EX.NO.3 TRANSPOSITION CIPHER

RAIL-FENCE CIPHER – ENCRYPTION AND DECRYPTION

CODE:

```
import java.util.Scanner;

public class RailFence {

 String plainText = new String();
 String cipherText = new String();
 int depth;

 char[][] enc_mat;
 char[][] dec_mat;

 void encrypt() {

  int len = plainText.length();
  boolean dir_down = false;
  int row = 0, col = 0;

  enc_mat = new char[depth][len];

  for (int i = 0; i < depth; i++)
    for (int j = 0;j < len;j++)
      enc_mat[i][j] = '\0';

  for (int i = 0; i < len; i++) {

   enc_mat[row][col++] = plainText.charAt(i);

   if (row == 0 || row == depth - 1)
    dir_down = !dir_down;

   if (dir_down)
    row++;
   else
    row--;
    }

    System.out.println("\nThe encryption matrix is \n");

    for (int i = 0; i < depth; i++){
       for (int j = 0;j < len;j++) {

          if(enc_mat[i][j] == '\0')
             System.out.print("  ");

          else{
          System.out.print(enc_mat[i][j]+" ");
          cipherText+=enc_mat[i][j];
          }

       }
       System.out.print("\n");
    }

    System.out.println("\nThe cipher text is "+cipherText);
 }

 void decrypt()
 {
    int len = cipherText.length();

    boolean dir_down = false;
    int row = 0, col = 0;

    dec_mat = new char[depth][len];

    for (int i=0; i < len; i++)
    {
       dec_mat[row][col++] = '*';

       if (row == 0 || row == depth - 1)
               dir_down = !dir_down;

       if (dir_down)
          row++;
    else
        row--;
}

int index = 0;
System.out.println("\nThe decryption matrix is \n");

for (int i=0; i<depth; i++) {
    for (int j=0; j<len; j++) {

        if (dec_mat[i][j] == '*' && index<len) {
            dec_mat[i][j] = cipherText.charAt(index++);
            System.out.print(dec_mat[i][j]+" ");
        }

        else{
            System.out.print("  ");
        }

    }
    System.out.print("\n");
}

row = 0;
col = 0;
plainText="";

for (int i=0; i< len; i++)
{
    if (dec_mat[row][col] != '*')
       plainText+=dec_mat[row][col++];

    if (row == 0 || row == depth - 1)
       dir_down = !dir_down;

    if (dir_down)
      row++;

    else
      row--;
}
```

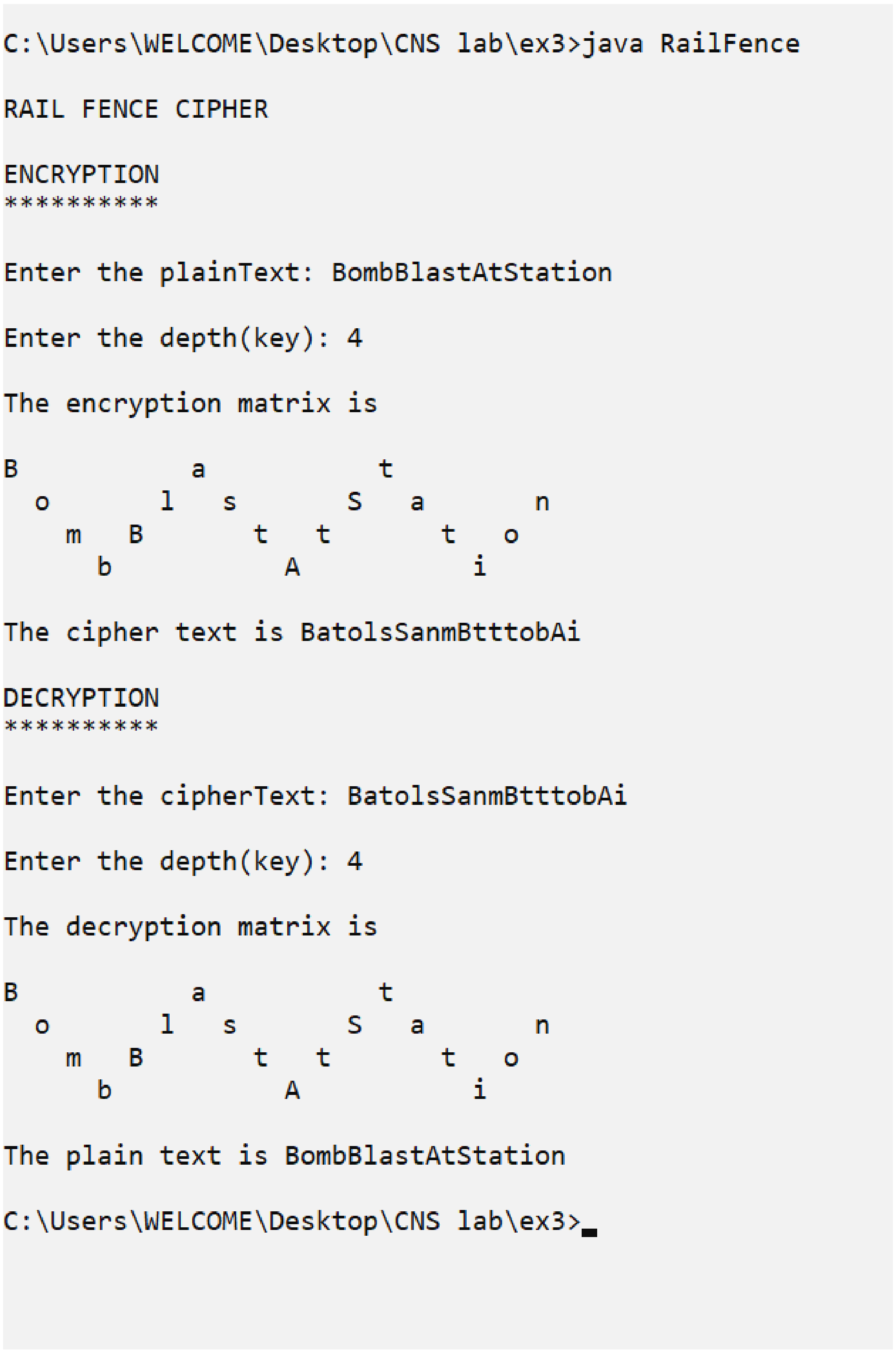
```
    System.out.println("\nThe plain text is "+plainText);

  }

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    RailFence rf = new RailFence();
    System.out.println("\nRAIL FENCE CIPHER");
    System.out.println("\nENCRYPTION");
    System.out.println("**********");
    System.out.print("\nEnter the plainText: ");
    rf.plainText = sc.next();
    System.out.print("\nEnter the depth(key): ");
    rf.depth = sc.nextInt();
    while (rf.depth<=0 || rf.depth>=rf.plainText.length()) {
      System.out.println("\nInvalid key");
      System.out.print("\nEnter the depth(key): ");
      rf.depth = sc.nextInt();
    }
    rf.encrypt();

    System.out.println("\nDECRYPTION");
    System.out.println("**********");
    System.out.print("\nEnter the cipherText: ");
    rf.cipherText = sc.next();
    System.out.print("\nEnter the depth(key): ");
    rf.depth = sc.nextInt();
    while (rf.depth<=0 || rf.depth>=rf.plainText.length()) {
      System.out.println("\nInvalid key");
      System.out.print("\nEnter the depth(key): ");
      rf.depth = sc.nextInt();
    }
    rf.decrypt();
    sc.close();
  }
}
```

<u>OUTPUT:</u>

<u>Example 1:</u>

```
C:\Users\WELCOME\Desktop\CNS lab\ex3>java RailFence

RAIL FENCE CIPHER

ENCRYPTION
**********

Enter the plainText: goodMorning

Enter the depth(key): 3

The encryption matrix is

g       M       i
  o   d   o   n   n
    o       r       g

The cipher text is gMiodonnorg

DECRYPTION
**********

Enter the cipherText: gMiodonnorg

Enter the depth(key): 3

The decryption matrix is

g       M       i
  o   d   o   n   n
    o       r       g

The plain text is goodMorning

C:\Users\WELCOME\Desktop\CNS lab\ex3>
```

Example 2:



```
C:\Users\WELCOME\Desktop\CNS lab\ex3>java RailFence

RAIL FENCE CIPHER

ENCRYPTION
**********

Enter the plainText: BombBlastAtStation

Enter the depth(key): 4

The encryption matrix is

B           a           t
  o       l   s       S   a       n
    m   B       t   t       t   o
      b           A           i

The cipher text is BatolsSanmBtttobAi

DECRYPTION
**********

Enter the cipherText: BatolsSanmBtttobAi

Enter the depth(key): 4

The decryption matrix is

B           a           t
  o       l   s       S   a       n
    m   B       t   t       t   o
      b           A           i

The plain text is BombBlastAtStation

C:\Users\WELCOME\Desktop\CNS lab\ex3>
```

Example 3:

```
RAIL FENCE CIPHER

ENCRYPTION
**********

Enter the plainText: hello

Enter the depth(key): 0

Invalid key

Enter the depth(key): -5

Invalid key

Enter the depth(key): 8

Invalid key

Enter the depth(key): 5

Invalid key

Enter the depth(key): 4

The encryption matrix is

h
  e
    l   o
      l

The cipher text is helol

DECRYPTION
**********

Enter the cipherText: helol

Enter the depth(key): 4

The decryption matrix is

h
  e
    l   o
      l

The plain text is hello

C:\Users\WELCOME\Desktop\CNS lab\ex3>
```

ROW-COLUMN CIPHER – ENCRYPTION AND DECRYPTION

CODE:

```
import java.util.Scanner;

public class RowColumn {
  String plainText = new String();
  String cipherText = new String();
  String key = new String();
  char[][] enc_mat;
  char[][] dec_mat;

  void encrypt() {
    int n_row, n_col,i, j,k=0;
    n_col = key.length();
    n_row = plainText.length() / n_col;
    if (plainText.length() % n_col>0)
        n_row += 1;

    enc_mat = new char[n_row][n_col];

    System.out.println("\nThe encryption matrix is \n");
    for ( j = 0; j < n_col; j++) {
    System.out.print(j+1+" ");
    }
    System.out.print("\n");
    for ( j = 0; j < n_col; j++) {
       System.out.print("--");
       }
    System.out.print("\n");

    for ( i = 0; i < n_row; i++) {
      for ( j = 0; j < n_col; j++) {
           if(k<plainText.length())
               enc_mat[i][j] = plainText.charAt(k++);
           else
               enc_mat[i][j] = '*';
        System.out.print(enc_mat[i][j] + " ");
      }
      System.out.print("\n");
    }
```

```
   System.out.println("\nAfter rearranging the columns using the key, the
encryption matrix  is \n");
   for ( j = 0; j < n_col; j++) {
     System.out.print(key.charAt(j)+" ");
     }
     System.out.print("\n");
     for ( j = 0; j < n_col; j++) {
        System.out.print("--");
        }
     System.out.print("\n");

   for ( i = 0; i < n_row; i++) {
     for ( j = 0; j < n_col; j++) {
        k = key.charAt(j) - 48 - 1;
