Rmit University - Semester 2, 2017
COSC2539 - Security in Computing and Information Technology
Instructor: Denis Rinfret
Date: August 5th, 2017

ASSIGNMENT 1 - CRYPTOGRAPHY

Programs How-To-Use Explanation & CryptAnalysis

Group Members

Tung Lam Minh - s3598768
Tuan Huynh Doan - s3463207

Table Of Contents

Question 1

A. CryptAnalysis

In Question 1, we need to decrypt the message with Ceasar Algorithm and the key unknown. However, we know one thing that key is between 1 and 50. Key cannot be 0 because with key 0, the decrypted message will be the same as the encrypted message. Also, we need to notice one thing that with Ceasar algorithm , we can shift forward or backward. With alphabet = `ABCDEFGHIJKLMNOPQRSTUVWXYZ .,:;()-!?$'"\n0123456789` We can shift from A → 9 or 9 → A

Shift from A→ 9 happens most rather than 9--> A ⇒ Therefore, we test Shift forward from A→9 first . If we do not get any message, we then test with shift backward from 9→A

At this point, we use our program to test each key from 1--> 50 with the situation: shifting from A-->9 . After a while testing with the key from 1→ 38 , we do not get any message. However, with the key 39, we get the message like below:

```
LIVING IN VIETNAM

LOCATED IN SOUTH EAST ASIA, VIETNAM OCCUPIES THE SOUTH OF THE INDOCHINA PENINSULA WITH THE SHAPE OF AN ELONGATED 'S'. A TROPICAL COUNTRY, VIETNAM HAS DIVERSE SCENERY: FROM SOARING MOUNTAINS TO WHITE SANDY BEACHES, FROM TROPICAL FORESTS TO MYSTERIOUS CAVES.

VIETNAM IS ALSO WHERE YOU CAN FIND A FEAST OF HISTORY AND CULTURE. VIETNAMESE CULTURE IS DISTINCT: A FUSION  OF CHINESE, JAPANESE, AMERICAN AND FRENCH CULTURES.

THE COUNTRY IS DIVIDED INTO THREE AREAS: NORTH VIETNAM, CENTRAL VIETNAM AND SOUTH VIETNAM. WHILE THE NORTH IS KNOWN FOR ITS OLD AND HISTORIC CAPITAL, CENTRAL VIETNAM IS FAMOUS FOR ITS SPECTACULAR BEACHES AND LAGOONS. COMING TO THE SOUTH, YOU WILL ENCOUNTER DYNAMIC HO CHI MINH CITY'S MODERN LIFE. LIVING AND STUDYING IN VIETNAM WILL YOU AN OPPORTUNITY TO EXPLORE A UNIQUE, HISTORIC, EXOTIC, YET RAPIDLY DEVELOPING COUNTRY.,
```

⇒ With the key = 39, we get decrypted message. Therefore, we do not need to test other key

⇒ For Question 1: Algorithm : Ceasar , Key = 39. Decrypted message:

LIVING IN VIETNAM

LOCATED IN SOUTH EAST ASIA, VIETNAM OCCUPIES THE SOUTH OF THE INDOCHINA PENINSULA WITH THE SHAPE OF AN ELONGATED 'S'. A TROPICAL COUNTRY, VIETNAM HAS DIVERSE SCENERY: FROM SOARING MOUNTAINS TO WHITE SANDY BEACHES, FROM TROPICAL FORESTS TO MYSTERIOUS CAVES.

VIETNAM IS ALSO WHERE YOU CAN FIND A FEAST OF HISTORY AND CULTURE. VIETNAMESE CULTURE IS DISTINCT: A FUSION OF CHINESE, JAPANESE, AMERICAN AND FRENCH CULTURES.

THE COUNTRY IS DIVIDED INTO THREE AREAS: NORTH VIETNAM, CENTRAL VIETNAM AND SOUTH VIETNAM. WHILE THE NORTH IS KNOWN FOR ITS OLD AND HISTORIC CAPITAL, CENTRAL VIETNAM IS FAMOUS FOR ITS SPECTACULAR BEACHES AND LAGOONS. COMING TO THE SOUTH, YOU WILL ENCOUNTER DYNAMIC HO CHI MINH CITY'S MODERN LIFE. LIVING AND STUDYING IN VIETNAM WILL YOU AN OPPORTUNITY TO EXPLORE A UNIQUE, HISTORIC, EXOTIC, YET RAPIDLY DEVELOPING COUNTRY.,

B. Program How-To-Use Explanation

```
#OPEN FILE "MSG1.ENC" , REMOVE "<" AND ">" AND CONVERT ALL LETTERS INTO UPPERCASE

file = open('msg1.enc', 'r')
s = file.read()
s1 = s.replace("<","")
message1 = s1.replace('>','')
message = message1.upper()

#######################

#ALPHABET LETTERS

alphabet="""ABCDEFGHIJKLMNOPQRSTUVWXYZ .,:;()-!?$'"\n0123456789"""

##############################
```

```
#Replace the key from 1->50 for testing purpose only.
# After getting the result, please leave the correct key here ( key =39)

key =39

###############
#CREATE STRING TO STORE DECRYPTED MESSSAGE
decrypt=''

################################
#FUNCTION TO DECRYPT CEASER CIPHER

for a in message:
   if a in alphabet:

#### In ceasear, we have 2 situation, shiting forward from A->9 and shifitng backward from 9->A:
#           For shifting forward from A->9: decrypt +=
alphabet[(alphabet.index(a)+key)%(len(alphabet))]
#           For shifting backward from 9->A: decrypt += alphabet[(alphabet.index(a)-
key)%(len(alphabet))]
#       In this case, we test with the situation shifting forward from A->9 first,
#       so we will use : decrypt += alphabet[(alphabet.index(a)+key)%(len(alphabet))]

       decrypt += alphabet[(alphabet.index(a)+key)%(len(alphabet))]

#######################################

#PRINT DECRYPTED MESSAGE
print('Your decrypted message is:
'+"\n"+"-----------------------------------------------------"+"\n"+decrypt)

#CLOSE FILE
file.close()

##################################################################################################
######################
```

In this program, we need to notice two parts:

- `key =39`
    - ○ We need to replace number “39” with the number between 1->50 for testing purpose. When we get the desire key, we will leave the correct key number (key=39) here

- `decrypt += alphabet[(alphabet.index(a)+key)%(len(alphabet))]`
    - ○ In ceasear, we have 2 situation, shiting forward from A->9 and shifitng backward from 9->A:

        - For shifting forward from A->9: `decrypt += alphabet[(alphabet.index(a)+key)%(len(alphabet))]`
        - For shifting backward from 9->A: `decrypt += alphabet[(alphabet.index(a)- key)%(len(alphabet))]`
    - In this case, we test with the situation shifting forward from A->9 first, so we will use : `decrypt += alphabet[(alphabet.index(a)+key)%(len(alphabet))]`

⇒ We test the key from 1->50 with the shifting forward from A->9 and get the decrypted message with key =39

Question 2

A. CryptAnalysis

In this question, we need to decrypt message with algorithm: Columnar Transposition and the key: unknown. However, we know 1 thing that in Columnar Transposition, the key is the divisor of the message length .
In message 2, the length of message is 882 ⇒ key is the divisor of 882 = **{1 2 3 6 7 9 14 18 21 42 49 63 98 126 147 294 441 882 }**

Also, the number of rows = length of message (882) / key

We then test with all possible keys above until we can find the correct key (correct key = 18)

⇒ For Question 2: Algorithm : Columnar Transposition, Key = 18.

Decrypted message:

HEALTH CARE SERVICES AND SAFETY CONCERN

MOVING AWAY FROM HOME, THE ALTERATIONS IN CLIMATE, IN LIVING ENVIRONMENT ARE SIGNIFICANT FACTORS AFFECTING ONE'S HEALTH CONDITION. IT IS IMPORTANT FOR INTERNATIONAL STUDENTS TO TAKE CARE OF THEIR HEALTH. IT IS RECOMMENDED THAT BEFORE TRAVELLING TO VIETNAM, YOU SHOULD PURCHASE PRIVATE HEALTH INSURANCE TO HELP COVER THE COST OF TREATMENT AT PRIVATE HEALTHCARE ESTABLISHMENTS.

AMONG THE WIDE RANGE OF HOSPITALS AND DIFFERENT STANDARDS OF HEALTHCARE, PRIVATE HOSPITALS ARE GENERALLY ON PAR WITH THOSE IN THE WESTERN COUNTRIES AND ACCEPT INTERNATIONAL HEALTH INSURANCE.

```
THE MAJORITY OF DOCTORS ARE FROM THE US, KOREA, JAPAN AND FRANCE,
AS WELL AS OVERSEAS-TRAINED VIETNAMESE DOCTORS.

AT RMIT VIETNAM, THE HEALTH AND SAFETY OF THE STUDENTS IS A TOP
PRIORITY. WE HAVE 24-HOUR SECURITY, HEALTH CENTRES AND COUNSELLING
SERVICES ON CAMPUS.
```

B. Program How-To-Use Explanation

```
#OPEN FILE "MSG2.ENC" , REMOVE "<" AND ">"
file = open('msg2.enc', 'r')
s = file.read()
s1 = s.replace("<","")
message = s1.replace('>','')
#############################

#Replace the key with the divisor of 882 ( because the length of message 2 is 882
) for testing purpose only.
# After getting the result, please leave the correct key here ( key =18)

#Divisor of 882 = {1   2   3   6   7   9   14 18 21 42 49 63 98 126    147    294
441    882 }

key= 18

###############################################

#CREATE STRING TO STORE DECRYPTED MESSAGE
decrypt=''

######## PRINT LENGTH OF MESSAGE 2

print("LENGTH OF MESSAGE 2: " + str(len(message))+
"\n"+"-----------------------------------------------------"+"\n")
######################

#FUNCTION TO DECRYPT COLUMNAR TRANSPOSITION CIPHER

#Note: the number of rows = length of message ( 882) / key

for i in range (int(len(message)/key)):
   for a in range (key):
       decrypt += message[i+(a*(int(len(message)/key)))]

###PRINT DECRYPTED MESSAGE
print('Your decrypted message is:
'+"\n"+"----------------------------------------------------"+"\n"+decrypt)

#CLOSE FILE
file.close()

#################################################################################
######################################
```

In this program, we need to notice 2 parts:

-
  ```
  print("LENGTH OF MESSAGE 2: " + str(len(message))+
  "\n"+"------------------------------------------------------"+"\n")
  ```
    - This part is to count the length of message 2 so that we can have all possible keys . After counting, we get the message 2 length = 882
    - ⇒ The key = divisor of 882 = **{1 2 3 6 7 9 14 18 21 42 49 63 98 126 147 294 441 882 }**
    - ⇒ Number of rows = length of message 2 (882) / key

-
  ```
  key =18
  ```
    - We need to replace number “18” with all possible keys (**1 2 3 6 7 9 14 18 21 42 49 63 98 126 147 294 441 882**) for testing purpose. When we get the desire key, we will leave the correct key number (key=18) here

