Vigenere with Cipher Block Chaining

This assignment involves implementing the Vigenere cipher as a block cipher and using the Cipher Block Chaining mode of operation, both of which we have studied in lecture and in our textbook. Specifically, your program must: (1) read in an input file, (2) strip off any characters that are not alphabetic letters, (3) change upper case letters to lower case, and (4) encrypt the resulting data using the Vigenere cipher as a block cipher in the Cipher Block Chaining mode of operation. The input plaintext file name, the Vigenere key, and the Initialization Vector will be supplied to your program as command line arguments. The input file will be in the format described below. The program output must be in the format described below. The program will be graded according to the Grading Rubric that appears at the bottom of this assignment.

Programming Language

The program must be written in C, C++, or Java, whichever you find more convenient. No other programming or scripting languages are permitted.

What You Should Submit

Your program source file should have a header identifying you as the program author. The header should use the following form:

```
//-----------------------------------------------------------------
// University of Central Florida
// CIS3360 - Fall 2016
// Program Author: [your_name]
//-----------------------------------------------------------------
```

Command Line Parameters

The program must read in three arguments or parameters. The arguments for this program are as follows:

1. The first argument will be the name of the plaintext input file. The file name will be a string. It may or may not have the ".txt" file extension. The file will have no more than 4991 characters in it. Your program must open the file, read it, and process it as described further below. Some sample file names are: "sample7.txt" and "file23" .

2. The second argument will be the Vigenere keyword. This will be a string using only alphabetic characters in lower case. The length of this string will determine the block size for encryption. For example, for the keyword "pizza", the block size will be 5 characters. The maximum length of the keyword will be 10 characters.

3. The third and final argument will be the initialization vector, which will also be a string using only alphabetic characters in lower case. The length of the initialization vector will be the same as the length of the Vigenere keyword. For the Vigenere keyword "pizza", for example, "gevrs" would be an acceptable initialization vector.

Input File Format

The plaintext file to be encrypted can be any valid text file with no more than 4991 characters in it. Thus, it is safe to store all characters in the file in a character array of size 5000, including any needed padding characters. The characters in the file will be in UTF-8 format. The file will contain the letters to be encrypted. Please note that the file will generally also have punctuation, numbers, special characters, and whitespace in it, which should be ignored.

Your program should encrypt only the alphabetic characters (i.e., letters) in the input file. The program should also convert all upper case letters to lower case. For example, the letter "M" should be converted to "m".

Since all input characters that are not letters should be ignored and all letters should be in lower case, you may wish to store only lower case letters in your input array (or whatever other data structure you may wish to use).

The following is one of the actual plaintext source files that will be used to test your program. Please note that the file contains many characters that are not letters, including white space, punctuation, and special characters. Your program must extract only the letters (alphabetic characters) and convert all upper case letters to lower case before performing the encryption processing.

CS: the science that deals with the theory and methods of processing information in digital computers, the design of computer hardware and software, and the applications of computers.

IT: the development, implementation, and maintenance of computer hardware and software systems to organize and communicate information electronically. Abbreviation: IT

Computers are man-made tools that aid us in solving other problems. A biologist is trying to figure out how life works, physicists and chemists are trying to figure out how items react in our universe, mathematicians are trying to figure out rules for man-made systems.

Any research problem that may improve a computer's capability of helping solve a problem or any research problem that sheds light about a new way to do something with a computer is part of CS.

Most exciting research: Medical Applications (Expert Systems for Diagnosis, Remote surgery, nano-devices with computing power to deliver medicine, etc.), We need help trying to create a comprehensive EMR accessible to the right people only, Cars that can drive themselves – seems like the best way we know how to solve lots of problems is by throwing lots of computing power at them, instead of looking for elegant solutions. This doesn't sound exciting, but it will be exciting when the results are achieved. (ie Watson)

CS students tend to find jobs where they program at least some.
In the process, they are solving problems.
Challenges: It's impossible to teach all the new languages/toys.
Ultimately, we just need to teach our students how to think, so that they can pick up new things on their own. Our biggest challenge is getting them to buy into that.

Ethical: Lots, with security etc.

Encryption Process

There are two parts to the encryption process for this program: (1) Cipher Block Chaining, and (2) the Vigenere cipher. Both require knowing the block size for the encryption. The block size is simply the number of characters in the Vigenere keyword. It is also the number of characters in the Initialization Vector (IV), since the IV and keyword will always have the same length.

