

Exercise 6

Rivest-Shamir-Adleman (RSA) Algorithm

CODE:

```
import java.io.IOException;
import java.math.BigInteger;
import java.util.*;

public class RSA {
    private BigInteger p;
    private BigInteger q;
    private BigInteger N;
    private BigInteger phi;
    private BigInteger e;
    private BigInteger d;
    private int bitlength = 1024;
    private Random r;

    public RSA() {
        r = new Random();
        p = BigInteger.probablePrime(bitlength, r);
        q = BigInteger.probablePrime(bitlength, r);
        N = p.multiply(q);
        phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
        e = BigInteger.probablePrime(bitlength / 2, r);
        while (phi.gcd(e).compareTo(BigInteger.ONE) > 0 && e.compareTo(phi) < 0)
            //gcd(e,phi)=1 and 1<e<phi
            {
                e.add(BigInteger.ONE);
            }
        d = e.modInverse(phi);
    }

    public byte[] encrypt(byte[] message) {
        BigInteger res = new BigInteger(message).modPow(e, N);
        System.out.println("\nThe cipher text is (in Big Integer) " + res);
        return res.toByteArray();
    }

    public byte[] decrypt(byte[] message) {
        BigInteger res = new BigInteger(message).modPow(d, N);
        System.out.println("\nThe plain text is (in Big Integer) " + res);
        return res.toByteArray();
    }

    public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(System.in);
```

```

System.out.println("\nRSA ALGORITHM");
System.out.println("*****");
RSA rsa = new RSA();
String plain, cipher;

System.out.println("\nKey Generation");
System.out.println("*****");
System.out.println("\nP is (in Big Integer)");
System.out.println("-----\n"+ rsa.p);
System.out.println("\nQ is (in Big Integer)");
System.out.println("-----\n"+ rsa.q);
System.out.println("\nN is (in Big Integer)");
System.out.println("-----\n"+ rsa.N);
System.out.println("\nPHI (N) is (in Big Integer)");
System.out.println("-----\n"+ rsa.phi);
System.out.println("\ne is (in Big Integer)");
System.out.println("-----\n"+ rsa.e);
System.out.println("\nThe private key 'd' is (in Big Integer)");
System.out.println("-----\n"+ rsa.d);

System.out.println("\nEncryption");
System.out.println("*****");
System.out.print("\nEnter the plain text: ");
plain = sc.nextLine();

byte[] plainB =plain.getBytes();

System.out.println(
    "\nThe plain text is (in Big Integer) " + new BigInteger(plainB)
);
System.out.println(
    "\nThe plain text is (in Base64) " +
    Base64.getEncoder().encodeToString(plainB)
);

byte[] cipherB = rsa.encrypt(plainB);

System.out.println(
    "\nThe cipher text is (in Base 64): " +
    Base64.getEncoder().encodeToString(cipherB)
);

System.out.println("\nDecryption");
System.out.println("*****");
System.out.print("\nEnter the cipher text (in Base64): ");
cipher = sc.nextLine();

cipherB = Base64.getDecoder().decode(cipher);

