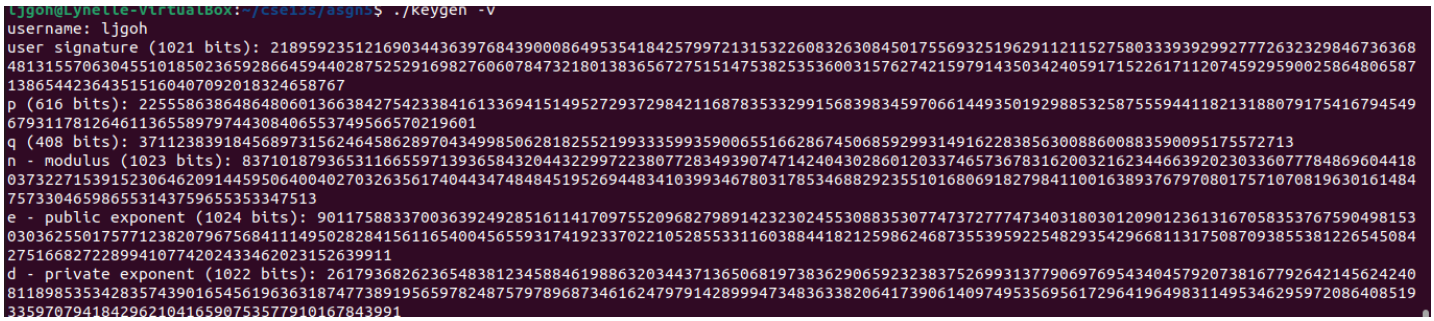
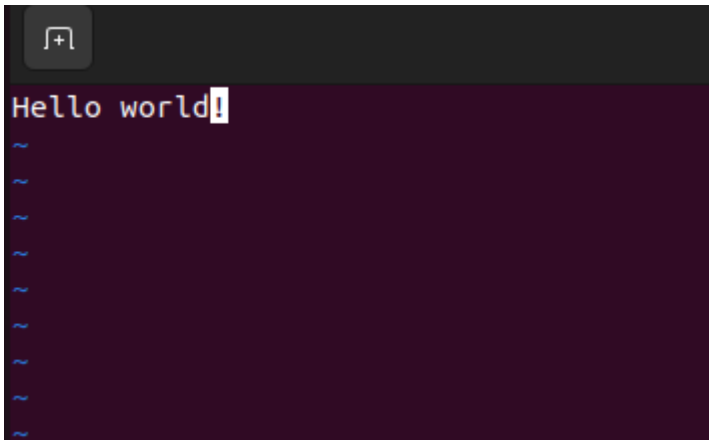
Lynelle Goh
ljgoh
Assignment 5



Using the test programs given to us, I was able to determine that my code does in fact work. The above is the output for my keygen using default values. Seeing as that the amount of bits for p and q add up to the total amount of bits, I can tell that that part of the code works. All of my bits are also equal to or less than 1024 bits. It also correctly follows the format of the test program.

Now to test my encryption and decryption programs.



This text file is called text.txt. We will use it to encrypt.

```
ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ ./encrypt -i text.txt -o text1.txt -v
username: ljgoh
user signature (1021 bits): 2189592351216903443639768439000864953541842579972131532260832630845017556932519629112115275803339392992777263232984673636848131557063045510185023659286645944028752529169827606078473218013836567275151475382535360031576274215979143503424059171522617112074592959002586480658713865442364351516040709201832465876 7
n - modulus (1023 bits): 837101879365311665597139365843204432299722380772834939074714240430286012033746573678316200321623446639202303360777848696044180373227153915230646209144595064004027032635617404434748484519526944834103993467803178534688292355101680691827984110016389376797080175710708196301614847573304659865531437596553535347513
e - public exponent (1024 bits): 901175883370036392492851611417097552096827989142323024553088353077473727774734031803012090123613167058353767590498153030362550175771238207967568411149502828415611654004565593174192337022105285533116038844182125986246873553959225482935429668113175087093855381226545084275166827228994107742024334620231526399 11
```