⇒ After testing with all possible keys , we get the decrypted message with the key = 18 and the algorithm : Columnar Transposition

Question 3

A. CryptAnalysis

- In this question, we need to decrypt message with unknown algorithm and the key = 20
- With the key =20 ⇒ we are sure that the message was not encrypted with Random Substitution and Vernam because those 2 algorithms provide the key in string of letters, not specific number ⇒ We therefore only need to test with Ceaser Cipher and Columnar Transposition
    - With Ceaser, we do not get any message with the key = 20. Below is what we get when decrypt with Ceaser + key=20 :

```
'-6""'6".,276:624(,'3U0XJYZ)-IY!()U"7.!,YYW!6'YW.6!",?-X4
6!?66
"!,)?6,(-"6U'"UWU,1",WY
Y6.?'UU'-"4
```

```
6YX?(X6"(;
-
-(,Y!.'6-0Y'6-V)U,)6?W06Y0

6?0 "6YY6-Y,"
?6!U,(,)VY0',"'X".0;(XY6.6
U7'Y;"!,16YY
6(--!Z,
6
UX-UY6
Y'-W66-,6--Y?"66"Y
,1,-0,'
Y!W-!.()UU
YW()6
-(6
0
661,W X-46-2Y
ZU?,)X.XY(! Y6Y2!-0!(YX66Y
D,
(6"6
6
6Y,06""'V
.6,XZU'Y,
-!V-HYY6Z6,6?)"WU6YU-Y.!I1YZ
""-.
6U
"X3
,YX0',6U-)6X"-!6
Y-,Y,6
Y""'Y6 ?U-)Y-!W
XU20.Y'."
W'
?Y66?Y
YY;,?-,16,,6;!U,ZUU,"",74",UYX'.6'Y
(
1,"1!,,'
?,61U,6U66!--'W6UY2,6Y!,YX0Y-)-;,"W!Y(
Z(61 "
Y,W
!,Y!64Z'Y?"U? D!6,Z664!66,'.Y6Z!Y'
--?Y4""Y6(!6'-.U6V'6'Y6YU,'Y,06Y.W6"8,6'U6"'Y7U(WY016)6
)?--W'

6YU!"6
W16Y

UJ16!Y-,??6J
```

○ We therefore, continue to test with Columnar Transposition + key=20. Now we get decrypted message:

RMIT UNIVERSITY VIETNAM PROVIDES PROGRAM SPECIFIC INTERNSHIPS. DURING THE INTERNSHIP STUDENTS ARE ENROLLED INTO SPECIFIC INDUSTRY-BASED COURSES THAT INTEGRATE THE STUDENT'S THEORETICAL KNOWLEDGE WITH WORKPLACE EXPERIENCE. STUDENTS TAKE ON A RANGE OF FINITE PROJECTS THAT EMPLOYERS CAN HELP IDENTIFY, WHILE RMIT UNIVERSITY VIETNAM PROVIDES CLOSE ACADEMIC SUPERVISION. THIS GIVES A RICHER EDUCATIONAL EXPERIENCE AND IMPORTANT SKILLS MAKING STUDENTS MORE WORK-READY UPON GRADUATION.

THE INTERNSHIPS HAVE PROVEN BENEFICIAL FOR A PLETHORA OF COMPANIES WHO HAVE BEEN ABLE TO "TRIAL RUN" RMIT UNIVERSITY VIETNAM STUDENTS FOR POSSIBLE OFFICIAL EMPLOYMENT LATER.

⇒ For Question 3: Algorithm : Columnar Transposition, Key = 20.

Decrypted message:

RMIT UNIVERSITY VIETNAM PROVIDES PROGRAM SPECIFIC INTERNSHIPS. DURING THE INTERNSHIP STUDENTS ARE ENROLLED INTO SPECIFIC INDUSTRY-BASED COURSES THAT INTEGRATE THE STUDENT'S THEORETICAL KNOWLEDGE WITH WORKPLACE EXPERIENCE. STUDENTS TAKE ON A RANGE OF FINITE PROJECTS THAT EMPLOYERS CAN HELP IDENTIFY, WHILE RMIT UNIVERSITY VIETNAM PROVIDES CLOSE ACADEMIC SUPERVISION. THIS GIVES A RICHER EDUCATIONAL EXPERIENCE AND IMPORTANT SKILLS MAKING STUDENTS MORE WORK-READY UPON GRADUATION.

THE INTERNSHIPS HAVE PROVEN BENEFICIAL FOR A PLETHORA OF COMPANIES WHO HAVE BEEN ABLE TO "TRIAL RUN" RMIT UNIVERSITY VIETNAM STUDENTS FOR POSSIBLE OFFICIAL EMPLOYMENT LATER.

B. Program How-To-Use Explanation

```
#OPEN FILE "MSG3.ENC" , REMOVE "<" AND ">" AND CONVERT ALL LETTERS INTO UPPERCASE


file = open('msg3.enc', 'r')
s = file.read()
s1 = s.replace("<","")
message1 = s1.replace('>','')
message = message1.upper()

###
```

```
##WE INPUT NUMBER 20 BELOW AS THE GIVEN KEY = 20
key= 20
###

#CREATE STRING TO STORE DECRYPTED MESSAGE
columnar=''

#PLEASE UNCOMMENT THE LINE BELOW WHEN TESTING CEASER CIPHER
#ceaser=''


################


##########PLEASE ONLY UNCOMMENT THE BELOW PART (BETWEEN ########  AND ###### )
TO TEST CEASER CIPHER ONLY#################


#FUNCTION TO DECRYPT CEASER CIPHER


#ALPHABET LETTERS

#alphabet="""ABCDEFGHIJKLMNOPQRSTUVWXYZ .,:;()-!?$'"\n0123456789"""


#for a in message:
#   if a in alphabet:

#### In ceasear, we have 2 situation, shiting forward from A->9 and shifitng
backward from 9->A:
#            For shifting forward from A->9: decrypt +=
alphabet[(alphabet.index(a)+key)%(len(alphabet))]
#            For shifting backward from 9->A: decrypt +=
alphabet[(alphabet.index(a)- key)%(len(alphabet))]
#        In this case, we test with the situation shifting forward from A->9
first,
#        so we will use : decrypt +=
alphabet[(alphabet.index(a)+key)%(len(alphabet))]

#         ceaser += alphabet[(alphabet.index(a)+key)%(len(alphabet))]


#PRINT CEASER DECRYPTED MESSAGE
#print('Your CEASER decrypted message is:
'+"\n"+"------------------------------------------------------"+"\n"+ceaser +"\n")

#print("##########################################################################
####"+"\n")


##########PLEASE ONLY UNCOMMENT THIS PART (BETWEEN ########  AND ###### ) TO
TEST CEASER CIPHER ONLY#################


############


# FUNCTION TO DECRYPT COLUMNAR TRANSPOSITION

for i in range (int(len(message)/key)):
   for a in range (key):
       columnar += message[i+(a*(int(len(message)/key)))]

#PRINT COLUMNAR TRANSPOSITION DECRYPTED MESSAGE
```

```
print('Your COLUMNAR decrypted message is:
'+"\n"+"------------------------------------------------------"+"\n"+columnar)

#CLOSE FILE
file.close()
####################################################################################
##
```

In this program, we need to notice 3 parts:

- **`key= 20:`**
    - We input number 20 here because the given key is 20

- **Ceaser Part:** Below is the Ceaser Part to test decrypting message 3 with Ceaser + Key 20. This Part is in commented mode. To use it, We need to uncomment all lines below:

```
        #ceaser=''

        #ALPHABET LETTERS

        #alphabet="""ABCDEFGHIJKLMNOPQRSTUVWXYZ .,:;()-!?$'"\n0123456789"""


        #for a in message:
        #   if a in alphabet:

        #### In ceasear, we have 2 situation, shiting forward from A->9 and
        shifitng backward from 9->A:
        #           For shifting forward from A->9: decrypt +=
        alphabet[(alphabet.index(a)+key)%(len(alphabet))]
        #           For shifting backward from 9->A: decrypt +=
        alphabet[(alphabet.index(a)- key)%(len(alphabet))]
        #       In this case, we test with the situation shifting forward
        from A->9 first,
        #       so we will use : decrypt +=
        alphabet[(alphabet.index(a)+key)%(len(alphabet))]

        #        ceaser += alphabet[(alphabet.index(a)+key)%(len(alphabet))]


        #PRINT CEASER DECRYPTED MESSAGE
        #print('Your CEASER decrypted message is:
        '+"\n"+"----------------------------------------------------"+"\n"+c
        easer +"\n")
```

- **Columnar Transposition Part:** Below is the Columnar Transposition Part. We use it to test decrypting message 3 with Columnar Transposition + Key 20.

```
columnar=''
```

```
for i in range (int(len(message)/key)):
    for a in range (key):
        columnar += message[i+(a*(int(len(message)/key)))]


print('Your COLUMNAR decrypted message is:
'+"\n"+"-----------------------------------------------------"+"\n"+columnar)
```

Question 4

A. CryptAnalysis

For this question, we need to decrypt message with algorithm : Random Substitution and unknown key

To decrypt message 4 with Random Substitution:

- We first count the character frequency in message 4
  - **'.': 30, 'O': 13, 'K': 38, '1': 40, 'D': 105, '\n': 8, 'L': 63, 'N': 42, '(': 35, ':': 27, 'R': 17, 'J': 23, 'P': 45, 'Q': 6, ',': 20, '8': 20, '?': 9, ' ': 13, 'Y': 16, 'I': 16, 'X': 7, '-': 4, '3': 8, 'T': 14, '2': 5, 'G': 4, 'S': 2, 'M': 1, '$': 3, '!': 1**
- We see that D is the character that appear most frequently in message 4 ⇒ We then replace it with the space " "
- We then continue testing and replacing all the characters that appear most frequently (from the most to least) in message 7 with the common characters: **e t a o i n s r h l d c u m f p g w y b v k x j q z**
- While replacing the characters, we also need to consider some specific words
  - 1-letter word: I , a
  - 2-letter words: it, is, an, he, me, us, at, as by, if , by, go, of , on ,so, we,etc
  - 3-letter words: are, the, and, for, but, not, any , all, can, you, our, one, etc.
- We then need to guess the words after replacing some characters
- After replacing most of the characters, we then proofread the message to replace the rest of characters

⇒ For Question 4: Algorithm : Random Substitution.