        System.out.print(enc_mat[i][k] + " ");
     }
     System.out.print("\n");
   }

   for (j = 0; j < n_col; j++) {
     k = key.charAt(j) - 48 - 1;
     for (i = 0; i < n_row; i++) {
       cipherText += enc_mat[i][k];
     }
   }
   System.out.println("\nThe cipher text is " + cipherText);
  }

  void decrypt() {

   int n_row, n_col,i, j,k=0;
   n_row = key.length();
   n_col = plainText.length() / n_row;
   if (plainText.length() % n_row>0)
      n_col += 1;

   dec_mat = new char[n_row][n_col];
   System.out.println("\nThe decryption matrix is \n");

   for ( i = 0; i < n_row; i++) {
     System.out.print(key.charAt(i) + "---> ");
     for ( j = 0; j < n_col; j++) {
       dec_mat[i][j]=cipherText.charAt(k++);
         System.out.print(dec_mat[i][j] + " ");
     }
     System.out.print("\n");
   }

   int row;
   k=0;

   System.out.println("\nAfter rearranging the rows using the key, the
decryption matrix  is \n");
   for( i=0;i<cipherText.length();i=i+n_col){
      row=key.charAt(k++)-48-1;
      for( j=0;j<n_col;j++)
      {
         dec_mat[row][j]=cipherText.charAt(i+j);
      }
   }

   for ( i = 0; i < n_row; i++) {
     for ( j = 0; j < n_col; j++) {
        System.out.print(dec_mat[i][j] + " ");
     }
     System.out.print("\n");
   }

   plainText = "";
   for ( i = 0; i < n_col; i++) {
      for ( j = 0; j < n_row; j++) {
         plainText += dec_mat[j][i];
      }
     }

   System.out.println("\nThe plain text is " + plainText+"\n");
  }

  boolean validateKey(){

   int len=key.length();
```

```
    if(len<=1 || len>=plainText.length() || !key.matches("[1-9]+"))
      return false;
    for(int i=0;i<len;i++){
      if(!key.contains(Integer.toString(i+1)))
        return false;
    }
    return true;
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    RowColumn rc = new RowColumn();
    System.out.println("\nROW COLUMN CIPHER");
    System.out.println("\nENCRYPTION");
    System.out.println("**********");
    System.out.print("\nEnter the plainText: ");
    rc.plainText = sc.next();
    System.out.print("\nEnter the key: ");
    rc.key = sc.next();
    while (!rc.validateKey()) {
        System.out.println("\nInvalid key");
        System.out.print("\nEnter the key: ");
        rc.key = sc.next();
    }
    rc.encrypt();

    System.out.println("\nDECRYPTION");
    System.out.println("**********");
    System.out.print("\nEnter the cipherText: ");
    rc.cipherText = sc.next();
    System.out.print("\nEnter the key: ");
    rc.key = sc.next();
    while (!rc.validateKey()) {
        System.out.println("\nInvalid key");
        System.out.print("\nEnter the key: ");
        rc.key = sc.next();
    }
    rc.decrypt();
    sc.close();
  }
}
```

