8) ~~RSA~~ b) Deffie Hellman:

```
#include <stdio.h>
# include <math.h>
long long int power (long long int a, long long intb,
                long long int p)                              or
{
    if (b==1)
        return a;
    else
        return (((long long int) pow (a, b)) % P);
}

int main ()
{
  long long int P, G, x, a, y, b, ka, kb;
   P = 23;
   printf ("Enter value of P :\n");
   scanf ("%lld, &P);

   G = 9
   printf ("Enter value of G : \n");
   scanf ("%lld", &G);
   a = 4;
   printf ("The private key for A, a: %lld \n", a);
   b = 3;
   printf ("The private key for B, b: %lld \n \n", b);
   y = power (G, b, P);
   Ka = power (y, a, P);
   Kb = power (x, b, P);
printf ("The secret for ~~Alice is:~~ both A & B is:
        %lld, %lld \n", ka, kb);
   return 0;
 }
```

RSA:

```
# include <iostream>
# include <stdlib.h>
# include <math.h>
# include <string.h>
using namespace std;
long int gcd (long int a, long int b)
{ if (a==0) return b;
  if (b==0) return a;
  return gcd (b, a%b);
 }
long int coprime (long in a)
{  int i;
   for (i=2; i<a; i++)
   { if (a%i)==0)
          return 0;
   }
  return 1; }
long  int encrypt~~ion~~ (char ch, long int n, long int e)
{  int i; long int temp = ch;
  for (i=1; i<e ; i++)
   temp = (temp * ch) % n;
   return temp;
 }
```

```
char decrypt (long int ch, long int n, long int d)
{ int i; long int temp=ch;
for (i=1; i<d; i++)
    ch= (temp * ch) % n;
return ch;

}
int main ()
{
long int i, len;
long int p, q, n, phi, e, d, cipher [50];
char text[50];
cout << " Enter the text to be encrypted:\n";
cin.getline (text, sizeof (text));
len= strlen(text);

do {
    p= rand () % 30;
  } while (!isprime(p));
do {
    q= rand() % 30;
  } while (!isprime (q));

n= p * q
phi= (p-1) * (q-1)*
do {
  e = rand () %. phi;
} while (gcd(phi, e) != 1);
do {
  d > rand () % phi
} while (((d*e)%.phi) != 1);
cout <<"n(p*q)="<<~~phi~~ p<<"*"<<q<<"="
   <<p*q << endl;
cout <<"(p-1)*(q-1)="<<phi << endl;
cout <<" PBK (n,e):("<<n<<","<<e<<")",

cout<<" PRK (n,d): ("<<n<<",:"<<d<<")"
for (i=0; i<len; i++)
cipher [i] = encrypt (text[i], n, e);

cout <<" Encrypted message:";
for (i=0; i<len; i++)
   cout << cipher [i];
for (i=0; i<len; i++)
  text[i] = decrypt (cipher[i], n, d);
cout << endl;
cout<<" Decrypted message: ");
for [l=0; i<len , i++)
cout << text[i];
cout << endl;  return 0; }
```