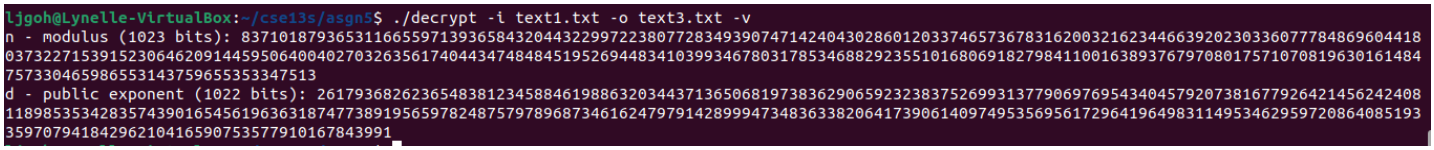
This is the output after running my code. We will output the encryption to a txt file called text1.txt.

```
ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ ./encrypt-dist -i text.txt -o text2.txt -v
username: ljgoh
user signature (1021 bits): 2189592351216903443639768439000864953541842579972131532260832630845017556932519629112115275803339392992777263232984673636848131557063045510185023659286645944028752529169827606078473218013836567275151475382535360031576274215979143503424059171522617112074592959002586480658713865442364351516040709201832465876 7
n - modulus (1023 bits): 837101879365311665597139365843204432299722380772834939074714240430286012033746573678316200321623446639202303360777848696044180373227153915230646209144595064004027032635617404434748484519526944834103993467803178534688292355101680691827984110016389376797080175710708196301614847573304659865531437596553535347513
e - public exponent (1024 bits): 901175883370036392492851611417097552096827989142323024553088353077473727774734031803012090123613167058353767590498153030362550175771238207967568411149502828415611654004565593174192337022105285533116038844182125986246873553959225482935429668113175087093855381226545084275166827228994107742024334620231526399 11
```

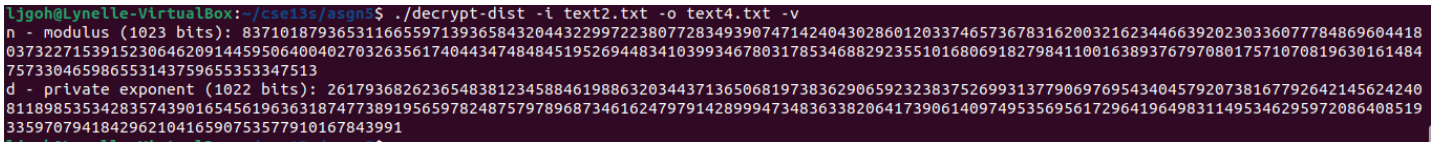
This is the output after running the test program for encryption. We output the encryption to a txt file called text2.txt.

```
ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ cat text1.txt
559ad96d02de67cde9508cc774e590d53a17f43215cac5b6c3a042771e07028fa69e6ac2a26ec19d10e8a764393f4b981821c2e06ceeeec93be5dead9d482a07cf9ded8e2ee6a0cddd771cd240eb6bdcad4e5ee9b84721baef98235784cf587a80e43ffa1f42e12311431b045e84c833f8c6985db50fce890164136f6075034a
ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ cat text2.txt
559ad96d02de67cde9508cc774e590d53a17f43215cac5b6c3a042771e07028fa69e6ac2a26ec19d10e8a764393f4b981821c2e06ceeeec93be5dead9d482a07cf9ded8e2ee6a0cddd771cd240eb6bdcad4e5ee9b84721baef98235784cf587a80e43ffa1f42e12311431b045e84c833f8c6985db50fce890164136f6075034a
ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$
```

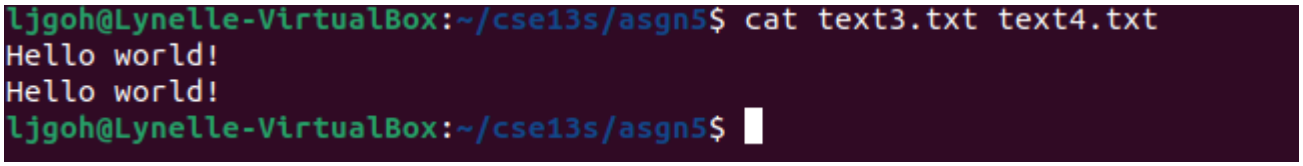
If we concatenate our two text files, we see that they have the same output.



I am running my decrypt file here and outputting the result to a text file called text3.txt. Its input is the file I encrypted using my own program, called text1.txt.

