Practical 3

Name:Satishkumar Rajan Nadar Exam seat:S.20.64

Rail Fence Cipher:-

package practical3;

import java.util.\*;

public class Practical3 {

public static void main(String[] args) {

System.out.print("Enter the number of rails: ");

Scanner sc = new Scanner(System.in);

int rails = sc.nextInt();

System.out.print("Enter the plaintext for encryption: ");

String plaintext = sc.next();

encryption(plaintext,rails);

System.out.println("-------------------------------Decryption process started--------------------------");

System.out.print("Enter the number of rails: ");

rails = sc.nextInt();

System.out.print("Enter the ciphertext for decryption: ");

String ciphertext = sc.next();

decryption(ciphertext,rails);

}

public static void encryption(String str,int rails){

boolean checkdown=false;

int j=0;

int row=rails;

int col=str.length();

char[][] a=new char[row][col];

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

if(j==0||j==row-1)

checkdown=!checkdown;

a[j][i]=str.charAt(i);

if(checkdown){

j++;

}

else

j--;

}

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

for(int k=0;k<col;k++){

System.out.print(a[i][k]+"\t");

}

System.out.println();

}

String en="";

System.out.println("--------------------------");

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

for(int k=0;k<col;k++){

if(a[i][k]!=0)

en=en+a[i][k];

}

}

System.out.println(en); //printing the ciphertext

}

public static void decryption(String str, int rails){

boolean checkdown=false;

int j=0;

int row=rails;

int col=str.length();

char[][] a=new char[row][col];

//first of all mark the rails position by \* in the matrix

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

if(j==0||j==row-1)

checkdown=!checkdown;

a[j][i]='\*';

if(checkdown)j++;

else j--;

}

int index=0;

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

for(int k=0;k<col;k++){

if(a[i][k]=='\*'&&index<str.length()){

a[i][k]=str.charAt(index++);

}

}

}

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

for (int k=0;k<col;k++){

System.out.print(a[i][k]+"\t");

}

System.out.println();

}

checkdown=false;

String s="";

j=0;

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

if(j==0||j==row-1)

checkdown=!checkdown;

s+=a[j][i];

if(checkdown)j++;

else j--;

