Ex. No.: 2(b) **DIFFIE-HELLMAN KEY EXCHANGE** **Reg. No.: 210701296**

Date:

AIM:

To implement Diffie-Hellman key exchange using C.

ALGORITHM:

1. Get a prime number q as input from the user.
2. Get a value xa and xb which is less than q.
3. Calculate primitive root α
4. For each user A, generate a key Xa < q
5. Compute public key, α pow(Xa) mod q
6. Each user computes Ya
7. Print the values of exchanged keys.

PROGRAM CODE:

```
//This program uses fast exponentiation function power instead of pow library function
#include <stdio.h>
#include <math.h>
int power( int,unsigned int,int);
int main()
{
int x,y,z,count,ai[20][20];
int alpha,xa,xb,ya,yb,ka,kb,q;
printf("\nEnter a Prime Number \"q\":");
scanf("%d",&q);
printf("\nEnter a No \"xa\" which is less than value of q:");
scanf("%d",&xa);
printf("\nEnter a No \"xb\" which is less than value of q:");
scanf("%d",&xb);
printf("\nEnter alpha:");
scanf("%d",&alpha);
ya = power(alpha,xa,q);
yb = power(alpha,xb,q);
ka = power(yb,xa,q);
kb = power(ya,xb,q);
printf("\nya = %d \nyb = %d \nka = %d \nkb = %d \n",ya,yb,ka,kb);
if(ka == kb)
printf("\nThe secret keys generated by User A and User B are same\n");
else
printf("\nThe secret keys generated by User A and User B are not same\n");
return 0;
}
int power(int x, unsigned int y, int p)
{
int res = 1; // Initialize result
```

```
x = x % p; // Update x if it is more than or equal to p
while (y > 0)
{
// If y is odd, multiply x with result
if (y & 1)
res = (res*x) % p;
// y must be even now
y = y>>1; // y = y/2 x
= (x*x) % p;
}
return res;
}
```

OUTPUT:

```
[root@localhost-live liveuser]# vi diffie.c
[root@localhost-live liveuser]# cc diffie.c
[root@localhost-live liveuser]# ./a.out

Enter a Prime Number "q":17

Enter a No "xa" which is less than value of q:3

Enter a No "xb" which is less than value of q:5

Enter alpha:7

ya = 15
yb = 15
ka = 4
kb = 4
```

RESULT: