
# TSN3251 COMPUTER SECURITY
# Assignment Report

## Trimester 1 2020/2021 (2010 RMCO)

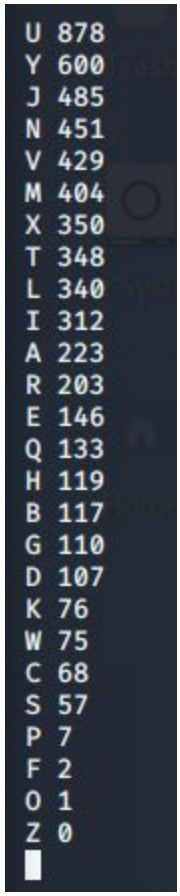

| Student ID | Name |
|---|---|
| 1181301724 | Sarah Batrisyia binti Ahmad Salman |

Complete the following sections for the TSN3251 assignment report.

PART 1

Terminal output results of the (sorted) frequency analysis on your encrypted text file (either copy/paste output or screenshot)

[2 marks]




Analysis of the ciphertext and frequency analysis obtained (i.e. highest likelihood of most common letter, possible ciphertext patterns to indicate common words etc). Briefly explain how the decryption script can be altered to decrypt the ciphertext using your findings. You may show multiple steps on the process to obtaining the final decrypted output … but once again, BRIEFLY.

[4 marks]

- Started off by identifying the most frequent letter in the ciphertext which is U followed by Y and J.
- The most frequent letter in the English language is E followed by T and A
- I substitute the characters of the ciphertext with the frequency of the characters in English language.
- The substitution took a long time since I had to play around with the characters.

- Everytime I substitute the ciphertext character with a new plaintext character, I have to rerun the program to see if there is any pattern.

Final mapping of plaintext to ciphertext in simple substitution cipher to decrypt content. You may need to run the decryption multiple times to get your mapping that decrypts all ciphertext correctly.

[3 marks]

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L | Y | V | F | U | Q | M | G | R | H | B | S | N | A | X | J | C | W | K | I | E | O | P | D | T | Z |

Your final bash decryption script for simple substitution in full using the mapping from the previous question

[4 marks]

```
#!/bin/bash
#subs.sh
cat $1 | tr '[:upper:]' '[:lower:]' | sed \
-e 's/a/L/g' \
-e 's/b/Y/g' \
-e 's/c/V/g' \
-e 's/d/F/g' \
-e 's/e/U/g' \
-e 's/f/Q/g' \
-e 's/g/M/g' \
-e 's/h/G/g' \
-e 's/i/R/g' \
-e 's/j/H/g' \
-e 's/k/B/g' \
-e 's/l/S/g' \
-e 's/m/N/g' \
-e 's/n/A/g' \
-e 's/o/X/g' \
-e 's/p/J/g' \
-e 's/q/C/g' \
-e 's/r/W/g' \
-e 's/s/K/g' \
-e 's/t/I/g' \
-e 's/u/E/g' \
-e 's/v/O/g' \
-e 's/w/P/g' \
-e 's/x/D/g' \
-e 's/y/T/g' \
-e 's/z/Z/g'
echo
exit 0
```

Decrypted (plaintext) output of the text file. There is no need to insert spaces or punctuation into the output

[2 marks]

