Homework 4
Deranged Alphabet Shifted Substitution Cipher

10 Points – Due December 5th by 11:59pm

Description:

For this assignment you are going to create a program that implements a complete deranged alphabet shifted substitution cipher, henceforth known as cipher.

It turns out that a map is a great way to store the cipher alphabet. When encrypting or decrypting, there is a mapping between the original character and what it is changed to. For instance, if we were to encrypt a message without entering a keyword and using a shift of 3 the mapping would be 'a' → 'd'. This means that if there is an 'a' character in the message it changes to a 'd' character. When no keyword is entered the cipher behaves as a Caesar Cipher.

The program presents the user with a menu of choices allowing encryption, decryption, or quitting the program. If encrypt or decrypt is selected the user will be asked to enter a message, a keyword, and a shift value. After entering the information the program will display the converted message to the user. On page four there are two examples of running the program.

To help you when testing there is a `printAlphabets()` function that you can use to display what is inside of the `cipher_alphabet` map. To use it you can put the line below anywhere in your program. Do not keep the line active in what you turn in.

```
printAlphabets(cipher_alphabet);
```

There is also a lot of code provided for you in cipher.cpp and it is worth understanding what it does. Who knows what help you may find.

Areas in the code that need to be completed are indicated by comments, where the first comment is in the form `// ** 1 **`. Please complete the code under the comments.

Make sure that there are **no warnings** during compilation. If you are not actually using the provided Makefile the warnings may not display. Also, do not leave output that you used for debugging in the code submitted. Lastly, make sure the code formatting is clean and easy to read.

What You Need To Do:

In the `main()` function you will need to create the deranged alphabet that is a valid keyword with the remaining unique letters afterward. This is similar to the deranged alphabet lab, however you may want to consider easier ways to generate the alphabet using what is provided for you. For instance you do not need to use a STL list for this homework.

Depending on whether the user selects the encrypt or decrypt option, setup the corresponding `cipher_alphabet` map that will be used by the `encrypt()` and `decrypt()` functions.

There are also four functions that you need to complete before the main function will work.

1. `string shift( string text, int n )`

Shift will take the text passed in as an argument and shift it n characters in the alphabet. The string text is the cipher deranged alphabet before shifting. Shift also needs to handle wrapping characters around to the front of the alphabet if the shift is greater than a 'z'.

Example using keyword "Mississippi" and shift of 2:

```
Original Alphabet = mispabcdefghjklnoqrtuvwxyz

Shifted Alphabet =  okurcdefghijlmnpqstvwxyzab
```

Notice how the character 'z' is now an 'a' and 'y' is now a 'b'.

2. `string encrypt( map<char, char> alpha, string text )`

The arguments for `encrypt()` are an alphabet map to use during encryption and a string of text that is the message to encrypt. Before encrypting the text message uppercase letters need to be made lowercase and all non 'a' – 'z' characters need to be removed.

Example of Encryption with keyword "mississippi" and shift of 2:

Map Contents (generated in `main( )` function)

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
Value:	o	k	u	r	c	d	e	f	g	h	i	j	l	m	n	p	q	s	t	v	w	x	y	z	a	b

Text Message

"Finding Nemo is a great movie!"

Encrypted Message Returned by Function

"dgmrgmemclngtoescovlnxgc"

3. `string decrypt( map<char, char> alpha, string text )`

The arguments for `decrypt()` are an alphabet map to use during decryption and a string of text that is the message to decrypt. Before encrypting the text message uppercase letters need to be made lowercase and all non 'a' – 'z' characters need to be removed. To make the decrypted message a little easier to read, put spaces between characters in the "decoded" string.

Example of Decryption with keyword mississippi and shift of 2:

Map Contents (generated in `main()` function)

```
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
Value:  y z e f g h i j k l b m n o a p q d r s c t u v w x
```

Text Message

"dgmrgmemclngtoescovlnxgc"

Decrypted Message Returned by Function

"f i n d i n g n e m o i s a g r e a t m o v i e"

4. `string processKeyword( string key )`

This function will process the keyword passed into the function based on the following rules:

- no special chars
- no duplicate chars
- all lowercase letters
- not larger than the 'a' - 'z' alphabet (26 characters) ← taken care of for you

Process Keyword Examples:

Keyword Passed to Function

"Mississippi" "funny zerbra story"

Returned String

"misp" "funyzebrasto"

Deliverables:

Zip your program code & Makefile, rename the zip file to your first name underscore last name, and submit it on D2L in the Homework 4 dropbox.

Extra Credit (1 point):

Make the cipher work with letters and numbers (0 – 9). This means that numbers will be part of the cipher alphabet and numeric values will be retained when encrypting / decrypting messages.

Tips: Don't try to tackle everything at once. Instead, decide on small chunks to change within the program and get those working first. Experiment to see what happens.

Program Execution Examples:

```
Menu:
1 - Encrypt
2 - Decrypt
3 - Quit
Selection: 1

Enter a text message:
The quick brown fox jumped over the moon, hitting a star, that fell down to the
ground.

Enter a keyword: sunny side up

Enter a shift value (0 - 25): 5

Message Entered:
The quick brown fox jumped over the moon, hitting a star, that fell down to the
ground.

Encoded Message:
vunqwfshzroamiobgwlpndoynrvunloomufvvfmjxtvxrvuxvinkkdoamvovunjrowmd
```

Figure 1: Example of Encryption

```
Menu:
1 - Encrypt
2 - Decrypt
3 - Quit
Selection: 2

Enter a text message:
vunqwfshzroamiobgwlpndoynrvunloomufvvfmjxtvxrvuxvinkkdoamvovunjrowmd

Enter a keyword: sunny side up

Enter a shift value (0 - 25): 5

Message Entered:
vunqwfshzroamiobgwlpndoynrvunloomufvvfmjxtvxrvuxvinkkdoamvovunjrowmd

Decoded Message:
t h e q u i c k b r o w n f o x j u m p e d o v e r t h e m o o n h i t t i n g
a s t a r t h a t f e l l d o w n t o t h e g r o u n d
```

Figure 2: Example of Decryption