Faculty Of Engineering and Technology

Electrical And Computer Engineering Department

Linux Lab

ENCS 313

Shell Scripting Project Report

Students Name_ID :

Rawan Yassin_1182224

Alaa Zuhd_1180865

Instructor Name : Dr. Aziz Qaroush

TA Name : Eng. Assel Awwad

Section : 2

Date : 14-8-2020

Abstract

The aim of this project is to be more familiar with shell script programming by building a shell script that dose a simple encryption/decryption based on Caesar-cipher algorithm for English-Based text messages .

Table Of Contents

Description of Tasks (codes and test cases for the outputs) :

We will use this test case to discuss the output of the following tasks .

```
rawan@ubuntu:~$ cat file.txt
My n123aMe IS Alaa
MY na#me i%s r&60AwAN
rawan@ubuntu:~$ ./SimpleEncryption.sh
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    ------------------------------------------------
Please enter e to do encryption or d to do decryprion
e
------------------------
Please enter the name of the plain file  you would like to get the text from:
file.txt
------------------------
Reading done successfully
------------------------
Removing none alphabet characters done:
My naMe IS Alaa
MY name is rAwAN
------------------------
Converting to lower case done:
my name is alaa
my name is rawan
------------------------
The sum of word characters frequencies:
my --- 6
name --- 16
is --- 4
alaa --- 22
my --- 6
name --- 16
is --- 4
rawan --- 19
------------------------
The shift value is: 22
------------------------
The encrypted text to be printed in a file is:
iu jwia eo whww
iu jwia eo nwswj
------------------------
Please enter the name of the file you would like to write the cipher text into:
encryption.txt
Writting the text into the file done successfully
------------------------
Thanks for using!
```

✓ The main menu :

It asks the user to enter ‘e’ for encryption or ‘d’ for decryption , and if he/she enters another character then the program will terminate , otherwise if the entered character is considered valid then the main menu will call the remaining tasks (functions) to achieve the required task .

```
#This is our main
#it prints a welcome message and then asks the user to choose between encryption and decryption
echo "                    Welcome to the most GENUIS Caesar Cipher Analyzer   "
echo "                    ------------------------------------------------   "
echo "Please enter e to do encryption or d to do decryprion "
read  choice
echo "------------------------"

if [ "$choice" = "E" -o "$choice" = "e" ]
then
readFile
removeNonAlphabetical
convertToLowerCase
countFrequency
CaculatingShiftValue
encryption
writeFile
elif [ "$choice" = "D" -o "$choice" = "d" ]
then
readFile
#no need to convert to lower case or to remove nonalphabetical characters since it is already ciphered
cat temp1.txt > temp11.txt
countFrequency
CaculatingShiftValue
decryption
writeFile
else
echo "Not Valid!"
exit 1
fi
echo "Thanks for using!"
```

✓ The output of the main menu :

```
rawan@ubuntu:~$ ./SimpleEncryption.sh
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    ------------------------------------------------
Please enter e to do encryption or d to do decryprion
e
------------------------
```

✓ readFile Function :

This function will do the task of asking the user to enter the name of the plain text file to read it , and it will handle any possible cases for the entry file name . for example if the entered name for the file to read from does not exist then it will ask the user to either continue the program and enter the name of the file again or terminate the program , also if the entered file name exists but it's not an ordinary file then it will do the same as the previous case . and finally when the user enter a valid name then it will read it and put it in temp1.txt file .

```
#readFile function asks the user to enter the name of the file to read the text from, then checks that it exits and that it is an ordinary file
readFile () {
#cheacking if choice is e , if it's then the type of the text to be written is plain else it's a cipher text
  if [ "$choice" = e ] ; then
     textType="plain"
  else
     textType="cipher"
  fi
#we are in a while statement, so as if the user enters an invalid file, he/she can choose between terminating or continue
  while true
  do
    echo "Please enter the name of the $textType file  you would like to get the text from:"
    read fileName
    if [ ! -e "$fileName" ] ; then
      echo "File dosn't exist!, would you like to try another name(y) or teminate (t): "
      #flag is used to read if reader wants to try again or terminate
      read flag
      if [ "$flag" = "y" ] ; then
        continue
      else
        exit 1
      fi
    else
      if [ ! -f "$fileName" ] ; then
        echo "It's not a file to read from!, would you like to try another name(y) or terminate (t): "
        read flag
        if [ "$flag" = "y" ] ; then
          continue
        else
          exit 1
        fi
      else
        echo "-------------------------"
        echo "Reading done successfully"
        echo "-------------------------"
        cat "$fileName" > temp1.txt
        break
      fi
    fi
  done
}
```