```

```

System.out.println(
    "\nThe cipher text is (in Big Integer) " + new BigInteger(cipherB)
);

plainB = rsa.decrypt(cipherB);

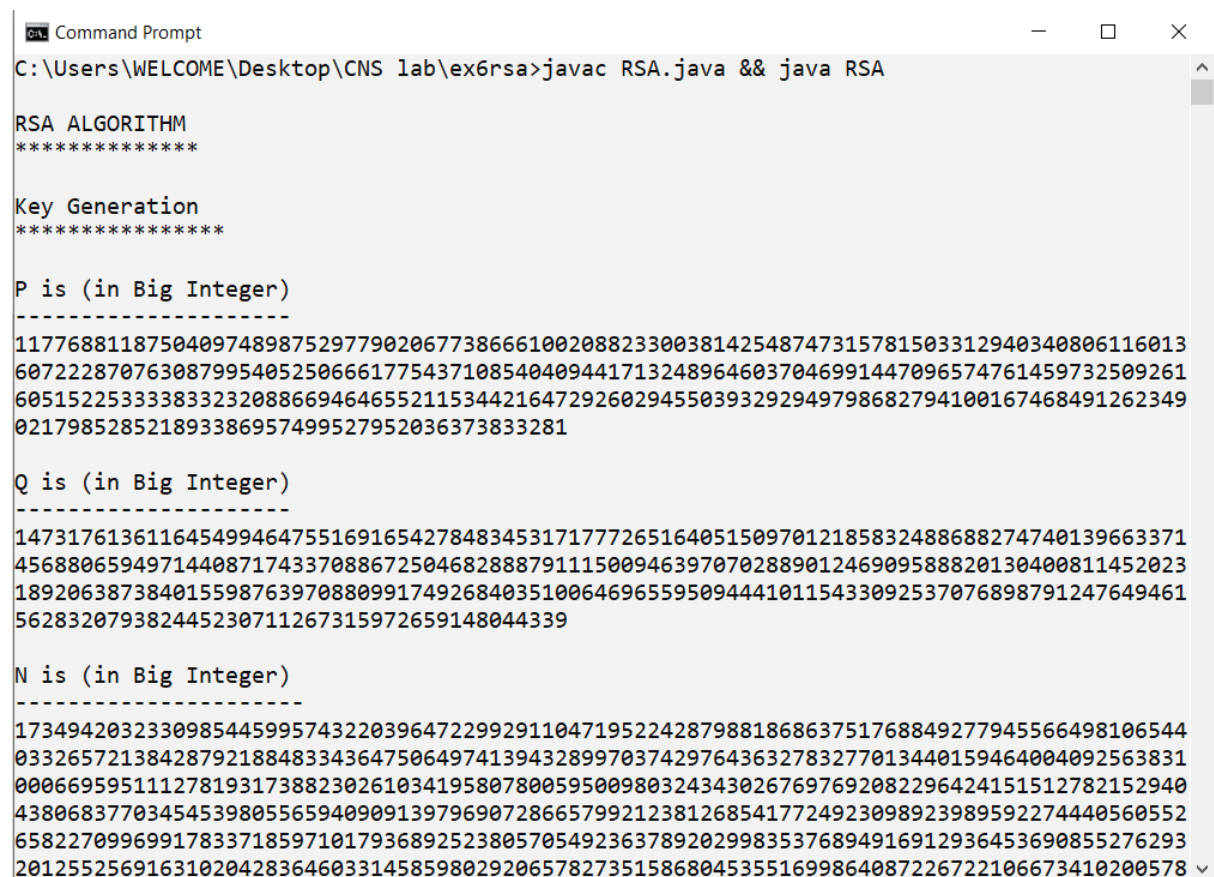
System.out.println(
    "\nThe plain Text is (in Base 64): " +
    Base64.getEncoder().encodeToString(plainB)
);

System.out.println("\nThe original plain Text is: " + new String(plainB));
}
}

```

OUTPUT:

Example 1:



```

C:\Users\WELCOME\Desktop\CNS lab\ex6rsa>javac RSA.java && java RSA

RSA ALGORITHM
*****

Key Generation
*****

P is (in Big Integer)
-----
11776881187504097489875297790206773866610020882330038142548747315781503312940340806116013
60722287076308799540525066617754371085404094417132489646037046991447096574761459732509261
60515225333383323208866946465521153442164729260294550393292949798682794100167468491262349
021798528521893386957499527952036373833281

Q is (in Big Integer)
-----
14731761361164549946475516916542784834531717772651640515097012185832488688274740139663371
45688065949714408717433708867250468288879111500946397070288901246909588820130400811452023
18920638738401559876397088099174926840351006469655950944410115433092537076898791247649461
562832079382445230711267315972659148044339

N is (in Big Integer)
-----
17349420323309854459957432203964722992911047195224287988186863751768849277945566498106544
03326572138428792188483343647506497413943289970374297643632783277013440159464004092563831
00066959511127819317388230261034195807800595009803243430267697692082296424151512782152940
43806837703454539805565940909139796907286657992123812685417724923098923989592274440560552
65822709969917833718597101793689252380570549236378920299835376894916912936453690855276293
20125525691631020428364603314585980292065782735158680453551699864087226722106673410200578

