CIS-7 Project Documentation

A. Team name, members:

Team name: Y&C

Members: Juan Yin, Jou-Chih Chang

B. Project Information and details:

- What problems are you solving in this project?

We are solving the Vigenere Cipher Decryption problem, the code was created to give user a new string utilizing the user's inputs.

- What solutions are you implementing in the project?

We used **do-while loop** to ask users to input the string they want to shift.

We used getline function to get the user's inputs and store them into **inputt** and **keyy** respectively.

We used **strcpy function** to copy the **keyy** value to a new array, which is named **key.**

We called the **c_str function** of the char **keyy** inside the **strcpy function**.

We used **for loop** to read the user's input one character by one character.

We used **toupper function** to make the user's input becomes uppercase.

We called the **length function** to get the length of the string.

We used two **if statements** to let users choose to encrypt or decrypt.

We used a math calculation to shift the char utilizing the user's input.

We output the original and new strings.

We create a file and store all of the data into the file each time we run it.

- Provide explanation of calculations and algorithm implementation.

Calculation:

```
((input[i] + newKey[i]) % 26) + 'A')
```

We utilized an ASCII table in this calculation. We add key value to input value, and this sum mod 26, so we could get the remainder. Then add the remainder value to character ‘A’, so we get our output char.

```
(((input[i] + newKey[i]) + 26) % 26) + 'A')
```

We utilized an ASCII table in this calculation. We add key value to input value, and this time we add an extra 26 to the sum since we need to decrypt instead of encrypting, and this sum mod 26, so we could get the remainder. Then add the remainder value to character ‘A’, so we get our output char.

Algorithm:

```
for(i = 0; i < inputt.length(); i++)
```

For loop was used to read each char and let them become uppercase.

```
if (choice == 1)
if (choice == 2)
```

These two if loop let the user choose decryption or encryption.

```
repo.open("repo.txt", fstream::app);
```

This code is going to open the file witch named “repo”.

```
repo<<
```

This line is going to store the text into the file.

```
repo.close();
```

This line is going to close the file.

- What are the program objectives? Explain how your program is interacting with the user and its purpose.

This program is going to translate some strings by the user’s key.

We will ask the user to input the original string, and key string. After that, the output will give the user the new string. What’s more, the whole process will be recorded in the “repo.txt” file, so we could see the user’s input, key-value, new string in that file.

- How is discrete structures implemented in the C++ program?

There are two math formulas that be used to get the new char:

((input[i] + newKey[i]) % 26) + 'A')

(((input[i] + newKey[i]) + 26) % 26) + 'A')

- What are the limitations of the program?

I think the limitation of our program is that users are unable to change their original string before they get a new string.

- Provide recommendations on improving the limitations of the program.

We could add a extra code to ask users “if you would like to change your original string/key value?” before they get the new string.

C. Flowchart OR Pseudocode. (30 points)

1. Declare variables – (int) “i”, “j”, “choice”, (string) “input”, “keyy”, and (ofstream)“repo”.

2. Ask the user to enter the original string, get the original string from the user, and store it in the string variable – “inputt”.

3. Declare the char variable – “input”, and use the function – “strcpy” to convert the string variable – “inputt” to the char variable, then store it in the char variable – “input”.

4. Use a for loop to change all of the elements in the char variable – “input” to uppercase(function – “toupper”). First, assign 0 to the int variable – “i”. When the int variable – “i” is less than the inputt length, enter the for loop. In the for loop, the “i” element in “input” will equal the uppercase “i” element in “input”.

5. Ask the user to enter the original keyword, get the original keyword from the user, and store it in the string variable – “keyy”.

6. Declare the char variable – “key”, and use the function – “strcpy” to convert the string variable – “keyy” to the char variable, then store it in the char variable – “key”.

7. Use a for loop to change all of the elements in the char variable – “key” to uppercase(function – “toupper”). First, assign 0 to the int variable – “i”. When the int variable – “i” is less than the keyy length, enter the for loop. In the for loop, the “i” element in “key” will equal to the uppercase “i” element in “key”.

8. Declare the char variables – “newKey” and “output”. Then, use a for loop to get the new generated key. First, assign 0 to the int variables – “i” and “j”. When the int variable – “i” is less than the inputt length, enter the for loop. In the for loop, if the int variable – “j” is equal to the keyy length, “j” becomes 0. Else, the “i” element in “newKey” will equal the “j” element in “key”. If it does not enter the for loop, the “i” element in “newKey” will become null.

9. Ask the user for encrypt or decrypt, enter 1 for encrypt, 2 for decrypt. Get the choice from the user, and store it in the int variable – “choice”.

10. If the user enters 1, do encrypt. Use a for loop to get the encrypted string. First, assign 0 to the int variables – “i”. When the int variable – “i” is less than the inputt length, enter the for loop. In the for loop, the “i” element in “output” will equal to the “i” element in “input” plus the “i” element in “newKey”, divide 26, then plus ‘A’. If it does not enter the for loop, the “i” element in “output” will become null. Next, show the output.

11. Use .open() function to open the file – “repo.txt”. Write all of the output into the file. Then, close the file.

12. If the user enters 2, do decrypt. Use a for loop to get the decrypted string. First, assign 0 to the int variables – “i”. When the int variable – “i” is less than the inputt length, enter the for loop. In the for loop, the “i” element in “output” will equal to the “i” element in “input”

plus the “i” element in “newKey” plus 26, divide 26, then plus ‘A’. If it does not enter the for loop, the “i” element in “output” will become null. Next, show the output.

13. Use .open() function to open the file – “repo.txt”. Write all of the output into the file. Then, close the file.

14. Exit the program.