THEREWASONCEUPONATIMEAKINGWHOHADTWELVEDAUGHTERSEACHONEMOREBEAUTIFU
LTHANTHEOTHERTHEYALLSLEPTTOGETHERINONECHAMBERINWHICHTHEIRBEDSSTOODSID
EBYSIDEANDEVERYNIGHTWHENTHEYWEREINTHEMTHEKINGLOCKEDTHEDOORANDBOLTEDI
TBUTINTHEMORNINGWHENHEUNLOCKEDTHEDOORHESAWTHATTHEIRSHOESWEREWORNO
UTWITHDANCINGANDNOONECOULDFINDOUTHOWTHATHADCOMETOPASSTHENTHEKINGCAU
SEDITTOBEPROCLAIMEDTHATWHOSOEVERCOULDDISCOVERWHERETHEYDANCEDATNIGHTS
HOULDCHOOSEONEOFTHEMFORHISWIFEANDBEKINGAFTERHISDEATHBUTTHATWHOSOEVER
CAMEFORWARDANDHADNOTDISCOVEREDITWITHINTHREEDAYSANDNIGHTSSHOULDHAVEFO
RFEITEDHISLIFEITWASNOTLONGBEFOREAKINGSSONPRESENTEDHIMSELFANDOFFEREDTOUN
DERTAKETHEENTERPRISEHEWASWELLRECEIVEDANDINTHEEVENINGWASLEDINTOAROOMA
DJOININGTHEPRINCESSESSLEEPINGCHAMBERHISBEDWASPLACEDTHEREANDHEWASTOOBSE
RVEWHERETHEYWENTANDDANCEDANDINORDERTHATTHEYMIGHTDONOTHINGSECRETLYOR
GOAWAYTOSOMEOTHERPLACETHEDOOROFTHEIRROOMWASLEFTOPENBUTTHEEYELIDSOFT
HEPRINCEGREWHEAVYASLEADANDHEFELLASLEEPANDWHENHEAWOKEINTHEMORNINGALL
TWELVEHADBEENTOTHEDANCEFORTHEIRSHOESWERESTANDINGTHEREWITHHOLESINTHES
OLESONTHESECONDANDTHIRDNIGHTSTHEREWASNODIFFERENCEANDTHENHISHEADWASST
RUCKOFFWITHOUTMERCYMANYOTHERSCAMEAFTERTHISANDUNDERTOOKTHEENTERPRISEB
UTALLFORFEITEDTHEIRLIVESNOWITCAMETOPASSTHATAPOORSOLDIERWHOHADAWOUNDA
NDCOULDSERVENOLONGERFOUNDHIMSELFONTHEROADTOTHETOWNWHERETHEKINGLIVED
THEREHEMETANOLDWOMANWHOASKEDHIMWHEREHEWASGOINGIHARDLYKNOWMYSELFAN
SWEREDHEANDADDEDINJESTIHADHALFAMINDTODISCOVERWHERETHEPRINCESSESDANCED
THEIRSHOESINTOHOLESANDTHUSBECOMEKINGTHATISNOTSODIFFICULTSAIDTHEOLDWOMA
NYOUMUSTNOTDRINKTHEWINEWHICHWILLBEBROUGHTTOYOUATNIGHTANDMUSTPRETEND
