

Transposition Cipher

1) RailFence:

Program:

```
import java.util.Scanner;

public class rail {
    int numRails;

    public rail(int numRails) {
        this.numRails = numRails;
    }

    String getDecryptedData(String data) {
        char[] decrypted = new char[data.length()];
        int n = 0;
        for(int k = 0 ; k < numRails; k ++ ) {
            int index = k;
            boolean down = true;
            while(index < data.length() ) {
                //System.out.println(k + " " + index+ " "+ n );
                decrypted[index] = data.charAt(n++);

                if(k == 0 || k == numRails - 1) {
                    index = index + 2 * (numRails - 1);
                }
                else if(down) {
                    index = index + 2 * (numRails - k - 1);
                    down = !down;
                }
                else {
                    index = index + 2 * k;
                    down = !down;
                }
            }
        }
        return new String(decrypted);
    }

    String getEncryptedData(String data) {
        char[] encrypted = new char[data.length()];
```

```
int n = 0;

for(int k = 0 ; k < numRails; k ++ ) {
    int index = k;
    boolean down = true;
    while(index < data.length() ) {
        //System.out.println(k + " " + index+ " "+ n );
        encrypted[n++] = data.charAt(index);

        if(k == 0 || k == numRails - 1) {
            index = index + 2 * (numRails - 1);
        }
        else if(down) {
            index = index + 2 * (numRails - k - 1);
            down = !down;
        }
        else {
            index = index + 2 * k;
            down = !down;
        }
    }
}
return new String(encrypted);
}

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

    System.out.println("Enter plain text:");
    String data=scan.nextLine();
    String str;
    int i,a;
    a = data.indexOf(" ");
    str = data.replaceAll("\\s", "");
    str = str.toLowerCase();
    System.out.println("Enter no of rails:");
    int n =scan.nextInt();

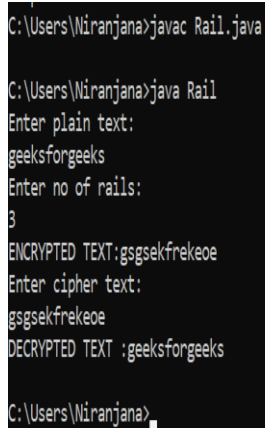
    rail railFenceCipher = new rail(n);

    String encrypted =railFenceCipher.getEncryptedData(str);

    System.out.println("Encryption:"+encrypted);
}
```

```
String decrypted =  
railFenceCipher.getDecryptedData(encrypted);  
System.out.println("Decryption:"+decrypted);  
  
}  
}
```

Screenshot:



```
C:\Users\Niranjana>javac Rail.java  
  
C:\Users\Niranjana>java Rail  
Enter plain text:  
geeksforgeeks  
Enter no of rails:  
3  
ENCRYPTED TEXT:gsgekferekeo  
Enter cipher text:  
gsgekferekeo  
DECRYPTED TEXT :geeksforgeeks  
  
C:\Users\Niranjana>
```

2) Rowcolumn:

Program:

```
import java.util.*;  
public class row{  
public static void main(String[] args) {  
String s, str;  
System.out.println("Enter The String:");  
Scanner scan = new Scanner(System.in);  
s=scan.nextLine();  
str = s.replaceAll("\\s", "");  
int row,col;  
System.out.println("Enter No of Rows:");  
row=scan.nextInt();  
System.out.println("Enter No of Columns:");  
col=scan.nextInt();  
char[][] mat = new char[row][col];  
int i,j,k=0,c=120;  
for(i=0;i<row;i++){  
for(j=0;j<col;j++){  
if(k<str.length())  
{  
mat[i][j]=str.charAt(k++);  
}  
}  
}
```

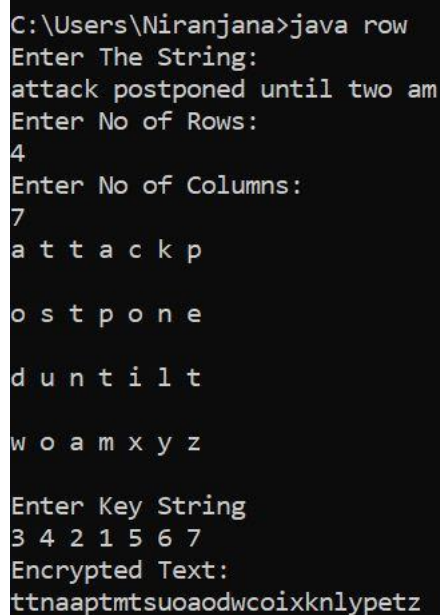
```
        else
        {
            mat[i][j]=(char)c;
            c++;
        }
    }

    for(i=0;i<row;i++){
        for(j=0;j<col;j++){
            System.out.print(mat[i][j]);
            System.out.print(" ");
        }
        System.out.println("\n");
    }

    int[] key = new int[col];
    System.out.println("Enter Key String");
    for(i=0;i<col;i++){
        key[i]=scan.nextInt();
    }

    System.out.println("Encrypted Text:");
    for(i=0;i<col;i++){
        for(j=0;j<row;j++){
            System.out.print(mat[j][key[i]-1]);
        }
    }
}
```

Screenshot:



```
C:\Users\Niranjana>java row
Enter The String:
attack postponed until two am
Enter No of Rows:
4
Enter No of Columns:
7
a t t a c k p
o s t p o n e
d u n t i l t
w o a m x y z

Enter Key String
3 4 2 1 5 6 7
Encrypted Text:
ttnaaptmtsuaodwcoixknlypetz
```