CIS 22B Lab 1 Simple Cryptography

200 points

Topics: New, Delete, Characters, Strings, C++ strings, C strings

-----------------------------------------------------------------------------------------------------------------------------------------

For all assignments in CIS 22B, use heading comments at the top with the following format:

/\*

Fred Flintstone

Summer 2019

Lab 1

Problem 0.0.0

Description of problem:

a few lines describing input, activity, and output of the program

\*/

----------------------------------------------------------------------------------------------------------------------

***Cryptography*** — the science of secret writing — is an old science; the first recorded use was well before

1900 B.C. An Egyptian writer used previously unknown hieroglyphs in an inscription.

We will use a simple substitution cypher called rot13 to encode and decode our secret messages.

**ROT13** ("**rotate by 13 places**", sometimes hyphenated **ROT-13**) is a simple letter substitution cipher that replaces a letter with the 13th letter after it, in the alphabet. ROT13 is a special case of the Caesar cipher which was developed in ancient Rome.

Decryption Key

**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**

(letter above equals below, and vice versa)

As you can see, A becomes N, B becomes O and so on.

Your job is to write a program, **with at least four functions**, including main, which must do the following:

1. Ask user whether they want to encode or decode a message – if no, then terminate
2. Take the input string from the user, store it in dynamic memory (use new)
3. As appropriate, encode or decode the message using Rot13.
4. Output the encoded/decoded message
5. Delete the input string from dynamic memory (use delete)

Input will be a string of no more than 25 characters. Blanks get replaced with blanks.

Do not worry about punctuation; there will be no punctuation in the string.

ALPHABET becomes NYCUNORG

Test your program with the following strings:

TAF VF

paddrpf

I’ll be using other strings to test your code.

Some suggestions (**NOT** requirements):

1. C++ string library functions would be useful here
2. You can process the C++ string one char at a time if you use the length member function

Main

-user input processing -repeat if wanted

-clean up of dynamic storage (delete)

|

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| | |

Input encrypt/decrypt output

-read string -read from dynamic storage -get results

-put into dynamic storage -encrypt/decrypt -output results

-store in dynamic storage