Erikson Sodergren CS462-01

1.
    a. n=33, Φ(33)=20, $e=7^{-1}\text{mod}20=3$, $y=5^{3}\text{mod}33=26$, $x=26^{7}\text{mod}33=5$
    b. n=55, Φ(55)=40, $d=3^{-1}\text{mod}40=27$, $y=9^{27}\text{mod}55=4$, $x=8^{3}\text{mod}55=9$
2. p=11, q=13, n=143, $c_p=6$, $c_q=6$
    a. (1, 0): [13*6]1 + [11*6]0 mod143 = 78mod143
    b. (4, 5): [13*6]4 + [11*6]5 mod143 = [78]4+[66]5 = 312+330mod143=70mod143
    c. (5, 4): [13*6]5 + [11*6]4 mod143 = [78]5+[66]4 = 390+264mod143=82mod143
3. p=7, q=11, n=77
    a. $x^2=1\text{mod}77$
    b. $x^2=1\text{mod}7$, -1 = 6mod7
    c. $x^2=1\text{mod}11$, -1 = 10mod11
    d. 34=6+4(7)=(-1, 1), 43=10+3(11)=(1, -1)
4. My fermats test occasionally(1 out of my 30ish tests) accidentally lets a non-carmichaels composite number out, but im accepting that rare mistake that only happens anyway due to the weakening of the function required to return every carmichael on purpose, as solving it seems computationally expensive.
    a. 10^6: [997633, 852841, 838201, 825265, 748657]
    b. 10^7: [9890881, 9613297, 9585541, 9582145, 9494101]

```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 10 18:01:56 2019

@author: Erikson
"""
from random import sample
from math import sqrt, gcd, floor
import time

def sqAndMul(base, exp, mod):
    result = 1
    binlist = [int(x) for x in '{:b}'.format(exp)]
    for x in binlist:
        result**=2
        result %=mod
        if x is 1:
            result *= base
            result %=mod
    return result

def isPrime(p):
    for x in range(2, floor(sqrt(p))):
        if p % x == 0:
            return False
    return True

def fermat(p, s):
    subset = sample(range(2, p-1), s)
    passedGcd = False
    for a in subset:
        if gcd(a, p) is 1:
            passedGcd = True
            if sqAndMul(a, p-1, p) is not 1:
                return False
    if passedGcd:#catch cases where all s values for a were thrown out, often
        return True#due to p being even
    else:
        return False

def isCarmichael(C):
    for a in range(C):
        if gcd(a, C)==1:
```

```
            if sqAndMul(a, C-1, C) is not 1:
                return False
    return True

def main():
    start_time = time.time()
    start = 10**7
    security = 15
    carmichaels = []
    while(True):
        if(fermat(start, security)):#likely prime
            if(not isPrime(start)):#not prime
                #if(isCarmichael(start)):#is car
                carmichaels.append(start)
                if(len(carmichaels)==5):
                    break
        start-=1
        if(start < security+3):
            break
    print(carmichaels)
    print("--- %s seconds total" % (time.time() - start_time))


if __name__ == "__main__":
    main()
```