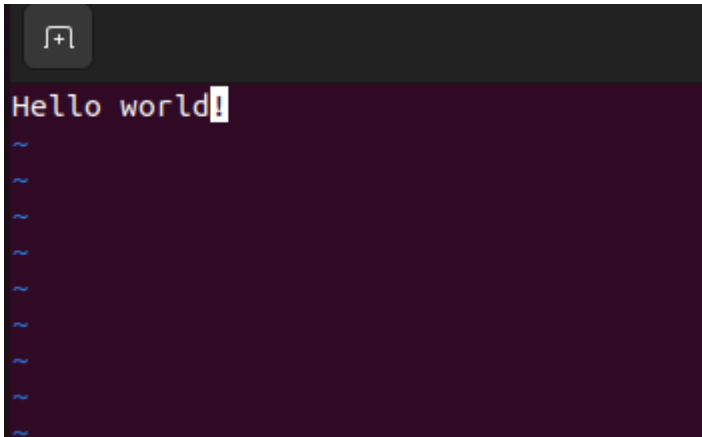


```
ljgoh@Lynelle-VirtualBox:~/cse138/assign5$ ./keygen -v
username: ljgoh
user signature (1021 bits): 21895923512169034436397684390008649535418425799721315322608326308450175569325196291121152758033393929927772632329846736368
481315570630455101850236592866459440287525291698276060784732180138365672751514753825353600315762742159791435034240591715226171120745929590025864806587
138654423643515160407092018324658767
p (616 bits): 2255586386486480601366384275423384161336941514952729372984211687835332991568398345970661449350192988532587555944118213188079175416794549
67931178126461136558979744308406553749566570219601
q (408 bits): 371123839184568973156246458628970434998506281825521993335993590065516628674506859299314916228385630088600883590095175572713
n - modulus (1023 bits): 83710187936531166559713936584320443229972238077283493907471424043028601203374657367831620032162344663920230336077784869604418
037322715391523064620914459506400402703263561740443474848451952694483410399346780317853468829235510168069182798411001638937679708017571070819630161484
757330465986553143759655353347513
e - public exponent (1024 bits): 901175883370036392492851611417097552096827989142323024553088353077473727774734031803012090123613167058353767590498153
030362550175771238207967568411149502828415611654004565593174192337022105285533116038844182125986246873553959225482935429668113175087093855381226545084
27516682722899410774202433462023152639911
d - private exponent (1022 bits): 26179368262365483812345884619886320344371365068197383629065923238375269931377906976954340457920738167792642145624240
811898535342835743901654561963631874773891956597824875797896873461624797914289994734836338206417390614097495356956172964196498311495346295972086408519
335970794184296210416590753577910167843991
```

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/asn5$ ./encrypt -i text.txt -o text1.txt -v
username: ljgoh
user signature (1021 bits): 21895923512169034436397684390008649535418425799721315322608326308450175569325196291121152758033393929927772632329846736368
481315570630455101850236592866459440287525291698276060784732180138365672751514753825353600315762742159791435034240591715226171120745929590025864806587
138654423643515160407092018324658767
n - modulus (1023 bits): 83710187936531166559713936584320443229972238077283493907471424043028601203374657367831620032162344663920230336077784869604418
037322715391523064620914459506400402703263561740443474848451952694483410399346780317853468829235510168069182798411001638937679708017571070819630161484
75733046598655314375965533347513
e - public exponent (1024 bits): 901175883370036392492851611417097552096827989142323024553088353077473727774734031803012090123613167058353767590498153
030362550175771238207967568411149502828415611654004565593174192337022105285533116038844182125986246873553959225482935429668113175087093855381226545084
27516682722899410774202433462023152639911
```

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