Cipher Block Chaining ("CBC")

You will recall that CBC uses the initialization vector (IV) to represent the "previous" block of ciphertext output before the first ciphertext block is computed. To compute the first ciphertext block, we use the formula: $C_1 = E_k(P_1 \oplus IV)$, where C_1 represents the ciphertext for the first block, E_k represents encryption using the key "k", P_1 is the first plaintext block, IV is the initialization vector, and $\oplus$ is the XOR operator. For blocks 2 and all blocks thereafter, we use the formula $C_i = E_k (P_i \oplus C_{i-1})$, where C_i represents the ciphertext for the i^{th} block and C_{i-1} represents the ciphertext for the previous block.

For this program, we will implement the XOR operator by simply adding, letter-by-letter, the numerical values of the letters in the input block and the IV, taking the result modulo 26 so that each result will also be a valid letter. The numerical values are zero-based, so that the letters a, b, c, ..., z will have the values 0, 1, 2, ..., 25.

Vigenere Cipher Encryption

For this assignment, We will use the Vigenere cipher as the encryption algorithm (E_k in the equations above). We will implement the Vigenere cipher in a way that is similar to the implementation of the XOR operation described above. Specifically, we will perform the encryption by simply adding, letter-by-letter, the numerical values of the letters of the Vigenere keyword and the letters of the output of the XOR operation above, taking the result modulo 26 so that each result will also be a valid letter.

Sample Encryption

As an example, suppose we have the Vigenere keyword "secret" and the initialization vector (IV) "fspqrd". Suppose also that the lower case letters of the cleaned up plaintext file begin with "csthesciencethatdealswith..." . Now, the block size is 6 because that is the length of the keyword (and also the IV), so we must divide the plaintext into blocks of that size. The first two plaintext blocks are therefore "csthes" and "cience". Using comma-separated numbers within square brackets to show the numerical codes for strings, this is how we use Vigenere in Cipher Block Chaining (CBC) mode to encrypt those two blocks:

Block 1 (for plaintext "csthes"):

output of XOR step = "csthes" + IV

= "csthes" + "fspqrd"

= [2,18,19,7,4,18] + [5,18,15,16,17,3]

= [7,36,34,23,21,21] mod 26 = [7,10,8,23,21,21] = "hkixvv"

output of Vigenere step

= "secret" + "hkixvv"

= [18,4,2,17,4,19] + [7,10,8,23,21,21]

= [25,14,10,40,25,40] mod 26

= [25,14,10,14,25,14] = "zokozo" = C_1

For Block 2 (for plaintext "cience"), we perform the same operations, except that we use the ciphertext for Block 1 instead of the initialization vector:

output of XOR step = "cience" + C_1 = "cience" + "zokozo"

= [2,8,4,13,2,4] + [25,14,10,14,25,14]

= [27,22,14,27,27,18] mod 26 = [1,22,14,1,1,18] = "bwobbs"

output of Vigenere step = "secret" + "bwobbs"

= [18,4,2,17,4,19] + [1,22,14,1,1,18]

= [19,26,16,18,5,37] mod 26 = [19,0,16,18,5,11] = "taqsfl" = C_2

For Block 3, we repeat the process, this time using the third plaintext block and C_2, the ciphertext for Block 2. Subsequent blocks are processed similarly.

Padding the Last Block

Because the input plaintext file is divided into blocks, it is possible that the last block is not completely full. In that case, we must add additional characters so that the last plaintext block has the required number of characters. If padding is needed, your program should use the character "x", with integer value 23, for each pad character.

Program Output

All program output should be to the screen (console) and should follow the format of the following sample program output for the input file shown above. You will observe that the program output contains four parts:

