CS 450 HW 3 Report

** Walkthrough of the problem first, written problem answers follow them.**

Programming question 1.

At first, I thought this would be a simple answer, 1/2x and 1/2y.
Upon rereading the requirement for an integer answer, I realized it
would be a lot harder than I had thought. I spent quite a while
just trying to figure out how to get ax+by=1 to work, and I watched
a lot of videos and read many threads. Specifically, the following links
are what helped me the most to understand what was needed and how.

https://www.youtube.com/watch?v=FjliV5u2IVw
https://stackoverflow.com/questions/4917003/whats-algorithm-used-to-solve-linear-diophantine-equation-ax-by-c
https://stackoverflow.com/questions/53030850/extended-euclidian-algorithm-in-scheme

After following these, there was a lot of guessing and checking, changing
variable placements, what is getting added or subtracted until I was able
to run the test case and get the correct answers.

Problem 1 Written Exercise:

> (solve-ax+by=1 233987973 41111687)
(-11827825 . 67318298)

Programming question 2.

If I'm perfectly honest, I don't fully understand how it worked, I just
randomly tried solutions and tried reversing the order of the procedures in
convert-list and it ended up working correctly after a few trial
and errors.

> (RSA-decrypt result1 test-private-key1)
"This is a test message. "

Written exercise 2:

> (define test-key-pair3 (generate-RSA-key-pair))
> (define test-key-pair4 (generate-RSA-key-pair))
> test-key-pair3
((630480229 . 552041671) 630480229 . 312831331)
> test-key-pair4

((450084457 . 193992087) 450084457 . 77224927)
> (define test-public-key3 (key-pair-public test-key-pair3))
> (define test-private-key3 (key-pair-private test-key-pair3))
> (define test-public-key4 (key-pair-public test-key-pair4))
> (define test-private-key4 (key-pair-private test-key-pair4))
> test-public-key3
(630480229 . 552041671)
> test-private-key3
(630480229 . 312831331)
> test-public-key4
(450084457 . 193992087)
> test-private-key4
(450084457 . 77224927)

Programming question ¾

For the encrypt-and-sign problem, I was overthinking it due to the line
in the instructions where it says "Start
by specifying a (very) simple data structure called a signed-message".

I had originally thought this meant we would need to declare something,
but after looking at the test and the discussion on piazza #77, I just
used cons on the encrypted message and signature which seemed to work.

One issue I had at first was trying to run the procedure without compressing.
I was getting the correct message but my signature was always wrong until I
added a compress into the procedure. I then had to call a (list x)
on the compressed message or I would get a contract violation error.

> (define result2
(encrypt-and-sign "Test message from user 1 to user 2"
test-private-key1
test-public-key2))
> result2
((499609777 242153055 12244841 376031918 242988502 31156692 221535122
463709109 468341391) . 15378444)

For problem 4 I simply tried to test and make sure the keys were correct
to begin decrypting. If the wrong private key is given, the message is
unintelligible. If the wrong public key is given, or does not match what
is attached to the message, then it returns false and we know it was not
from the correct person.

```
> (authenticate-and-decrypt result2 test-public-key1 test-private-key2)
"Test message from user 1 to user 2  "
> (authenticate-and-decrypt result2 test-public-key1 test-private-key1)
"0ADK\u0002\u001C\a\u0001D\u0002C0V\u0001\u0001`#/\u000E\u0001zg\f\u000F
        \u0002_1X\u0014\u0001\b*\u0014\u001EX\u0017mX\u0001=X2I\u0002"
> (authenticate-and-decrypt result2 test-public-key2 test-private-key2)
#f
```

Written Exercise 3:
The message is: "Put your mask on! There is a deadly pandemic outside. "
The message was sent by kamala harris.

To do this, I used cons on the mystery message and signature to get them in one list so I could use it as an argument for authenticate-and-decrypt. I then randomly tested running the procedure using joe-biden-private-key and randomly using the provided public keys. I actually used the right public key first, but tested with a few others to make sure.

```
> (define mysterysignedmessage (cons received-mystery-message
        received-mystery-signature))
> (authenticate-and-decrypt mysterysignedmessage kamala-harris-public-key
        joe-biden-private-key)

"Put your mask on! There is a deadly pandemic outside.  "
> (authenticate-and-decrypt mysterysignedmessage bernie-sanders-public-key
        joe-biden-private-key)
#f
> (authenticate-and-decrypt mysterysignedmessage michael-cohen-public-key
        joe-biden-private-key)
#f
>
```

Programming Question 5.
For this, I knew we were trying to compute d, so I went and watched the provided video by Obyat however I couldn't figure out how to implement what was shown in the video into the program. I then looked up some RSA explanations and found this website:
https://www.di-mgt.com.au/rsa_alg.html
On this website I found d = modulo inverse (e m) which upon testing, gave the correct answers.