✓ The output of The readFile function is :

```
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    ------------------------------------------------
Please enter e to do encryption or d to do decryprion
e
-------------------------
Please enter the name of the plain file  you would like to get the text from:
n.txt
File dosn't exist!, would you like to try another name(y) or teminate (t):
y
Please enter the name of the plain file  you would like to get the text from:
encs
It's not a file to read from!, would you like to try another name(y) or terminate (t):
y
Please enter the name of the plain file  you would like to get the text from:
file.txt
-------------------------
Reading done successfully
-------------------------
```

✓ removeNonAlphabetical Function :

This function will do the task of removing all non-alphabetical characters from the text in temp1.txt using sed command .

```
#removeNonAlphabetical function uses the sed command to delete all nonaphabetical characters
removeNonAlphabetical () {
  sed -i 's/[^a-zA-Z ]//g' temp1.txt
  echo "Removing none alphabet characters done: "
  cat temp1.txt
  echo "........................."
}
```

✓ The output of the removeNonAlphabetical function is :

```
-------------------------
Removing none alphabet characters done:
My naMe IS Alaa
MY name is rAwAN
-------------------------
```

✓ convertToLowerCase Function :

This function will do the task of converting all the upper case letters in the text temp1.txt file into lower case letters . also here temp11.txt file is used to store the text after conversion and removing all non-alphabetical characters to use it later .

```
#convertToLowerCase function as its name indicates, it changes the alphabets to their lower case using tr and stores the result is temp1 file
convertToLowerCase () {
  cat temp1.txt | tr '[A-Z]' '[a-z]' > temp2.txt
  mv temp2.txt temp1.txt
  cp "temp1.txt" "temp11.txt"
  echo "Converting to lower case done: "
  cat temp1.txt
  echo "----------------------"
}
```

✓ The output of convertTolowerCase function is :

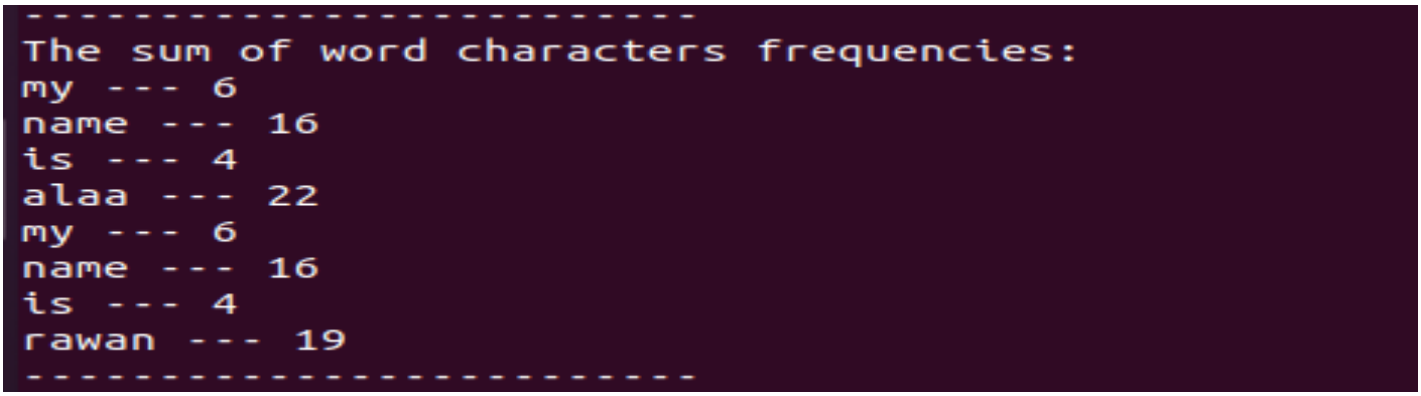
```
--------------------------
Converting to lower case done:
my name is alaa
my name is rawan
--------------------------
```

✓ countFrequency Function :