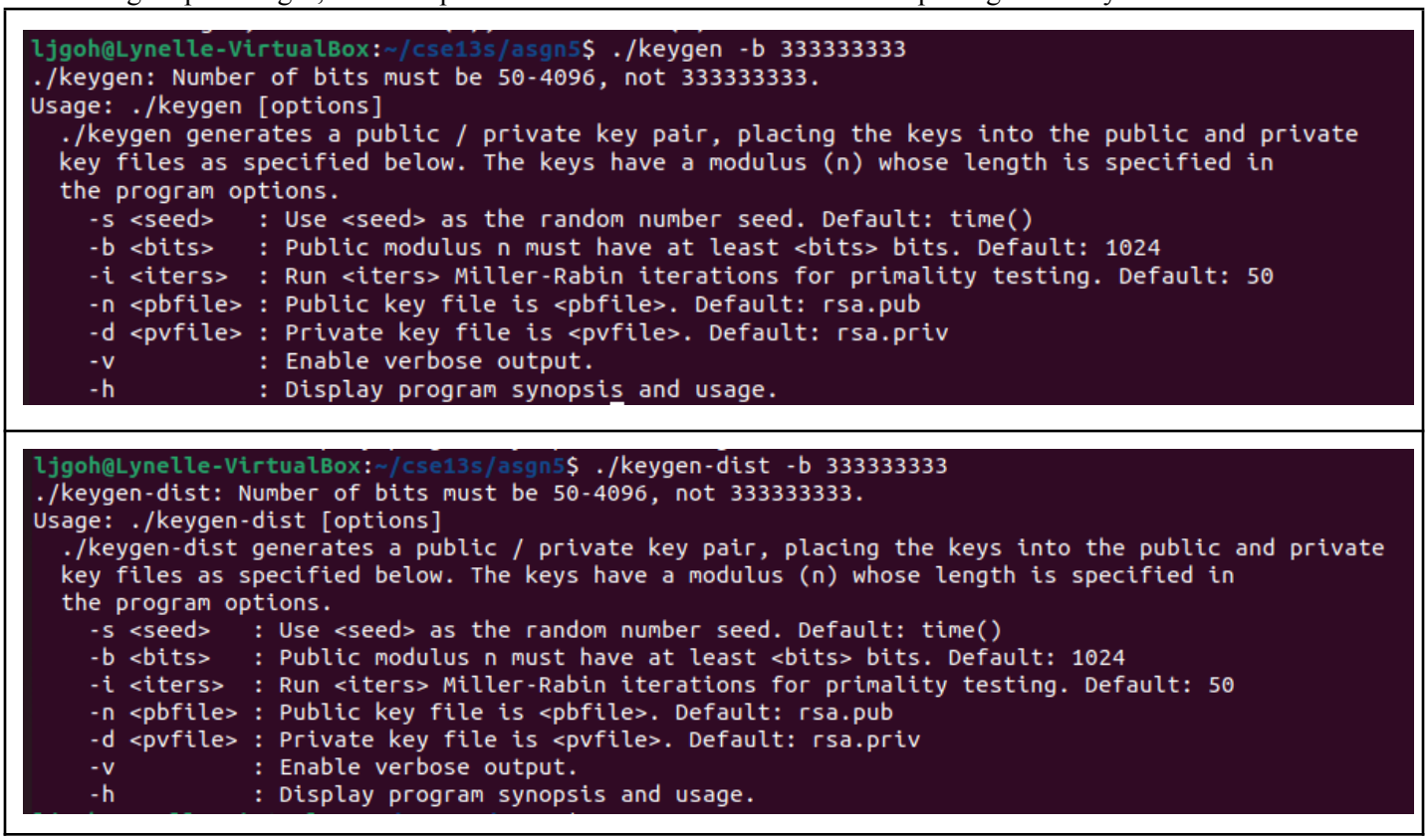


I am running the test decrypt program here and outputting the result to a text file called text4.txt. Its input is the file I encrypted using my own program, called text2.txt.



And here we can see that it correctly concatenates the message we inputted.

Concerning help messages, I also outputted the correct format. Below is for inputting too many number of bits.



Resources

https://www.w3resource.com/c-programming/c-printf-statement.php
I used this resource to help me remember printing formats.

https://gmplib.org/manual/
Though it was given to us in the assignment document, I shall still cite it here.

https://www.tutorialspoint.com/c_standard_library/c_function_fread.htm
I used this link to read the file.

https://www.tutorialspoint.com/c_standard_library/c_function_fseek.htm
I used this link to understand how seeking in files work.

https://www.tutorialspoint.com/c_standard_library/c_function_ftell.htm
I used the above link to help me figure out the size of the files, specifically the bytes.

https://www.tutorialspoint.com/c_standard_library/c_function_rewind.htm
I used the above link to figure out how to reset back to the beginning of the file after going through it.

https://www.gnu.org/software/libc/manual/html_node/Permission-Bits.html
I used this link to figure out how to set file permissions.

https://www.tutorialspoint.com/c_standard_library/c_function_feof.htm
I used this link to learn about end-of-file indicator for the given stream..