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 plain 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 cipher plainText: ");

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();

}

}



