

```
# File: Project3.py
# Student: Ramsay Ward
# UT EID: rjw2777
# Course Name: CS303E
#
# Date: 11/23/2022
# Description of Program: This program will encrypt and decrypt files with a user
inputted or random encryption key.
```

```
import random
import os
```

```
LETTERS = "abcdefghijklmnopqrstuvwxyz"
```

```
def isLegalKey( key ):
    key = key.lower()
    return ( len(key) == 26 and all( [ ch in key for ch in LETTERS ] ) )
```

```
def makeRandomKey():
    lst = list( LETTERS )
    random.shuffle( lst )
    return ''.join( lst )
```

```
def makeConversionDictionary( key1, key2 ):
    dict1 = dict(zip(key1, key2))
    return dict1
```

```
def convertCharacter( ch, d ):
    return d[ch]
```

```
def convertText( text, d ):
    string = ""
```

```
    for e in text:
        flag = False
        if e.isupper():
            flag = True
            e = e.lower()
```

```

    if e in d:
        z = (convertCharacter(e,d))
        if flag == True:
            z = z.upper()
        string += z
    else:
        string += (e)
return string

```

These nicely encapsulate the actions of encrypting and decrypting strings,
and separate that from the details of file manipulation. """

```

class SubstitutionCipher:
    def __init__( self, key = makeRandomKey() ):
        self.__key = key
        self.__d = makeConversionDictionary(LETTERS,key)

    def getKey( self ):
        return self.__key

    def setKey( self, newKey ):
        if newKey == "random" or newKey == "Random":

            self.__key = makeRandomKey(LETTERS)
            self.__d = makeConversionDictionary(LETTERS,self.__key)
        else:
            while True:

                if isLegalKey(newKey):

                    self.__d = makeConversionDictionary(LETTERS,newKey)
                    self.__key = newKey
                    return
                else:
                    print(input("Enter a valid key. "))

    def encryptFile( self, inFile, outFile ):
        inFile = open(inFile,"r")
        outFile = open(outFile,"w")
        line = inFile.readline()
        while line:
            outFile.write(convertText(line,self.__d))
            line = inFile.readline()

        outFile.close()
        return outFile

```

```

def decryptFile( self, inFile, outFile ):
    inFile = open(inFile,"r")
    outFile = open(outFile,"w")
    reverseD = makeConversionDictionary(self.__key, LETTERS)
    line = inFile.readline()
    while line:
        outFile.write(convertText(line,reverseD))
        line = inFile.readline()

    outFile.close()
    return outFile

def main():
    SC = SubstitutionCipher()
    while True:
        command = (input("Enter a command (getKey, changeKey, encryptFile,
decryptFile, quit): "))

        command = command.lower()

        if command == "getKey":
            print(("    Current cipher key: ") + str(SC.getKey()))
            print()

        elif command == "changekey":
            while True:
                command2 = (input("    Enter a valid cipher key, 'random' for a
random key, or 'quit' to quit: "))
                if command2 == "random":

                    SC = SubstitutionCipher(makeRandomKey())
                    print(("    New cipher key: ") + str(SC.getKey()))
                    print()
                    break

                elif isLegalKey(command2):
                    SC.setKey(command2)
                    print("    New cipher key: " + str(SC.getKey()))
                    print()
                    break

                elif command2 == "quit":
                    print()

                    break

                elif isLegalKey(command2) is False:
                    print("    Illegal key entered. Try again!")

```

```

elif command == "encryptfile":
    ext = "-Enc"
    while True:
        file = input(" Enter a filename: ")
        if not os.path.isfile(file):
            print("File does not exist")
            print()
            break
        if file.endswith(".txt"):
            outfile1 = file[:-4] + ext + ".txt"
        else:
            outfile1 = file + ext
        if os.path.isfile(file):
            SC.encryptFile(file,outfile1)
            print("The encrypted output filename is " + outfile1)
            print()
            break
elif command == "decryptfile":
    ext2 = "-Dec"
    while True:
        file = input(" Enter a filename: ")
        if not os.path.isfile(file):
            print("File does not exist")
            print()
            break
        if file.endswith(".txt"):
            outfile2 = file[:-4] + ext2 + ".txt"
        else:
            outfile2 = file + ext2
        if os.path.isfile(file):
            SC.decryptFile(file,file[:-4]+"-Dec"+file[-4:])
            print("The decrypted output filename is " + outfile2)
            print()
            break
elif command == "quit":
    print("Thanks for visiting!")
    break
else:
    print(" Command not recognized. Try again!")
    print()

```

main()