Homework #2

Due date: 20/10/2017
Notes:

- Computer programs and other soft material must be submitted through sucourse.
- Winzip your programs and add a readme.txt document to explain the programs and how to use them.
- Name your winzip file as "cs411_507_hw02_yourname.zip"

1. (**20 pts**) Consider the following modulus:

n=93923989829536838645425792878391884185852090015047096928325495680735077974573254200941142654635659889069058397106827931666572379715390977199861730622956934976456672365327932404671265952251676873946441183498231232238528978514016592480741429133646590466736838628474537927484115146306435052269979805285188280829

Consider also the following number:

y=56944778095026064508803320557324194828217604358002584860993983127246257172572625621288924208876440674553832158479040031724187898214806257072823966168352349606611537362177018477860328420295741255602485512749523176875861988864327723267007959112818744831884377494800464304976689851182957989115066715277845807162

The integer y is a quadratic residue modulus n, and its square roots are known:

{
11539293044780349664317193739835099161008802876501927386196632728052884864570713532203427639431106901813684612368583348366278480903942058202903897004649103055743285751895970690426609694759832511976035062765586527404034628524567695175048634551582051994897593554954834163440853023500476963332472457537871514030,

65462078929074573896561682473134429960343753900612807859205872292206976673766764795638688642515250572910065281537469802520465145301193070320295080324024966677495122426689606204830245523869232443939539387029183651868429281477875679397889633306037656227303018935833418527282673149878942564121394686954099450002,

284619109004622647488641104052574542255083361144342890691196233885281013008064894053024540121204093161589931155693581291461072344141979068795666502989319682989615499386383261998410204283824444300069017964690475803

7009969703614091308285179582760893423943381969264111940020144199642749248814858511833108883O827,

82384696784756488981108599138556785024843287138545169542128862952682193110002540668737715015204552987255373784738244583300293898811448918996957833618307831920713386613431961714244656257491844361970406120732644704834494349989448897305692794582064538471839245073519703764043262122805958088937507347747316766799

}

Factor n.

Use greatest common divisor of a-b and the modulus as follows:

p = gcd(a-b, n) =
8799871096643187569636362221703137117970705734262560672209772353211410091242059304273666872097572116972793282600477392267607320573024412287806194663768831

q = n/p =
10673337006648339312925067672751218131836069322217857757055585879216897770517306431254639065397004795737064426817789349478902907246196721814345105935705859

2. (**20 pts**) Consider the group Z_{46}^*.

a. How many elements are there in the group Z_{46}^*? List the elements of Z_{46}^*. (**5 pts**)

Z_{46}^* is the group that consists of numbers, which are relatively prime to 46. Therefore, the number of elements can be computed using Euler's totient function

$\phi(46) = \phi(2\times23) = 1\times22 = 22$

They are {1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45}

b. Find all the generators in the group Z_{46}^*. (**5 pts**)

Z_{46}^* is a cyclic group, therefore there are generators which are

{5, 7, 11, 15, 17, 19, 21, 33, 37, 43}

c. How many elements are there in the group Q_{46}^*? The group Q_{46}^* consists of all residues in Z_{46}^*, which are relatively prime to 46. List the elements of Q_{46}^*. (**5 pts**)

The group Q_{46}^* consists of elements in Z_{46}^*which are quadratic residues, i.e. elements that have square roots modulo 46. These elements are

{1, 3, 9, 13, 25, 27, 29, 31, 35, 39, 41}

There are 11 elements in Q_{46}^{*}.

d. Find the generators in Q_{46}^{*}. (**5 pts**)

{3, 9, 13, 25, 27, 29, 31, 35, 39, 41} are generators since their powers generate all the elements in Q_{46}^{*}. 1 is a non generator since its powers can generate only the following elements: 1.

3. (**30 pts**) Consider the following numbers:

p=
12997110888321186459316436989724265349460712629389513072528028592113499927549315535209275673470089822303733552137264381101263928058260190003283395786606613

q=
12208677561088985820124024567796306546470708824672729385334541873189398794633650052335324309711480763919028490727764288604694220087933148985261608483250761

n=p×q

m=
104508383527831908823036869386648261753406819764116432937577039021334848453871469297253914394880121355608067237189088530241064709118310089323613713098921642285063603825756456138538154719968501891170428872546150841926912027053577225264111903314505091217903294998448178692295392732881494063202944747727969714684

e = 67

a. Compute c ≡ m^e mod n (Hint: you can use `pow(m, e, n)` function of Python) . (**5 pts**)

n=
158677536061232204658393773935896900727298645787572781799648383107628926980714927153614878935279780362427564853712851616037597223887361677691793706045284159279949143629365339119850917863751677664046513081392152647862570804301588857974508794857377645185972146884225226694319403547128686211473900355543339882493

c ≡ m^e mod n =
11261646976760858481160916642548827106258644725211512553756486319304453

8537395472094406209420709926485510934377621008687907559465706333197400364258608745396555196216103894265082687542702158480984946698751962784029154848419122472663111880510768289690233009425382958375171148628817677521343074095737592803335878

b. Compute the number d such that $e \times d \equiv 1 \mod \phi(n)$, compute $m' \equiv c^d \mod n$, and show that $m = m'$. (**5 pts**)

Using EEA we can find

d =
7578628587999150073236717561117463915333666644810880859533556111106353184819069685308598894461984650711672765952406741988106136782023487852796993932076004333927495746480267309559097477339597245432864009146122882527572776789428859705753432958755564711826113013579635422399855564886228068362942954287317026923

