COMP 3350 Project #5

Possible points: 100 Due: April 28, 2023 **11:59pm CST**

Requirements:

- Read the design section and write a program. Submit source file .asm to Canvas.
- Please start early. ZERO point for late submission. After the **11:59pm** on the due day, you can't submit your assignment anymore.

Deliverables:

- Save your source of assembly program as a **.**asm file.
- Name document of source code as a "**Firstname_Lastname.asm**".
- (**5 points**) At the top of your ".asm" file, give a brief description of the program, author name and last modified date in the form of comments.
- Submit your "**Firstname_Lastname.asm**" and "**Firstname_Lastname.png**" through the Canvas system.

Rebuttal period:

- You will be given a period of 2 **business** days to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

Objective:

This program has two objectives. The first objective is to create a procedure that will take plaintext and an encryption key, use the Vigenère cipher, and encrypt the plaintext. The second objective is to create a procedure that will take cypher-text, an encryption key, use the Vigenère cipher, and decrypt the cypher-text.

Assume that all characters will be uppercase letters, no spaces, symbols, etc.

All "high level" directives are not allowed on this homework. (e.g. .IF .ENDIF .REPEAT, etc)

Design:

Create a BYTE array with the label 'input'. This array may be of any length between 2 and 100. In the case of encryption 'input' will hold the plaintext and in the case of decryption 'input' will hold cypher-text.

Create a BYTE array with the label ‘key’. This array will have length between 1 and (LENGHTOF input -1). Meaning the lower bound for ‘key’ is one character and the upper bound is one less than the number of characters in ‘input’.

Create a BYTE variable named “options”. This variable will be used to determine which procedures should be executed. If ‘options’ contain the value 1 it means the program should execute encryption procedure. If ‘options’ contain the value 0 (or any other value than 1) the program should execute decryption procedure.

Create a BYTE array with the label ‘output’. This array will have a dynamic length equal to LENGTHOF input. When executing the program using encryption, the variable ‘output’ will hold the cypher-text and in the case of decryption, ‘output’ will hold plaintext.

You may create any other values you deem necessary.

You must have three procedures for this homework. The main procedure, a procedure to handle encryption, and another to handle the decryption. You may create other procedures if you wish.

Example of encryption / decryption:

For more information check the Wiki link: (https://en.wikipedia.org/wiki/ Vigenère_cipher)

The example below will use the plaintext word MEMORY and the key BAD.

First you “line up” your plaintext word with your key by writing the key directly beneath the plaintext word. Continue to repeat the key under each plaintext character:

MEMORY

BADBAD

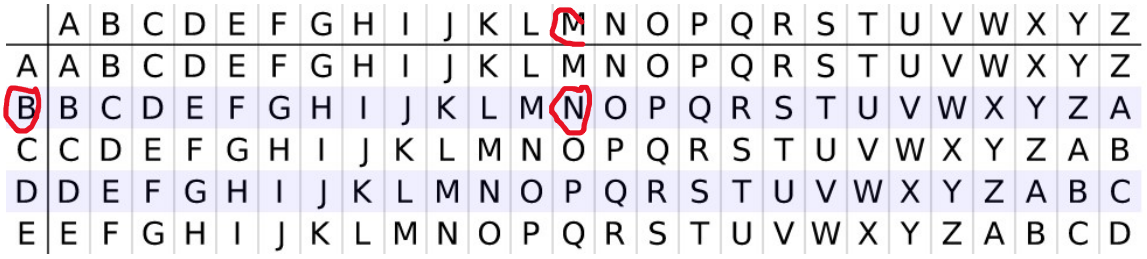
In the above example MEMORY is longer than BAD, therefore, BAD was repeated. Next choose each character in the plaintext word, starting with the first and encrypt it.

To encrypt a character, you first find the column on the chart corresponding to the character in the plaintext, e.g. the character M (circled in red). Next, find the row starting with the key character, e.g. the character B (circled in red). Finally, find where the plaintext column and the key row intersect, e.g. the character N (circled in red). Continue this for each character.

MEMORY

BADBAD

N

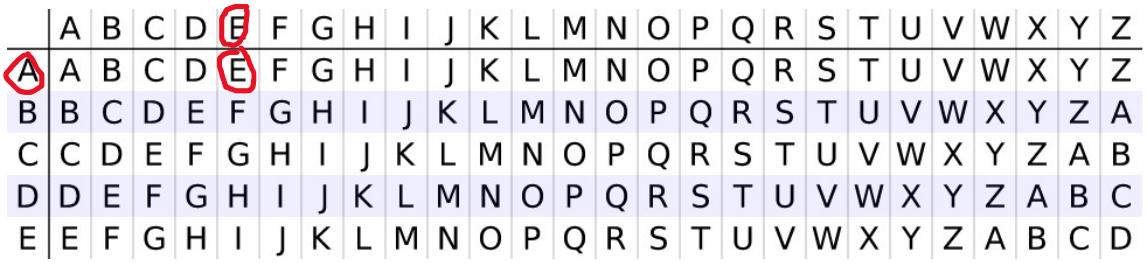


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T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

MEMORY

BADBAD

NE



	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

MEMORY

BADBAD

NEP

	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

Continue until all characters have been encrypted:

Input = MEMORY (plaintext)

Key = BADBAD (key)

output= NEPPRB (cypher-text)

Decrypting is the opposite of encrypting. To decrypt NEPPRB first write the key under the cypher-text. Then find the row starting with the key character. Next, move across that row until you find the cypher-text character. The cypher-text character's column is the plaintext column.

