EXPERIMENT-7

AIM: To implement RSA encryption-decryption.

THEORY:

The RSA algorithm is an asymmetric cryptographic technique that uses a pair of keys: a public key for encryption and a private key for decryption. It relies on the mathematical properties of large prime numbers and modular arithmetic.

Two large primes, p and q, are chosen, and their product n = p×q is computed. The value of Euler’s Totient function is then calculated as ϕ(n)=(p−1)(q−1). Next, a public exponent eee is selected such that eee is coprime with ϕ(n). The private key d is then determined as the modular inverse of e modulo ϕ(n).

Encryption:
C = M^e mod n

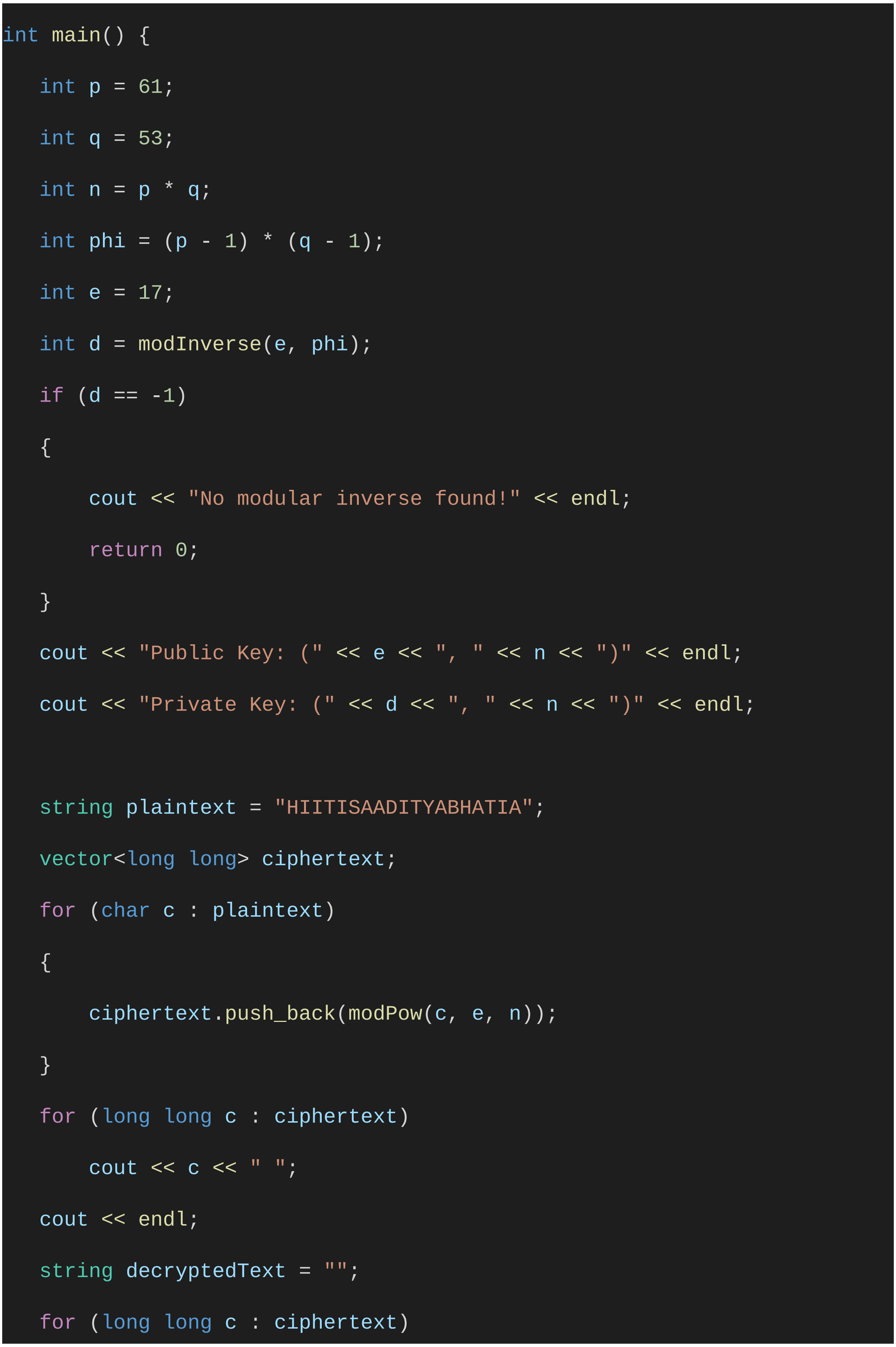
Decryption:
M = C^d mod n

This process ensures secure data exchange, as only the private key can correctly decrypt the ciphertext generated using the public key.

CODE:

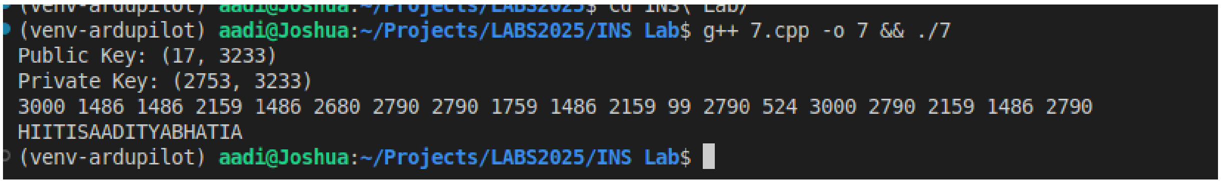
```
// RSA Encryption and Decryption in C++ by Aaditya Bhatia 23/CS/004


#include <iostream>

#include <vector>

using namespace std;
```

```
int gcd(int a, int b) {
   return b == 0 ? a : gcd(a, b % a);
}

long long modPow(long long base, long long exp, long long mod) {
   long long result = 1;
   base %= mod;
   while (exp > 0)
   {
       if (exp % 2 == 1)
           result = (result * base) % mod;
       base = (base * base) % mod;
       exp /= 2;
   }
   return result;
}

int modInverse(int e, int phi) {
   for (int d = 2; d < phi; d++)
