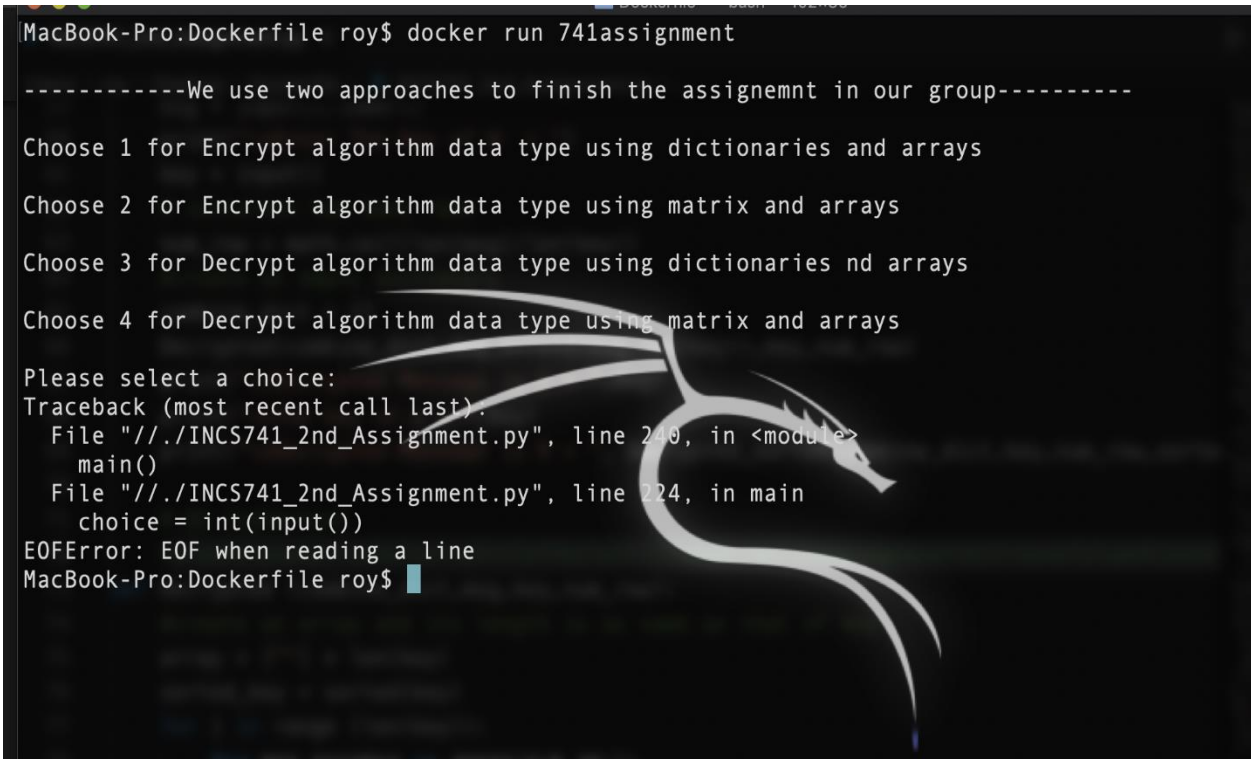
Explanation: The reason using “.” and “ -f ” in the command means that. “ ” will build the docker image to the working directory, “ -f ” will follow the specific name of docker file content as well.

Step 3: Docker run image command

“ docker run –it 741assignment.”

