

#include <iostream>

#include <string>

using namespace std;

// проверка на простое число

bool isPrime(long int prime)

{

long int i, j;

j = (long int)sqrt((long double)prime);

for (i = 2; i <= j; i++)

{

if (prime % i == 0)

{

return false;

}

}

return true;

}

//взаимно простые

long int greatestCommonDivisor(long int e, long int t)

{

while (e > 0)

{

long int myTemp;

myTemp = e;

e = t % e;

t = myTemp;

}

return t;

}

long int calculateE(long int t)

{

// Выбирается целое число e ( 1 < e < t ) // взаимно простое со значением функции Эйлера (t)

long int e;

for (e = 2; e < t; e++)

{

if (greatestCommonDivisor(e, t) == 1)

{

return e;

}

}

return -1;

}

long int calculateD(long int e, long int t)

{

long int d;

long int k = 1;

while (1)

{

k = k + t;

if (k % e == 0)

{

d = (k / e);

return d;

}

}

}

long int encrypt(long int i, long int e, long int n)

{

long int current, result;

current = i - 97;

result = 1;

for (long int j = 0; j < e; j++)

{

result = result \* current;

result = result % n;

}

return result;

}

long int decrypt(long int i, long int d, long int n)

{

long int current, result;

current = i;

result = 1;

for (long int j = 0; j < d; j++)

{

result = result \* current;

result = result % n;

}

return result +97;

}

int main()

{

long int p, q, n, t, e, d;

long int encryptedText[100];

long int decryptedText[100];

bool flag;

string msg;

// Выбираются два различных случайных простых числа p и q заданного размера

do

{

cout << "Enter a Prime number p :" << endl;

cin >> p;

flag = isPrime(p);

if (flag == false)

{

cout << "\nWRONG INPUT (This number is not Prime. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself)\n" << endl;

}

} while (flag == false);

do

{

cout << "Enter a Prime number q :" << endl;

cin >> q;

flag = isPrime(q);

if (flag == false)

{

cout << "\nWRONG INPUT (This number is not Prime. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself)\n" << endl;

}

} while (flag == false);

n = p \* q;

cout << "\nResult of computing n = p\*q = " << n << endl;

//Вычисляется значение функции Эйлера от числа n

t = (p - 1) \* (q - 1);

cout << "Result of computing Euler's totient function:\t t = " << t << endl;

e = calculateE(t);

d = calculateD(e, t);

cout << "\nRSA public key is (n = " << n << ", e = " << e << ")" << endl;

cout << "RSA private key is (n = " << n << ", d = " << d << ")" << endl;

cout << "\nEnter Message to be encryped:" << endl;

cin.ignore();

getline(cin, msg);

cout << "\nThe message is: " << msg << endl;

// encryption

for (long int i = 0; i < msg.length(); i++)

{

encryptedText[i] = encrypt(msg[i], e, n);

}

cout << "\nTHE ENCRYPTED MESSAGE IS:" << endl;

for (long int i = 0; i < msg.length(); i++)

{

printf("%c", (char)encryptedText[i]);

}

//decryption

for (long int i = 0; i < msg.length(); i++)

{

decryptedText[i] = decrypt(encryptedText[i], d, n);

}

cout << "\n\nTHE DECRYPTED MESSAGE IS:" << endl;

for (long int i = 0; i < msg.length(); i++)

{

printf("%c", (char)decryptedText[i]);

}

cout << endl << endl;

return 0;

}