NEPPRB

BADBAD

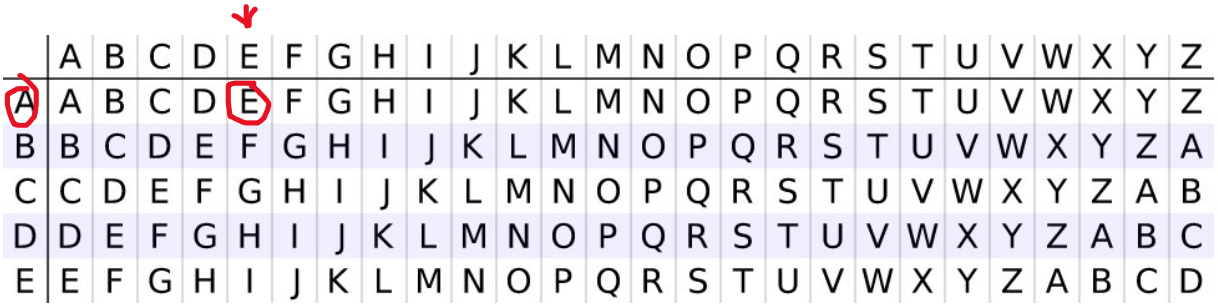
M

	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

NEPPRB

BADBAD

ME

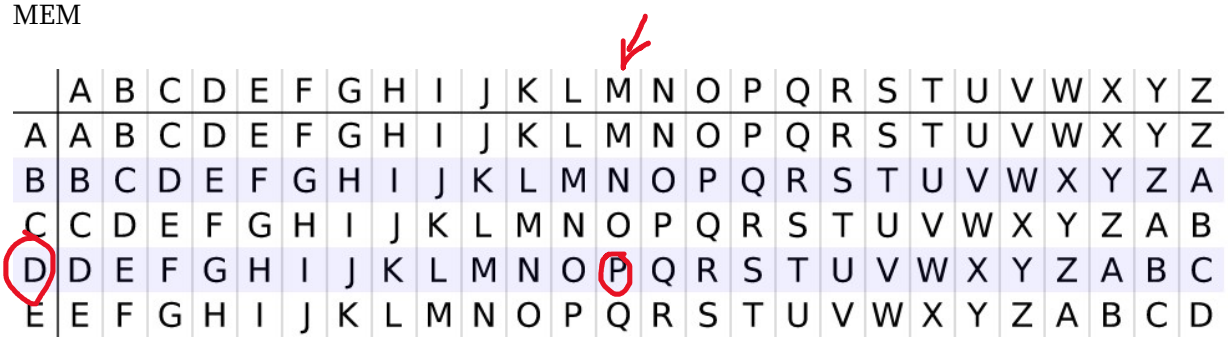


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T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

NEPPRB

BADBAD

MEM



	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

Continue until the cypher-text has been decrypted:

Input = NEPPRB

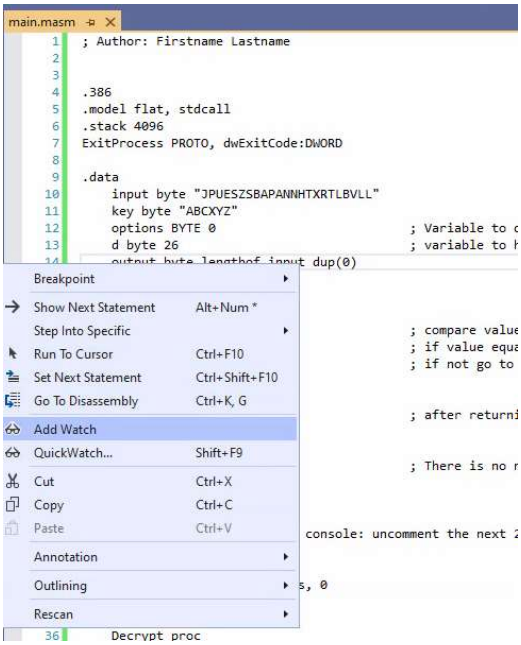
Key = BADBAD

Output = MEMORY

(**80 points**): Read the comments in the template file “project5_template.asm” and finish (1) - (8)

(**15 points**) A **screenshot** of the result after you compile and run program, correctly demonstrate the decryption result of the input "JPUESZSBAPANNHTXRTLBVLL". The screenshot file should be named as "**Firstname_Lastname.png**" or "**Firstname_Lastname.jpg**" or any other valid picture format suffix.

To take a screenshot, right-click on output, select Add Watch.



Change the Name of output into &output,30. The result should appear in the value column.