```cpp
// C++ PROGRAM FOR EL-GAMAL CRYPTOGRAPHIC SYSTEM
// CODED BY PRAJWAL SUNDAR, CSE-B, THIRD YEAR, ROLL NO 106121092
#include "bits/stdc++.h"
using namespace std;
int inverse(int R2, int R1) // Inverse using Extended Euclidean Algorithm
{
    int D = R1, T1 = 0, T2 = 1, Q, R, T;
    while (R2) // Run algorithm as long as R2 != 0
    {
        Q = R1 / R2; R = R1 % R2;
        T = T1 - Q*T2;
        R1 = R2; R2 = R;
        T1 = T2; T2 = T;
    }
    if (R1 != 1) return 0; // HCF != 1, so inverse does not exist
    else if (T1 > 0) return T1 % D; // positive remainder
    else return D - (-T1 % D); // convert negative remainder to positive
remainder
}
int mod(int b, int e, int q) // Modular Exponentiation
{
    vector<int> V = {b}; // store results of powers of 2
    int r = b;
    while (e >= pow(2, V.size()))
    {
        r = (r*r) % q; // repeated squaring and taking modulus
        V.push_back(r);
    }
    int p = V.size()-1, n = e; r = 1;
    while (p >= 0)
    {
        if (n >= pow(2, p)) // consider the current power
        {
            n -= pow(2, p);
            r = (r*V[p]) % q; // consider remainder
        }
        p--;
    }
    return r;
}
int main() // Main Function
{
    cout << "Welcome to C++ ElGamal Cryptosystem !" << endl << endl;

    int q, a; // global public elements
    cout << "Enter the global prime number 'q' : ";
    cin >> q;
    cout << "Enter the primitive root of q, 'a' : ";
```

```
    cin >> a;
    int M; // plain text
    cout << "Enter your plain text (less than q) : ";
    cin >> M;

    // Key Generation
    int XA;
    cout << "Enter your choice of private key (less than q-1) : ";
    cin >> XA;
    int YA = mod(a, XA, q); // public key

    // Encryption using Public Key
    int k; // random number
    cout << "Enter your choice of 'k' (less than q) : ";
    cin >> k;
    int K = mod(YA, k, q); // one-time-key
    int C1 = mod(a, k, q); // cipher-text-part-1
    int C2 = (K*M) % q; // cipher-text-part-2
    cout << endl << "Cipher Text = (" << C1 << "," << C2 << ")";

    // Decryption using Private Key
    int KD = mod(C1, XA, q); // re-generate one-time-key
    int PT = (C2 * inverse(K, q)) % q; // plain-text
    cout << endl << "Regenerated Plain Text = " << PT;

    cout << endl << endl << "Thank you for using C++ ElGamal Cryptosystem. Bye
Bye !";
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

prajw@Prajwal_DELL MINGW64 ~/OneDrive/Desktop/NITT/Sem6/CSPC63 - Crypto
$ g++ elgamal.cpp -o elgamal

prajw@Prajwal_DELL MINGW64 ~/OneDrive/Desktop/NITT/Sem6/CSPC63 - Crypto
$ ./elgamal.exe
Welcome to C++ ElGamal Cryptosystem !

Enter the global prime number 'q' : 19
Enter the primitive root of q, 'a' : 10
Enter your plain text (less than q) : 17
Enter your choice of private key (less than q-1) : 5
Enter your choice of 'k' (less than q) : 6

Cipher Text = (11,5)
Regenerated Plain Text = 17

Thank you for using C++ ElGamal Cryptosystem. Bye Bye !
prajw@Prajwal_DELL MINGW64 ~/OneDrive/Desktop/NITT/Sem6/CSPC63 - Crypto
$
```

```cpp
// C++ PROGRAM FOR RC4 CRYPTOGRAPHIC SYSTEM
// CODED BY PRAJWAL SUNDAR, CSE-B, THIRD YEAR, ROLL NO 106121092

#include "bits/stdc++.h"
using namespace std;

int main()
{
    cout << "Welcome to C++ RC4 Cyrptosystem Key Stream Generator !" << endl
<< endl;

    int n; // size of state array
    cout << "Enter the size of your state array : ";
    cin >> n;
    vector<int> S (n); // State-Array
    for (int i = 0; i < n; i++) S[i] = i;

    int k; // size of key array
    cout << "Enter the size of your key array : ";
    cin >> k;
    vector<int> K (k); // Key-Array
    cout << "Enter the elements of your key array : ";
    for (int i = 0; i < k; i++) cin >> K[i];

    vector<int> T (n); // T-array
    for (int i = 0; i < n; i++) T[i] = K[i%k];

    // Key-Generation
    int i, j = 0;
    for (i = 0; i < n; i++)
    {
        j = (j + S[i] + T[i]) % n;
        swap(S[i], S[j]);
    }

    // Stream-Generation
    int l;
    cout << "Enter the length of your desired key-stream : ";
    cin >> l;
    i = j = 0;
    vector<int> keys (l);
    for (int x = 0; x < l; x++)
    {
        i = (i+1) % n;
        j = (j+S[i]) % n;
        swap(S[i], S[j]);
        int t = (S[i] + S[j]) % n;
        keys[x] = S[t];
```

```
    }

    cout << "The Generated Key Stream is : ";
    for (int & key : keys) cout << key << " ";

    cout << endl << endl << "Thank you for using C++ RC4 Cyrptosystem Key
Stream Generator. Bye Bye !";
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                Python: des

prajw@Prajwal_DELL MINGW64 ~/OneDrive/Desktop/NITT/Sem6/CSPC63 - Crypto
$ g++ rc4.cpp -o rc4

prajw@Prajwal_DELL MINGW64 ~/OneDrive/Desktop/NITT/Sem6/CSPC63 - Crypto
$ ./rc4.exe
Welcome to C++ RC4 Cyrptosystem Key Stream Generator !

Enter the size of your state array : 256
Enter the size of your key array : 6
Enter the elements of your key array : 1 2 3 4 5 6
Enter the length of your desired key-stream : 20
The Generated Key Stream is : 134 185 98 8 122 95 17 197 127 192 23 76 92 132 216 138 219 7 63 186

Thank you for using C++ RC4 Cyrptosystem Key Stream Generator. Bye Bye !
prajw@Prajwal_DELL MINGW64 ~/OneDrive/Desktop/NITT/Sem6/CSPC63 - Crypto
$
```