The encryption in this program is based on the ascii values assigned to the text. The program iterates backwards through the string and for each letter subtracts the previous (in the string) ascii value from the current and then adds the ascii value of the key to that. The key is not always as long as the text, so it takes the value of the key at position of i%key.length() to ensure there isn’t a bounds error. The program then adds 31 to the final value. It does not encrypt the first value of the string. The encrypted values are stored as integers in a list separated by spaces. Decrypting the text is done iterating forwards through the string. The first value was left alone so using that and the key the program can obtain the value of the second value by reversing what was previously done. The program then continues to iterate through the string reversing the previous operations until the original text is restored.

The encryption and decryption functions are all located in the “Encrypter” class. These include:

* The encrypt functions
  + Int\* encrypt(std::string)
    - takes an input but no key and generates a key to then return the array from encrypt(std::string, std::string) using the input and new key
  + Int\* encrypt(std::string, std::string)
    - takes an input and a key and outputs an int array which is the encrypted text
* The decrypt function
  + Std::string decrypt(std::string, std::string)
    - takes the file names for the encrypted text and the key and then decrypts the text
* And the genKey function
  + Std::string genKey()
    - Generates a random 25 character key, outputs it to the “key.txt” file, and returns it

The user navigation functions are included in the “Menu” class. These include

* Prompt()
  + Primary prompt that asks user what they want to do and then calls the corresponding function
* fileEncrypt()
  + prompts user for the name of their input file then reads it into a string. Then asks if they have a key they want to use. If not then it calls the Encrypt::encrypt(std::string) function, otherwise it reads the key into a string and calls Encrypter::encrypt(std::string, std::string). It then outputs the result to the file “encrypted.txt”
* fileDecrypt()
  + prompts user for input file name, then prompts user for key file name. calls Encrypter::decrypt(std::string, std::string) with those values. It then outputs the decrypted text to the file “decrypted.txt”
* textEncrypt()
  + similar to fileEncrypt() but prompts user to enter text to encrypt rather than provide a file name to encrypt