TOBESOUNDASLEEPWITHTHATSHEGAVEHIMALITTLECLOAKANDSAIDIFYOUWEARTHISYOUWI
LLBEINVISIBLEANDTHENYOUCANSTEALAFTERTHETWELVEWHENTHESOLDIERHADRECEIVED
THISGOODADVICEHEFELLTOINEARNESTTOOKHEARTWENTTOTHEKINGANDANNOUNCEDHIM
SELFASASUITORHEWASASWELLRECEIVEDASTHEOTHERSANDROYALGARMENTSWEREPUTUP
ONHIMHEWASCONDUCTEDTHATEVENINGATBEDTIMEINTOTHEANTECHAMBERANDASHEWAS
ABOUTTOGOTOBEDTHEELDESTCAMEANDBROUGHTHIMACUPOFWINEBUTHEHADTIEDASPON
GEUNDERHISCHINANDLETTHEWINERUNDOWNINTOITWITHOUTDRINKINGADROPTHENHELA
YDOWNANDWHENHEHADLAINAWHILEHEBEGANTOSNOREASIFINTHEDEEPESTSLEEPTHETW
ELVEPRINCESSESHEARDTHATANDLAUGHEDANDTHEELDESTSAIDHETOOMIGHTASWELLHAVE
SAVEDHISLIFEWITHTHATTHEYGOTUPOPENEDWARDROBESPRESSESCUPBOARDSANDBROUGH
TOUTPRETTYDRESSESDRESSEDTHEMSELVESBEFORETHEMIRRORSSPRANGABOUTANDREJOIC
EDATTHEPROSPECTOFTHEDANCEONLYTHEYOUNGESTSAIDIKNOWNOTHOWITISYOUAREVERY
HAPPYBUTIFEELVERYSTRANGESOMEMISFORTUNEISCERTAINLYABOUTTOBEFALLUSYOUARE
AGOOSEWHOAREALWAYSFRIGHTENEDSAIDTHEELDESTHAVEYOUFORGOTTENHOWMANYKIN
GSSONSHAVEALREADYCOMEHEREINVAINIHADHARDLYANYNEEDTOGIVETHESOLDIERASLEEP
INGDRAUGHTTHEBOOBYWOULDNOTHAVEAWAKENEDANYWAYWHENTHEYWEREALLREADYT
HEYLOOKEDCAREFULLYATTHESOLDIERBUTHEHADCLOSEDHISEYESANDDIDNOTMOVEORSTIR
SOTHEYFELTTHEMSELVESSAFEENOUGHTHEELDESTTHENWENTTOHERBEDANDTAPPEDITWH
EREUPONITIMMEDIATELYSANKINTOTHEEARTHANDONEAFTERTHEOTHERTHEYDESCENDED
THROUGHTHEOPENINGTHEELDESTGOINGFIRSTTHESOLDIERWHOHADWATCHEDEVERYTHIN
GTARRIEDNOLONGERPUTONHISLITTLECLOAKANDWENTDOWNLASTWITHTHEYOUNGESTHAL
FWAYDOWNTHESTEPSHEJUSTTRODALITTLEONHERDRESSSHEWASTERRIFIEDATTHATANDCR