This function will do the task of counting the frequency of each character and store them into an array called countOfFrequency , and then calculate the frequency of the characters in each word , by taking the summation of characters frequency and using the ascii value of each character by using printf , and then obtain it's index in the countOfFrequency array by subtract 97 from the ascii value , and this process will continue for each character in each word , then the frequency of the words will be store in wordFreq array , and after that it will print the words with their frequency (using wordFreq array) . The comments in the following snaps contain all the information for the commands and steps to obtain the frequency of each character and then calculate the frequency of each word , and finally print the words with their frequency

```
#countFrequency function aims mainly to calculate the frequencies of letters and words as described below
countFrequency(){
  wc -w temp1.txt > temp2.txt #count words in the file that has the text to be encrypted or decrypted
  sed -i 's/\(.\)/\1\n/g' temp1.txt #seperating each letter in a line so the counting process is easier
  index=0
  #after having each letter printed in a line, counting the letter would be easy by counting the lines that has the letter using grep and wc
  #countOfFrequency array has the frequency of each character, where for example the index for letter a would be 0 and so on
  for i in {a..z}
   do
    countOfFrequency["$index"]=$(grep "$i" temp1.txt| wc -l)
    index=$((index+1))
   done
  countOfFrequency[$index]=$(cut -d" " -f1 temp2.txt) #the last element of the countOfFrequency array is the count of words in the required file
  #Counting the frequency of each seperate word
  index=0
  while read line
  do
    temp=$(printf "%d" "'$line")   #obtaining the asci value of the each seperate letter
     if [ "$temp" -ge 97 -a "$temp" -le 122 ] ; then #checking that it is a letter, not a space or new line
     temp=$((temp-97))   #to obtain the index associated with the countOfFrequency array
     sum=$((sum+${countOfFrequency["$temp"]}))
    else
    #wordFreq array stores the frequency of each word after summing all its letters
      wordFreq["$index"]=$sum
      index=$((index+1))
      sum=0
      continue
    fi
  done < temp1.txt
 #printing the word freq
 echo "The sum of word characters frequencies: "
 i=0
 while read line
  do
   temp=$(printf "%d" "'$line")
   if [ $temp -eq 0 ] ; then  #if a zero ascii letter is reached, then we have done with this word and need to print its sum of letters frequency
    echo -n " --- "
    echo "${wordFreq["$i"]}"
    i=$((i+1))
   else                #if a non zero letter (space) is reached, print the letters to have the word
    echo -n $line
   fi
done < temp1.txt
 echo "-------------------------"
}
```

✓ **The output of the countOfFreqeuncy function Is :**



According to the above test case , Each letter frequency is :
f(m)=4, f(y)=2, f(n)=3, f(a)=7, f(e)=2, f(i)=2, f(s)=2, f(l)=1, f(r)=1, f(w)=1

And each word frequency :
f(my)= 4+2 = 6
f(name)= 3+7+4+2 = 16
f(is)= 2+2 = 4
f(alaa)= 7+1+7+7 = 22
f(my) 4+2 = 6
f(name)= 3+7+4+2 = 16
f(is)= 2+2 = 4
f(rawan)= 1+7+1+7+3 = 19 ➔ The result is exactly as the output in the above snap .

✓ calculatingShiftValue Function :

This function will do the task of calculating the shift value by finding the maximum between each word frequency and then calculate the shift value using the following equation ➔ shiftValue = max(wordFreq) % 26 .

```
#CalculatingShiftValue funtion works as described below
CaculatingShiftValue(){
#first, we will be looping through the wordFreq array to get the maximum number
i=1
max="${wordFreq[0]}"
for i in "${wordFreq[@]}"
 do
  if [ "$i" -gt "$max" ] ; then
   max="$i"
  fi
 done
#then, calculating the shift value,which is equal to the max value mod 26
shift=$( expr $max % 26 )
echo "The shift value is: $shift "
echo "------------------------"
}
```

✓ The output of the caculatingShiftValue function is :

```
------------------------
The shift value is: 22
------------------------
```

- Shift value = { max[(4+2),(3+7+4+2),(2+2),(7+1+7+7),(4+2),(3+7+4+2),(2+2),(1+7+1+7+3)] mod 26} = 22 . and it's exactly as the one we get from the program in the above test case .

