1.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main() {
  char str1[20];
  char str[]="Hello World";
  int l=strlen(str);
  for(int i=0;i<l;i++){
            str1[i]=str[i]^0;
            printf("%c", str1[i]);
  }
}
```

Output: Hello World

2.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main() {
    char str1[20],str2[20];
    char str[]="Hello World";
    int l=strlen(str);
    for(int i=0;i<l;i++){
        str1[i]=str[i]&127;
        printf("%c", str1[i]);
    }
    for(int i=0;i<l;i++){
        str2[i]=str[i]^127;
        printf("%c",  str2[i]);
    }
}
```

```
Output: Hello World
7_(
```

3. 3.b)

```
import java.io.*;
import java.util.*;
public class Substitution{
  public static void main(String[] args)throws IOException{
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    String alpha="abcdefghijklmnopqrstuvwxyz";
    String subs="qwertyuiopasdfghjklzxcvbnm";
    System.out.println("Enter a string: ");
    String str=br.readLine();
    String encrypted="";
    for(int i=0;i<str.length();i++){
      int j;
      char c;
      c=str.charAt(i);
      j=alpha.indexOf(c);
      encrypted=encrypted+subs.charAt(j);
    }
    System.out.println("Encrypted string is: "+encrypted);
    String decrypted="";
    for(int i=0;i<encrypted.length();i++){
      int j;
      char c;
      c=encrypted.charAt(i);
      j=alpha.indexOf(c);
      decrypted=decrypted+subs.charAt(j);
    }
    System.out.println("Decrypted string is: "+decrypted);
  }
}
```

```
Output: Enter a String: hello
        Encrypted string is: itssg
        Decrypted string is: hello
```

3.a)Ceaser Cipher

```
import java.io.*;

public class Ceaser{

  public static void main(String[] args)throws IOException{
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Input string: ");
```

```
      String str=br.readLine();
      String alpha="abcdefghijklmnopqrstuvwxyz";
      String encrypted=encrypt(str,alpha);
      System.out.println("Encrypted String: "+encrypted);
      String decrypted=decrypt(encrypted,alpha);
      System.out.println("Decrypted String: "+decrypted);
   }
   public static String encrypt(String str, String alpha){
      String encrypted="";
      for(int i=0; i<str.length(); i++){
         int c,j;
         c=str.charAt(i);
         j=alpha.indexOf(c);
         j+=3;
         if(j>25){
            j=j%26;
         }
         encrypted+=alpha.charAt(j);
      }
      return encrypted;
   }
   public static String decrypt(String encrypted, String alpha){
      String decrypted="";
      for(int i=0; i<encrypted.length(); i++){
         int c,j;
         c=encrypted.charAt(i);
         j=alpha.indexOf(c);
         j-=3;
         if(j<0){
            j=j+26;
         }
         decrypted+=alpha.charAt(j);
      }
      return decrypted;
   }}
```

```
Output: Input string:
hello
Encrypted String: khoor
```

```
Decrypted String: hello
```

4. DES Algorithm:

```
import java.io.*;
import java.util.*;
import javax.crypto.*;
public class DES{
   public static void main(String[] args){
      try{
         Scanner sc=new Scanner(System.in);
         System.out.println("Enter a message: ");
         String input=sc.nextLine();
         KeyGenerator kg=KeyGenerator.getInstance("DES");
         SecretKey sk=kg.generateKey();
         Cipher c=Cipher.getInstance("DES");
         c.init(Cipher.ENCRYPT_MODE,sk);
         byte[] encrypt=c.doFinal(input.getBytes());
         System.out.println("Encrypted: "+new String(encrypt));
         c.init(Cipher.DECRYPT_MODE,sk);
         byte[] decrypt=c.doFinal(encrypt);
         System.out.println("\nDecrypted: "+new String(decrypt));
      }
      catch(Exception e){
         System.out.println(e.getMessage());
      }
   }
}
```

Output:

```
hello
Encrypted: Y*ë?ruî?
Decrypted: hello
```

5.

```
import java.io.*;
import java.util.*;
import javax.crypto.*;
public class Blowfish{
   public static void main(String[] args)throws Exception{
        KeyGenerator kg=KeyGenerator.getInstance("BLOWFISH");
        SecretKey sk=kg.generateKey();
        Cipher c=Cipher.getInstance("BLOWFISH");
        c.init(Cipher.ENCRYPT_MODE,sk);
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the words to encrypt: ");
        String input=sc.nextLine();
        byte[] encrypt=c.doFinal(input.getBytes());
        c.init(Cipher.DECRYPT_MODE,sk);
        byte[] decrypt=c.doFinal(encrypt);
        System.out.println("After Encryption: "+new String(encrypt));
        System.out.println("After Decryption: "+new String(decrypt));
    }
}
```

Output:

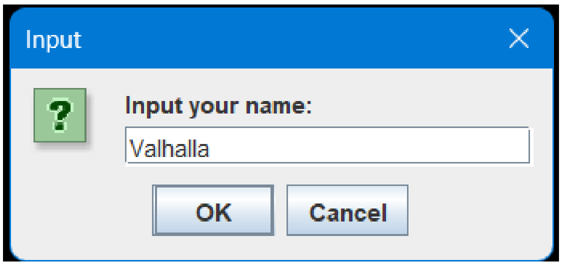
```
Enter the words to encrypt:
hello
After Encryption: |v£k?oP→
After Decryption: hello
```

6. 7th program Blowfish:::::

```
import java.io.*;
import java.util.*;
import javax.crypto.*;
import javax.swing.*;
public class Blowfish7{
    public static void main(String[] args)throws Exception{
        KeyGenerator kg=KeyGenerator.getInstance("BLOWFISH");
        SecretKey sk=kg.generateKey();
        Cipher c=Cipher.getInstance("BLOWFISH");
        c.init(Cipher.ENCRYPT_MODE,sk);
        String input=JOptionPane.showInputDialog("Input your name: ");
        byte[] encrypt=c.doFinal(input.getBytes());
        c.init(Cipher.DECRYPT_MODE,sk);
        byte[] decrypt=c.doFinal(encrypt);
        JOptionPane.showMessageDialog(JOptionPane.getRootFrame(), "\nEncrpted:
"+new String(encrypt)+"\nDecrypted: "+new String(decrypt));
```

```
        System.exit(0);
    }
}
```

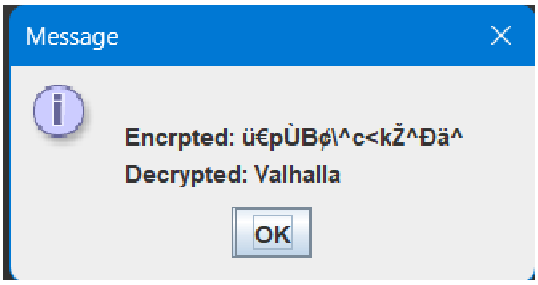
Output:1)



Input your name:

Valhalla

2)



Encrpted: ü€pÙB¢\^c<kŽ^Ðä^

Decrypted: Valhalla

7. 8th Program RSA

```
import java.io.*;
import java.util.*;
public class RSAlgorithm{
public static void main(String[] args){
Scanner sc=new Scanner(System.in);
System.out.println("Enter prime no.1: ");
int p=sc.nextInt();
System.out.println("Enter prime no.2: ");
int q=sc.nextInt();
int n=p*q;
int phi=(p-1)*(q-1);
int d=0,e=0;
for(int i=0;i<phi;i++){
if(gcd(i,phi)==1){
if(i==3){
continue;
}
else{
e=i;
break;
}}}
for(int k=0;;k++){
if(((k*e)%phi)==1){
d=k;
break;
}}
System.out.println("Value of e: "+e);
System.out.println("Value of d: "+d);
System.out.println("\n");
System.out.println("Public key: {"+e+","+n+"}");
System.out.println("Private key: {"+d+","+n+"}");
sc.close();
}
public static int gcd(int a, int b){
if(b==0){
return a;}
else{
return gcd(b,a%b);}
}
}
```

Output:

```
Enter prime no.1:
3
Enter prime no.2:
11
Value of e: 1
Value of d: 1
```