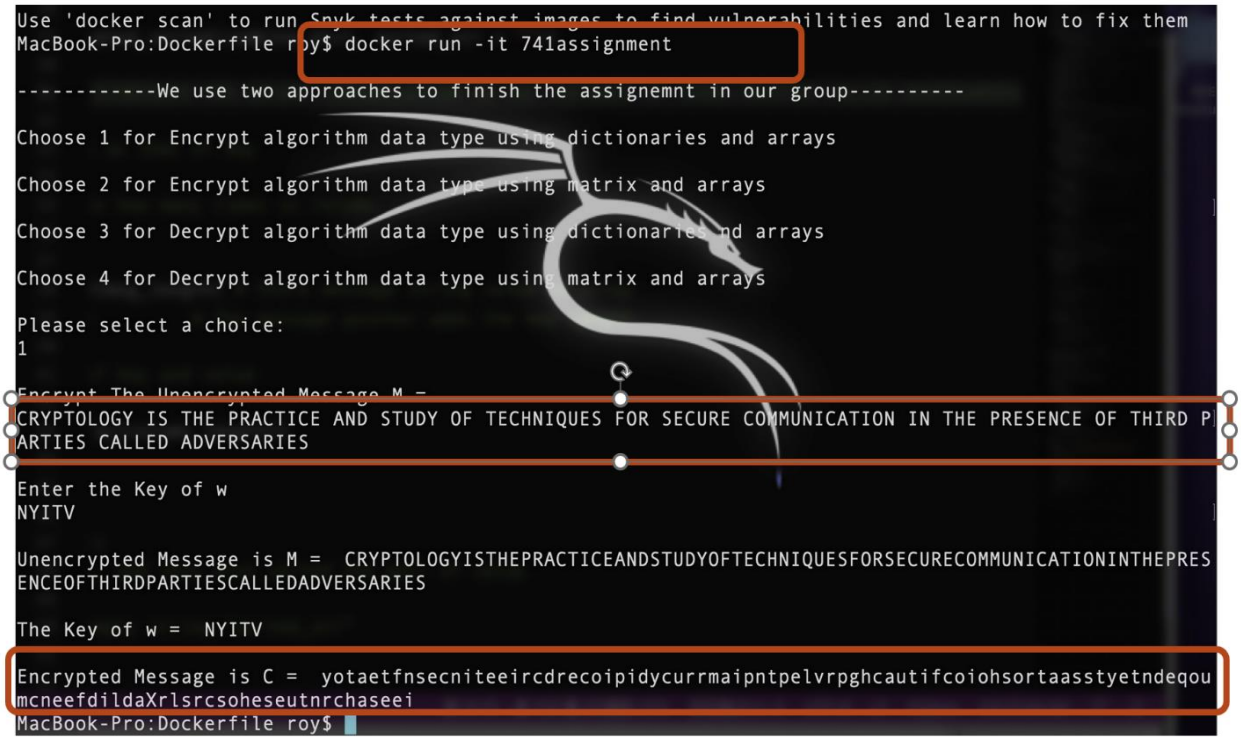
Because our python code involved the interactive mode, we need to use “ - it ” before the image name. Otherwise, this will have EOF reading line errors (Docker EOF error).

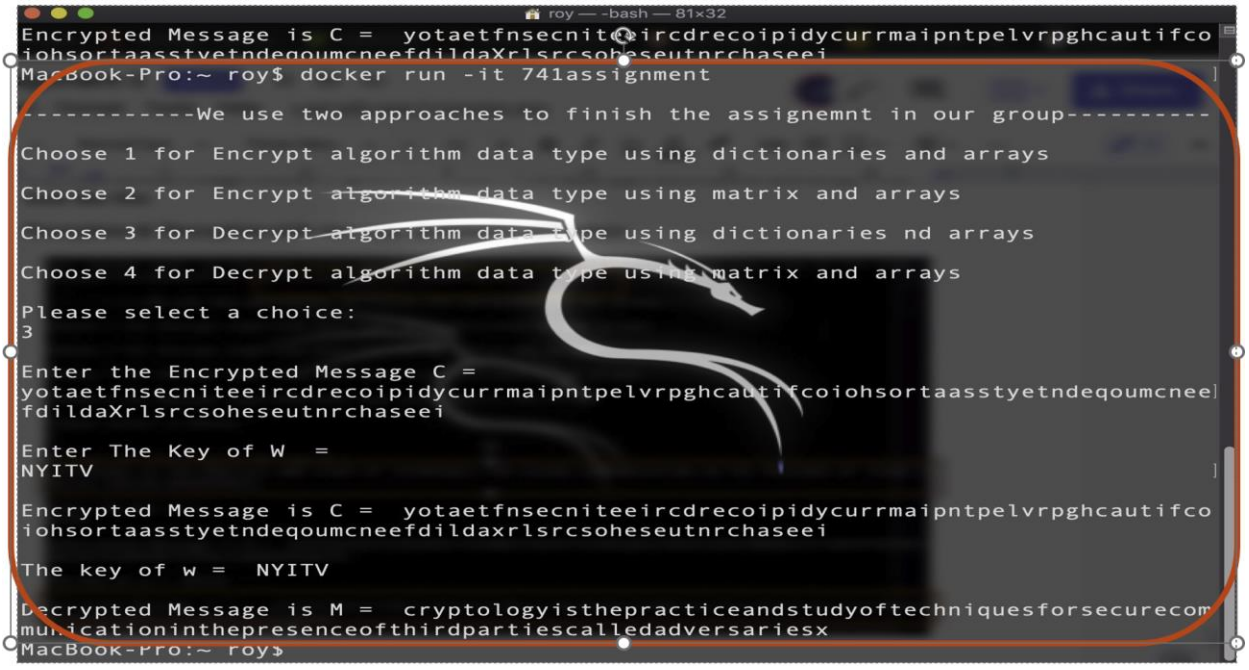
EOF Error Screenshot:



The final docker image results of questions 1 and 4 with data type dictionaries and arrays will look like this:

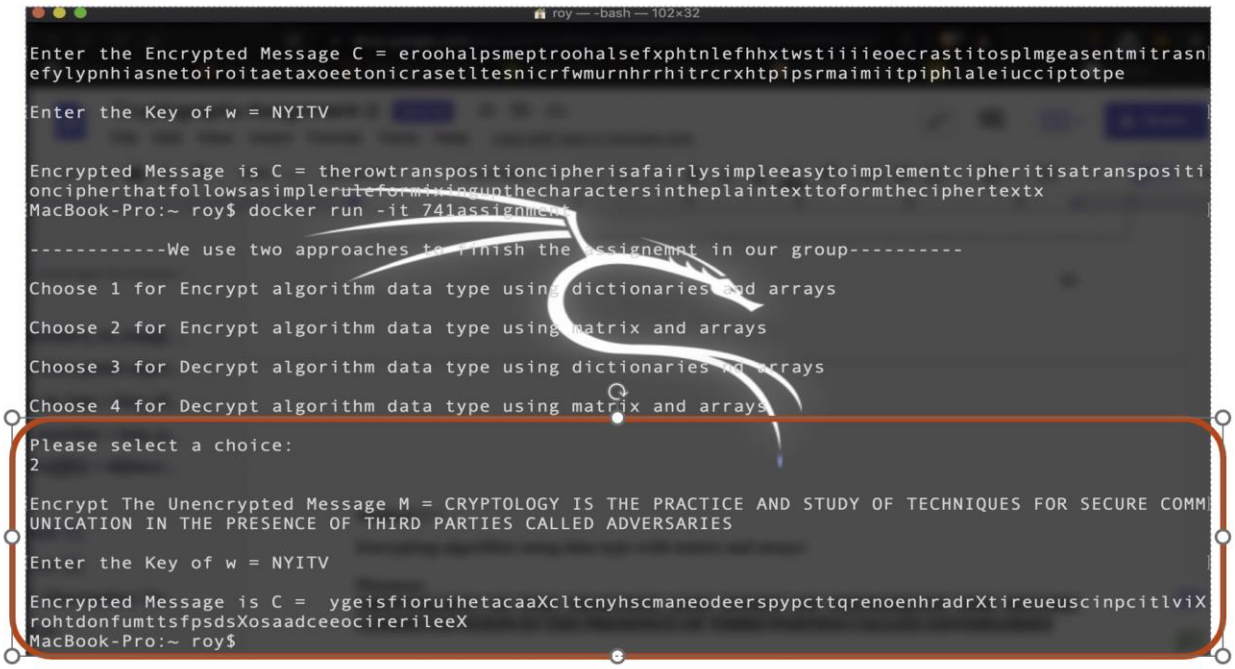
Encryption & Decryption with data type dictionaries and arrays:

