The bruteForce() method accepts the dictionary as an input, which is created in the main thread by calling a makeDictionary() function. There is a dictionary hard coded but if a file with the name of “dictionary.txt” is found in the program directory it uses that instead. The bruteForce method asks the user for a filename, first it attempts to decrypt with the value 0 to ensure that the file is encrypted, then it attempts decryption with 50 characters from the file, checking all keys and combinations possible from the space character to ~. After it attempts to decrypt the 50 characters it sends its output, and the dictionary to the DecryptCheck method. The DecryptCheck method checks if the input is 40% letters or numbers and if it finds it is not returns a false value. If it finds that 40% is then it attempts to find words, after finding that 80% of the strings that it has are words it will return that the encryption is correct. Otherwise it returns false. If 30% of the characters found during the first run are not numbers or letters, then it returns false. For a larger dictionary each loop to check for words is saved so that it doesn’t continuously find the same word. All outputs from the program are hard coded.

If you start with a dictionary.txt in the directory then enter 3, then enter the filename it will attempt to brute force if a key is found then it outputs the key, otherwise it communicates that no key was found and no file was written. If the program finds that the file is not encrypted then it tells the user that the file wasn’t encrypted from the beginning, and does not write an output file. If the file it is attempting to create already exists then it prompts the user before overwriting.