✓ encryption Function :

This function will do the task of encryption a given text , by finding the character that corresponding to ‘a’ , and then finding the character that corresponding to ‘z’ , which can be done by finding the ascii value of the character that correspond to ‘a’ (which = 97+shiftValue) , and then the character that correspond to ‘z’ (which is = shiftCaue+96) . The comments in the below code describe the whole process exactly .

```
#The encryption function is called after calculating the shif value and having the text converted to small letters and with no nonalphabetical characters
#it does the encryption as the following steps:
#1) it calcuates the ascii value of the letter that would replace the letter 'a', which is equal to the shift value and the 'a' ascii value
#2) it then converts the "start" ascii value to the needed letter
#3) it calculates the ascii value of the letter that would replace the letter 'z', which is equal to the letter right before the stating character
#4) it obtains the ending character after converting the ascii value
#5)) Now, the encryption is done using the tr command, in which the range to be replaced is [a-z] by [begeningCharacter-za-endingCharacter], and this command would be replacing
# letter 'a' with the first in the second range, which is the begeningCharacter, and then the letter 'b' by the second character in the second range, and so on, until it reaches
# 'z' then starts over.
encryption(){

  if [ "$shift" != 0 ] ; then
     start=$((shift+97))
     begChar=$(printf \\$(printf "%03o" "$start"))
     end=$((start-1))
     endChar=$(printf \\$(printf "%03o" "$end"))
     cat temp11.txt | tr '[a-z]' [$begChar-za-$endChar] > temp1.txt
     echo "The encrypted text to be printed in a file is:"
     cat temp1.txt
     echo "----------------------"
  fi
}
```

✓ The output of the encryption function is : note the text in the before encryption was “my name is alaa

my name is rawan “

and the shift value was 22 , so output in the following case is 100% true .

```
--------------------------
The encrypted text to be printed in a file is:
iu jwia eo whww
iu jwia eo nwswj
--------------------------
```

The following is a decryption example, using the file encrypted in the above case, to verify that the decryption function works well

```
rawan@ubuntu:~$ cat encryption.txt
iu jwia eo whww
iu jwia eo nwswj
rawan@ubuntu:~$ ./SimpleEncryption.sh
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    ------------------------------------------------
Please enter e to do encryption or d to do decryprion
d
------------------------
Please enter the name of the cipher file  you would like to get the text from:
encryption.txt
------------------------
Reading done successfully
------------------------
The sum of word characters frequencies:
iu --- 6
jwia --- 16
eo --- 4
whww --- 22
iu --- 6
jwia --- 16
eo --- 4
nwswj --- 19
------------------------
The shift value is: 22
------------------------
The decrypted text to be printed in a file is:
my name is alaa
my name is rawan
------------------------
Please enter the name of the file you would like to write the plain text into:
decryption.txt
Writting the text into the file done successfully
------------------------
Thanks for using!
```

✓ decryption Function :

this function decrypt the cipher text, using exactly the same way of encryption a given text except that the ranges in tr command were switched.

```
#The decryption function is performed using the same concept used in encryption, excpet that is would be replacing the opposite ranges
decryption(){
  if [ "$shift" != 0 ] ; then
      start=$((shift+97))
      begChar=$(printf \\$(printf "%03o" "$start"))
      end=$((start-1))
      endChar=$(printf \\$(printf "%03o" "$end"))
      cat temp11.txt | tr [$begChar-za-$endChar] '[a-z]'> temp1.txt
      echo"The decrypted text to be printed in a file is:"
      cat temp1.txt
      echo "----------------------"
  fi
}
```

✓ The output of the decryption function is :

Note the input file for this case was the same as the output of the previous task (encryption task) and as we note the output here is the same as the input in that task , which means the output is 100% true .

```
--------------------------
The decrypted text to be printed in a file is:
my name is alaa
my name is rawan
--------------------------
```

✓ writeFile function :

This function will do the task of asking the user to enter the name of the file to write a text on , and it handles all the possible cases , for example if the file exists then it will ask the user to either overwrite the file or no , and if he/she dosen't want to overwrite the file then it will give the user a chance to either continue the program and try to enter aonther file name or terminate from the program , another possible case that the function handle is that if the file exists but it's not an ordinary file then it asks the user to either contiue the program and try aonther name or to terminate . Finally if the file name is valid then it will write the desired text into it .

```
#writeFile function writes the processed text into a file
writeFile() {
  #checking if choice is e , if it's then the type of the text to be written is cipher else it's a plain text
  if [ "$choice" = e ] ; then
     textType="cipher"
  else
     textType="plain"
  fi
  while true
  do
    #ask the user to enter the name of the file to write the text into
    echo "Please enter the name of the file you would like to write the $textType text into: "
    read fileName
     if [ -e "$fileName" ] ; then
          # checking if the given file is an ordinary file or not
       if [ -f "$fileName" ] ; then
          #ask the user if he want to override the file or not, as it exists
          echo "The file already exists . Do you want to override the file ?(yes/no) "
          read ans
          anss=$(echo "$ans" | tr '[A-Z]' '[a-z]' )
          # if the answer is anything other than yes then exit
          if [ $anss != yes ] ; then
            echo "Do you want to continue and try again(y) or terminate(t) ? : "
            read flag
            echo "-------------------------"
            if [ $flag = y ] ; then
               continue
            else
               exit 1
            fi
          else
            break
          fi
        else  # the given file is not an ordinary file so we can't override it or creat a file with the same name
          echo "The given file is exist , but it's not an ordinary file so we can't write the text into it."
          echo "Do you want to continue and try again(y) or terminate(t) ? : "
          read flag
          echo "-------------------------"
          if [ $flag = y ] ; then
             continue
          else
             exit 1
          fi
        fi
        else
          break
     fi
  done
   #writing the text into the file from temp1.txt as it has the encrypted\decrypted text
   cat temp1.txt > "$fileName"
   echo "Writting the text into the file done successfully "
   echo "------------------------"
}
```