IEDOUTWHATISTHATWHOISPULLINGMYDRESSDONTBESOSILLYSAIDTHEELDESTYOUHAVECA
UGHTITONANAILTHENTHEYWENTALLTHEWAYDOWNANDWHENTHEYWEREATTHEBOTTOMT
HEYWERESTANDINGINAWONDERFULLYPRETTYAVENUEOFTREESALLTHELEAVESOFWHICHW
EREOFSILVERANDSHONEANDGLISTENEDTHESOLDIERTHOUGHTIMUSTCARRYATOKENAWAY
WITHMEANDBROKEOFFATWIGFROMONEOFTHEMONWHICHTHETREECRACKEDWITHALOUDR
EPORTTHEYOUNGESTCRIEDOUTAGAINSOMETHINGISWRONGDIDYOUHEARTHECRACKBUTTHE
ELDESTSAIDITISAGUNFIREDFORJOYBECAUSEWEHAVEGOTRIDOFOURPRINCESOQUICKLYAFTE
RTHATTHEYCAMEINTOANAVENUEWHEREALLTHELEAVESWEREOFGOLDANDLASTLYINTOATH
IRDWHERETHEYWEREOFBRIGHTDIAMONDSHEBROKEOFFATWIGFROMEACHWHICHMADESUC
HACRACKEACHTIMETHATTHEYOUNGESTSTARTEDBACKINTERRORBUTTHEELDESTSTILLMAI
NTAINEDTHATTHEYWERESALUTESTHEYWENTONANDCAMETOAGREATLAKEWHEREONSTOO
DTWELVELITTLEBOATSANDINEVERYBOATSATAHANDSOMEPRINCEALLOFWHOMWEREWAITI
NGFORTHETWELVEANDEACHTOOKONEOFTHEMWITHHIMBUTTHESOLDIERSEATEDHIMSELFB
YTHEYOUNGESTTHENHERPRINCESAIDIWONDERWHYTHEBOATISSOMUCHHEAVIERTODAYISH
ALLHAVETOROWWITHALLMYSTRENGTHIFIAMTOGETITACROSSWHATSHOULDCAUSETHATSAI
DTHEYOUNGESTBUTTHEWARMWEATHERIFEELVERYWARMTOOONTHEOPPOSITESIDEOFTHE
LAKESTOODASPLENDIDBRIGHTLYLITCASTLEFROMWHENCERESOUNDEDTHEJOYOUSMUSICOF
TRUMPETSANDKETTLEDRUMSTHEYROWEDTHEREENTEREDANDEACHPRINCEDANCEDWITHT
HEGIRLHELOVEDBUTTHESOLDIERDANCEDWITHTHEMUNSEENANDWHENONEOFTHEMHADA
CUPOFWINEINHERHANDHEDRANKITUPSOTHATTHECUPWASEMPTYWHENSHECARRIEDITTOH
ERMOUTHTHEYOUNGESTWASALARMEDATTHISBUTTHEELDESTALWAYSSILENCEDHERTHEYD
ANCEDTHERETILLTHREEOCLOCKINTHEMORNINGWHENALLTHESHOESWEREDANCEDINTOHO
LESANDTHEYWEREFORCEDTOLEAVEOFFTHEPRINCESROWEDTHEMBACKAGAINOVERTHELAK
EANDTHISTIMETHESOLDIERSEATEDHIMSELFBYTHEELDESTONTHESHORETHEYTOOKLEAVEO
FTHEIRPRINCESANDPROMISEDTORETURNTHEFOLLOWINGNIGHTWHENTHEYREACHEDTHES
TAIRSTHESOLDIERRANONINFRONTANDLAYDOWNINHISBEDANDWHENTHETWELVEHADCOM
EUPSLOWLYANDWEARILYHEWASALREADYSNORINGSOLOUDLYTHATTHEYCOULDALLHEARHI
MANDTHEYSAIDSOFARASHEISCONCERNEDWEARESAFETHEYTOOKOFFTHEIRBEAUTIFULDRES
SESLAIDTHEMAWAYPUTTHEWORNOUTSHOESUNDERTHEBEDANDLAYDOWNNEXTMORNINGT
HESOLDIERWASRESOLVEDNOTTOSPEAKBUTTOWATCHTHEWONDERFULGOINGSONANDAGAI
NWENTWITHTHEMASECONDANDATHIRDNIGHTTHENEVERYTHINGWASJUSTASITHADBEENTH
EFIRSTTIMEANDEACHTIMETHEYDANCEDUNTILTHEIRSHOESWEREWORNTOPIECESBUTTHET
HIRDTIMEHETOOKACUPAWAYWITHHIMASATOKENWHENTHEHOURHADARRIVEDFORHIMTO
GIVEHISANSWERHETOOKTHETHREETWIGSANDTHECUPANDWENTTOTHEKINGBUTTHETWEL
VESTOODBEHINDTHEDOORANDLISTENEDFORWHATHEWASGOINGTOSAYWHENTHEKINGPUT
THEQUESTIONWHEREHAVEMYTWELVEDAUGHTERSDANCEDTHEIRSHOESTOPIECESINTHENIG
HTHEANSWEREDINANUNDERGROUNDCASTLEWITHTWELVEPRINCESANDRELATEDHOWITHA
DCOMETOPASSANDBROUGHTOUTTHETOKENSTHEKINGTHENSUMMONEDHISDAUGHTERSAND
ASKEDTHEMIFTHESOLDIERHADTOLDTHETRUTHANDWHENTHEYSAWTHATTHEYWEREBETR
AYEDANDTHATFALSEHOODWOULDBEOFNOAVAILTHEYWEREOBLIGEDTOCONFESSALLTHERE
UPONTHEKINGASKEDWHICHOFTHEMHEWOULDHAVETOWIFEHEANSWEREDIAMNOLONGERY
OUNGSOGIVEMETHEELDESTTHENTHEWEDDINGWASCELEBRATEDONTHESELFSAMEDAYAND
THEKINGDOMWASPROMISEDHIMAFTERTHEKINGSDEATHBUTTHEPRINCESWEREBEWITCHED
FORASMANYDAYSASTHEYHADDANCEDNIGHTSWITHTHETWELVEONESEVENTWOFOUR

