Roll No: 20BCE204
Course Code and Course Name:
Practical No. 1
Aim: To implement digital signature to sign and verify authenticated user. Also, show a message when tampering is detected.

Code:

```
#include <bits/stdc++.h>
using namespace std;

bool isPrime(long long n){
    if(n <= 1){
        return false;
    }
    for(long long i = 2; i * i <= n; i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}

long long generatePrimeNumber(){
    srand(time(NULL));
    long long p = rand() % 2048;
    while(!isPrime(p)){
        p = rand() % 2048;
    }
    return p;
}

long long generatePublicKey(long long phi_n){
    srand(time(NULL));
    long long e = rand() % phi_n + 1;
    while(__algo_gcd(e, phi_n) != 1) {
        e = rand() % phi_n + 1;
    }
    return e;
}

long long binpow(long long a, long long b, long long m) {
    a = a % m;
    long long res = 1;
    while(b > 0){
        if (b & 1){
```

```
            res = res * a % m;
        }
        a = a * a % m;
        b = b/2;
    }
    return res;
}

void RSA_encrypt_decrypt() {
    long long p = generatePrimeNumber();
    long long q = generatePrimeNumber();
    while(p == q){
        q = generatePrimeNumber();
    }
    long long n = p * q;
    long long phi_n = (p - 1) * (q - 1);
    long long e = generatePublicKey(phi_n);
    long long d, i = 1;
    while((phi_n * i + 1) % e != 0){
        i++;
    }
    d = (phi_n * i + 1) / e;

    cout << "P: " << p << endl;
    cout << "Q: " << q << endl;
    cout << "Public Key: " << e << " " << n << endl;
    cout << "Private Key: " << d << " " << n << endl;

    string mssg = "Dhyan";
    vector<int> pt;
    vector<int> ct;
    cout << "Message: " << mssg << endl;
    for(auto it : mssg){
        pt.push_back(it - 'A');
    }
    for(auto it : pt){
        long long enc = binpow(it, e, n);
        ct.push_back(enc);
    }

    cout << "ENCRYPTED: ";
    for(auto it : ct){
        cout << it << " ";
    }
```

```
    cout << endl;

    string decrypted_text = "";
    for(auto it : ct){
        long long enc = binpow(it, d, n);
        decrypted_text = decrypted_text + (char)(enc + 'A');
    }
    cout << "DECRYPTED: " << decrypted_text << endl;
}

int main() {
    RSA_encrypt_decrypt();
    return 0;
}
```

Output:
P: 1693
Q: 491
Public Key: 37801 831263
Private Key: 561961 831263
Message: Dhyan
ENCRYPTED: 557790 451210 724281 24372 623888
DECRYPTED: Dhyan