8. Write a program to encrypt and decrypt the data using RSA and exchange key securely using Deffie-Hellman key exchange.

// both the programs use a common function which is separately compiled

```
// exponentiation.h
#include <stdio.h>
int exponentiation (int a, int x, int n);
```

```
// exponentiation.c
#include "exponentiation.h"

int exponentiation (int a, int x, int n)
{
    int dp[1024];
    dp[0] = 1;   dp[1] = a%n;
    for (int i = 2; i<x; i++)
    {
        dp[i] = (dp[i/2] * dp[i/2]) %n;
        if (i%2)  dp[i] = (dp[i] * dp[1])%n;
    }
    if (x>=0)
        return dp[x];

    return 0;
}
```

OUTPUT

```
$ gcc -o rsa rsa.c exponentiation.o

$ ./rsa

Format: ./a.out p q x d

$ ./rsa 11 13 23 47
Enter a string of not more than 9 english alphabets
HELLO
5                    // program prints the length
Encrypted message:
2 75 110 110 27
Decrypting the same message:
HELLO

$ ./rsa 11 13 23 47
Enter a string of not more than 9 english alphabets
morning
7
Encrypted msgae
99 41 4 89 79 80 103
Decrypting the same message:
morning.

$
```

# R.V. COLLEGE OF ENGINEERING

## OBSERVATION / DATA SHEET

Date _____ Name M.C. SOHAN

Dept./Lab _____ Class _____ Expt./No. 8

Title RSA

```
// RSA Encryption - decryption
#include "exponentiation.h"
#include <stdlib.h>
#include <string.h>

int main (int argc, char * argv[])
{
    int p, q, n, a, x, d, len;
    if (argc != 5)
    {
        printf("Format: ./a.out p q x d");
        return 0;
    }
    char input[10];
    int arr[10];
    printf("Enter a string of not more than 9
            english alphabets\n");
```

Signature of
Teacher incharge

```
scanf("%s", input)
len = strlen(input);
printf("%d", len);
p = atoi(argv[1]);
q = atoi(argv[2]);
x = atoi[argv[3]);
d = atoi(argv[4]);
n = p*q;
printf("Encrypted message: \n");
for(int i=0; i<len; i++)
{
    a = input[i] - 'A';
    arr[i] = exponentiation(a, x, n);
    printf("%d", arr[i]);
}
printf("Decrypting the same encrypted
        message: \n");
for(int i=0; i<len; i++)
{
    printf("%c", (char)(exponentiation(
        arr[i], d, n) + 'A'));
}
return 0;
} // end of program.
```

// code on datasheet

RSA Encryption.

OUTPUT

```
$ gcc -o dhke dhke.c -lpowermation.o
$./dhke 23 9 4 3

  Ya = 6
  Yb = 16
  Key computed by A : 9
  Key computed by B : 9

$ ./dhke

 Format: ./a.out q alpha xa xb

$
$./dhke 11 2 9 4
  YA = 6
  Yb = 5
  Key computed at A : 9
  Key computed at B : 9

$
```

# R.V. COLLEGE OF ENGINEERING

## OBSERVATION / DATA SHEET

Date _____ Name M. C. SOHAN

Dept./Lab _____ Class _____ Expt./No. 8

Title Deffie Hellman key exchange

```
#include "exponentiation.h" // created file
#include <stdlib.h>

int main ( int argc, char* argv[])
{
        int q, alpha, xa, xb; // input
        int ya, yb, key ; //calculated
        if (argc != 5)
        {
            printf ("format: ./a.out q alpha xa xb");
            return 0;
        }
        q = atoi (argv[1]);
        alpha = atoi (argv[2]);
        xa = atoi (argv[3]);
        xb = atoi (argv[4]);

        //continued :
```

Signature of
Teacher incharge

```
ya = exponentiation (alpha, xa, q);
yb = exponentiation (alpha, xb, q);
printf ("\n ya = %d \n yb = %d \n", ya, yb);
printf ("\n Key computed at side A: %d",
                exponentiation (yb, xa, q));
printf ("\n Key computed at side B: %d",
                exponentiation (ya, xb, q));
printf ("END\n");
return 0;
} // end of program.
```

Pt. No. 8

Diffie Hellman key exchange

// Code on Datasheet.