✓ The output cases for writeFile function :

First Case : test.txt is not empty so you can select yes to overwrite or no if you don't want to overwrite it , here we choose to overwrite it .

```
rawan@ubuntu:~$ cat test.txt
Linux Lab Project 1
rawan@ubuntu:~$ ./SimpleEncryption.sh
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    ------------------------------------------------
Please enter e to do encryption or d to do decryprion
e
------------------------
Please enter the name of the plain file  you would like to get the text from:
file.txt
Removing none alphabet characters done:
My naMe IS Alaa
MY name is rAwAN
------------------------
Converting to lower case done:
my name is alaa
my name is rawan
------------------------
The sum of word characters frequencies:
my --- 6
name --- 16
is --- 4
alaa --- 22
my --- 6
name --- 16
is --- 4
rawan --- 19
------------------------
The shift value is: 22
------------------------
The encrypted text to be printed in a file is:
iu jwia eo whww
iu jwia eo nwswj
------------------------
Please enter the name of the file you would like to write the cipher text into:
test.txt
The file already exists . Do you want to override the file ?(yes/no)
yes
Writting the text into the file done successfully
------------------------
Thanks for using!
rawan@ubuntu:~$ cat test.txt
iu jwia eo whww
iu jwia eo nwswj
```

Second Case : here encs is a directory (not an ordinary file) so we have to choose either to continue the program and try to enter another file name or choose to terminate the program , here we chose to continue and enter another name which is (e.txt) .

```
rawan@ubuntu:~$ ./SimpleEncryption.sh
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    ------------------------------------------------
Please enter e to do encryption or d to do decryprion
e
------------------------
Please enter the name of the plain file  you would like to get the text from:
file.txt
Removing none alphabet characters done:
My naMe IS Alaa
MY name is rAwAN
------------------------
Converting to lower case done:
my name is alaa
my name is rawan
------------------------
The sum of word characters frequencies:
my --- 6
name --- 16
is --- 4
alaa --- 22
my --- 6
name --- 16
is --- 4
rawan --- 19
------------------------
The shift value is: 22
------------------------
The encrypted text to be printed in a file is:
iu jwia eo whww
iu jwia eo nwswj
------------------------
Please enter the name of the file you would like to write the cipher text into:
encs
The given file is exist , but it's not an ordinary file so we can't write the text into it.
Do you want to continue and try again(y) or terminate(t) ? :
y
------------------------
Please enter the name of the file you would like to write the cipher text into:
e.txt
Writting the text into the file done successfully
------------------------
Thanks for using!
```

General test Cases for the whole program :

