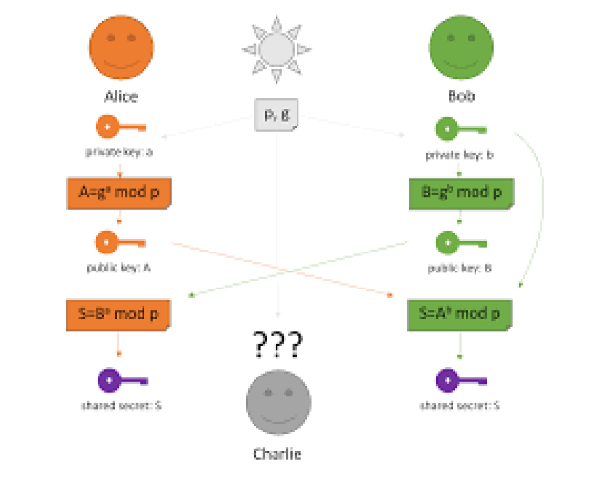
**AIM: To implement the Diffie-Hellman Key Exchange algorithm using C language.**

DESCRIPTION: Diffie–Hellman Key Exchange establishes a shared secret between two parties that can be used for secret communication for exchanging data over a public network. It is primarily used as a method of exchanging cryptography keys for use in symmetric encryption algorithms like AES. The algorithm in itself is very simple. The process begins by having the two parties, Alice and Bob. Let's assume that Alice wants to establish a shared secret with Bob.

**EXAMPLE:**

****

**ALGORITHM:**

STEP-1: Both Alice and Bob share the same public keys g and p.

STEP-2: Alice selects a random public key a.

STEP-3: Alice computes his secret key A as ga mod p.

STEP-4: Then Alice sends A to Bob.

STEP-5: Similarly, Bob also selects a public key b and computes his secret key as B and sends the same back to Alice.

STEP-6: Now both of them compute their common secret key as the other one’s secret key power of a mod p.

**PROGRAM: (Diffie Hellman Key Exchange)**

#include<stdio.h>

#include<conio.h>

long long int power(int a, int b, int mod)

{

long long int t;

if(b==1)

return a;

t=power(a,b/2,mod);

if(b%2==0)

return (t\*t)%mod;

else

return (((t\*t)%mod)\*a)%mod;

}

long int calculateKey(int a, int x, int n)

{

return power(a,x,n);

}

void main()

{

int n,g,x,a,y,b;

clrscr();

printf("Enter the value of n and g : ");

scanf("%d%d",&n,&g);

printf("Enter the value of x for the first person : ");

scanf("%d",&x);

a=power(g,x,n);

printf("Enter the value of y for the second person : ");

scanf("%d",&y);

b=power(g,y,n);

printf("key for the first person is :

%lld\n",power(b,x,n));

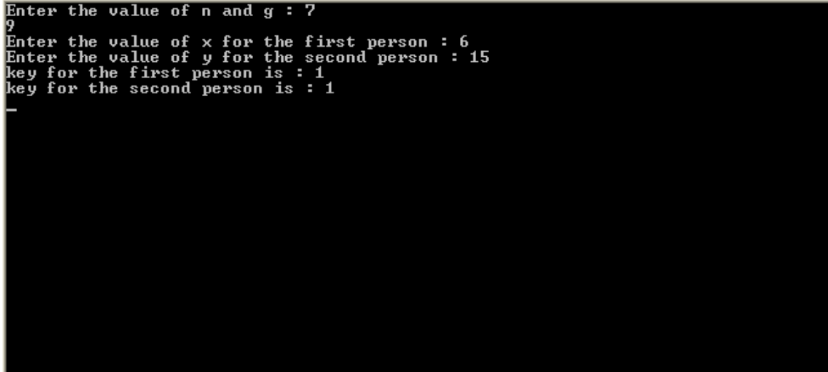
printf("key for the second person is :

%lld\n",power(a,y,n));

getch();

}

**OUTPUT:**

****