⇒ Keys:

"D" ⇒ " "	"L" ⇒ "E"	"." ⇒ "R"

"O"	⇒	"M"
"K"	⇒	"I"
"1"	⇒	"T"
"("	⇒	"A"
"\n"	⇒	"V"
":"	⇒	"S"
"\n"	⇒	"V"
","	⇒	"U"
"8"	⇒	"Y"
"R"	⇒	"G"
"J"	⇒	"L"
"P"	⇒	"O"
"Q"	⇒	"B"
"?"	⇒	"F"
" "	⇒	"C"
"Y"	⇒	"H"
"I"	⇒	"D"
"X"	⇒	"."
"-"	⇒	"\n"
"3"	⇒	"W"
"2"	⇒	","
"G"	⇒	"K"
"!"	⇒	"J"
"$"	⇒	"'"
"T"	⇒	"P"
"S"	=>	"-"
"M"	⇒	"X"

⇒ Decrypted Message:

RMIT VIETNAM IS A GLOBAL UNIVERSITY OF TECHNOLOGY AND DESIGN...

TECHNOLOGY NOW UNDERPINS SO MANY ASPECTS OF MODERN LIFE. WHETHER YOU ARE READING THE NEWS FROM YOUR LAPTOP, BUYING MOVIE TICKETS ON YOUR SMARTPHONE ON THE WAY TO THE CINEMA, OR USING A RECIPE APP ON YOUR TABLET WHILE YOU COOK - TECHNOLOGY HAS BECOME INTEGRATED WITH OUR DAILY EXISTENCE.

TAKING A DEGREE IN A AN IT-RELATED FIELD MEANS LEARNING SKILLS THAT HELP TO SOLVE PEOPLE'S PROBLEMS IN CREATIVE, INNOVATIVE WAYS, USING TECHNOLOGY TO TRANSFORM PEOPLE'S WORLDS. IF YOU'RE INTERESTED IN BEING A LEADER IN A CUTTING EDGE INDUSTRY, YOU MAY HAVE JUST FOUND YOUR FUTURE.

B. Program How-To-Use Explanation

```python
#OPEN FILE "MSG4.ENC" , REMOVE "<" AND ">" AND CONVERT ALL LETTERS INTO UPPERCASE


file = open('msg4.enc', 'r')
s = file.read()
s1 = s.replace("<","")
s2=s1.upper()
message = s2.replace('>','')

####

########## Count Character Frequency
def char_frequency(str):
   count = {}
   for n in str:
       keys = count.keys()
       if n in keys:
           count[n] += 1
```

```
        else:
            count[n] = 1
    return count

print(char_frequency(message))

print("\n")
###################################
######Decrypt Start Here

#CREATE STRING TO STORE DECRYPT MESSAGE
mes0=list(message)
mes1=list(message)
####


for i in range(len(mes0)):

#Replace the letter in " " in 2 line of code below with any testing character .
# The first one is the letter in the message, the 2nd one is the new replaced
character
    if mes0[i]=="D":
        mes1[i]=" "

    if mes0[i]=="L":
        mes1[i]="E"

    if mes0[i]==".":
        mes1[i]="R"

    if mes0[i]=="O":
        mes1[i]="M"

    if mes0[i]=="K":
        mes1[i]="I"

    if mes0[i]=="1":
        mes1[i]="T"

    if mes0[i]=="(":
        mes1[i]="A"

    if mes0[i]=="\n":
        mes1[i]="V"

    if mes0[i]==":":
        mes1[i]="S"

    if mes0[i]=="\n":
        mes1[i]="V"

    if mes0[i]==",":
        mes1[i]="U"

    if mes0[i]=="8":
        mes1[i]="Y"

    if mes0[i]=="R":
        mes1[i]="G"

    if mes0[i]=="J":
        mes1[i]="L"

    if mes0[i]=="P":
        mes1[i]="O"
    if mes0[i]=="Q":
        mes1[i]="B"

    if mes0[i]=="?":
        mes1[i]="F"

    if mes0[i]==" ":
        mes1[i]="C"

    if mes0[i]=="Y":
        mes1[i]="H"

    if mes0[i]=="I":
        mes1[i]="D"

    if mes0[i]=="X":
        mes1[i]="."

    if mes0[i]=="-":
        mes1[i]="\n"

    if mes0[i]=="3":
        mes1[i]="W"

    if mes0[i]=="2":
        mes1[i]=","

    if mes0[i]=="G":
        mes1[i]="K"

    if mes0[i]=="!":
        mes1[i]="J"

    if mes0[i]=="$":
        mes1[i]="'"

    if mes0[i]=="T":
        mes1[i]="P"

    if mes0[i]=="S":
        mes1[i]="-"

    if mes0[i]=="M":
        mes1[i]="X"


############################
####################

#PRINT DECRYPTED MESSAGE
print("Decrypt Message: "+"\n")
print("".join(mes1))

######

#CLOSE FILE
file.close()

###########################################################
```