✓ Test Case#1 :

```
rawan@ubuntu:~$ cat file.txt
Project i^&s DoN$e Finall123y
Jerusalem iS THE CAPITAL of PAlestine
```

```
rawan@ubuntu:~$ ./SimpleEncryption.sh
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    ------------------------------------------------
Please enter e to do encryption or d to do decryprion
e
------------------------
Please enter the name of the plain file  you would like to get the text from:
file.txt
------------------------
Reading done successfully
------------------------
Removing none alphabet characters done:
Project is DoNe Finally
Jerusalem iS THE CAPITAL of PAlestine
------------------------
Converting to lower case done:
project is done finally
jerusalem is the capital of palestine
------------------------
The sum of word characters frequencies:
project --- 23
is --- 9
done --- 14
finally --- 26
jerusalem --- 34
is --- 9
the --- 12
capital --- 29
of --- 5
palestine --- 43
------------------------
The shift value is: 17
------------------------
The encrypted text to be printed in a file is:
gifavtk zj ufev wzerccp
aviljrcvd zj kyv trgzkrc fw grcvjkzev
------------------------
Please enter the name of the file you would like to write the cipher text into:
output1.txt
Writting the text into the file done successfully
------------------------
Thanks for using!
```

```
rawan@ubuntu:~$ ./SimpleEncryption.sh
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    -------------------------------------------------
Please enter e to do encryption or d to do decryprion
d
------------------------
Please enter the name of the cipher file  you would like to get the text from:
output1.txt
------------------------
Reading done successfully
------------------------
The sum of word characters frequencies:
gifavtk --- 23
zj --- 9
ufev --- 14
wzerccp --- 26
aviljrcvd --- 34
zj --- 9
kyv --- 12
trgzkrc --- 29
fw --- 5
grcvjkzev --- 43
------------------------
The shift value is: 17
------------------------
The decrypted text to be printed in a file is:
project is done finally
jerusalem is the capital of palestine
------------------------
Please enter the name of the file you would like to write the plain text into:
output2.txt
Writting the text into the file done successfully
------------------------
Thanks for using!
rawan@ubuntu:~$ cat output2.txt
project is done finally
jerusalem is the capital of palestine
```

- We encrypted the text in file.txt and save the encrypted text into output1.txt file , then decrypted the text in output1.txt and store the result in output2.txt and the result In output2.txt was the same as the one in file.txt except the conversion from uppercase to lowercase for the letters which is done in the encryption process , which supports the validity of our program code .

✓ Test case#2 :

```
rawan@ubuntu:~$ ./SimpleEncryption.sh
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    ------------------------------------------------
Please enter e to do encryption or d to do decryprion
d
------------------------
Please enter the name of the cipher file  you would like to get the text from:
n.txt
File dosn't exist!, would you like to try another name(y) or teminate (t):
t
```

- Here in the decryption process we have entered a file name which does not exist then we have chosen ‘t’ to terminate the program instead of continue and try to enter another file name .

✓ Test case#3 :

```
rawan@ubuntu:~$ cat file.txt
Project i^&s DoN$e Finall123y
Jerusalem iS THE CAPITAL of PAlestine
```

```
                    Welcome to the most GENUIS Caesar Cipher Analyzer
                    -------------------------------------------------
Please enter e to do encryption or d to do decryprion
e
-------------------------
Please enter the name of the plain file  you would like to get the text from:
n.txt
File dosn't exist!, would you like to try another name(y) or teminate (t):
y
Please enter the name of the plain file  you would like to get the text from:
file.txt
-------------------------
Reading done successfully
-------------------------
Removing none alphabet characters done:
WE are testing a Random Case
-------------------------
Converting to lower case done:
we are testing a random case
-------------------------
The sum of word characters frequencies:
we --- 5
are --- 10
testing --- 14
a --- 4
random --- 11
case --- 11
-------------------------
The shift value is: 14
-------------------------
The encrypted text to be printed in a file is:
ks ofs hsghwbu o fobrca qogs
-------------------------
Please enter the name of the file you would like to write the cipher text into:
encs
The given file is exist , but it's not an ordinary file so we can't write the text into it.
Do you want to continue and try again(y) or terminate(t) ? :
y
-------------------------
Please enter the name of the file you would like to write the cipher text into:
result.txt
Writting the text into the file done successfully
-------------------------
Thanks for using!
```

- Here, firstly n.txt file did not exist, so we asked the user to try another name, then a valid file was given(file.txt) and then when writing the results, encs was firstly detected not an ordinary file to write on, then after asking the user to terminate or continue we have chosen to try another name and then a message indicating that writing done successfully appeared

Conclusion and Future Work

In this project, we have learned how to do simple encryption and decryption using the Caesar cipher, we have practiced the shell programming and tried to include as much as possible cases. Building the program have motivated us to be familiar with encryption and decryption and to understand their real benefits.

As a future plan, this project has motivated us to consider other cipher techniques and to consider how errors could be handled when receiving data.

References

1. https://www.tutorialspoint.com/unix/unix-shell-functions.htm
   [Accessed on 11/Aug/2020 at 11:00 am]
2. https://www.tutorialspoint.com/unix/unix-using-arrays.htm
   [Accessed on 11/Aug/2020 at 1:00 pm]
3. https://linuxhint.com/read_file_line_by_line_bash/
   [Accessed on 14/Aug/2020 at 1:30 pm]