**IT NETWORK SECURITY**

**LAB WORK- DIFFIE HELLMAN EXCHANGE**

NAME: OJASVI SINGH CHAUHAN

ROLL NO. R134218140

BRANCH: BTECH CSE-CSF-B3

SAP ID: 500068818

**INDEX**

* AIM
* INPUT/OUTPUT
* THEORY
* ALGORITHM
* CODE
* OUTPUT

**EXPERIMENT – 4**

**DIFFIE HELLMAN KEY EXCHANGE TECHNIQUE**

**AIM:** To implement Diffie Hellman Key Exchange

**INPUT:**  PRIME NUMBER – Q , ALPHA- PRIMITIVE ROOT FOR Q

**OUTPUT:** KEYS FOR SENDER AND RECIEVER

**THEORY: Diffie-Hellman key exchange**is a method of securely exchanging cryptographic keys over a public channel and was one of the first [public-key protocols](https://en.wikipedia.org/wiki/Public-key_cryptography) as conceived by [Ralph Merkle](https://en.wikipedia.org/wiki/Ralph_Merkle) and named after [Whitfield Diffie](https://en.wikipedia.org/wiki/Whitfield_Diffie) and [Martin Hellman](https://en.wikipedia.org/wiki/Martin_Hellman). DH is one of the earliest practical examples of public [key exchange](https://en.wikipedia.org/wiki/Key_exchange) implemented within the field of [cryptography](https://en.wikipedia.org/wiki/Cryptography).

Traditionally, secure encrypted communication between two parties required that they first exchange keys by some secure physical means, such as paper key lists transported by a trusted [courier](https://en.wikipedia.org/wiki/Courier). The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a [shared secret](https://en.wikipedia.org/wiki/Shared_secret) key over an [insecure channel](https://en.wikipedia.org/wiki/Insecure_channel). This key can then be used to encrypt subsequent communications using a [symmetric key](https://en.wikipedia.org/wiki/Symmetric_key) [cipher](https://en.wikipedia.org/wiki/Cipher).

Diffie–Hellman is used to secure a variety of [Internet](https://en.wikipedia.org/wiki/Internet) services. However, research published in October 2015 suggests that the parameters in use for many DH Internet applications at that time are not strong enough to prevent compromise by very well-funded attackers, such as the security services of large governments.

**ALGORITHM:**

Suppose users A and B wish to exchange the key.

* 1. **k= (YA)XB mod q -> same as calculated by B**
  2. **Global Public Elements**

**q; prime number**

**α; α < q and it is primitive root of q**

* 1. **USER A KEY GENERATION**

**Select Private key  XA XA<q**

**Calculation of Public key YA YA=αXA mod q**

* 1. **USER B KEY GENERATION**

**Select Private key  XB XB<q**

**Calculation of Public key  YB YB= αXB mod q**

**Calculation of Secret Key by A**

**k=(YB)XA mod q**

* 1. **Calculation of Secret Key by B**

**k=(YA)XB mod q**

The result is that two sides have exchanged a secret value.

**CODE:**

#include<stdio.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 long int calculateKey(int a,int x,int n)

{

 return power(a,x,n);

}

int main()

{

 int n,g,x,a,y,b;

 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);

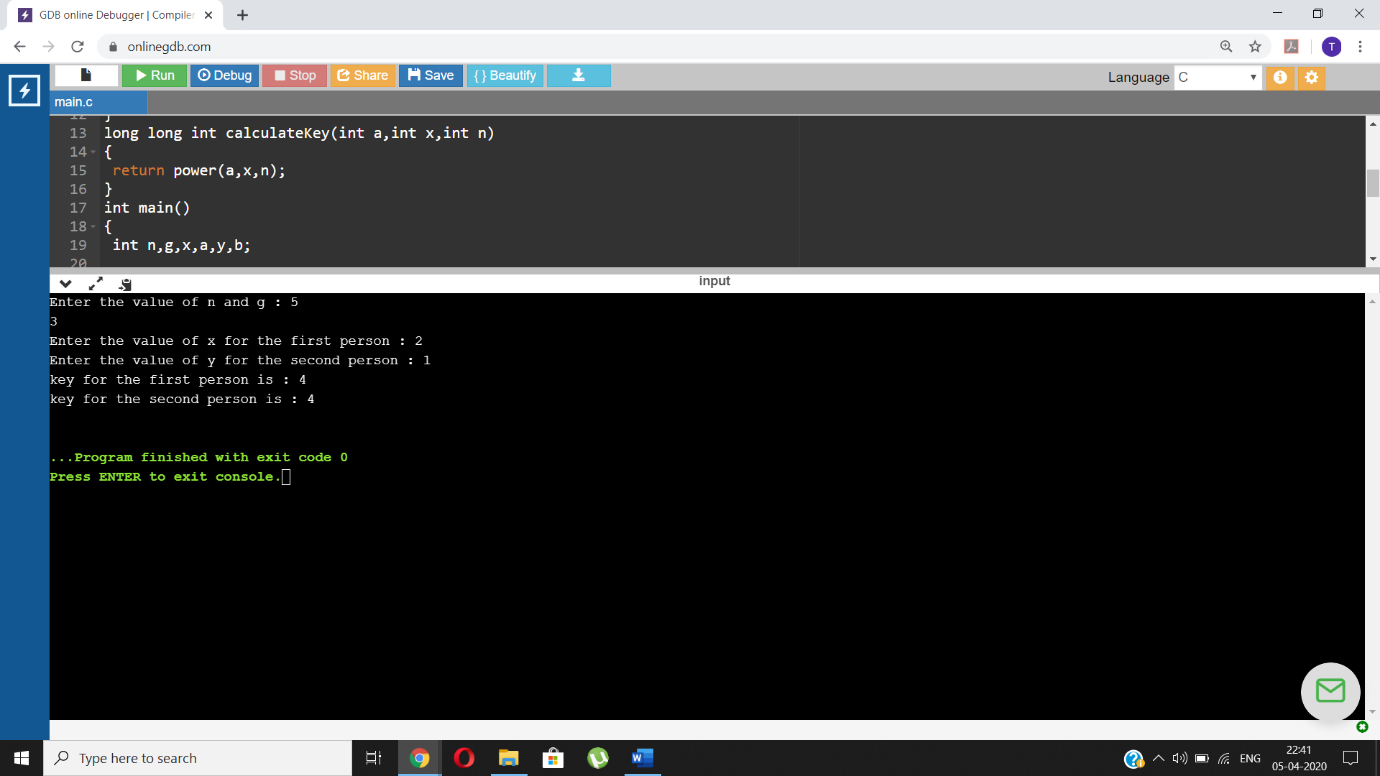
 printf("key for the first person is : %lld\n",power(b,x,n));

 printf("key for the second person is : %lld\n",power(a,y,n));

 return 0;

}

**OUTPUT SNAPSHOT:**

****