8. 9th Program Diffie Hellman

```
import java.io.*;
import java.util.*;
public class DiffieHellman{
  public static void main(String[] args)throws Exception{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a prime no.: ");
    int q=sc.nextInt();
    System.out.println("Enter alpha Value: ");
    int al=sc.nextInt();
    System.out.println("Enter A's Private key: ");
    int xa=sc.nextInt();
    System.out.println("Enter B's Private key: ");
    int xb=sc.nextInt();
    int ya=(int) Math.pow(al,xa)%q;
    int yb=(int) Math.pow(al,xb)%q;
    System.out.println("A's public key: "+ya);
    System.out.println("B's public key: "+yb);
    int ka=(int) Math.pow(yb,xa);
    int kb=(int) Math.pow(ya,xb);
    System.out.println("Secret key of A: "+ka);
    System.out.println("Secret key of B: "+kb);
    if(ka==kb){
      System.out.println("Both A and B can Communicate!");
    }
    else{
      System.out.println("Cannot communicate");
    }
  }
}
```

Output:

```
Enter a prime no.:
5
Enter alpha Value:
2
Enter A's Private key:
69
Enter B's Private key:
69
A's public key: 2
B's public key: 2
Secret key of A: 2147483647
Secret key of B: 2147483647
Both A and B can Communicate!
```

9. 10th program MD5

```
import java.math.BigInteger;
import java.security.MessageDigest;
public class MD5Algorithm{
  public static void main(String[] main) throws Exception{
    MessageDigest md=MessageDigest.getInstance("MD5");
    System.out.println("Message Digest Object Information: ");
    System.out.println("\nAlgorithm: "+md.getAlgorithm());
    System.out.println("\nProvider: "+md.getProvider());
    System.out.println("\nTo String: "+md.toString());
    String input="";
    md.update(input.getBytes());
    byte[] output=md.digest();
    System.out.println("\nMD5"+"("+input+")"+"="+bytesToHex(output));
    input="abc";
    md.update(input.getBytes());
    output=md.digest();
    System.out.println("\nMD5"+"("+input+")"+"="+bytesToHex(output));
    input="abcdefghijklmnopqrstuvwxyz";
```

```
        md.update(input.getBytes());
        output=md.digest();
        System.out.println("\nMD5"+"("+input+")"+"="+bytesToHex(output));
    }
    public static String bytesToHex(byte[] output){
        BigInteger n=new BigInteger(1,output);
        String hashtext=n.toString(16);
        return hashtext;
    }
}
```

```
Output:
Message Digest Object Information:
Algorithm: MD5
Provider: SUN version 17
To String: MD5 Message Digest from SUN, <initialized>
MD5()=d41d8cd98f00b204e9800998ecf8427e
MD5(abc)=900150983cd24fb0d6963f7d28e17f72
MD5(abcdefghijklmnopqrstuvwxyz)=c3fcd3d76192e4007dfb496cca67e13b
```

10. 11th Program SHA-1

```
import java.math.BigInteger;
import java.security.MessageDigest;
public class SHA1{
    public static void main(String[] args) throws Exception{
        MessageDigest md=MessageDigest.getInstance("SHA1");
        System.out.println("Message digest object information \n");
        System.out.println("Algorithm: "+md.getAlgorithm());
        System.out.println("Provider: "+md.getProvider());
        System.out.println("To String: "+md.toString());
        String input="";
        md.update(input.getBytes());
        byte[] output=md.digest();
        System.out.println("SHA1"+"("+input+")"+"="+bytesToHex(output));
        input="abc";
        md.update(input.getBytes());
        output=md.digest();
        System.out.println("SHA1"+"("+input+")"+"="+bytesToHex(output));
        input="abcdefghijklmnopqrstuvwxyz";
        md.update(input.getBytes());
        output=md.digest();
        System.out.println("SHA1"+"("+input+")"+"="+bytesToHex(output));
    }
    public static String bytesToHex(byte[] output){
        BigInteger n=new BigInteger(1,output);
        String hasht=n.toString(16);
        return hasht;
    }
}
```

Output:

```
Algorithm: SHA1
Provider: SUN version 17
To String: SHA1 Message Digest from SUN, <initialized>
SHA1()=da39a3ee5e6b4b0d3255bfef95601890afd80709
SHA1(abc)=a9993e364706816aba3e25717850c26c9cd0d89d
SHA1(abcdefghijklmnopqrstuvwxyz)=32d10c7b8cf96570ca04ce37f2a19d84240d3a89
```