Thành viên 1: Thái Hoàng Nhân

ID: 18520326

Thành viên 2: Nguyễn Đức Phú

ID: 18520333

Bài thực hành Lab1- Nhóm 7

Task 1.1.1:

*   :

+ Số 0 là đáp án và đứng sai vị trí (ở hình là vị trí 3)

* Số 0 đứng ở 1 hoặc 2.

+ Không có số 7, 3, 8

* :

Số 0 là đáp án nhưng sai vị trí (ở hình là vị trí thứ 2)

* Số 0 đứng ở 1.
* :
* Số 2 là đáp án và đứng ở 3
*  :

Do số 0 đứng ở 1 và số 2 đứng ở 3

* Không có số 6
*  :

Nếu số 1 là đáp án => số 1 đứng ở 1 hoặc 3 (là vị trí của số 0 và 2)

* Không có số 1
* Số 4 đứng ở giữa.

Vậy đáp án là 042

Task 1.1.2:

**Task 1.2:**

def Encrypt\_By\_Algorithm (text, key):

result = ""

for i in range(len(text)):

letter = text[i]

if (letter.isupper()):

result += chr((ord(letter) + key - 65 ) % 26 + 65)

else:

result += chr((ord(letter) + key - 97 ) % 26 + 97)

return result

def Decrypt\_By\_Algorithm (text, key):

result = ""

for i in range(len(text)):

letter = text[i]

if (letter.isupper()):

result += chr((ord(letter) - key - 65 ) % 26 + 65)

else:

result += chr((ord(letter) - key - 97 ) % 26 + 97)

return result

def BruteForce (text, anphabet):

for key in range(len(anphabet)):

translated = ''

translated = Decrypt\_By\_Algorithm(text, 25 - key)

print('Key %s: %s' % (25 - key, translated))

anphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

plaintext = input("Nhập plaintext: ")

key = int(input("Nhập key: ").strip())

print("-" \* 100)

ciphertext = Encrypt\_By\_Algorithm(plaintext, key)

print("Ciphertext: ", ciphertext)

ori\_text = Decrypt\_By\_Algorithm(ciphertext, key)

print("Original text: ", ori\_text)

print("-" \* 100)

print("Brute-force: ")

BruteForce(ciphertext, anphabet)

**Task 1.3:**

**Ciphertext:**

Crbcyr jub fhpprrq unir zbzraghz. Gur zber gurl fhpprrq, gur zber gurl jnag gb fhpprrq naq gur zber gurl svaq n jnl gb fhpprrq. Fvzvyneyl, jura fbzrbar vf snvyvat, gur graqrapl vf gb trg ba n qbjajneq fcveny gung pna rira orpbzr n frysshysvyyvat cebcurpl.

**Plain text:**

People who succeed have momentum. The more they succeed, the more they want to succeed and the more they find a way to succeed. Similarly, when someone is failing, the tendency is to get on a downward spiral that can even become a self-fulfilling prophecy.

**\*\*\* Điểm đặc biệt \*\*\***  
Đây là đoạn thông điệp được mã hóa bằng phương pháp mã hóa Caesar với key = 13 (hay còn được gọi với tên gọi ROT13). Loại mã hóa này đặc biệt ở chỗ sau khi plaintext được mã hóa tạo thành ciphertext, thì có thể dùng chính ciphertext vừa được mã hóa tiếp tục sử dụng thuật toán mã hóa với key = 13 sẽ trở lại thành plaintext ban đầu.

