 

 

 

 

 

 



import java.awt.Point;

import java.util.Scanner;

public class PlayfairCipher {

private static char[][] charTable;

private static Point[] positions;

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String key = prompt("Enter an encryption key (min length 6): ", sc, 6);

String txt = prompt("Enter the message: ", sc, 1);

String jti = prompt("Replace J with I? y/n: ", sc, 1);

boolean changeJtoI = jti.equalsIgnoreCase("y");

createTable(key, changeJtoI);

int i,j;

for(i=0;i<5;i++){

for(j=0;j<5;j++){

System.out.print(charTable[i][j]);

}

System.out.println("");

}

String enc = encode(prepareText(txt, changeJtoI));

System.out.printf("%nEncoded message: %n%s%n", enc);

System.out.printf("%nDecoded message: %n%s%n", decode(enc));

}

private static String prompt(String promptText, Scanner sc, int minLen) {

String s;

do {

System.out.print(promptText);

s = sc.nextLine().trim();

} while (s.length() < minLen);

return s;

}

private static String prepareText(String s, boolean changeJtoI) {

s = s.toUpperCase().replaceAll("[^A-Z]", "");

return changeJtoI ? s.replace("J", "I") : s.replace("Q", "");

}

private static void createTable(String key, boolean changeJtoI) {

charTable = new char[5][5];

positions = new Point[26];

String s = prepareText(key + "ABCDEFGHIJKLMNOPQRSTUVWXYZ", changeJtoI);

int len = s.length();

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

char c = s.charAt(i);

if (positions[c - 'A'] == null) {

charTable[k / 5][k % 5] = c;

positions[c - 'A'] = new Point(k % 5, k / 5);

k++;

}

}

}

private static String encode(String s) {

StringBuilder sb = new StringBuilder(s);

for (int i = 0; i < sb.length(); i += 2) {

if (i == sb.length() - 1)

sb.append(sb.length() % 2 == 1 ? 'X' : "");

else if (sb.charAt(i) == sb.charAt(i + 1))

sb.insert(i + 1, 'X');

}

return codec(sb, 1);

}

private static String decode(String s) {

return codec(new StringBuilder(s), 4);

}

private static String codec(StringBuilder text, int direction) {

int len = text.length();

for (int i = 0; i < len; i += 2) {

char a = text.charAt(i);

char b = text.charAt(i + 1);

int row1 = positions[a - 'A'].y;

int row2 = positions[b - 'A'].y;

int col1 = positions[a - 'A'].x;

int col2 = positions[b - 'A'].x;

if (row1 == row2) {

col1 = (col1 + direction) % 5;

col2 = (col2 + direction) % 5;

} else if (col1 == col2) {

row1 = (row1 + direction) % 5;

row2 = (row2 + direction) % 5;

} else {

int tmp = col1;

col1 = col2;

col2 = tmp;

}

text.setCharAt(i, charTable[row1][col1]);

text.setCharAt(i + 1, charTable[row2][col2]);

}

return text.toString();

}

}