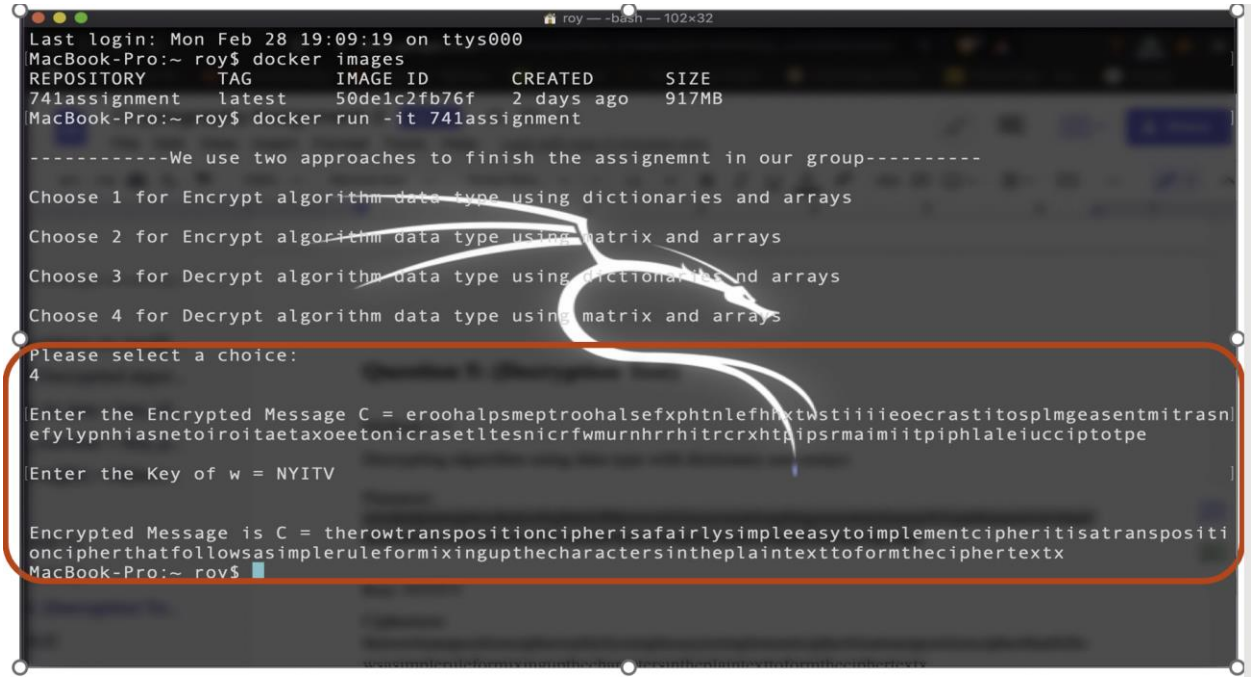




The final docker image results of questions 2 and 5 with data type matrix and arrays will look like this:

Encryption & Decryption with data type matrix and arrays:





V. CONCLUSION

1. We created the executable file that can work on the macos and windows system.
2. It was a challenge to place the logical ideas and coding problems However, it was fascinating as a team to get deeper into this and apply it to crypto science.
3. For the decryption algorithm data type using matrix and arrays, if some text is trying to be decrypted with spaces, it would have some possibilities of errors. So the text may be decrypted has to contain no spaces.
4. The assignment was very interesting in the sense that, which includes many challenges, activities, and instructions to deliver at the end. Those activities include some other new popular approaches like Docker image, which play as a good complement for understanding new area development.
5. The assignment pushes us to go deeper into a programming language again for some of the participants in the team. In contrast to other basic programs done, this assignment specifies a good challenge.
6. The final decision of the team was to include all the hard work done in this document, so as a result two codes were compiled and handy out into the assignment, and explained in this report.

VI. REFERENCES

1. Transposition Cipher (August 10, 2018). *Transposition Cipher in Python - Cryptography with Python.* YouTube Video. https://www.youtube.com/watch?v=BzaaUgcqrL4

2. Python w3schools (n.d.). *Python Dictionaries.* Python Website. https://www.w3schools.com/python/python_dictionaries.asp

3. Docker EOF error (Dec 7, 2018). *Trying to integrate Python with Docker and getting errors on the input portion of code.* stack overflow Website. https://stackoverflow.com/questions/53674490/trying-to-integrate-python-with-docker-getting-error-on-input-portion-of-code/53674593.