```

```
Command Prompt

N is (in Big Integer)
-----
17349420323309854459957432203964722992911047195224287988186863751768849277945566498106544
03326572138428792188483343647506497413943289970374297643632783277013440159464004092563831
00066959511127819317388230261034195807800595009803243430267697692082296424151512782152940
43806837703454539805565940909139796907286657992123812685417724923098923989592274440560552
65822709969917833718597101793689252380570549236378920299835376894916912936453690855276293
20125525691631020428364603314585980292065782735158680453551699864087226722106673410200578
23865468245202024633597529746853994802808479001494528050902490318068932283981846259

PHI (N) is (in Big Integer)
-----
17349420323309854459957432203964722992911047195224287988186863751768849277945566498106544
03326572138428792188483343647506497413943289970374297643632783277013440159464004092563831
00066959511127819317388230261034195807800595009803243430267697692082296424151512782152940
43806837703454539805565940909139796907286392905698325998943361414951856494005263023174002
84144052324158332104605100578608306601185485132848660067752797307162062888059948023217112
41238809365682782071679208422725436330780988376517962604720847223741579761303848252843278
73364130542136792858266352680594255890997894370886623712284821551225007588459968640

e is (in Big Integer)
-----
12545678540723446982259157974485595914830777176447327085760209094494309512522179831845733
077572531045646564859435402278388039511573811300577448185929611143
```

```
Command Prompt

The private key 'd' is (in Big Integer)
-----
52581352005780918991355945729542116233446886023975563578412426871184101734688402757324466
76327294658908167617535172873533316050374864732443485847858707668199945277019569027151137
95822163693787312902773191298305916870641971789836973367359927070537305829304277206915993
74437522195473290704144308832503134715573389724872151753902390765188546543110606740715789
13205929779378052223848206133806699830890801352057781203577880275318345318230467091093655
80576520174450100808724751497532167155274797412653902124642460566409835793103606790910069
0113535626368954522605193160916943378089774789731852984210586030634212004821084087

Encryption
*****

Enter the plain text: Hello 123 World!

The plain text is (in Big Integer) 96231036770496460398885810566817670177

The plain text is (in Base64) SGVsbG8gMTIzIFdvcmxkIQ==

The cipher text is (in Big Integer) 26522802315299759778436812959016757065165016640845732
50591989872276581212025915644980893815978831909043627232036399985315825584650566914780583
32887911262966335759082117246873225331800713781992401710050727602545373731753243024638466
99147840162884845306915343466503440393331338300565575229806281047540563819481310236053531
90965084066588633163034926528468550674256132442418986208557020644652228588058704292215479
53203961957438778980751039792157699025352539493462604857860830717386075226254348191599967
47804553664629917282361970859882496237242358532451759251040623471387527652330847567582366
93581578589480807017878888033

The cipher text is (in Base 64): FQKWVuiiqC7V8kit+9zXagBJHrBRYQs1Tz9oxjl2FDHzYAPzrzLoig0T
BlmH4f4ZWnsrjw4a2ffbukohYPopq2rtdfyHTzdhP313J3tjkVgF8pMPixkIhyyyWL1uvQ5zXj4TD2hTB0ue/TX0p
```

```
Command Prompt

The cipher text is (in Base 64): FQKWVuiiqC7V8kit+9zXagBJHrBRYQs1Tz9oxjl2FDHzYAPzrzLoig0T
B1mH4f4ZWnsrjw4a2ffbukohYPopq2rdtfyHTzdhP313J3tjkVgF8pMPixkIhyyyWL1uvQ5zXj4TD2hTB0ue/TX0p
/2ANpyqoaSCkpBuCn97zbvrtKKK0hK4D+BRxwuJdenpmlVB/gkpqnUYsqf3BTGJg9QN8FXnOuY6T8+8H+1kDg+Cuk
nrRD0n7a+kP4cJYIekAoHiTOpBvbFfMern/FBBYqs76gXmd1iI+kYUWrGrulSCEaP7xX24BSSDxxHadDccLwLkJRr
3H+ZIQpkZNsCFpYOeYQ==

Decryption
*****

Enter the cipher text (in Base64): FQKWVuiiqC7V8kit+9zXagBJHrBRYQs1Tz9oxjl2FDHzYAPzrzLoig
0TB1mH4f4ZWnsrjw4a2ffbukohYPopq2rdtfyHTzdhP313J3tjkVgF8pMPixkIhyyyWL1uvQ5zXj4TD2hTB0ue/TX
0p/2ANpyqoaSCkpBuCn97zbvrtKKK0hK4D+BRxwuJdenpmlVB/gkpqnUYsqf3BTGJg9QN8FXnOuY6T8+8H+1kDg+C
uknrRD0n7a+kP4cJYIekAoHiTOpBvbFfMern/FBBYqs76gXmd1iI+kYUWrGrulSCEaP7xX24BSSDxxHadDccLwLkJ
Rr3H+ZIQpkZNsCFpYOeYQ==

The cipher text is (in Big Integer) 26522802315299759778436812959016757065165016640845732
50591989872276581212025915644980893815978831909043627232036399985315825584650566914780583
32887911262966335759082117246873225331800713781992401710050727602545373731753243024638466
99147840162884845306915343466503440393331338300565575229806281047540563819481310236053531
90965084066588633163034926528468550674256132442418986208557020644652228588058704292215479
53203961957438778980751039792157699025352539493462604857860830717386075226254348191599967
47804553664629917282361970859882496237242358532451759251040623471387527652330847567582366
93581578589480807017878888033

The plain text is (in Big Integer) 96231036770496460398885810566817670177

The plain Text is (in Base 64): SGVsbG8gMTIzIFdvcmxkIQ==

The original plain Text is: Hello 123 World!
```