1. Echo of the author's name and the command line parameters (plaintext file name, Vigenere keyword, and initialization vector);
2. Clean Plaintext echo: the alphabetic characters of the plaintext input file, all in lower case, displayed in lines of exactly 80 characters;
3. Ciphertext output: the ciphertext output for the entire clean plaintext input, including encryption of any needed pad characters at the end of the last block. This section is also displayed in lines of exactly 80 characters;
4. Statistics section, reporting the total number of characters in the clean plaintext file (before padding), the block size, and the number of pad characters added, if any.

```
CBC Vigenere by [student-name]
Plaintext file name: ramble.txt
Vigenere keyword: secret
Initialization vector: pdqist

Clean Plaintext:

csthesciencethatdealswiththetheoryandmethodsofprocessinginformationindigitalcomp
utersthedesignofcomputerhardwareandsoftwareandtheapplicationsofcomputersitthedev
elopmentimplementationandmaintenanceofcomputerhardwareandsoftwaresystemstoorgani
zeandcommunicateinformationelectronicallyabbreviationitcomputersaremanmadetoolst
hataidusinsolvingotherproblemsabiologististryingtofigureouthowlifeworksphysicist
sandchemistsaretryingtofigureouthowitemsreactinouruniversemathematiciansaretryin
gtofigureoutrulesformanmadesystemsanyresearchproblemthatmayimproveacomputerscapa
bilityofhelpingsolveaproblemoranyresearchproblemthatshedslightaboutanewwaytodoso
methingwithacomputerispartofcsmostexcitingresearchmedicalapplicationsexpertsyste
msfordiagnosisremotesurgerynanodeviceswithcomputingpowertodelivermedicineetcwene
edhelptryingtocreateacomprehensiveemraccessibletotherightpeopleonlycarsthatcandr
ivethemselvesseemslikethebestwayweknowhowtosolvelotsofproblemsisbythrowinglotsof
computingpoweratthemsteadoflookingforelegantsolutionsthisdoesntsoundexcitingbu
titwillbeexcitingwhentheresultsareachievediewatsoncsstudentstendtofindjobswheret
heyprogramatleastsomeintheprocesstheyaresolvingproblemschallengesitsimpossibleto
teachallthenewlanguagestoysultimatelywejustneedtoteachourstudentshowtothinksotha
ttheycanpickupnewthingsontheirownourbiggestchallengeisgettingthemtobuyintothatet
hicallotswithsecurityetc

Ciphertext:

jzlgaedlrkgbowtunyglnhzkfiwcwkbapraqwqvblxrmlxehxsrgaiccbkjplsduvwqesohvqbuqncuu
qavmerzvcjodhzuhivvfsteqewnpsuxrrvnqnnokfltefgqwzvyotidmvpzzxrekfpocbholymyhksey
glqlstxxqbjvfcmsemcdsjgqjikvvarocmzvjmwenbhyszwscugdpngabzfjdptulsfaeczbchkxhsti
bfjwadhvxhrebzscdkyrlfhwyjaawtsgtfnumkghpnfpzcooxmjhfpiszkyctastcmpquxsfkypckjuv
ythcucuctieuhofocmcmnyjhsecvnduwocqvcvkoqrxdmpvbsivqtzldndekbqcebszvxhlliktifxcu
pbkegsxgkabbbqgxuqnweukmzvqpqrxhphoquetdhlyzepwqwjqggcetaoutbtwewdnowilmrysfebde
qcuxpdcxacnplvnxjnrqboasjxhxcdufvggjkabpoctgkvjjmvqygjesepikilxuhharnxaulzgggnsd
jfvodfzbglgyjfqwmngauevfjccjbalgetxwrxzowpkcessffaiecqlueicaueiciypvzzwyrkwgrqlp
mdcbvtwscyekczgraukjgnrlfvmwvuvcwtbibdhfqfwarnsncavvxmityzosxusdjrpaclixxknhiguh
rdplpdrhxphordqkxvclkvsuygkzwaenqlivajokfvubcqdhuskwvwqnfbcttzcwxypgmpjelohijbqw
ncuytpikwzyzdpcjudykgeatpsoewkjrnbsacitwowwhyaktmrjxtfupgfpxlromumlkswejubprwazj
bfsxierqomfznbkuticcmcogkwiyjtonwgvxgcvythynbnqcnumjgygesottwpskwitpbcuhkoeupqcm
ymdwayydlcsnuynmpnqoxqlzmscvudskoulcpchpazngwzwgqeryomabaxkigjubhtmhjvpjgteugizr
ztdxzgmlluomwvxammfqomluylkzorscfwgxbbnkcelwndnabatoxxxubuitspsqtlzojmudcxjmqyih
ujgrbmetyhdwtqqdyflvmpnpwstkdieddhmdgwmpgivkmgooqrfvurmmezhtuschbsrdcxsxnoccfskv
nkhlvoqzcjdumzpautydxvqfefrgfrevtqnvuvzqlgfmflssljlcysrhelvfmptvrfspoaofsgajgrrk
vtrohqxxkpbetdbmpbccslztrmdizohzupljkyeohffrphkulvzekqtqwqkhbahnoersfeovmxcegssh
nfawfjfrkwptqcgclbcowkgude

Number of characters in clean plaintext file: 1384
Block size = 6
Number of pad characters added: 2
```