Tobie Collins

Assignment 3

3/8/21

PseudoCode:

stringInBounds() Method:

* For each character in the string input(or length of the string){
* Check if the character at the index is within the bounds.
* Return true if it is within.
* Return false if not.}

encryptCaesar() Method:

* Create an int placeholder for characters out of bounds.
* Create an empty string to hold the return value.
* For the length of the string{
* Set the placeholder equal to the ascii value of the character at an index + the key value.
* While loop to check if the placeholder is within the range.
* Subtract by the range until the placeholder is within the range.
* Add the ascii values, converted to characters, to the empty string to form the encryption.}
* Return encrypted string.

encryptBellaso() Method:

* Create an int placeholder for the offset value and empty string for the return value.
* For the length of the input string{
* While loop to extend the key string to the length of the input string.
* Set the offset value to the sum of the two string’s ascii values at an index.
* While loop to correct the offset value to within the range.
* Add the ascii values, converted to characters, to the empty string to form the encryption.}
* Return the encrypted string.

decryptCaesar() Method:

* Create an int placeholder and empty string.
* For the length of the encrypted string{
* Set the offset value to the ascii value – the key value.
* While loop to adjust the value to within the range.
* Add the ascii values, converted to characters, to the empty string to form the decryption.}
* Return decrypted string.

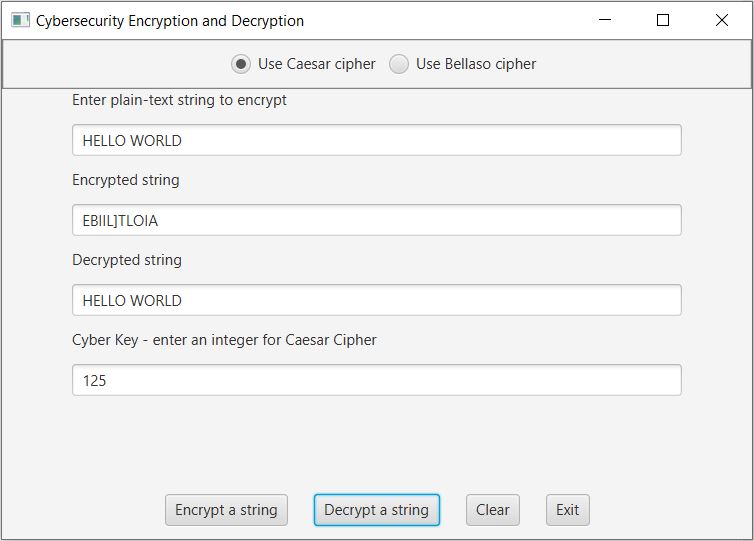
decryptBellaso() Method:

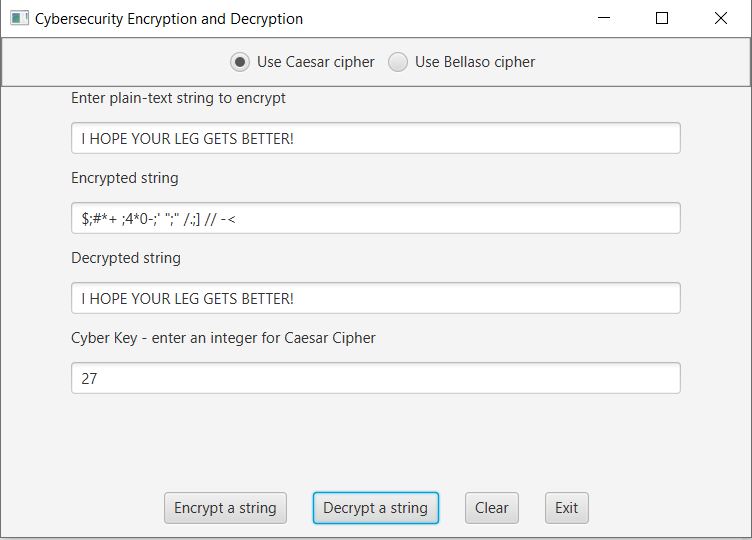
* Create an int placeholder and an empty string.
* For the length of the encrypted string{
* While loop to extend the key string to the length of the input string.
* Set the offset value to the ascii value of the input string – the ascii value of the key string.
* While loop to adjust the value to within the range.
* Add the ascii values, converted to characters, to the empty string to form the decryption.}
* Return decrypted string.