In this program, we need to notice 1 part:

```
if mes0[i]=="D":
   mes1[i]=" "

if mes0[i]=="L":
   mes1[i]="E"

if mes0[i]==".":
   mes1[i]="R"

if mes0[i]=="O":
   mes1[i]="M"

if mes0[i]=="K":
   mes1[i]="I"

if mes0[i]=="1":
   mes1[i]="T"

if mes0[i]=="(":
   mes1[i]="A"

if mes0[i]=="\n":
   mes1[i]="V"

if mes0[i]==":":
   mes1[i]="S"

if mes0[i]=="\n":
   mes1[i]="V"

if mes0[i]==",":
   mes1[i]="U"

if mes0[i]=="8":
   mes1[i]="Y"

if mes0[i]=="R":
   mes1[i]="G"

if mes0[i]=="J":
   mes1[i]="L"

if mes0[i]=="P":
   mes1[i]="O"

if mes0[i]=="Q":
   mes1[i]="B"

if mes0[i]=="?":
   mes1[i]="F"

if mes0[i]==" ":
   mes1[i]="C"

if mes0[i]=="Y":
   mes1[i]="H"

if mes0[i]=="I":
   mes1[i]="D"

if mes0[i]=="X":
   mes1[i]="."
```

```
if mes0[i]=="-":
   mes1[i]="\n"

if mes0[i]=="3":
   mes1[i]="W"

if mes0[i]=="2":
   mes1[i]=","

if mes0[i]=="G":
   mes1[i]="K"

if mes0[i]=="!":
   mes1[i]="J"

if mes0[i]=="$":
   mes1[i]="'"

if mes0[i]=="T":
   mes1[i]="P"

if mes0[i]=="S":
   mes1[i]="-"

if mes0[i]=="M":
   mes1[i]="X"
```

⇒ These are the keys.

For example of 2 lines of code below:

```
  if mes0[i]=="M":
     mes1[i]="X"
```

Replace the letter in " " in 2 line of code with any testing character .
The first one is the letter in the message 7, the 2nd one is the new replaced character

Question 5

A. CryptAnalysis

B. Program How-To-Use Explanation

Question 6

A. CryptAnalysis

- In this question, we need to decrypt message through 2 steps : First is Columnar Transposition and Second is Ceaser Cipher with the unknown key. As stated in the assignment description sheet, the message is encrypted through first Ceaser cipher and then columnar transposition⇒ when decrypting, we need to decrypt through columnar transposition first and then ceaser

- The key for both ceaser cipher and columnar are the same ⇒ We just need to find the key for ceaser and will then match it with columnar transposition
- For Ceaser cipher, we have 2 situations: shift forward or backward. With alphabet = `ABCDEFGHIJKLMNOPQRSTUVWXYZ .,:;()-!?$'"\n0123456789` We can shift from A → 9 or 9 → A
    - We therefore first need to test decrypting message with ceaser shiting forward from A-->9 and columnar transposition with all possible keys from 1-->50.
    - After testing a while, we do not get any desire message, we therefore move to the 2nd situation: shiting backward from 9-->A with all possible keys from 1→ 50 and columnar transposition.
    - With the key 10, and shiting backward from 9→ A for ceaser We now get decrypted message:

> THE BACHELOR OF INFORMATION TECHNOLOGY PROGRAM IS ACCREDITED AT THE PROFESSIONAL LEVEL BY THE AUSTRALIAN COMPUTER SOCIETY (ACS), AN ORGANISATION THAT ACCREDITS INFORMATION AND COMMUNICATION TECHNOLOGY RELATED PROGRAMS IN AUSTRALIA.QQQQQQQQQ

⇒ For Question 6: Algorithm : Columnar Transposition First, Ceaser Second (Ceaser shifting backward from 9-->A), Key = 10.

Decrypted message:

> **THE BACHELOR OF INFORMATION TECHNOLOGY PROGRAM IS ACCREDITED AT THE PROFESSIONAL LEVEL BY THE AUSTRALIAN COMPUTER SOCIETY (ACS), AN ORGANISATION THAT ACCREDITS INFORMATION AND COMMUNICATION TECHNOLOGY RELATED PROGRAMS IN AUSTRALIA.QQQQQQQQQ**

B. Program How-To-Use Explanation

```
#OPEN FILE "MSG6.ENC2" , REMOVE "<" AND ">" AND CONVERT ALL LETTERS INTO
UPPERCASE
file = open('msg6.enc2', 'r')
s = file.read()
s1 = s.replace("<","")
message1 = s1.replace('>','')
message = message1.upper()

#############

#ALPHABET LETTERS
```

```
alphabet="""ABCDEFGHIJKLMNOPQRSTUVWXYZ .,:;()-!?$'"\n0123456789"""

#Replace the key from 1->50 ( with ceaser cipher, the key will be between 1->50 )
for testing purpose only.
# After getting the result, please leave the correct key here ( key =10)


#We do not need to count the length of message and then find out the key by
finding out the divisor of  message length
#Like we normally find the key for columnar transposition cipher as we also have
ceaser cipher in this question and both
#columnar transposition and ceaser cipher has the same key ==> We just need to
find out the key for ceaser cipher and
#then match it with the key for columnar transposition


key = 10

###############
#CREATE STRING TO STORE DECRYPTED MESSSAGE

decrypt=''
columnar=''

########

#Columnar Cipher

#Note: the number of rows = length of message / key
for i in range (int(len(message)/key)):
   for a in range (key):
       columnar += message[i+(a*(int(len(message)/key)))]

#Caesar Cipher

#### In ceasear, we have 2 situations, shiting forward from A->9 and shifitng
backward from 9->A:
#           For shifting forward from A->9: decrypt +=
alphabet[(alphabet.index(a)+key)%(len(alphabet))]
#           For shifting backward from 9->A: decrypt +=
alphabet[(alphabet.index(a)- key)%(len(alphabet))]
#   In this case, we test with the situation shifting forward from A->9 first.
#   After first testing (with key between 1--> 50) , we do not get any decrypted
message
#   ==> We will move to the 2nd situation: shifting backward from 9->A: decrypt
+= alphabet[(alphabet.index(a)-key)%(len(alphabet))]


for a in columnar:
   if a in alphabet:
       decrypt += alphabet[(alphabet.index(a)-key)%(len(alphabet))]

#PRINT DECRYPTED MESSAGE
print('Your decrypted message is: '+"\n"+decrypt)

###CLOSE FILE
file.close()

##########################
```