Here are sample outputs and the actual keys called after to confirm.
> (crack-rsa joe-biden-public-key)
(718616329 . 129033029)
> joe-biden-private-key
(718616329 . 129033029)
> (crack-rsa test-public-key1)
(816898139 . 301956869)
> test-private-key1
(816898139 . 301956869)
> (crack-rsa test-public-key2)
(513756253 . 462557987)
> test-private-key2
(513756253 . 462557987)

Written Problem 4.

> (define donald-trump-private-key (crack-rsa donald-trump-public-key))
> (define mike-pence-private-key (crack-rsa mike-pence-public-key))
> (define ivanka-trump-private-key (crack-rsa ivanka-trump-public-key))
> (define michael-cohen-private-key (crack-rsa michael-cohen-public-key))

> (define trump-to-pence (encrypt-and-sign "what are you having for lunch?"
donald-trump-private-key mike-pence-public-key))

> (authenticate-and-decrypt trump-to-pence donald-trump-public-key
mike-pence-private-key)
"what are you having for lunch? "

> (define pence-to-trump (encrypt-and-sign "I'm not hungry, planning to
work through lunch you go on without me." mike-pence-private-key
donald-trump-public-key))

> (authenticate-and-decrypt pence-to-trump mike-pence-public-key
donald-trump-private-key)
"I'm not hungry, planning to work through lunch you go on without me."

> (define pence-to-cohen (encrypt-and-sign "I gave that fool an excuse,
lets go grab a bite somewhere nice." mike-pence-private-key
michael-cohen-public-key))

> (authenticate-and-decrypt pence-to-cohen mike-pence-public-key
michael-cohen-private-key)
"I gave that fool an excuse, lets go grab a bite somewhere nice. "

```
> (define cohen-to-ivanka (encrypt-and-sign "Ditching Don. Want to grab a
        bite with Pence and I?" michael-cohen-private-key
        ivanka-trump-public-key))

> (authenticate-and-decrypt cohen-to-ivanka michael-cohen-public-key
        ivanka-trump-private-key)
"Ditching Don. Want to grab a bite with Pence and I? "
```

Written problem 5.

The time to find the smallest divisors when using 5 digit primes is almost instant. When going up to just 10 digit primes, the time is already in the scale of around 10 minutes. This is about 600,000 times the amount of time if you consider 5 digit primes taking from 0 to 1 ms, and 10 digit primes taking about 600,000 ms. It's not very accurate, but we can estimate that doubling from 10 digit primes to 20 digit primes would provide a time of atleast 600,000 times that of ten minutes, so 360,000,000,000ms.
This equates to 360,000,000 seconds, or 6,000,000 minutes or 100,000 hours and eventually, 11.4 years. Going larger, to 50, or even hundred digit primes would probably take well over hundreds of years to calculate, possibly even thousands or millions once the number gets large enough.

Programming problem 6.

This problem was easy, enough trump2pence would just take a message, and call the encrpyt-and-sign procedure with that message, donald trump's private key generated using crack-rsa on his public key, and mike pence's public key. We can then decode it using mike pence's private key, which we get once again with crack-rsa.

My only issue, is with the test case on gradescope I cannot get the quote to decrypt properly. I am not sure why, as that is the only symbol that is giving me errors when decrypting.

Upon further testing and speaking with Obyat, he gave me a hint at where my program was giving the wrong return. I had made trump2pence a procedure which can take any message as an argument and encrypt it, however we just needed it to take the supplied message and store it in trump2pence. I redid this as below and it now works properly.

code:

```
(define donald-trump-private-key (crack-rsa donald-trump-public-key))
(define mike-pence-private-key (crack-rsa mike-pence-public-key))
(define trump2pence
```

(encrypt-and-sign "Announce that we're increasing taxes by 100%! Biggest increase ever! TREMENDOUS increase!" donald-trump-private-key mike-pence-public-key))

sample output:

```
> trump2pence
((224243101 208047674 564765069 295837598 149056572 536048142 153487972 314310349
2834656 204403445 3147582 106402978 462103370 51248842 172393441 606011332
210316189 320993850 490336219 648117848 274341017 262862335 63310884)
 .
 612334283)

> (authenticate-and-decrypt trump2pence donald-trump-public-key mike-pence-private-key)
"Announce that we're increasing taxes by 100%! Biggest increase ever! TREMENDOUS
increase!  "
>
```