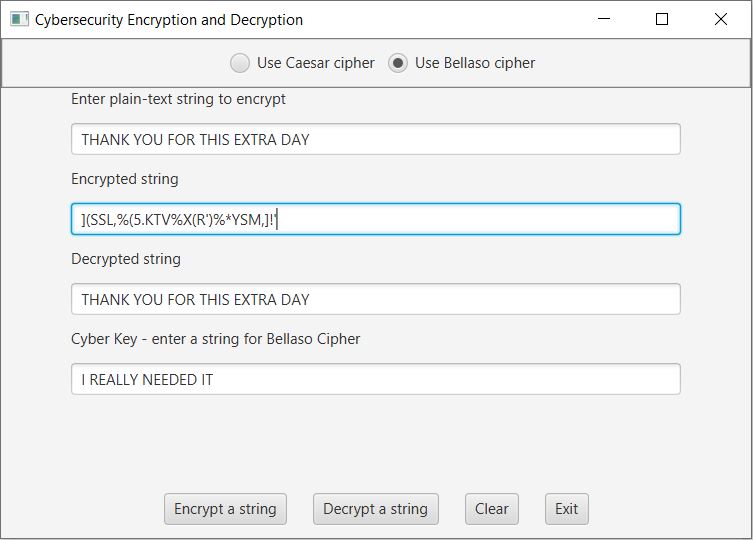
**Test Table:**

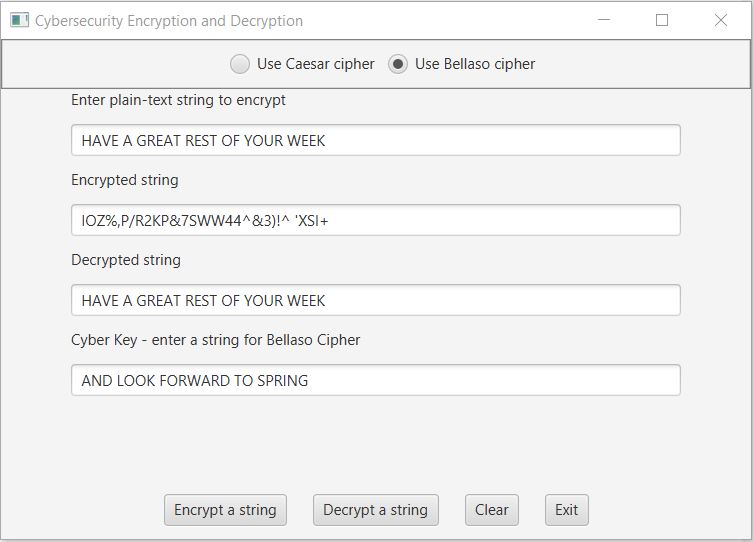
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Actual Output** | **Did it pass? (y/n)** |
| **1** | Caesar:  Hello World  125 | Encrypted: EBIIL]TLOIA  Decrypted:  Hello World | Encrypted: EBIIL]TLOIA  Decrypted:  Hello World | **Y** |
| **2** | Caesar:  I hope your leg gets better!  27 | Encrypted:  $;#\*+ ;4\*0-;’ “;” /.;] // -<  Decrypted:  I hope your leg gets better! | Encrypted:  $;#\*+ ;4\*0-;’ “;” /.;] // -<  Decrypted:  I hope your leg gets better! | **Y** |
| **3** | Bellaso:  Thank you for this extra day  Key:  I really needed it | Encrypted:  ](SSL,%(5.KTV%X  (R')%\*YSM,]!'  Decrypted:  Thank you for this extra day | Encrypted:  ](SSL,%(5.KTV%X  (R')%\*YSM,]!'  Decrypted:  Thank you for this extra day | **Y** |
| **4** | Bellaso:  Have a great rest of your week  Key:  And look forward to spring | Encrypted:  IOZ%,P/R2KP&7S  WW44^&3)!^ 'XSI+  Decrypted:  Have a great rest of your week | Encrypted:  IOZ%,P/R2KP&7S  WW44^&3)!^ 'XSI+  Decrypted:  Have a great rest of your week | **Y** |
| **5** | Caesar:  {}{}{}{}  50 | Fail | Encrypted:  -/-/-/-/  Decrypted:  ;=;=;=;= | **N** |

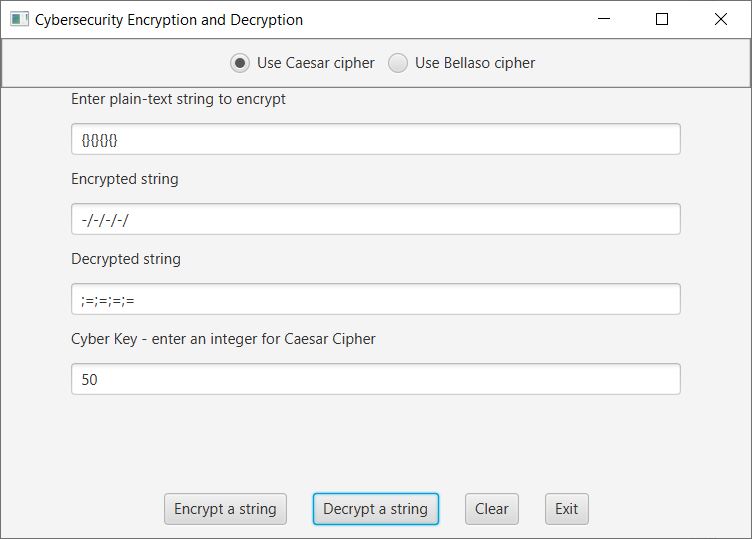
**Screenshots:**

****

****

****

****

****

**Lessons Learned:**

This assignment was particularly difficult for me. I understood the ascii values concept and how the encryption process works, however the implementation was tricky. At first as you may have seen by my first submission I tried to create the encrypted string using substrings of the rest of the string after changing one character at an index in that string. However this did not work. I knew that a character array would work however I did not want to go that route for some reason. I realized that I needed neither but simple for loops and while loops and a single type cast.

The biggest problem I faced was figuring out how much to adjust the ascii values back using their key to secure a proper decryption. Eventually I realized it was quite simple just the opposite of the encryption however it took me a while to get there. Also, I took a long time to realize that having return true in the for loop causes the method to return true without finishing all the characters. My solution to this problem was the continue; statement. I learned a lot about interacting with mixes of strings, characters and integers.