Example 2:

```
Command Prompt

C:\Users\WELCOME\Desktop\CNS lab\ex6rsa>javac RSA.java && java RSA

RSA ALGORITHM
*****

Key Generation
*****

P is (in Big Integer)
-----
15308195487307735571390408855550195946934331819840757696206929412272097511131325270844195
85345584076735630671915506848159643804622998146777317280295986945389229063799424264184252
10607234240752115614094764467553514312720466871657204519295151146181520892125581194024533
603236164272197712485300496183630847602543

Q is (in Big Integer)
-----
11407894185513152763907747778806928271601498090832552528022814716988904895651335330386991
88677299026413493061007186800662556465580160998643629948371987242350920784735696519080790
94110004809068071286917136759306201446740402807298430985377534629294285527641090962498116
187087398040980371526924297856993145202393

N is (in Big Integer)
-----
17463427429035660075513437393888179726762027135817417316439147250383539670263334953757963
18894844135359291333952519431681645815815389395817937612986970420749886512328028202737938
17500535127358436117094929775657250462415180231825261547024547533249244611017077408592563
88716541896630472046291668547908072112672408066678217105396494648770515801948528994019326
47263011409029979339723353129182320227667356717552453423647102324205258597481489331167476
43543903588892949379210186487917356890869624121766757653782602618238193774794719437545736
```



```
Command Prompt

N is (in Big Integer)
-----
17463427429035660075513437393888179726762027135817417316439147250383539670263334953757963
18894844135359291333952519431681645815815389395817937612986970420749886512328028202737938
17500535127358436117094929775657250462415180231825261547024547533249244611017077408592563
88716541896630472046291668547908072112672408066678217105396494648770515801948528994019326
47263011409029979339723353129182320227667356717552453423647102324205258597481489331167476
43543903588892949379210186487917356890869624121766757653782602618238193774794719437545736
91423219931312927896891152269406172482823801780347890081444826617283633706556485399

PHI (N) is (in Big Integer)
-----
17463427429035660075513437393888179726762027135817417316439147250383539670263334953757963
18894844135359291333952519431681645815815389395817937612986970420749886512328028202737938
17500535127358436117094929775657250462415180231825261547024547533249244611017077408592563
88716541896630472046291668547908072112672140905781488896513141667204172230706343635720219
73952787179285850078720946346521718996479616488721421932409773097268770375478787299576022
22596674920918761639060337952796573625826576949376259451913592499225925177637124828848947
35787715258627152421084732502734015960174011456785576903360814392489593082563680464

e is (in Big Integer)
-----
11309525844223232980506424662831420051042556813495265065126438329874710553792490621535757
159470674276254776822820929533468262062492059813893928078026529811
```

```
Command Prompt

The private key 'd' is (in Big Integer)
-----
22301696846416451269095712502822276658728815648372925341234419016592015900344670068817628
83888524399796060005967214955610808138772747297233089764532958507611248977239894267637012
32049166957673167417559901669091874790787645049520550166447949274067620398616901246593124
08459917564533872285435752800716052276564566854148026242639981963219198460279097694075088
11761043191363860972880969704960523315256592813917838225187959596005125707642447423123924
11848979711525298577168549118434369297202527710661161554103348079489751716286622109819039
0313053681500267702542891282605193017277470717489776100399075370286790606368962427

Encryption
*****

Enter the plain text: RUN

The plain text is (in Big Integer) 5395790

The plain text is (in Base64) UlVO

The cipher text is (in Big Integer) 68049402944128944511709999301882689328354370788248736
57571422055061477623577767502344638547208308427474459011612431878198912708682929673418853
88178465067466910112239707912222529379294465506589131079346947207210318912369284918813887
95771485422322952187546836296880105371714716614421070298674387001167503289607040152828330
87713207531754016362533411277891517574397783202021110047767331272459850556929139917348744
95208448485886040554735940903891341699899214990956995750924284798976821097420106983854935
68705029347520694835189917228042093878594650198806424359985486768753043255386380038841725
36918652085986958780520811463
```

