**Practical : 9**

**Write a function to find out multiplicative inverse using Extended Euclidean algorithm and perform encryption and decryption using affine cipher algorithm.**

* **Extended Euclidean Algorithm**
* **CODE :-**

import math

def extendedeuclidian(r1,r2):

q=0

r=0

t1=0

t2=1

s1=1

s2=0

s=0

t=0

i=0

while r2!=0:

if i>0:

r1=r2

r2=r

if r2==0:

q=0

break

q=math.floor(r1/r2)

r=r1-(q\*r2)

if i>0:

t1=t2

t2=t

s1=s2

s2=s

t=t1-(q\*t2)

s=s1=(q\*s2)

i+=1

t1=t2

t2=t

s1=s2

s2=s

return r1,t1

a=int(input("Enter first value : "))

b=int(input("Enter second value : "))

r1=max(a,b)

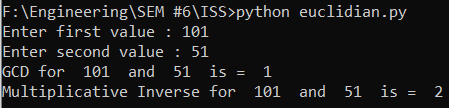
r2=min(a,b)

gcd,inverse=extendedeuclidian(r1,r2)

print("GCD for ",r1," and ",r2," is = ",gcd)

print("Multiplicative Inverse for ",r1," and ",r2," is = ",inverse)

* **OUTPUT :-**

****

* **Affine Cipher**
* **CODE :-**

import math

pt=list(input("Enter plaintext : "))

a=int(input("Enter value of a = "))

b=int(input("Enter value of b = "))

def encryption(pt,a,b):

en=[]

i=0

while i<len(pt):

k=0

if pt[i].isupper():

k=65

else:

k=97

x=ord(pt[i])-k

fox=chr((((a\*x)+b)%26)+k)

en.append(fox)

i+=1

return en

def euclidian\_inverse(r1,r2):

q=0

r=0

t1=0

t2=1

t=0

i=0

while r2!=0:

if i>0:

r1=r2

r2=r

if r2==0:

q=0

break

q=math.floor(r1/r2)

r=r1-(q\*r2)

if i>0:

t1=t2

t2=t

t=t1-(q\*t2)

i+=1

t1=t2

t2=t

inverse=t1%26

return inverse

def decryption(en,c,b):

de=[]

i=0

while i<len(en):

k=0

if en[i].isupper():

k=65

else:

k=97

x=ord(en[i])-k

fox=chr(((c\*(x-b))%26)+k)

de.append(fox)

i+=1

return de

en=encryption(pt,a,b)

#print(en)

print("Encrypted text is : ")

i=0

while i<len(en):

print(en[i],end="")

i+=1

print()

c=euclidian\_inverse(26,a)

de=decryption(en,c,b)

print("Decrypted text is : ")

i=0

while i<len(de):

print(de[i],end="")

i+=1

* **OUTPUT :-**

