COURSEWORK 1: KH4061CEM PROGRAMMING AND ALGORITHMS 1

Begad Hatem Diab 202101453

Vigenère Cipher

Date of Submission

Table of Contents

Introduction

In this report I will introduce to the user what is Vigenère ciphering which is the type of algorithm used in the program, how the ciphering works, and why I chose this specific type.

Vigenère cipher is a method that uses a method called polyalphabetic substitution that helps encrypt the alphabet, but in this case, it'll be used to encrypt all printable characters by increasing the range of the ascii code characters. Its way of ciphering is quite interesting. It takes each letter from the key and runs it over each letter from the text given to be encrypted and encrypts it according to the range given in the encrypting function, similarly in the deciphering process with a small change, instead of entering the text and running through the procedure the user has to enter the ciphered text for the program to ,according to the range of the ascii code and the equation used, trace back to the (32) placement in ascii so it can reset and go through it furthermore decipher the ciphered text.

The reason I chose this ciphering algorithm is because as it is as entertaining as it is challenging to find a way to modernize the old ways this cipher was used and turn it into a tool that can be used in our modern society.

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

This Diagram represents how the Vigenère cipher was used in the past. It only included alphabet, unlike how it was used in the program which included all printable characters.

Figure 1

Instructions for Running the Program

Firstly, the user will be prompted to choose between encrypt, decrypt, or if he/she wants to exit the program by reading the prompted questions the user will find what to type in between single quotations, depending on the user's answers and inputs the program will execute the right code for the job. In the case the user chooses to encrypt, they will be asked if they want to encrypt their own self typed text which can be chosen by typing 'text' or choose an existing file that contains text by typing 'file', then (in the case of choosing the 'file' command) the user will be prompted to type the name of the file with respect of spelling and typing the file type following the name. After that the user will be prompted to choose a method of using the key to encrypt whether to generate a random key that will generate a key that consist of not less than 3 and not more than 9 characters so it'll be a reasonable number of characters not too large and not so little that encryption of the text would be easy to decrypt which can be accessed by typing '1', self-type in the key which can be accessed by typing '2', or to import a key from an existing file or created previously by the user which can be accessed by typing '3'. And of course, all this with respect of spelling and typing the file type following the name, furthermore whichever method the user will choose the key will be saved in a file named as the user wants and in addition to the ciphered text in a separate file. As for how the key works, a key generator function was developed to ensure that the key has the same length as the text, and if the key's length already matches that of the text's the function will just use the key without editing it. The encryption works using the encrypt function which is defined in the program. It adds the placement of each character's value in the ascii code list from the text to each character's value in the ascii code list in the key and with the help of the math equation given and range given.

(If the user misspells words and forget to type the file type the program will prompt the user to retype the phrases, he/she wrote wrong).

Next if the user chooses to decrypt the program will ask the user to input the name of the file they want to decrypt, then the program opens the file and reads its content, next the user will be prompted to type in the file that has the key that was used to encrypt the phrase in the previously chosen file, furthermore the program opens the key file and then reads its content. Finally, the program takes both files decrypts them using the decrypt function and lists the text in a new file. The decrypt function is the opposite of the encrypt function instead of adding each character from the text to each character in the key it subtracts them so it can trace back the placement in the range give in the ascii code list and with the help of the math equation given.

Unit Tests

Encrypting unit test:

Phrase	Key Used	Results
Hell0 w0rld :)	!TWp\	iZd}-Al($i&t2:
1 am hungry :(	W5,p^5!M	)U.~~>7_HF19]
G00d bye world :(	-?d-.	to52N0YjMF=Rq2Ngg
My@credit_card-numbe'is-->2378011645378003	ahgfsdfgsakyl!234@#	O#Hj'jkq)ao{ &_BJN&g0qzA2E:G9DJ>RhgisZ:98:
My C0d3 w0rk5 :))	ikZ$psC	W&zgAxv*$+7\|IcD5$D
I have 13 more years to live?!	{ftq	e'}s3l5CO'#"/l5,"h(&<{%2)p,w[(

Figure 2

Decrypting unit test:

Encryption	Key Used	Results
iZd}-Al($i&t2:	!TWp\	Hell0 w0rld :)
)U.~~>7_HF19]	W5,p^5!M	1 am hungry :(
to52N0YjMF=Rq2Ngg	-?d-.	G00d bye world :(
O#Hj'jkq)ao{ &_BJN&g0qzA2E:G9DJ>RhgisZ:98:	ahgfsdfgsakyl!234@#	My@credit_card-numbe'is-->2378011645378003
My C0d3 w0rk5 :))	ikZ$psC	W&zgAxv*$+7\|IcD5$D
e'}s3l5CO'#"/l5,"h(&<{%2)p,w[(	{ftq	I have 13 more years to live?!

Figure 3

Description of work

The equations and range used where inspired from self-study and the lab's coursework

Encryption equation:

"Ascii value for each character from text" = ("ascii value for each character from text" + "ascii value for each character from key" - 32) % 95) + 32

Decryption equation:

"Ascii value for each character from text" = ("ascii value for each character from text" - "ascii value for each character from key" - 32) % 95) + 32

Each equation looks the same, but the difference is in encrypting the values are added together, however; in decrypting they are subtracted them because to decrypt it needs to trace back to the first value given in the range.

Range given:

127 > "Ascii value for each character from text" > 31

The numbers represents where the ascii code table starts tracing from and the value it will end the tracing at, which is in this case is 32 and 127.

As for the algorithm research was put into how to construct the program and when the most suitable method to me was found, I used it and came out better than expected. For how it cam to understanding the algorithm itself, it is from studying all cipher types and methods and understanding them with the help of labs and assignments. They helped a lot in constructing everything top to bottom.

The main problem and issues faced was tracing and researching errors that didn't make any sense at first glance, but eventually got the resolution to all. And had some issues with indenting all the loops under each other because of everything getting pent up together and all the loops that were mixed up with each other. Implementing the files method was a bit challenging but, in the end, it worked without any errors.

Other than the mentioned problems and issues everything ran smoothly.

Source Code

Repository link:

Coventry-TKH/coursework-1-Elit3-Looser27: coursework-1-Elit3-Looser27 created by GitHub Classroom

Python file:

Kindly find the python folder her

→ https://drive.google.com/drive/folders/17TVfEmJDCzTSwXffi2kj-U3f6J_V1zlW?usp=sharing

References

Blog. bobbyhadz. (n.d.). Retrieved November 9, 2022, from https://bobbyhadz.com/

A computer science portal for geeks. GeeksforGeeks. (n.d.). Retrieved November 9, 2022, from https://www.geeksforgeeks.org/

Where developers learn, share, & build careers. Stack Overflow. (n.d.). Retrieved November 9, 2022, from https://stackoverflow.com/

Labs