```
ljgoh@Lynelle-VirtualBox:~/cse13s/asn5$ ./encrypt-dist -i text.txt -o text2.txt -v
username: ljgoh
user signature (1021 bits): 21895923512169034436397684390008649535418425799721315322608326308450175569325196291121152758033393929927772632329846736368
481315570630455101850236592866459440287525291698276060784732180138365672751514753825353600315762742159791435034240591715226171120745929590025864806587
138654423643515160407092018324658767
n - modulus (1023 bits): 83710187936531166559713936584320443229972238077283493907471424043028601203374657367831620032162344663920230336077784869604418
037322715391523064620914459506400402703263561740443474848451952694483410399346780317853468829235510168069182798411001638937679708017571070819630161484
75733046598655314375965533347513
e - public exponent (1024 bits): 901175883370036392492851611417097552096827989142323024553088353077473727774734031803012090123613167058353767590498153
030362550175771238207967568411149502828415611654004565593174192337022105285533116038844182125986246873553959225482935429668113175087093855381226545084
27516682722899410774202433462023152639911
```

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/asn5$ cat text1.txt
559ad96d02de67cde9508cc774e590d53a17f43215cac5b6c3a042771e07028fa69e6ac2a26ec19d10e8a764393f4b981821c2e06ceeeec93be5dead9d482a07cf9ded8e2ee6a0cddd771c
d240eb6bdcad4e5ee9b84721baef98235784cf587a80e43ffa1f42e12311431b045e84c833f8c6985db50fce890164136f6075034a
ljgoh@Lynelle-VirtualBox:~/cse13s/asn5$ cat text2.txt
559ad96d02de67cde9508cc774e590d53a17f43215cac5b6c3a042771e07028fa69e6ac2a26ec19d10e8a764393f4b981821c2e06ceeeec93be5dead9d482a07cf9ded8e2ee6a0cddd771c
d240eb6bdcad4e5ee9b84721baef98235784cf587a80e43ffa1f42e12311431b045e84c833f8c6985db50fce890164136f6075034a
ljgoh@Lynelle-VirtualBox:~/cse13s/asn5$
```

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

```

ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ ./decrypt -i text1.txt -o text3.txt -v
n - modulus (1023 bits): 83710187936531166559713936584320443229972238077283493907471424043028601203374657367831620032162344663920230336077784869604418
037322715391523064620914459506400402703263561740443474848451952694483410399346780317853468829235510168069182798411001638937679708017571070819630161484
7573304659865531437596553347513
d - public exponent (1022 bits): 261793682623654838123458846198863203443713650681973836290659232383752699313779069769543404579207381677926421456242408
118985353428357439016545619636318747738919565978248757978968734616247979142899947348363382064173906140974953569561729641964983114953462959720864085193
335970794184296210416590753577910167843991

```

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.

```

ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ ./decrypt-dist -i text2.txt -o text4.txt -v
n - modulus (1023 bits): 83710187936531166559713936584320443229972238077283493907471424043028601203374657367831620032162344663920230336077784869604418
037322715391523064620914459506400402703263561740443474848451952694483410399346780317853468829235510168069182798411001638937679708017571070819630161484
7573304659865531437596553347513
d - private exponent (1022 bits): 261793682623654838123458846198863203443713650681973836290659232383752699313779069769543404579207381677926421456242408
811898535342835743901654561963631874773891956597824875797896873461624797914289994734836338206417390614097495356956172964196498311495346295972086408519
335970794184296210416590753577910167843991

```

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.

```

ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ cat text3.txt text4.txt
Hello world!
Hello world!
ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$

```

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.

```

ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ ./keygen -b 333333333
./keygen: Number of bits must be 50-4096, not 333333333.
Usage: ./keygen [options]
./keygen generates a public / private key pair, placing the keys into the public and private
key files as specified below. The keys have a modulus (n) whose length is specified in
the program options.
-s <seed> : Use <seed> as the random number seed. Default: time()
-b <bits> : Public modulus n must have at least <bits> bits. Default: 1024
-i <iters> : Run <iters> Miller-Rabin iterations for primality testing. Default: 50
-n <pbfile> : Public key file is <pbfile>. Default: rsa.pub
-d <pvfile> : Private key file is <pvfile>. Default: rsa.priv
-v : Enable verbose output.
-h : Display program synopsis and usage.

```

```

ljgoh@Lynelle-VirtualBox:~/cse13s/asgn5$ ./keygen-dist -b 333333333
./keygen-dist: Number of bits must be 50-4096, not 333333333.
Usage: ./keygen-dist [options]
./keygen-dist generates a public / private key pair, placing the keys into the public and private
key files as specified below. The keys have a modulus (n) whose length is specified in
the program options.
-s <seed> : Use <seed> as the random number seed. Default: time()
-b <bits> : Public modulus n must have at least <bits> bits. Default: 1024
-i <iters> : Run <iters> Miller-Rabin iterations for primality testing. Default: 50
-n <pbfile> : Public key file is <pbfile>. Default: rsa.pub
-d <pvfile> : Private key file is <pvfile>. Default: rsa.priv
-v : Enable verbose output.
-h : Display program synopsis and usage.

```

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..