PART 2

Paste your bash script program to perform *Viginère Autokey* encryption/decryption in full into the space below. Add comments into your script where/when necessary to indicate portions for (1) user input of keyword, (2) user input of plain/ciphertext to en/decrypt and (3) file output save options (if used, else overwrite file). You may include screenshots of your program running (recommended in case I am not able to get your script to run) and sample the outputs generated from the terminal window

[8 marks]

```
#!/bin/bash
op="+"


# "Tests"

while getopts edk: flags
do
  case $flags in
  e)  continue=1;;
  d)  continue=1; op="-";;
  k)  key=$(echo "$OPTARG" | tr A-Z a-z | sed s/[^a-z]//g); option=1;;
# Argument is taken as key.
  esac                        # "op" will be used to determine if -k was
used and for "error" messages
done

read -t 0 < /dev/stdin
if [[ "$?" = 0 ]]
then
  continue=1            # Continue -> 1, as default to indicate to break
out from the loop
  pipe=1                # Pipe -> 1
  if [[ -z "$key" ]]
  then
    echo "-k flag with an argument containing at least 1 letter required
when piping."  # Error message will be shown if there is pipping
    exit 1              # Exit from the script
  fi
fi


#USER INPUT IN DETERMINING WHETHER THEY WANT TO ENCRYPT OR DECRYPT

while [[ "$continue" != 1 ]]  # Loop will keep moving if the continue is
not equal to 1
do
  echo "----------------------------------------"
  echo "ENCODING/DECODING USING VIGENERE CIPHER"
  echo "Encode -> 1"
  echo "Decode -> 2"
  echo "----------------------------------------"
```

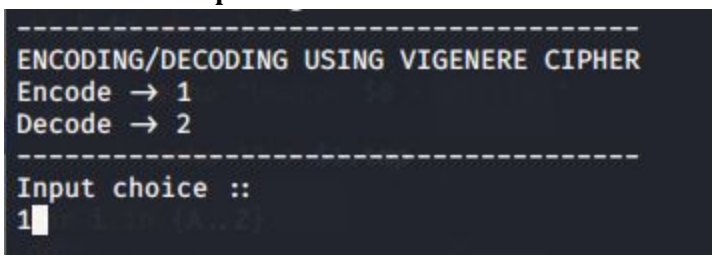
```
  echo "Input choice :: "
  read -n 2  choice                        # User input will be read
  echo
  if [[ "$choice" = 2 ]] # When user input 2
  then
    continue=1            # Break the loop
    op="-"                # Moving forward to Decryption
  fi
  if [[ "$choice" = 1 ]] # If 0 was entered (no):
  then
    continue=1            # Break the loop
    op="+"                # Moving forward to Encryption
  fi
  if [[ "$continue" != 1 ]]     #Check the continue value
  then
    echo "Invalid Input! Please Enter Again : "  #Error message will be
shown when user input invalid value
  fi
done


#USER INPUT MESSAGE TO BE ENCODE OR DECODE

if [[ "$pipe" = 1 ]]
then
  read pt < /dev/stdin
else
  echo "---------------------------------"
  echo Input Message to be Encode/Decode ::           # Ask user to
input message to be encrypt or decrypt
  read pt                                              # Read the input
given by the user
fi
pt=$(echo "$pt" | tr a-z A-Z | sed s/[^A-Z]//g) # Change all characters
to uppercase and remove all spacing


#USER INPUT KEYWORD

while test -z "$key"                                  # To check if key is
empty
do
  if [[ "$option" = 1 ]]
  then
    echo "-k Argument need at least ONE letter"
  fi
  echo "---------------------------------"
  echo Input Key to be Used ::                        # Ask the user to
input the key to be used
  read key                                            # Read the input
given by the user
  key=$(echo "$key" | tr a-z A-Z | sed s/[^A-Z]//g) # Change all
characters to uppercase and remove all spacing
```

```
  if [[ -z "$key" ]]                                    # The input must
have input, or else error message will be shown
  then
    echo "Invalid! Atleast one letter is needed :: "
  fi
done

length=${#key}  # Length of the key
step=0


# ENCODE / DECODE USING THE KEY GIVEN

while test -n "$pt"                      # While the plaintext is not
zero, the loop while keep working
do
  char=${pt:0:1}                         # Set the position of char to 1
  loop=25                                       # Loop set to 25
(representing num of characters)
  for letter in {Z..A}                          # Loop through the letter
from Z .. A
  do
    char=$(echo $char | sed s/$letter/$loop/)
    loop=$((loop-1))
  done

  loop=25               # Reset the loop
  shift=${key:$step:1}
  for letter in {Z..A}
  do
    shift=$(echo $shift | sed s/$letter/$loop/)
    loop=$((loop-1))
  done

  # Step will be increase +1 and will be mod with the length of key
  step=$(($(($step+1))%$length))


  code=$(($char$op$shift))
  if [[ $code -lt 0 ]]      # If the output is < 0
  then
    code=$((code+26))       # will be add 26 so that it will loop
  fi
  if [[ $code -gt 25 ]]     # If result > 25
  then
    code=$((code-26))       # - 26
  fi

  # Convert number -> alphabet
  loop=25
  for letter in {Z..A}
  do
    code=$(echo $code | sed s/$loop/$letter/)
    loop=$((loop-1))
```

