```
#Evaluate the following

# a.
x=7503
y=81
print (x % y)
```

```
51
```

```
# b.
x=-7503
y=81
print (x % y)
```

```
30
```

```
#c.
x=81
y=7503
print (x % y)
```

```
81
```

```
#d
x=-81
y=7503
print (x % y)
```

```
7422
```

```
#### Exercie 2.5
###Use exhaustive key search to decrypt the following ciphertext, which was encrypted using a Shift Cipher:
# BEEAKFYDJXUQYHYJIQRYHTYJIQFBQDUYJIIKFUHCQD.

# Let's define the characters that we have in our alphabets and assumption is we are first changing into lowercase
letters = "abcdefghijklmnopqrstuvwxyz"
# Initialize variable for encrypted text
ciphertext = "BEEAKFYDJXUQYHYJIQRYHTYJIQFBQDUYJIIKFUHCQD"
# Initialize integer for loop execution
x = 0
# Enter in the loop
for a in range(1,26):
  x = x+1
  to_decrypt=ciphertext
  alphabets = letters.upper()
  to_decrypt=to_decrypt.upper()
  shift=int(x)
  decrypted=""
  for character in to_decrypt:
      position = alphabets.find(character)
      newposition = position-shift
      if character in alphabets:
          decrypted = decrypted + alphabets[newposition]
      else:
          decrypted = decrypted + character
  shift =str(shift)
  print(" Key="+shift )
  print("Plain text:")
  print(decrypted)
  print("#######################################\n")
```

```
 Key=1
Plain text:
ADDZJEXCIWTPXGXIHPQXGSXIHPEAPCTXIHHJETGBPC
#######################################

 Key=2
Plain text:
ZCCYIDWBHVSOWFWHGOPWFRWHGODZOBSWHGGIDSFAOB
#######################################
```

```
 Key=3
Plain text:
YBBXHCVAGURNVEVGFNOVEQVGFNCYNARVGFFHCREZNA
#######################################

 Key=4
Plain text:
XAAWGBUZFTQMUDUFEMNUDPUFEMBXMZQUFEEGBQDYMZ
#######################################

 Key=5
Plain text:
WZZVFATYESPLTCTEDLMTCOTEDLAWLYPTEDDFAPCXLY
#######################################

 Key=6
Plain text:
VYYUEZSXDROKSBSDCKLSBNSDCKZVKXOSDCCEZOBWKX
#######################################

 Key=7
Plain text:
UXXTDYRWCQNJRARCBJKRAMRCBJYUJWNRCBBDYNAVJW
#######################################

 Key=8
Plain text:
TWWSCXQVBPMIQZQBAIJQZLQBAIXTIVMQBAACXMZUIV
#######################################

 Key=9
Plain text:
SVVRBWPUAOLHPYPAZHIPYKPAZHWSHULPAZZBWLYTHU
#######################################

 Key=10
Plain text:
RUUQAVOTZNKGOXOZYGHOXJOZYGVRGTKOZYYAVKXSGT
#######################################

 Key=11
Plain text:
QTTPZUNSYMJFNWNYXFGNWINYXFUQFSJNYXXZUJWRFS
#######################################

 Key=12
Plain text:
```

```
####### 2.8 List all the invertible elements in Zm for m = 28, 33, and 35.

# When m=28
import math
m=28
for a in range (1,m):
  gcd =math.gcd(a,m)  # we are checking if there is common divisor or not
  if gcd==1:
    print(a)
```

```
1
3
5
9
11
13
15
17
19
23
25
27
```

```
# When m=33
import math
m=33
for a in range (1,m):
  gcd =math.gcd(a,m)
  if gcd==1:
    print(a)
```

```
1
2
```

```
4
5
7
8
10
13
14
16
17
19
20
23
25
26
28
29
31
32
```

```
# When m=35
import math
m=35
for a in range (1,m):
  gcd =math.gcd(a,m)
  if gcd==1:
    print(a)
```

```
1
2
3
4
6
8
9
11
12
13
16
17
18
19
22
23
24
26
27
29
31
32
33
34
```

```
# 2.9 For 1 ≤ a ≤ 28, determine a -1 mod 29 by trial and error

# we can find this by finding out relatively prime

m = 29
for a in range(1, 29):
    i = 1
    while (a * i) % m != 1:  # We are checking if the reminder is 1 which means if it is perfectly divisible or not
        i += 1
    print(" When a = " +str(a) +" value is "+ str(i))
```

```
 When a = 1 value is 1
 When a = 2 value is 15
 When a = 3 value is 10
 When a = 4 value is 22
 When a = 5 value is 6
 When a = 6 value is 5
 When a = 7 value is 25
 When a = 8 value is 11
 When a = 9 value is 13
 When a = 10 value is 3
 When a = 11 value is 8
 When a = 12 value is 17
 When a = 13 value is 9
 When a = 14 value is 27
 When a = 15 value is 2
 When a = 16 value is 20
```

```
When a = 17 value is 12
When a = 18 value is 21
When a = 19 value is 26
When a = 20 value is 16
When a = 21 value is 18
When a = 22 value is 4
When a = 23 value is 24
When a = 24 value is 23
When a = 25 value is 7
When a = 26 value is 19
When a = 27 value is 14
When a = 28 value is 28
```

2.3 Prove that a mod m = b mod m if and only if a ≡ b (mod m).

Here we need to prove that a mod m = b mod m

Given, a ≡ b (mod m) [congruency] or, m | (a-b) or, (a - b) = m (x1-x2) [x1 and x2 are integers] or, a - b = x1m - x2m
or, a = x1m + r and b = x2m + r [they will have common reminder. Adding common reminder r] which implies, a mod m = b mod m

Proved//