AP Computer Science Name: _____

Coding Assignment 3 – Caesar Cipher **Due Date: Tuesday, November 30, 2021 @ 12:01 a.m.**

You are a military spy and need to get a message across enemy lines to ensure the safety of the troops. You want to create a program that translates your message into something only someone from your side has the key to break. You decide to use a Caesar cipher. The Caesar cipher is one of the earliest known and simplest ciphers (codes). **It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet.** For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is named after Julius Caesar, who apparently used it to communicate with his generals.

Example

To pass an encrypted message from one person to another, it is first necessary that both parties have the 'key' for the cipher, so that the sender may encrypt it and the receiver may decrypt it. For the Caesar cipher, the key is the number of characters to shift the cipher alphabet.

Here is a quick example of the encryption and decryption steps involved with the Caesar cipher. The text we will encrypt is 'defend the east wall of the castle', with a shift (key) of 1.

```
plaintext:  defend the east wall of the castle
ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf
```

Obviously, if a different key is used, the cipher alphabet will be shifted a different amount.

Program

Your goal for this program is given a key and a plain text message, translate it into an encrypted message. You will then also need to be able to decrypt an encrypted message. The shift for encrypting should **always be moved forward** (+). The program should work for any key (shift). What could you do if a user entered a key over 26? What if they entered a key of a multiple of 26? You need to handle these situations. If the key is 0 or any multiple of 26, the user should be prompted for a new key, since there will be no shift.

An approach to this would be to create a String of the alphabet to associate each letter with a position in the String. Make one String for lower case letters and one String for upper case letters. Use charAt() and indexOf() to help you.

Some things you will have to watch for:

1) If the current character is a space or punctuation, skip the translation and just add the existing character to the translated string.
2) If you subtract the key from the current value and your result is past the beginning of the alphabet you need to account for "wrapping". For example the letter 'b' with a decryption key of 5 would go to w.

First ask the user if they want to "encrypt" or "decrypt" their message (use that to call the correct method below). Take in Strings shown. Your code should ask for the message and key in the main method then send them to the following method to do the encryption or decryption (use method header **EXACTLY**):

```
public static String encrypt(String message, int key)
```

To get full credit, your code should also be able to decrypt a message. Ask the user if they are encrypting or decrypting, then call either method appropriately. For decrypting, use this method header:

```
public static String decrypt(String message, int key)
```

The method headers are in the template, please use the template as given.

Grading rubric, out of 65 points:

Header comment block with your name, period, date and honor code statement: **required to be graded**

Program must encrypt/decrypt messages, be written to handle lowercase and uppercase, and must handle going past end of alphabet back to beginning (wrapping)

- Keys with values that don't shift get a prompt and re-entered: 5 points
- Key modified to value between 1 and 26 and shift matches: 5 points
- Encrypting a coded message works correctly: 15 points
- Wrap around works correctly: 5 points
- Spaces are handled correctly: 5 points
- Handles uppercase letters and punctuation/symbols: 5 points
- Decrypting message works correctly: 10 points
- Output is clear, SPELLED correctly, and spaced nicely: 3 points
- Program Description and Algorithm are detailed, and code commented on every line: 5 points
- Proper Java variable naming convention used: 2 points
- Braces are aligned and code is indented/tabbed properly with respect to curly braces for all methods, loops, etc: 3 points
- Code is organized into sections: 2 points
- **You must only use concepts we have covered in class so far this year. You may not use arrays. (10 points will be deducted)**
- **SEND (Name).JAVA file to pclaguna@fcps.edu**

`Welcome to Caesar Cipher! Do you want to encrypt or decrypt a message?`
Encrypt

Sample Run 1:

`Please enter the key to your cipher:`
`1`

`Please enter the message you want to encrypt:`
defend the east wall of the castle!

`Your encrypted message is:`
efgfoe uif fbtu xbmm pg uif dbtumf!

Sample Run 2:

`Please enter the key to your cipher:`
`8`

`Please enter the message you want to encrypt:`
the test is on Tuesday

`Your encrypted message is:`
bpm bmab qa wv bcmalig

Sample Run 3:

Please enter the key to your cipher:
25

Please enter the message you want to encrypt:
meet me at the corner of fourth and main

Your encrypted message is:
ldds ld zs sgd bnqmdq ne entqsg zmc lzhm

Sample Run 4:

Please enter the key to your cipher:
5

Please enter the message you want to encrypt:
ranger is not going to like this yogi //if doing only lowercase
RANGER IS NOT GOING TO LIKE THIS YOGI //if handling uppercase

Your encrypted message is:
WFSLJW NX STY LTNSL YT QNPJ YMNX DTLN //for uppercase

Welcome to Caesar Cipher! Do you want to encrypt or decrypt a message?
Decrypt

Sample Run 1:

Please enter the key to your cipher:
1

Please enter your coded message:
efgfoe uif fbtu xbmm pg uif dbtumf!

The original message is:
defend the east wall of the castle!

Sample Run 2:

Please enter the key to your cipher:
8

Please enter your coded message:
bpm bmab qa wv bcmalig

The original message is:
the test is on tuesday

Sample Run 3:

Please enter the key to your cipher:
25

Please enter your coded message:
ldds ld zs sgd bnqmdq ne entqsg zmc lzhm

The original message is:
meet me at the corner of fourth and main

Sample Run 4:

Please enter the key to your cipher:
5

Please enter your coded message:
WFSLJW NX STY LTNSL YT QNPJ YMNX DTLN //for uppercase

The original message is:
ranger is not going to like this yogi //if doing only lowercase

RANGER IS NOT GOING TO LIKE THIS YOGI //if handling uppercase