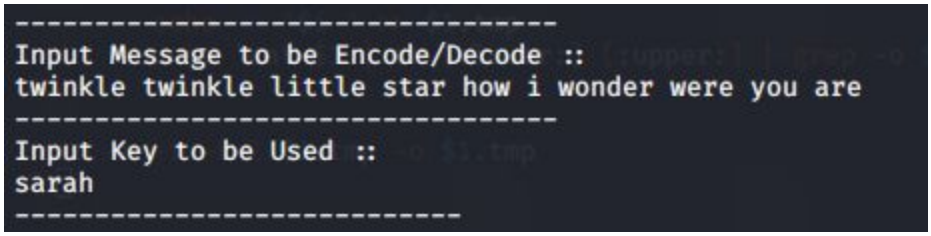
```
  done


  # The encoded / decoded message will be save into message
  message=$message$code
  # Remove the character of plaintext one-by-one
  pt=${pt:1}
done




#SAVE THE ENCODED OR DECODED MESSAGE INTO A TEXTFILE
#AT THE SAME TIME, THE FREQUENCY ANALYSIS WILL BE DONE
#IF SAVE THE FILE, WE CAN GET THE FREQUENCY ANALYSIS
echo "----------------------------"
echo "Save output into a textfile?"
echo "Yes -> 1"
echo "No -> 2"
read -n 2 save
if [[ "$save" = 2 ]]
 then
   echo "----------------------------"
   echo The Encoded/Decoded Message ::
   echo $message
   echo "Thank You :)) "
   echo "----------------------------"
elif [[ "$save" = 1 ]]
 then
   echo "----------------------------"
   echo "Enter the Text File Name :: "
   read filename
   if [ -f $filename.txt ]
    then
      echo "--------------------------"
      echo "The file already existed. "
      echo "Overwritting in process..."
      echo $message > $filename.txt
      grep -o . $filename.txt | sort | uniq -c | sort -rn >
$filename.txt.tmp
      echo "Done!"
      echo "--------------------------"
   else
     echo $message > $filename.txt
     grep -o . $filename.txt | sort | uniq -c | sort -rn >
$filename.txt.tmp
    fi
else
 echo The Encoded/Decoded Message ::
 echo $message

fi
```

1) The user will be given option whether they want to encode(1) or decode(2)

