**Code to Demonstrate RSA algorithm**

#include <bits/stdc++.h>

using namespace std;

// Returns gcd of a and b

int gcd(int a, int h)

{

int temp;

while (1) {

temp = a % h;

if (temp == 0)

return h;

a = h;

h = temp;

}

}

// Code to demonstrate RSA algorithm

int main()

{

// Two random prime numbers

double p = 3;

double q = 7;

// First part of public key:

double n = p \* q;

// Finding other part of public key.

// e stands for encrypt

double e = 2;

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

while (e < phi) {

// e must be co-prime to phi and

// smaller than phi.

if (gcd(e, phi) == 1)

break;

else

e++;

}

// Private key (d stands for decrypt)

// choosing d such that it satisfies

// d\*e = 1 + k \* totient

int k = 2; // A constant value

double d = (1 + (k \* phi)) / e;

// Message to be encrypted

double msg = 12;

printf("Message data = %lf", msg);

// Encryption c = (msg ^ e) % n

double c = pow(msg, e);

c = fmod(c, n);

printf("\nEncrypted data = %lf", c);

// Decryption m = (c ^ d) % n

double m = pow(c, d);

m = fmod(m, n);

printf("\nOriginal Message Sent = %lf", m);

return 0;

}

**Output**

