Question 1:

Exam 2 Sean Post

1. p= 26981 , q= 62549
$\phi(n) = (p-1)(q-1)$
= 1,687,545,040 → The decryption part is the inverse of b mod $\phi(n)$
~~I don't know how to calculate this.~~
= 1401973341

Originally, I couldn't find a way to calculate this inverse, so I tried to use my program that I've added to past assignments (I'll add it to the end) that will find *all* the invertible elements and their inverses, given a modulus. However, running an n^2 algorithm on the number 1.6 billion didn't produce any results in a timely manner.

So, I wrote a different program that, given a number and the modulus, will find the inverse, if it exists, in n time complexity:

```
def findInverse(mod, b):
#Find an inverse of a given number within a given modulus.

for i in range(mod):
      if (b * i) % mod == 1:
            print(i)
            return True
return False

if __name__ == '__main__':
      mod = 1687545040
      b = 2021
      print(findInverse(mod, b))
```

This still took a few minutes, but it worked far quicker than the other to find the inverse.

```
1401973341
True
```

As for b = 2020, this same program output no value and returned false. I believe this to mean that 2020 has no inverse given the modulus, meaning that the encrypted message can't be decrypted.

```
False
```

I now realize I could have also just used the inv function on the V200.

Question 2:

2. Seeing as how p & q are primes, if a factor of n is found, it means that either p or q has been found. This allows us to compromise bobs key by calculating the other p or q that we don't have with $\frac{n}{x}$, then calculate as above in Q1.
The probability would be $\frac{1}{n}$ if you just blindly guessed. However, you could utilize a prime generating algorithm (see: Sieve of Atkin) and make a more educated guess, dramatically increasing your chances.

Question 4:

4. $m_1 = 31$ $a_1 = 2$

$m_2 = 32$ $a_2 = 12$

$m_3 = 33$ $a_3 = 21$

$M = m_1 \cdot m_2 \cdot m_3 = 31 \cdot 32 \cdot 33 = 32,736$

$M_i = M / m_i$

$M_1 = \frac{32736}{31} = 1056$

$M_2 = \frac{32736}{32} = 1023$

$M_3 = \frac{32736}{33} = 992$

[V200] egcd(M_i, m_i)

egcd$(1056, 31) = [1 \quad -15 \quad 511]$

egcd$(1023, 32) = [1 \quad -1 \quad 32]$

egcd$(992, 33) = [1 \quad -16 \quad 481]$

$x = \overbrace{(2 \cdot 1056 \cdot -15)}^{-31,680} + \overbrace{(12 \cdot 1023 \cdot -1)}^{-12,276} + \overbrace{(21 \cdot 992 \cdot -16)}^{-333,312} = -377,268 \bmod 32,736$

$\boxed{= 15,564 \bmod 32,736}$

■ mod(-377268 , 32736) 15564

mod(⁻377268,32736)

CRYPTO RAD AUTO FUNC 14/30

Question 5:

All ops are % 3

5.

	$f(1)$	$f(2)$	irreducible
x^2+1	2	$5 \to 2$	1
x^2+2	$3 \to 0$	$6 \to 0$	0
x^2+x+1	$3 \to 0$	$7 \to 1$	0
x^2+x+2	$4 \to 1$	$8 \to 2$	1
x^2+2x+1	$4 \to 1$	$9 \to 0$	0
x^2+2x+2	$5 \to 2$	$10 \to 1$	1

The polynomials are irreducible if they have no roots in $\mathbb{Z}_3$, meaning no 0s.

Irreducible Polynomials:

x^2+1

x^2+x+2

x^2+2x+2

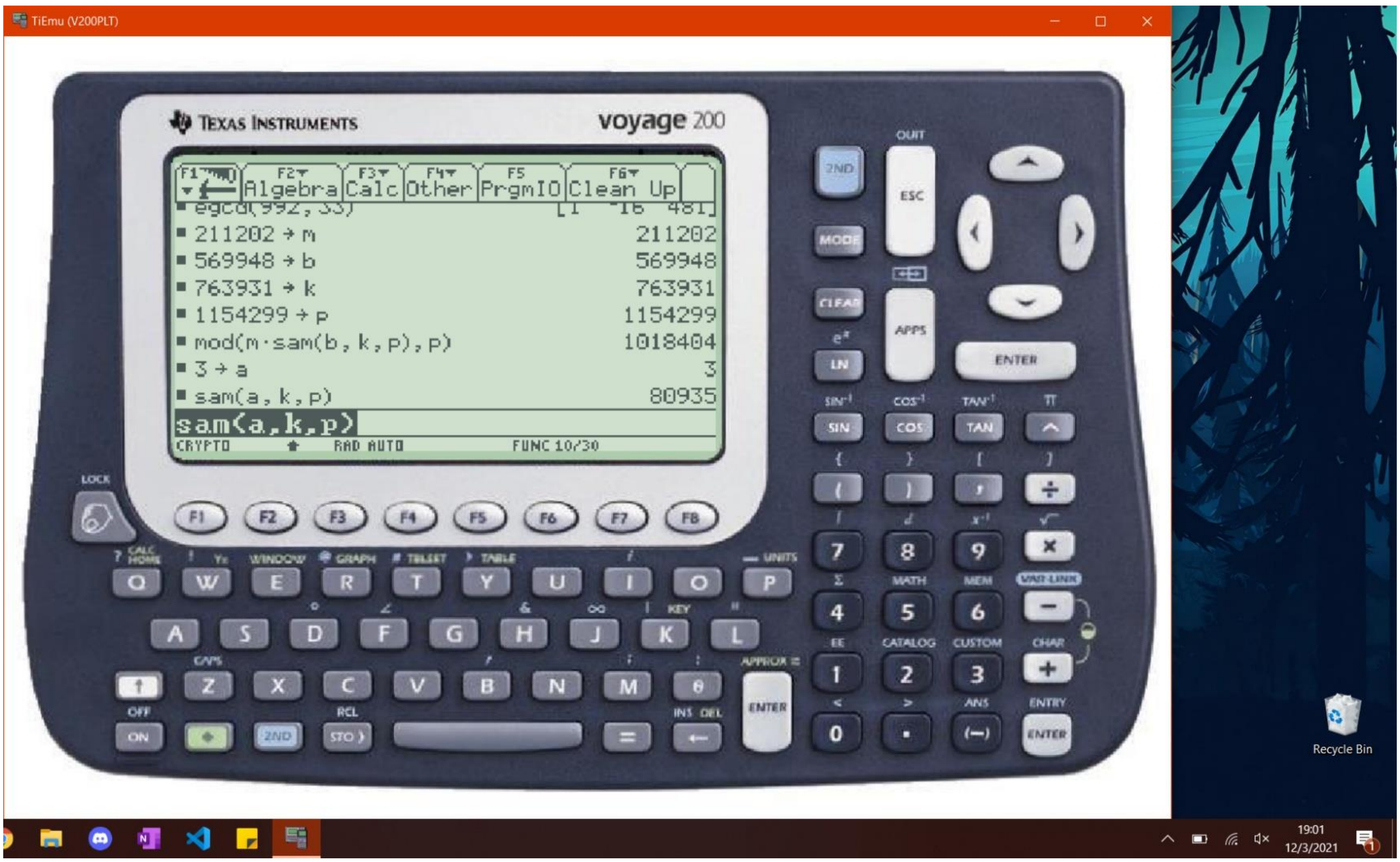
Question 6:

6. $p = 1154299$ [V200] $mod(m * sam(\beta, k, p), p) \to y_2$

$\alpha = 3$ $\beta = 569948$ [V200] $sam(\alpha, k, p) \to y_1$

$k = 763931$ $m = 211202$

$\boxed{(80{,}935\ ,\ 1{,}018{,}404)}$



The last line is y_1 and the third to last line is y_2.

modinv program:

```
# Find all invertible elements and their inverses
# given a modulo.

import math
mod = 1687545040

invertibleAndInverses = {'Invertible': [], 'Inverse': []}
for i in range(mod):
    if math.gcd(i, mod) == 1:
        #A number is invertible if the gcd of the number and the modulo is 1.
        invertibleAndInverses['Invertible'].append(i)
        for j in range(mod):
            if (i * j) % mod == 1:
                #A number's inverse occurs when the number is multiplied by a
number
                #, % 26, and it equals 1.
                invertibleAndInverses['Inverse'].append(j)
                break

# print(f'Invertibles: {invertibleAndInverses["Invertible"]}')
# print(f'Inverses:    {invertibleAndInverses["Inverse"]}')
# print(f'Number n:    {len(invertibleAndInverses["Invertible"])}')

#Uncomment to write to a file if output is too long.
fileout = open('./output.txt', 'w')
fileout.write(f'Modulo: {mod}\n\n')
fileout.write('%10s' % 'Invertibles')
fileout.write('%10s\n' % 'Inverses')
for i in range(len(invertibleAndInverses['Invertible'])):
    fileout.write('%10d' % invertibleAndInverses['Invertible'][i])
    fileout.write('%10d\n' % invertibleAndInverses['Inverse'][i])
```