**Assignment 11**

**Aim :** To Implement Diffie-Hellman Key Exchange Algorithm.

**Theory:**

The Diffie-Hellman algorithm is being used to establish a shared secret that can be used for secret communications while exchanging data over a public network using the elliptic curve to generate points and get the secret key using the parameters.

**Code:**

#include<bits/stdc++.h>

using namespace std;

long long int power(long long int x, long long int y, long long int p)

{

long long int res = 1;

x = x % p;

if (x == 0) return 0;

while (y > 0)

{

if (y & 1)

res = (res\*x) % p;

y = y>>1;

x = (x\*x) % p;

}

return res;

}

int main()

{

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

cout<<"enter a prime number: ";

cin>>P;

cout<<"enter primitive root of P: ";

cin>>G;

cout<<"The value of P : "<<P<<"\n";

cout<<"The value of G : "<<G<<"\n\n";

cout<<"enter first private key: ";

cin>>a;

cout<<"The private key a : "<<a<<"\n";

x = power(G, a, P);

cout<<"enter second private key: ";

cin>>b;

cout<<"The private key b : "<<b<<"\n\n";

y = power(G, b, P);

ka = power(y, a, P);

kb = power(x, b, P);

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

cout<<"Secret Key for the Bob is : "<<kb<<"\n\n";

return 0;

}

**Output**