**Task 1.4:**

Ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZVUEP HZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSZUFP OMBZWPFUPZHMDJUDTMOHMQ

Plaintext:

IT WAS DISCLOSED YESTERDAY THAT SEVERAL INFORMAL BUT DIRECT CONTACTS HAVE BEEN MADE WITH POLITICAL REPRESENTATIVES OF THE VIETCONG IN MOSCOW

**Task 1.6:**

def Generate\_Key(text, key):

key = list(key)

if (len(text) == len(key)):

return

for i in range(len(text) - len(key)):

key.append(key[i % len(key)])

return ("".join(key))

def Encrypt\_By\_Algorithm(text, key):

result = list()

for i in range(len(text)):

temp = (ord(text[i]) + ord(key[i])) % 26

temp += ord('A')

result.append(chr(temp))

return ("".join(result))

def Decrypt\_By\_Algorithm(text, key):

result = list()

for i in range(len(text)):

temp = (ord(text[i]) - ord(key[i]) + 26) % 26

temp += ord('A')

result.append(chr(temp))

return ("".join(result))

plaintext = input("Nhập plaintext (Viết in hoa): ").strip()

keyword = input("Nhập keyword (Viết in hoa): ").strip()

print("-" \* 100)

key = Generate\_Key(plaintext, keyword)

ciphertext = Encrypt\_By\_Algorithm(plaintext,key)

print("Ciphertext :", ciphertext)

print("Original Text :", Decrypt\_By\_Algorithm(ciphertext, key))

**Task 1.9:**

Ciphertext:

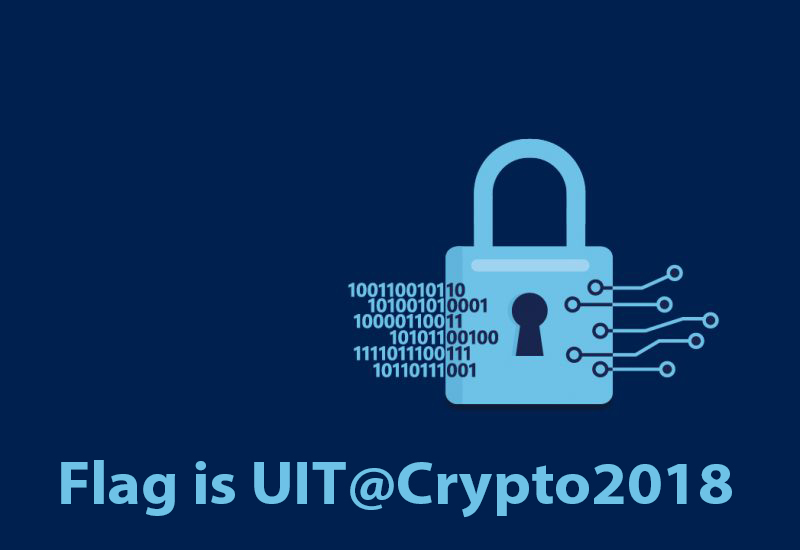
4C6520666C6167206465206365206368616C6C656E6765206573743A2 03261633337363438316165353436636436383964356239313237356433323465

Plaintext: Le flag de ce challenge est: 2ac376481ae546cd689d5b91275d324e

Flag: 2ac376481ae546cd689d5b91275d324e

**Task 1.10:**

**Flag: UIT@Crypto2018**

****

**Hướng giải:**

Nhận thấy file ảnh bị lỗi nhưng có đuôi .jpg suy ra sẽ có signature là ff d8 ff, hexdump file thấy 3 bytes đầu là 8a b1 8b. Xor 2 đoạn với nhau sẽ ra 75, 69, 74, convert ra decimal là 117, 105, 116 và tra mã ASCII sẽ ra phần đầu key là uit.

Phần key còn lại đọc file ở dạng binary thấy “UITHCM” xuất hiện nhiều.

* Có key là “uithcm” và dùng script Python để viết ra file ảnh đúng.

**Source code :**

import sys

fileIn = sys.argv[1]

keyString = sys.argv[2]

fileOut = 'solved.jpg'

key = [ord(char) for char in keyString]

with open(fileIn, 'rb') as file\_in:

    file\_out = open(fileOut, 'wb')

    index = 0

    for crypted\_bytes in file\_in:

        xor = []

        i = 0

        for byte in crypted\_bytes:

            xor.append(byte ^ key[(index + i) % len(key)])

            i += 1

        iterated\_bytes = file\_out.write(*bytes*(xor))

        index += iterated\_bytes

    file\_out.close()