The cipher text is (in Base 64): NefPi/tJgSqVdyW0orLQgAHuWnVvakyKuA0GZI/IC0jwvVE0ce17ecbt ^
pYT62N4dApBayrmx8DBpg60kDtThRmNaggaxws2eePobbsbaR2635MSriLXJcr77N4Zoegwqm4ZDuXyon0WpHBDp2
ii8UNHCee6td5IsxQSLgpcVuh/zPquzLLNNe9JWMJZHhCe7otpX4NbGYmE8LY8dRc9/rA2JUJMok+725s5tPRYs
T90SsqETNirsHJUmdp/BMpMe1zewRknjBY33gqU1ArKz1HI/eDQHEDpudw9UqTKJHZEsgTr3hLey9aGzOIaksjEa3
Znh9zQTYWyeshyplLxw==

Decryption

Enter the cipher text (in Base64): NefPi/tJgSqVdyW0orLQgAHuWnVvakyKuA0GZI/IC0jwvVE0ce17ec
btpYT62N4dApBayrmx8DBpg60kDtThRmNaggaxws2eePobbsbaR2635MSriLXJcr77N4Zoegwqm4ZDuXyon0WpHBD
p2ii8UNHCee6td5IsxQSLgpcVuh/zPquzLLNNe9JWMJZHhCe7otpX4NbGYmE8LY8dRc9/rA2JUJMok+725s5tPR
YsT90SsqETNirsHJUmdp/BMpMe1zewRknjBY33gqU1ArKz1HI/eDQHEDpudw9UqTKJHZEsgTr3hLey9aGzOIaksjE
a3Znh9zQTYWyeshyplLxw==

The cipher text is (in Big Integer) 68049402944128944511709999301882689328354370788248736
57571422055061477623577767502344638547208308427474459011612431878198912708682929673418853
88178465067466910112239707912222529379294465506589131079346947207210318912369284918813887
95771485422322952187546836296880105371714716614421070298674387001167503289607040152828330
87713207531754016362533411277891517574397783202021110047767331272459850556929139917348744
95208448485886040554735940903891341699899214990956995750924284798976821097420106983854935
68705029347520694835189917228042093878594650198806424359985486768753043255386380038841725
36918652085986958780520811463

The plain text is (in Big Integer) 5395790

The plain Text is (in Base 64): UlVO

The original plain Text is: RUN