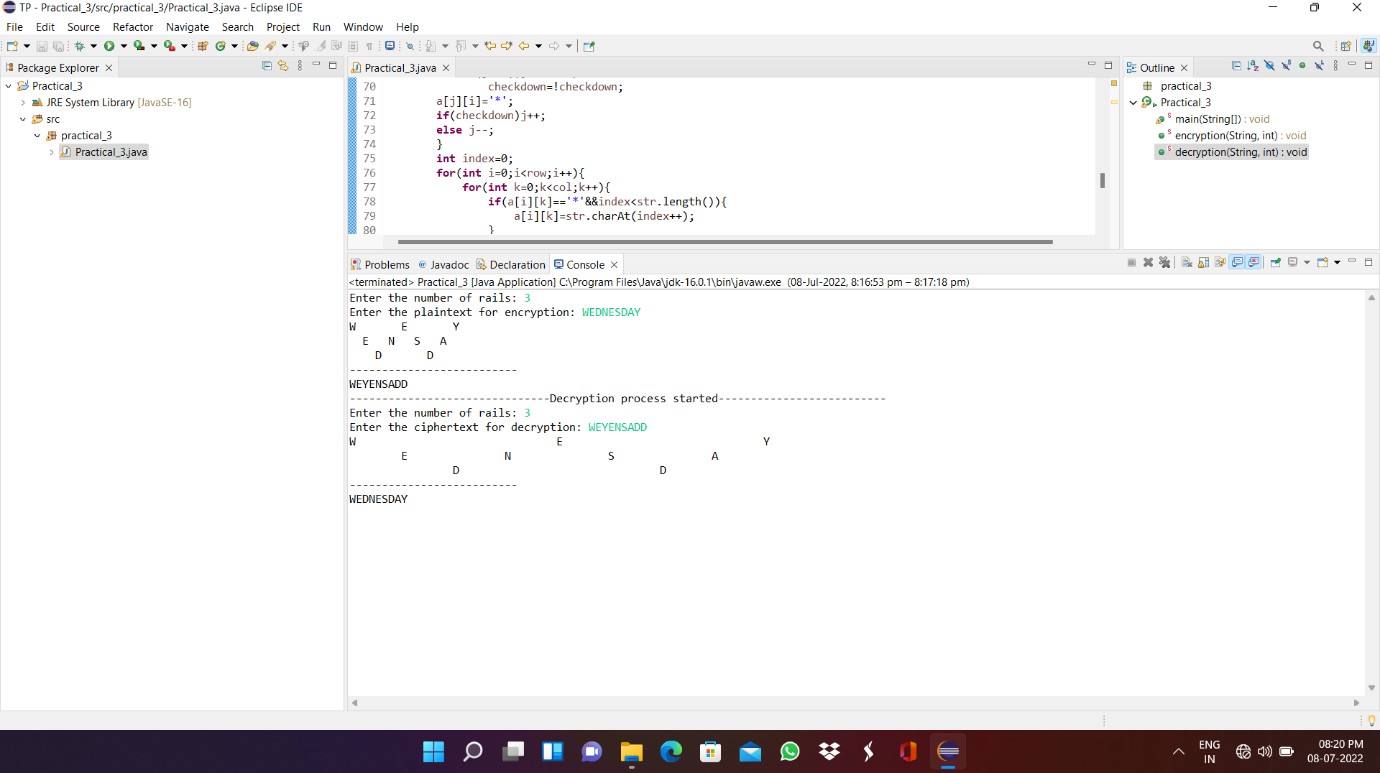
}

System.out.println("--------------------------");

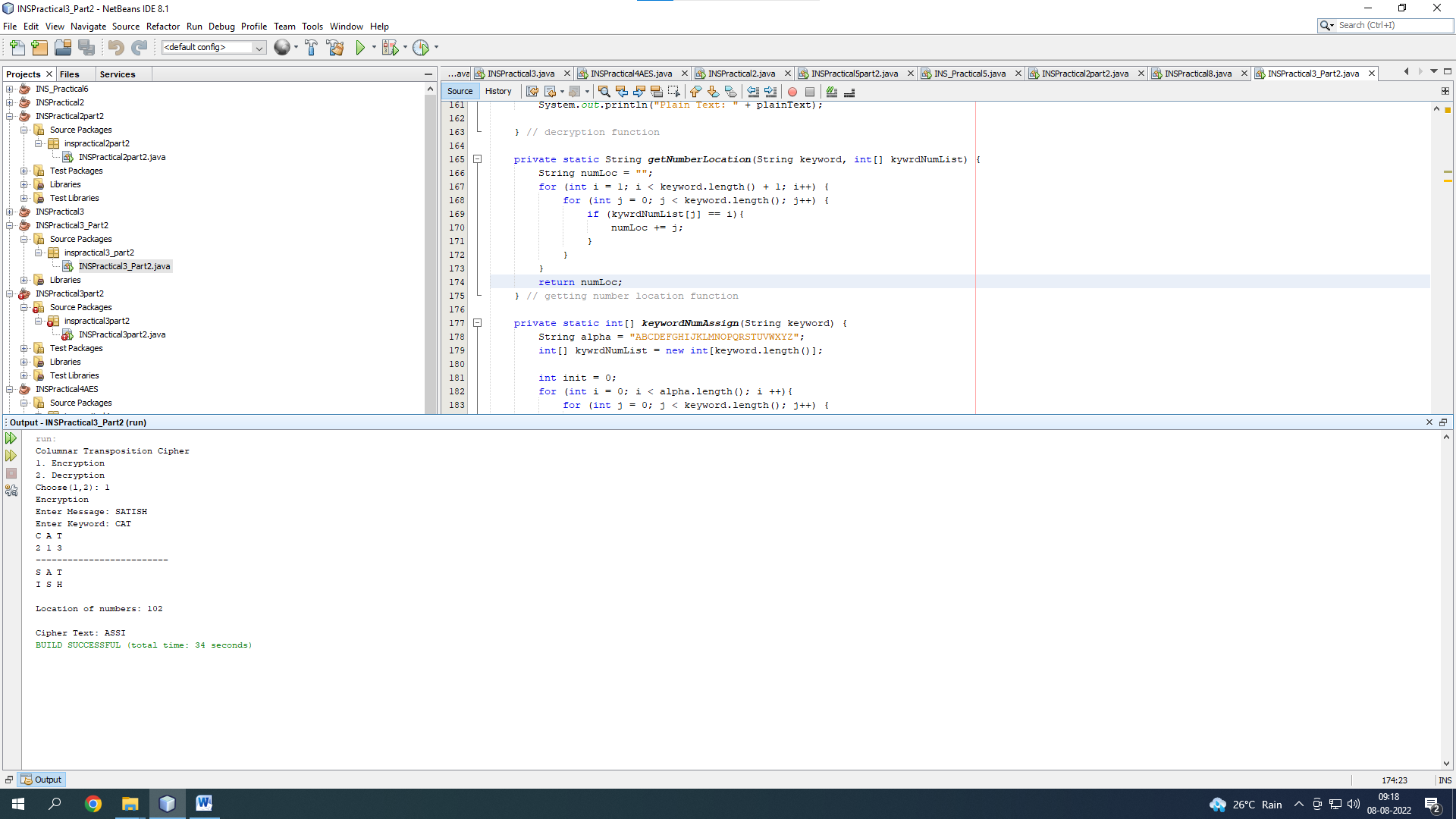
System.out.println(s);