OUTPUT:

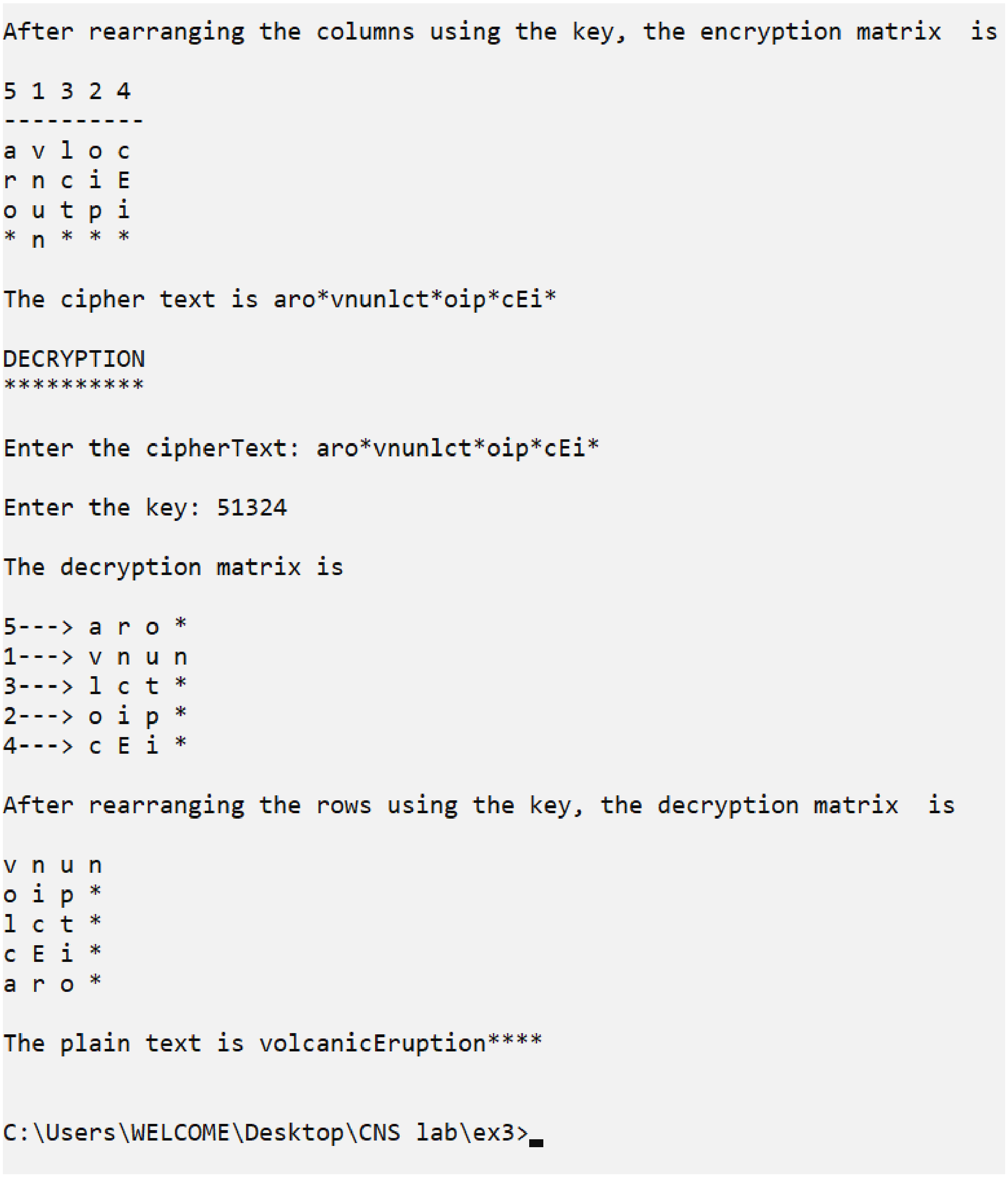
Example 1:

```
C:\Users\WELCOME\Desktop\CNS lab\ex3>java RowColumn

ROW COLUMN CIPHER

ENCRYPTION
**********

Enter the plainText: santaclaus

Enter the key: 1

Invalid key

Enter the key: 1298765403

Invalid key

Enter the key: 3A1

Invalid key

Enter the key: 333

Invalid key

Enter the key: 301

Invalid key

Enter the key: 312

The encryption matrix is

1 2 3
------
s a n
t a c
l a u
s * *

After rearranging the columns using the key, the encryption matrix  is
3 1 2
------
n s a
c t a
u l a
* s *
```

```
The cipher text is ncu*stlsaaa*

DECRYPTION
**********

Enter the cipherText: ncu*stlsaaa*

Enter the key: 312

The decryption matrix is

3---> n c u *
1---> s t l s
2---> a a a *

After rearranging the rows using the key, the decryption matrix  is

s t l s
a a a *
n c u *

The plain text is santaclaus**


C:\Users\WELCOME\Desktop\CNS lab\ex3>
```

Example 2:

```
C:\Users\WELCOME\Desktop\CNS lab\ex3>java RowColumn

ROW COLUMN CIPHER

ENCRYPTION
**********

Enter the plainText: volcanicEruption

Enter the key: 51324

The encryption matrix is

1 2 3 4 5
----------
v o l c a
n i c E r
u p t i o
n * * * *
```



```
After rearranging the columns using the key, the encryption matrix  is

5 1 3 2 4
----------
a v l o c
r n c i E
o u t p i
* n * * *

The cipher text is aro*vnunlct*oip*cEi*

DECRYPTION
**********

Enter the cipherText: aro*vnunlct*oip*cEi*

Enter the key: 51324

The decryption matrix is

5---> a r o *
1---> v n u n
3---> l c t *
2---> o i p *
4---> c E i *

After rearranging the rows using the key, the decryption matrix  is

v n u n
o i p *
l c t *
c E i *
a r o *

The plain text is volcanicEruption****


C:\Users\WELCOME\Desktop\CNS lab\ex3>
```