In this program, we need to notice 3 parts:

- **`key = 10`**

    - Replace the key from 1->50 (with ceaser cipher, the key will be between 1->50) for testing purpose only. After getting the result, please leave the correct key here (key =10)

    - We do not need to count the length of message and then find out the key by finding out the divisor of message length like we normally find the key for columnar transposition cipher as we also have ceaser cipher in this question and both columnar transposition and ceaser cipher has the same key ==> We just need to find out the key for ceaser cipher and then match it with the key for columnar transposition

- **Columnar Transposition Part**

Note: the number of rows = length of message / key

```
for i in range (int(len(message)/key)):
   for a in range (key):
       columnar += message[i+(a*(int(len(message)/key)))]
```

- **Ceaser Cipher Part**

Note:

In ceasear, we have 2 situations, shiting forward from A->9 and shifitng backward from 9->A:

- For shifting forward from A->9:
  decrypt += alphabet[(alphabet.index(a)+key)%(len(alphabet))]
- For shifting backward from 9->A:
  decrypt += alphabet[(alphabet.index(a)- key)%(len(alphabet))]

In this case, we test with the situation shifting forward from A->9 first.
After first testing (with key between 1--> 50) , we do not get any decrypted message

==> We will move to the 2nd situation: shifting backward from 9->A:
decrypt += alphabet[(alphabet.index(a)-key)%(len(alphabet))]

```
for a in columnar:
   if a in alphabet:
       decrypt += alphabet[(alphabet.index(a)-key)%(len(alphabet))]
```

Question 7

A. CryptAnalysis

- For this question, we need to decrypt message with unknown algorithm (but not Vernam) and unknown key
- First of all, We will need to test decrypting the message with both Ceaser Cipher and Columnar Transposition. ⇒ We do not get any decrypted message with both Ceaser Cipher and Columnar Transposition
- Secondly, We do not need to test decrypting message with Vernam because as stated, the algorithm is not Vernam
  ⇒ We finally have Random Substitution for the algorithm

To decrypt message 7 with Random Substitution:

- We first count the character frequency in message 7
  - **'H': 81, 'T': 105, 'J': 77, '7': 33, 'K': 17, '$': 82, '2': 71, '-': 20, 'F': 141, '4': 34, '0': 38, 'A': 55, 'Q': 35, '1': 19, 'N': 2, 'U': 42, '"': 43, '9': 10, 'P': 2, 'L': 23, 'Y': 9, ' ': 6, 'B': 3, 'Z': 15, 'D': 8, 'C': 1, 'S': 3, '6': 1, '?': 1, '8': 2, '\n': 1**
- We see that F is the character that appear most frequently in message 7 ⇒ We then replace it with the space “ “
- We then continue testing and replacing all the characters that appear most frequently (from the most to least) in message 7 with the common characters: **e t a o i n s r h l d c u m f p g w y b v k x j q z**
- While replacing the characters, we also need to consider some specific words
  - 1-letter word: I , a
  - 2-letter words: it, is, an, he, me, us, at, as by, if , by, go, of , on ,so, we,etc
  - 3-letter words: are, the, and, for, but, not, any , all, can, you, our, one, etc.
- We then need to guess the words after replacing some characters
- After replacing most of the characters, we then proofread the message to replace the rest of characters

⇒ For Question 7: Algorithm : Random Substitution.

Key:

```
"F" → " "
"T" → "E"
"J" → "A"
"2" → "N"
"4" → "D"
"H" → "T"
```

```
"7" → "C"
"K" → "H"
"$" → "I"
"-" → "G"
"0" → "L"
"A" → "R"
```

```
"Q" → "M"
"1" → "V"
"N" → "\n"
"U" → "S"
'''"''' → "O"
"9" → "P"
```

"L" → "U"	"B" → "W"	"?" → "X"
	"Z" → "F"	"S" → ","
"P" → "Q"	"D" → "B"	"6" → "'"
" " → "."	"C" → "K"	"8" → "-"

Decrypted message:

TEACHING AND LEARNING AT RMIT VIETNAM

RMIT VIETNAM IS COMMITTED TO PROVIDING QUALITY EDUCATION. THE DEGREE YOU WILL RECEIVE AFTER SUCCESSFUL COMPLETION OF YOUR STUDIES IN VIETNAM IS A GLOBAL RMIT UNIVERSITY DEGREE. THE LEARNING OUTCOMES AND ASSESSMENTS FOR COURSES TAUGHT IN VIETNAM ARE EQUIVALENT TO THOSE DELIVERED AT RMIT MELBOURNE. ACADEMIC STAFF EMPLOYED AT RMIT INTERNATIONAL UNIVERSITY VIETNAM WORK IN COLLABORATION WITH ACADEMIC STAFF FROM RMIT UNIVERSITY MELBOURNE TO CUSTOMISE THE MATERIALS THAT ARE PREPARED IN MELBOURNE FOR DELIVERY IN VIETNAM, AND TO FACILITATE THE TEACHING AND LEARNING PROCESSES AT RMIT VIETNAM. THE ACADEMIC'S ROLE AT RMIT VIETNAM IS FOCUSED ON ADDING VALUE TO THE TEACHING AND LEARNING PROCESS THROUGH CREATIVE FACILITATED LEARNING. THE TEACHING AND LEARNING MODEL AT RMIT VIETNAM IS BASED ON INTERNATIONAL BEST PRACTICE AND INCLUDES CLASSROOM LEARNING EXPERIENCES, FACILITATED ON-LINE AND SELF-DIRECTED LEARNING BY STUDENTS, AND PEER LEARNING.