   {
       if ((e * d) % phi == 1)
           return d;
   }
   return -1;
}
```



```
int main() {
    int p = 61;
    int q = 53;
    int n = p * q;
    int phi = (p - 1) * (q - 1);
    int e = 17;
    int d = modInverse(e, phi);
    if (d == -1)
    {
        cout << "No modular inverse found!" << endl;
        return 0;
    }
    cout << "Public Key: (" << e << ", " << n << ")" << endl;
    cout << "Private Key: (" << d << ", " << n << ")" << endl;

    string plaintext = "HIITISAADITYABHATIA";
    vector<long long> ciphertext;
    for (char c : plaintext)
    {
        ciphertext.push_back(modPow(c, e, n));
    }
    for (long long c : ciphertext)
        cout << c << " ";
    cout << endl;
    string decryptedText = "";
    for (long long c : ciphertext)
    {
        decryptedText += (char)modPow(c, d, n);
    }
    cout << decryptedText << endl;
    return 0;
}
```

OUTPUT:



```
(venv-ardupilot) aadi@Joshua:~/Projects/LABS2025$ cd INS\ Lab/
(venv-ardupilot) aadi@Joshua:~/Projects/LABS2025/INS Lab$ g++ 7.cpp -o 7 && ./7
Public Key: (17, 3233)
Private Key: (2753, 3233)
3000 1486 1486 2159 1486 2680 2790 2790 1759 1486 2159 99 2790 524 3000 2790 2159 1486 2790
HIITISAADITYABHATIA
(venv-ardupilot) aadi@Joshua:~/Projects/LABS2025/INS Lab$
```

LEARNING OUTCOME:

To understand and implement RSA encryption and decryption using modular arithmetic and key pair generation for secure communication.