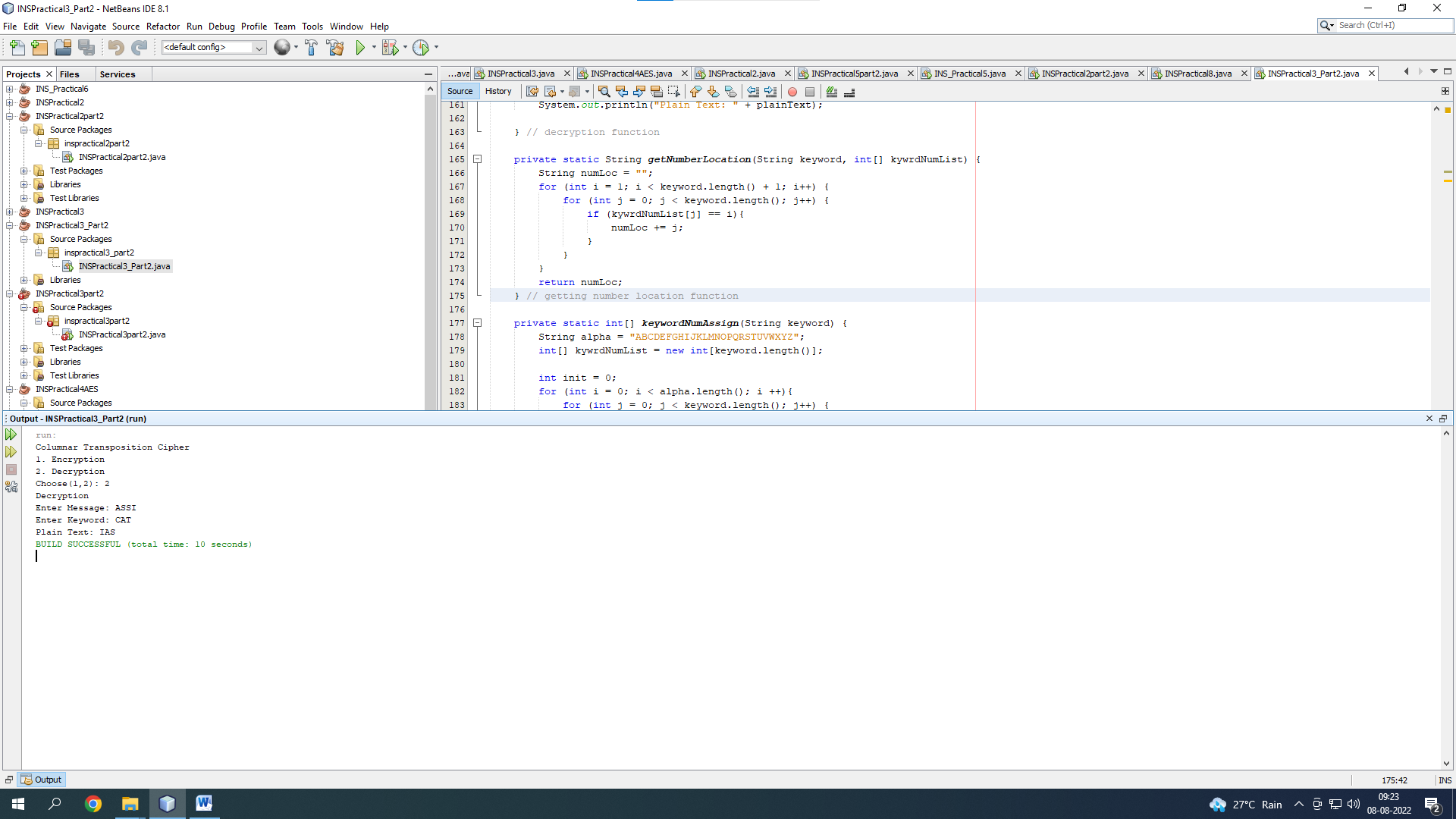
}

}



Simple columnar Technique:





package inspractical3\_part2;

import java.util.Scanner;

public class INSPractical3\_Part2 {

private static Scanner in;

public static void main(String[] args){

System.out.println("Columnar Transposition Cipher");

in = new Scanner(System.in);