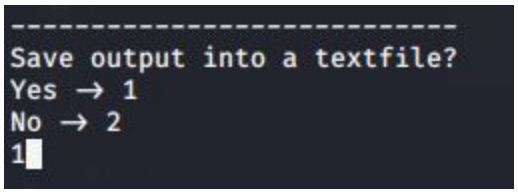
**ENCODE part :**



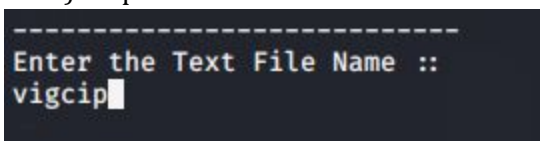

2) Input the message and key to be used in the encoding




3) User can choose to save the output in a text file by input 1

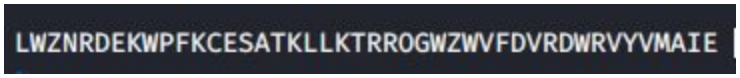



4) Input the name of the text file

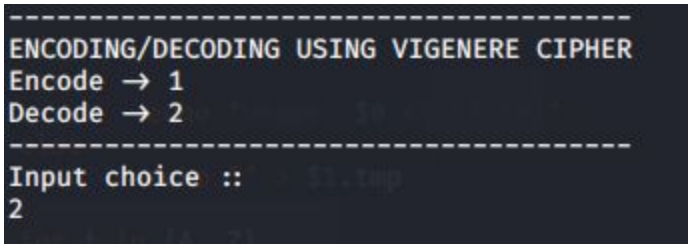



5) To see the encoded output, the user need to type :


6) The output will be shown in the text file




**DECODE part :**



1) Input the message and key to be decode

```
---------------------------------
Input Message to be Encode/Decode ::
lwznrdekwpfkcesatkllktrrogwzwvfdvrdwrvyvmaie
---------------------------------
```

```
---------------------------------
Input Key to be Used ::
sarah
---------------------------
```

2) Choose whether to save the output in a text file

```
---------------------------
Save output into a textfile?
Yes → 1
No → 2
1
---------------------------
```

3) Write the file name

```
---------------------------
Enter the Text File Name ::
dcrypt
```

4) To see the output, type:

```
vi dcrypt.txt
```

5) The output of the decoded message

```
TWINKLETWINKLELITTLESTARHOWIWONDERWEREYOUARE
```

Below are the output if the user wanted to perform encoding or decoding and to overwrite an existing text file.

```
-------------------------------------
ENCODING/DECODING USING VIGENERE CIPHER
Encode → 1
Decode → 2
-------------------------------------
Input choice ::
2

---------------------------------
Input Message to be Encode/Decode ::
lhvwowecoulhvbbkgfezjolnksnurvmnu
---------------------------------
Input Key to be Used ::
sarah
----------------------------
Save output into a textfile?
Yes → 1
No → 2
1
--------------------------
Enter the Text File Name ::
dcrypt
-------------------------
The file already existed.
Overwritting in process ...
Done!
-------------------------
```

```
vi dcrypt.txt
```

```
THEWHEELONTHEBUSGOESROUNDANDROUND
```

Terminal output results of the frequency analysis on your encrypted text file (either copy/paste output or screenshot)

[2 marks]

vigcip.txt.tmp

```
5 W
5 R
4 V
4 K
3 L
3 E
3 D
2 Z
2 T
2 F
2 A
1 Y
1 S
1 P
1 O
1 N
1 M
1 I
1 G
1 C
```

Brief explanation on the frequency analysis of the ciphertext generated by the Viginère Autokey cipher. Compare and contrast the results with the analysis from the simple substitution earlier and explain how it affects cryptanalysis of the ciphertext.

[5 marks]

The frequency analysis obtained above shows that the frequency of each character seems to have almost the same number (examples: W and R have a frequency of 5).

| Viginère Autokey Cipher | Simple Substitution |
|---|---|
| - uses a simple form of polyalphabetic substitution.<br>- Not open to the frequency analysis because the cipher rotates in different shifts.<br>- Plaintext not will not be encrypted with the same letter of ciphertext<br>- Disadvantage : repeating of the key, can easily be broken | - Characters from plaintext will be given a fixed characters to be substituted (based on the key)<br>- The security of the text can be increased by the variety multilateral.<br>- Disadvantage : length of message and the time taken to transmit the output will increase if the plaintext is substitute with more than one character of the ciphertext |

ADDITIONAL NOTES (OPTIONAL – NO MARKS ALLOTTED)

Any additional information required to run your script (e.g. what type of linux environment you used, what bash engine etc), what script to call to run which function (or what menu option to choose to execute a function)

Linux environment used : Virtual Box Kali
Script to call :
Part 1 : count.sh (when run : **./count.sh 1181301724.txt** )
subs.sh
Part 2 : vig2.sh

SUBMISSION INSTRUCTIONS

Fill in the sections above with your results, save the file and zip up this report file with your bash scripts before submitting into the assignment section in Google classroom.