$m' \equiv c^d \mod n$ =
1045083835278319088230368693866482617534068197641164329375770390213348484538714692972539143948801213556080672371890885302410647091183100893236137130989216422850636038257564561385381547199685018911704288725461508419269120270535772252641119033145050912179032949984481786922953927328814940632029447477279697146 84

c. Compute $c^d \mod n$ using Chinese Remainder theorem and p and q. You are allowed to perform exponentiation operation only in mod p and mod q. List the following values: $c_p = c \mod p$, $c_q = c \mod q$, $d_p = d \mod (p-1)$, $d_q = d \mod (q-1)$, $p^{-1} \mod q$, $q^{-1} \mod p$, $c_p^{d_p} \mod p$, and $c_q^{d_q} \mod q$. (**15 pts**)

$d_p = d \mod (p-1)$ =
3297774404499405519528051176497201655833315144770473466163828150237753712960274091023249051477485477300947319199007380279425175775976466120236085498094215

$d_q = d \mod (q-1)$ =
1093314408455730073443942498608624466848123178328901138985182854315468548773162691253909639675654993783793596184575906442211422694441774535993576879097083

$m_p = m \mod p$ =
7811512201739981695601181636669341084176351564109063915850725869370120077773470337824374673837545983579190734805159776461191295061224280922787521081025538

$m_q = m \mod q$ =
4140992233405111872830127805583727890844221847150031363372273846094805 8

90795421530767321150592953424216065213031145948095563512112130675224752
020408206822

$c_p \equiv m_p^{d_p} \bmod p$; $c_q \equiv m_q^{d_q} \bmod q$

c_p=
11792348330928035228208991070946382506194288360813574455958248260956522
09778390289698602070888336486037874421839672513698478948255274319805011
2790652329507

c_q=
21857157832899349480312697964977714099956714494476288565832939475201365
30429843834953363721243428807803221996057231855934964172118007401489722
093005810481

$c=CRT(c_p, c_q, p, q)=c_p \times q \times (q^{-1} \bmod p) + c_q \times p \times (p^{-1} \bmod q) \bmod n$ =
13275463994535999330575235142582723353389094361416623471736307584636808
62107087125463177765703044638200942455302101290000349311437986905899424
47470332728543276423298031608227339028725337871979606662280926207435995
20630449352998526722517037209183798171904699703698756197677763883762412
979789401669370274008194

d. Measure the execution times of the exponentiation methods in (b) and (c) using the Python function as follows:

```
import timeit
import time

…

iter = 100
t1 = time.clock()
for i in range(0, iter):
    pow(c, d, n)
t2 = time.clock()
print t2-t1
```

Which one is faster? (<u>Warning</u>: In exponentiation with CRT, you should not include the execution times of values that can be precomputed, such as $p^{-1} \bmod q$ and $q^{-1} \bmod p$

Exponentiation in (b) takes 1.65 ms whereas the one in (c) takes 0.47 ms. Of course, this is my computer; these values change from computer to computer. If the algorithms are implemented correctly, the exponentiation with CRT is significantly faster than the classic exponentiation.

4. (**15 pts**) Solve the following equations of the form ax ≡ b mod n and find all solutions for x if a solution exists. In case there is no solution, your answer must be "NO SOLUTION", and explain why there is no solution.

   a. n = 1200320707477907914300008804988
      a = 7211941535834517096225500817
      b = 102092299425228521972149597163 (**5 pts**)

      d= gcd(a, n) = 1. There is a solution.
      solution : 83912145255473489949903796379

   b. n = 1200320707477907914300008804988
      a = 44575693167043501900449109190
      b = 84664078284205068314514580089 (**5 pts**)

d= gcd(a, n) = 2 and d does not divide b then there is no solution.

   c. n = 1200320707477907914300008804988
      a = 404
      b = 2124884389680246530198080982220 (**5 pts**)

d=gcd(a, n) = 4 and therefore there are four solutions.
One solution can be obtained as follows:

$$\tilde{a} = \frac{a}{d} = 101$$

$$\tilde{n} = \frac{n}{d} = 30008017686947697857502201247$$

d divides b

$$\tilde{b} = \frac{b}{d} = 531221097420061632549520245555$$

Now we have an equation:

$101\tilde{x} \equiv 531221097420061632549520245555$ mod $30008017686947697857502201247$

$\tilde{x} \equiv$
$\{29622559878710622345195056909, 59630577565658320202697258156, 89638595252606018060199459403, 119646612939553715917701660650\}$

5. (**15 pts**) Consider the following two polynomials over GF(2) :

$p_1(x) = x^6 + x + 1$
$p_2(x) = x^6 + x^2 + 1$

a. Are they irreducible over GF(2)? Explain your answer. (**5 pts**)

$p_1(x)$ is irreducible

$p_2(x) = x^6 + x^2 + 1 = (x^3 + x + 1)(x^3 + x + 1) \Rightarrow p_1(x)$ is reducible.

b. Are they primitive over GF(2)? You need to show whether the roots of these polynomials generate all nonzero elements of the field GF(2^6) which has 64 elements. (**10 pts**)

$p_2(x)$ is reducible, therefore it cannot be primitive.

If $p_1(x)$ is a primitive polynomial, its roots need to generate all the nonzero elements of the binary extension field GF(2^5).

Let α be a root of $p_1(x) = x^6 + x + 1$. Then $p_2(\alpha) = \alpha^6 + \alpha + 1 = 0 \Rightarrow \alpha^6 = \alpha + 1$.

α
α^2,
α^3,
α^4,
α^5,
$\alpha^6 = \alpha + 1$,
$\alpha^7 = \alpha^2 + \alpha$,