EXPNO:5 **DATE:**

DIFFIE-HELLMANKEYEXCHANGE

Aim: ToimplementDiffie-HellmankeyexchangeusingC.

Algorithm:

- Step1:ChoosealargeprimenumberPandaprimitiverootmodulo(P), denoted as (G). Both parties agree on these values.
- Step2:Alicechoosesaprivatekey(a),whileBobchoosesa privatekey(b). These private keys are kept secret.
- Step3:Alicecalculatesherpublickey(x)using(x=G^amodP),andBob calculates his public key (y) using (y = G^b mod P).
- Step4:Alicesendsherpublickey(x)toBob,andBobsendshispublickey (y) to Alice.
- Step 5: Using the received public keys,Alice computes the secret key (ka) using(ka=y^amodP), andBobcomputesthesecretkey(kb) using(kb= x^b mod P).
- Step6:BothAliceandBobnowhavethesamesharedsecretkey.
- Step7:Theycannowcommunicatesecurelyusingthesharedsecretkeyfor encryption and decryption.
- Step8:ThesecurityoftheDiffie-HellmanKeyExchangereliesonthe difficulty of calculating discrete logarithms in finite fields.

Program:

```
#include<math.h>
#include<stdio.h>
longlongintpower(longlonginta,longlongintb,longlongintP)
{
        if(b==1)
                returna;
        else
                return(((longlongint)pow(a,b))%P);
```

```
}
intmain()
{
        longlongintP,G,x,a,y,b,ka,kb; P =
        23;
        printf("ThevalueofP:%lld\n",P); G
        = 9;
 printf("The value of G : %lld\n\n", G);a = 4;
        printf("TheprivatekeyaforAlice:%lld\n",a);
x       =power(G,a,P); b
        = 3;
        printf("TheprivatekeybforBob:%lld\n\n",b); y
        = power(G, b, P);
        ka=power(y,a,P);
        kb=power(x,b,P);
        printf("Secret key for theAlice is : %lld\n", ka);
 printf("SecretKeyfortheBobis:%lld\n",kb);return0;
}
```

Output:

```
/tmp/6Ex6MzCUmw.o
The value of P : 21
The value of G : 7

The private key a for Alice : 3
The private key b for Bob : 3

Secret key for the Alice is : 7
Secret Key for the Bob is : 7


=== Code Execution Successful ===
```

Result: