**ASSIGNMENT NO: 8**

**NAME: Paras Wane**

**PRN : 21510041**

**Q - Implement the Diffie-Hellman Key Exchange algorithm for a given problem.**

#include<bits/stdc++.h>

using namespace std;

// Power function to return value of a ^ b mod P

long long int power(long long int a, long long int b,

                    long long int P)

{

    if (b == 1)

        return a;

    else

        return (((long long int)pow(a, b)) % P);

}

// Driver program

int main()

{

    long long int P, G, x, a, y, b, ka, kb;

    // Both the persons will be agreed upon the

    // public keys G and P

    P = 23; // A prime number P is taken

    cout << "The value of P : " << P << endl;

    G = 9; // A primitive root for P, G is taken

    cout << "The value of G : " << G << endl;

    // Alice will choose the private key a

    a = 4; // a is the chosen private key

    cout << "The private key a for Alice : " << a << endl;

    x = power(G, a, P); // gets the generated key

    // Bob will choose the private key b

    b = 3; // b is the chosen private key

    cout << "The private key b for Bob : " << b << endl;

    y = power(G, b, P); // gets the generated key

    // Generating the secret key after the exchange

    // of keys

    ka = power(y, a, P); // Secret key for Alice

    kb = power(x, b, P); // Secret key for Bob

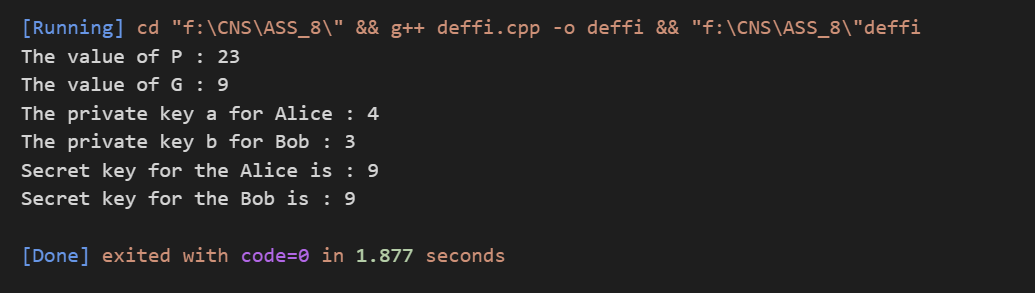
    cout << "Secret key for the Alice is : " << ka << endl;

    cout << "Secret key for the Bob is : " << kb << endl;

    return 0;

}

**OUTPUT :**

****