

(b) Simple Columnar Technique

```
import java.util.*;

class simplecolumnar{

    public static void main(String sap[]){
        Scanner sc = new Scanner(System.in);

        System.out.print("\nEnter plaintext(enter in lower case): ");
        String message = sc.next();

        System.out.print("\nEnter key in numbers: ");
        String key = sc.next();

        int columnCount = key.length();

        int rowCount = (message.length() + columnCount)/columnCount;

        int plainText[][] = new int[rowCount][columnCount];
        int cipherText[][] = new int[rowCount][columnCount];

        System.out.print("\n-----Encryption-----\n");
        cipherText = encrypt(plainText, cipherText, message, rowCount, columnCount, key);

        String ct = "";
        for(int i=0; i<columnCount; i++)
        {
            for(int j=0; j<rowCount; j++)
            {
                if(cipherText[j][i] == 0)
                    ct = ct + 'x';
                else{
                    ct = ct + (char)cipherText[j][i];
                }
            }
        }
    }
}
```

```
}  
}  
System.out.print("\nCipher Text: " + ct);  
  
System.out.print("\n\n\n----Decryption----\n");  
  
plainText = decrypt(plainText, cipherText, ct, rowCount, columnCount, key);  
  
String pt = "";  
for(int i=0; i<rowCount; i++)  
{  
    for(int j=0; j<columnCount; j++)  
    {  
        if(plainText[i][j] == 0)  
            pt = pt + " ";  
        else{  
            pt = pt + (char)plainText[i][j];  
        }  
    }  
}  
System.out.print("\nPlain Text: " + pt);  
  
System.out.println();  
}
```

```
static int[][] encrypt(int plainText[], int cipherText[], String message, int rowCount, int  
columnCount, String key){
```

```
    int i,j;
```

```
    int k=0;
```

```
        for(i=0; i<rowCount; i++)
```

```
{
```

```
for(j=0; j<columnCount; j++)
```

```
{
```

```
    if(k < message.length())
```

```
    {
```

```
        plainText[i][j] = (int)message.charAt(k);
```

```
        k++;
```

```
    }
```

```
    else
```

```
    {
```

```
        break;
```

```
    }
```

```
}
```

```
}
```

```
    for(i=0; i<columnCount; i++)
```

```
    {
```

```
int currentCol = ( (int)key.charAt(i) - 48 ) - 1;
```

```
    for(j=0; j<rowCount; j++)
```

```
    {
```

```
        cipherText[j][i] = plainText[j][currentCol];
```

```
    }
```

```
}
```

```
System.out.print("Cipher Array(read column by column): \n");
```

```
for(i=0; i<rowCount; i++){
```

```
    for(j=0; j<columnCount; j++){
```

```
        System.out.print((char)cipherText[i][j] + "\t");
```

```
    }
```

```
    System.out.println();
```

```
}
```




```
    return cipherText;
}

static int[][] decrypt(int plainText[], int cipherText[], String message, int rowCount, int
columnCount, String key){
    int i,j;
    int k=0;

    for(i=0; i<columnCount; i++)
    {
        int currentCol= ( (int)key.charAt(i) - 48 ) -1;
        for(j=0; j<rowCount; j++)
        {
            plainText[j][currentCol] = cipherText[j][i];
        }
    }

    System.out.print("Plain Array(read row by row): \n");
    for(i=0; i<rowCount; i++){
        for(j=0; j<columnCount; j++){
            System.out.print((char)plainText[i][j] + "\t");
        }
        System.out.println();
    }

    return plainText;
}
```

→ Output

```
E:\shraddha>javac simplecolumnar.java
E:\shraddha>java simplecolumnar
Enter plaintext (enter in lower case) :
networksecurity
Enter key in numbers : 25314
----Encryption----
Cipher array (read column by column) :
e o t n w
k c s r e
r y i u t
cipher Text : ekrxocyxtsixnruxwetx
----Decrypti56on----
Plain array (read row by row) :
n e t w o
r k s e c
u r i t y
```

➤ **Program 4 :** Write program to encrypt and de

Ans. :

(a) DES algorithm

```
import javax.crypto.*;
import java.io.*;
import java.security.InvalidAlgorithmParameterException;
import java.security.spec.*;
import javax.crypto.spec.IvParameterSpec;
import java.lang.*;
public class DesEncrypter
{
    Cipher ecipher;
    Cipher dcipher;
    DesEncrypter(SecretKey key)
    {
        try
```




```
{
    ecipher = Cipher.getInstance("DES");
    dcipher = Cipher.getInstance("DES");
    ecipher.init(Cipher.ENCRYPT_MODE, key);
    dcipher.init(Cipher.DECRYPT_MODE, key);
}
catch (javax.crypto.NoSuchPaddingException e){}
catch (java.security.NoSuchAlgorithmException e){}
catch (java.security.InvalidKeyException e){}
}
public String encrypt(String str)
{
    try
    {
        byte[] utf8 = str.getBytes("UTF8");
        byte[] enc = ecipher.doFinal(utf8);
        return new sun.misc.BASE64Encoder().encode(enc);
    }
    catch (javax.crypto.BadPaddingException e){}
    catch (IllegalBlockSizeException e){}
    catch (UnsupportedEncodingException e){}
    catch (java.io.IOException e){}
    return null;
}
public String decrypt(String str)
{
    try
    {
        byte[] dec = new sun.misc.BASE64Decoder().decodeBuffer(str);
        byte[] utf8 = dcipher.doFinal(dec);
        return new String(utf8, "UTF8");
    }
    catch (javax.crypto.BadPaddingException e){}
    catch (IllegalBlockSizeException e){}
```

```
catch (UnsupportedEncodingException e) {}  
catch (java.io.IOException e) {}  
return null;  
}  
  
public static void main(String args[])  
{  
    System.out.println();  
    System.out.println("----*--Encrypting string using DES--*----");  
    System.out.println();  
    try  
    {  
        SecretKey key = KeyGenerator.getInstance("DES").generateKey();  
        DesEncrypter encrypter = new DesEncrypter(key);  
        String s="Don't tell anybody!";  
        String d="Hello";  
        String encrypted = encrypter.encrypt(s);  
        String decrypted = encrypter.decrypt(encrypted);  
        System.out.println("Original string is : " + s);  
        System.out.println("Encrypted string is: " + encrypted);  
        System.out.println("Decrypted string is: "+decrypted);  
    }  
    catch (Exception e) {}  
}
```

→ Output

```
E:\Ashraddha>java DesEncrypter  
----*--Encrypting string using DES--*----  
Original string is : Don't tell anybody!  
Encrypted  
zvCwwlSUnaBwGV16e8GP7gVPLwT8JTPR  
Decrypted string is : Don't tell anybody !
```