Caesar Cipher

Darin Critchlow
CSIS 2430-001

Objective:

Implement a program that will take ANY such formula for the Caesar Cipher. You will use\need this for the first Midterm exam.

What Worked:

Everything worked properly

What Didn't Work:

I had to make sure that I was always working with uppercase strings

Comments:

It was definitely more fun to implement the affine cipher version.

Code:

```
#!/usr/bin/env python

def gcd(a, b):
    x, y = a, b
    while y != 0:
        r = x % y
        x = y
        y = r
    return x

def int_from_char(c):
    return ord(c) - 65

def char_from_int(i):
    return chr(65 + i)

def encipher(a, b, x, mod):
    return (a * x + b) % mod

def inv(a, mod):
    return next(i for i in xrange(1, 26) if (i * a) % mod == 1)

def decipher(a, b, x, mod):
    return (inv(a, mod) * (x - b)) % mod

def encrypt(plaintext, a, b):
    return ’’.join(char_from_int(encipher(
        a, b, int_from_char(letter), mod)) for letter in plaintext)

def decrypt(ciphertext, a, b):
    return ’’.join(char_from_int(decipher(
        a, b, int_from_char(letter), mod)) for letter in ciphertext)

if __name__ == "__main__":
    print ’Caser Cipher with "affine cipher" f (p) = (ap + b) mod 26’
    plaintext = raw_input(
        ’\nEnter plain text\n’).replace(" ","").upper()
    a = int(raw_input(’\nEnter "a" eg. "(ap + b)"\n’))
    b = int(raw_input(’\nEnter "b" eg. "(ap + b)"\n’))
    mod = int(raw_input(
        ’\nEnter "mod" value for your language. eg "26" for English\n’))
    if gcd(a,b) == 1:
        ciphertext = encrypt(plaintext, a, b)
        decryptedtext = decrypt(ciphertext, a, b)
        print ’\nPlain text:’,plaintext
        print ’\nEncrypted text:’,ciphertext
        print ’\nDecrypted text:’,decryptedtext
    else:
        print ’GCD of "a" and "b" must equal 1, try again’
```