System.out.print("1. Encryption\n2. Decryption\nChoose(1,2): ");

int choice = in.nextInt();

in.nextLine();

if (choice == 1){

System.out.println("Encryption");

encryption();

} else if (choice == 2){

System.out.println("Decryption");

decryption();

} else {

System.out.println("Invalid Choice");

System.exit(0);

}

}

private static void encryption(){

System.out.print("Enter Message: ");

String plainText = in.nextLine().toUpperCase().replace(" ", "");

StringBuilder msg = new StringBuilder(plainText);

System.out.print("Enter Keyword: ");

String keyword = in.nextLine().toUpperCase();

int[] kywrdNumList = keywordNumAssign(keyword);

// printing keyword

for (int i = 0, j = 1; i < keyword.length(); i++, j++) {

System.out.print(keyword.substring(i, j) + " ");

}

System.out.println();

for (int i: kywrdNumList){

System.out.print(i + " ");

}

System.out.println();

System.out.println("-------------------------");

int extraLetters = msg.length() % keyword.length();

int dummyCharacters = keyword.length() - extraLetters;

if (extraLetters != 0){

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

msg.append(".");

}

}

int numOfRows = msg.length() / keyword.length();

// Converting message into a grid

char[][] arr = new char[numOfRows][keyword.length()];

int z = 0;

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

for (int j = 0; j < keyword.length(); j++) {

arr[i][j] = msg.charAt(z);

z++;

}

}

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

for (int j = 0; j < keyword.length(); j++) {

System.out.print(arr[i][j] + " ");

} // inner for loop

System.out.println();

} // for loop

// cipher text generation

StringBuilder cipherText = new StringBuilder();

System.out.println();

// getting locations of numbers

String numLoc = getNumberLocation(keyword, kywrdNumList);

System.out.println("Location of numbers: " + numLoc);

System.out.println();

for (int i = 0, k = 0; i < numOfRows; i++, k++) {

int d;

if (k == keyword.length()){

break;

} else {

d = Character.getNumericValue(numLoc.charAt(k));

}

for (int j = 0; j < numOfRows; j++) {

cipherText.append(arr[j][d]);

}

}

System.out.println("Cipher Text: " + cipherText);

} // encryption method

private static void decryption(){

System.out.print("Enter Message: ");

String msg = in.nextLine().toUpperCase().replace(" ", "");

System.out.print("Enter Keyword: ");

String keyword = in.nextLine().toUpperCase();

int numOfRows = msg.length() / keyword.length();

// array with dummy values

char[][] arr = new char[numOfRows][keyword.length()];

// assigning numbers to keywords

int[] kywrdNumList = keywordNumAssign(keyword);

String numLoc = getNumberLocation(keyword, kywrdNumList);

for (int i = 0, k = 0; i < msg.length(); i++, k++) {

int d = 0;

if (k == keyword.length()){

k = 0;

} else {

d = Character.getNumericValue(numLoc.charAt(k));

}

for (int j = 0; j < numOfRows; j++, i++) {

arr[j][d] = msg.charAt(i);

}

--i;

}

StringBuilder plainText = new StringBuilder();

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

for (int j = 0; j < keyword.length(); j++) {

plainText.append(arr[i][j]);

}

}

System.out.println("Plain Text: " + plainText);

} // decryption function

private static String getNumberLocation(String keyword, int[] kywrdNumList) {

String numLoc = "";

for (int i = 1; i < keyword.length() + 1; i++) {

for (int j = 0; j < keyword.length(); j++) {

if (kywrdNumList[j] == i){

numLoc += j;

}

}

}

return numLoc;

}

private static int[] keywordNumAssign(String keyword) {

String alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

int[] kywrdNumList = new int[keyword.length()];

int init = 0;

for (int i = 0; i < alpha.length(); i ++){

for (int j = 0; j < keyword.length(); j++) {

if (alpha.charAt(i) == keyword.charAt(j)){

init++;

kywrdNumList[j] = init;

}

}

}

return kywrdNumList;

}

}