B. Program How-To-Use Explanation

```
#OPEN FILE "MSG7.ENC" , REMOVE "<" AND ">" AND CONVERT ALL LETTERS INTO UPPERCASE
file = open('msg7.enc', 'r')
s = file.read()
s1 = s.replace("<","")
message1 = s1.replace('>','')
message=message1.upper()
decrypt=''
#######

##########Count Character Frequency

def char_count(str):
    count = {}
    for char in str:
        keys = count.keys()
        if char in keys:
            count[char] += 1
        else:
            count[char] = 1
    return count

print(char_count(message))


print("\n")
###################################

######Decrypt Start Here

#CREATE STRING TO STORE DECRYPT MESSAGE
mes0=list(message)
mes1=list(message)
##

for i in range(len(mes0)):
#Replace the letter in " " in 2 line of code below with any testing character .
# The first one is the letter in the message, the 2nd one is the new replaced character
    if mes0[i]=="F":
        mes1[i]=" "

    if mes0[i]=="T":
        mes1[i]="E"

    if mes0[i]=="J":
        mes1[i]="A"

    if mes0[i]=="2":
        mes1[i]="N"

    if mes0[i]=="4":
        mes1[i]="D"

    if mes0[i]=="H":
        mes1[i]="T"

    if mes0[i]=="7":
        mes1[i]="C"

    if mes0[i]=="K":
        mes1[i]="H"

    if mes0[i]=="$":
        mes1[i]="I"

    if mes0[i]=="-":
        mes1[i]="G"

    if mes0[i]=="0":
        mes1[i]="L"

    if mes0[i]=="A":
        mes1[i]="R"

    if mes0[i]=="Q":
        mes1[i]="M"
```

```
   if mes0[i]=="1":
       mes1[i]="V"

   if mes0[i]== "N":
       mes1[i]="\n"

   if mes0[i]=="U":
       mes1[i]="S"

   if mes0[i]=='''"''':
       mes1[i]="O"

   if mes0[i]=="9":
       mes1[i]="P"

   if mes0[i]=="L":
       mes1[i]="U"

   if mes0[i]=="P":
       mes1[i]="Q"

   if mes0[i]==" ":
       mes1[i]="."

   if mes0[i]=="B":
       mes1[i]="W"

   if mes0[i]=="Z":
       mes1[i]="F"

   if mes0[i]=="D":
       mes1[i]="B"

   if mes0[i]=="C":
       mes1[i]="K"

   if mes0[i]=="?":
       mes1[i]="X"

   if mes0[i]=="S":
       mes1[i]=","

   if mes0[i]=="6":
       mes1[i]="'"

   if mes0[i]=="8":
       mes1[i]="-"

############################
####################

#PRINT DECRYPTED MESSAGE
print("Decrypt Message: "+"\n")
print("".join(mes1))

###############################################################

#CLOSE FILE
file.close()

############################
```

In this program, we need to notice 1 part:

```
if mes0[i]=="F":
       mes1[i]=" "

   if mes0[i]=="T":
       mes1[i]="E"

   if mes0[i]=="J":
       mes1[i]="A"

   if mes0[i]=="2":
       mes1[i]="N"

   if mes0[i]=="4":
       mes1[i]="D"

   if mes0[i]=="H":
       mes1[i]="T"

   if mes0[i]=="7":
       mes1[i]="C"

   if mes0[i]=="K":
       mes1[i]="H"

   if mes0[i]=="$":
       mes1[i]="I"

   if mes0[i]=="-":
       mes1[i]="G"

   if mes0[i]=="0":
       mes1[i]="L"

   if mes0[i]=="A":
       mes1[i]="R"

   if mes0[i]=="Q":
       mes1[i]="M"

   if mes0[i]=="1":
       mes1[i]="V"

   if mes0[i]== "N":
       mes1[i]="\n"

   if mes0[i]=="U":
       mes1[i]="S"

   if mes0[i]=='''"''':
       mes1[i]="O"

   if mes0[i]=="9":
       mes1[i]="P"

   if mes0[i]=="L":
       mes1[i]="U"

   if mes0[i]=="P":
       mes1[i]="Q"

   if mes0[i]==" ":
       mes1[i]="."
```

```
if mes0[i]=="B":
    mes1[i]="W"

if mes0[i]=="Z":
    mes1[i]="F"

if mes0[i]=="D":
    mes1[i]="B"

if mes0[i]=="C":
    mes1[i]="K"

if mes0[i]=="?":
    mes1[i]="X"

if mes0[i]=="S":
    mes1[i]=","

if mes0[i]=="6":
    mes1[i]="'"

if mes0[i]=="8":
    mes1[i]="-"
```

⇒ These are the keys.

For example of 2 lines of code below:

```
if mes0[i]=="8":
    mes1[i]="-"
```

Replace the letter in " " in 2 line of code with any testing character .
The first one is the letter in the message 7, the 2nd one is the new replaced character