**ADITHYA KRISHNA.R**

**AM.SC.P2CSC19005**

**Cryptography and Network Security**

**ASSIGNMENT**

**PART A**

**2. Find the multiplicative inverse of all the elements in Z5 and Z11**

**Ans:** Multiplicative Inverse of:

Z5->

a -> 1 2 3 4

a-1 -> 1 3 2 4

Z11->

a 1 2 3 4 5 6 7 8 9 10

a-1 1 6 4 3 9 2 8 7 5 10

**3. find the gcd of 56245 and 43159**

**Ans:** 56245=43159\*1+13086

43159=13086\*3+3901

13086=3901\*3+1383

3901=1383\*2+1135

1383=1135\*1+248

1135=248\*4+143

248=143\*1+105

143=105\*1+38

105=38\*2+29

38=29\*1+9

29=9\*3+2

9=2\*4+1

2=**1**\*2+0

gcd(56245,43159)=1.

**4. Compute phi(n) for 34 and 210**

**Ans:** According to Euler’s product formula

Phi(34)=34\*(1-(1/3))

=81\*2/3

=54.

Phi(210)=210\*(1-(1/2))

=1024\*1/2

=512.

**5. Compute 3100 mod(31319)**

**Ans:** Here e=100 =>26+25+22

30 mod 31319=3

32 mod 31319=9

34 mod 31319=81

38 mod 31319=6561

316 mod 31319=14418

332 mod 31319=21979

364 mod 31319=12185

3100 mod(31319)=12185\*21979\*81 mod 31319

=5346\*81 mod 31319 = 25879.

**PART B**

**3. RSA ALGORITHM**

#include<iostream>

#include<math.h>

using namespace std;

int gcd(int a, int b) {

int t;

while(1) {

t= a%b;

if(t==0)

return b;

a = b;

b= t;

}

}

int main() {

double p = 13;

double q = 11;

double n=p\*q;

double tr;

double phi= (p-1)\*(q-1);

double e=7;

while(e<phi) {

tr = gcd(e,phi);

if(tr==1)

break;

else

e++;

}

double d1=1/e;

double d=fmod(d1,phi);

double message = 9;

double c = pow(message,e);

double m = pow(c,d);

c=fmod(c,n);

m=fmod(m,n);

cout<<"Original Message = "<<message;

cout<<"\n"<<"p = "<<p;

cout<<"\n"<<"q = "<<q;

cout<<"\n"<<"n = pq = "<<n;

cout<<"\n"<<"phi = "<<phi;

cout<<"\n"<<"e = "<<e;

cout<<"\n"<<"d = "<<d;

cout<<"\n"<<"Encrypted message = "<<c;

cout<<"\n"<<"Decrypted message = "<<m;

return 0;

}