**Diffie-Hellman Algorithm**

**Code:**

#include <cmath>

#include <iostream>

using namespace std;

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);

}

int main()

{

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

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;

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

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

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:

The value of P : 23

The value of G : 9

The private key a for Alice : 4

The private key b for Bob : 3

Secret key for the Alice is : 9

Secret Key for the Bob is : 9