Deciphering Messages

Ciphering messages has been a popular way to relay secret messages since ancient civilization. One of the most famous examples of a cipher is Caesar's Cipher. Caesar’s Cipher is an example of a *monoalphabetic* cipher - where a single alphabet is used to encrypt and decrypt messages. Typically in these alphabets, the same English letters are used, but there is some type of algorithm that is used to shift the order. In the traditional case of Caesar’s Cipher, the alphabet is shifted to the “right”, or “down”, 3 spots. In this case, Code:You would end up as “Frgh:Brx”.

Key:
Normal Alphabet:

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Caesar’s Cipher Alphabet:

0	1	2	3	4	5	6	7	8	9	10	11	12
D	E	F	G	H	I	J	K	L	M	N	O	P
13	14	15	16	17	18	19	20	21	22	23	24	25
Q	R	S	T	U	V	W	X	Y	Z	A	B	C

There are several other examples of cipher usage in history. For example, in WWII, Germany used a computer called the Enigma Machine to encrypt military communications.
https://www.cia.gov/legacy/museum/artifact/enigma-machine/
This cipher proved exceedingly difficult due to the use of a polyalphabetic cipher system, which contains multiple different alphabets in a single message. Alan Turing, a British mathematician and one of the most notable figures in computer science, was able to break the messages encrypted by the Enigma Machine using his own machine called the Bombe. Using the Bombe, messages from Germany were able to be deciphered and the Americans were able to gain additional intelligence during the war.

In this exercise, you will create a .py file that utilizes an *affine* cipher, which is a type of monoalphabetic cipher. Instead of shifting the alphabet over where given P is the character in “plain-text”, or the normal alphabet, and K is the “key”, or the amount we are shifting to result in the cipher result C = (P + 3) mod 26 as in Caesar’s Cipher, we will be adding multiplication. In this affine cypher, given P, the “plain-text” character, K1, the multiplicative key, and K2, the additive key, C = (P * K1 + K2) mod 26. *Please keep in mind the order of operations.*

In your program, the user should be asked to provide values for K1, the multiplicative key, and K2, the additive key, and a message to encrypt. Using these values, your program should calculate the cipher key and use that key to encrypt the message. The user should also be given the option to decrypt an encrypted message using the following P = ((C - K2)/K1) mod 26:

$C = P * K1 + K2$

$C - K2 = P * K1$

$\frac{C - K2}{K1} = P$ where C = Cipher value, K2 = additive key, and K1 = multiplicative key

You should work with a partner on this project and decide how to split the work. Each partner should have at least 5 commits and at least 5 pull requests on separate branches on their respective repositories. This program should be tested with 3-5 different values.

Expected sample output:

```
(program start)
Welcome to the affine cypher!

Would you like to encrypt or decrypt a message?
          Encrypt

Please enter a message to encrypt:
          Hello world

Please enter a multiplicative key value (1 - 26):
          5

Please enter an additive key value (1 - 26):
          13

Encrypting… please wait.

[Your program should use the following cipher with these values.
These do NOT get printed to the user.]

Start:

0*5+13 = 13
9*5 = 45 + 13 = 58 mod 26 = 6
```

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

```
End:
```

0	1	2	3	4	5	6	7	8	9	10	11	12
N	S	X	C	H	M	R	W	B	G	L	Q	V
13	14	15	16	17	18	19	20	21	22	23	24	25
A	F	K	P	U	Z	E	J	O	T	Y	D	I

```
Encrypted message:
Whqqf tfuqc
```

```
Would you like to run again? (Y/N)
           Y

Would you like to encrypt or decrypt a message?
           Decrypt

Please enter a message to encrypt:
           Whqqf tfuqc

Please enter a multiplicative key value (1 - 26):
           5

Please enter an additive key value (1 - 26):
           13

Decrypting… please wait.

Decrypted message:
Hello world

Would you like to run again? (Y/N)
           N

Goodbye!
(program end)
```