PROGRAM:

CaesarCipher.java

```
class caesarCipher {
public static String encode(String enc, int offset) {
offset = offset % 26 + 26;
StringBuilder encoded = new StringBuilder();
for (char i : enc.toCharArray())
{
if (Character.isLetter(i))
{
if (Character.isUpperCase(i))
{
encoded.append((char) ('A' + (i - 'A' + offset) % 26)); }
else
{
encoded.append((char) ('a' + (i - 'a' + offset) % 26)); } }
else
{
encoded.append(i);
} }
return encoded.toString();
}
public static String decode(String enc, int offset) {
return encode(enc, 26 - offset); }
public static void main(String[] args) throws java.lang.Exception {
String msg = "Anna University";
System.out.println("Simulating Caesar Cipher\n ---------------------- ");
System.out.println("Input : " + msg);
System.out.printf("Encrypted Message : ");
System.out.println(caesarCipher.encode(msg, 3));
System.out.printf("Decrypted Message : ");
System.out.println(caesarCipher.decode(caesarCipher.encode(msg,        3), 3));
}
}
```

OUTPUT:

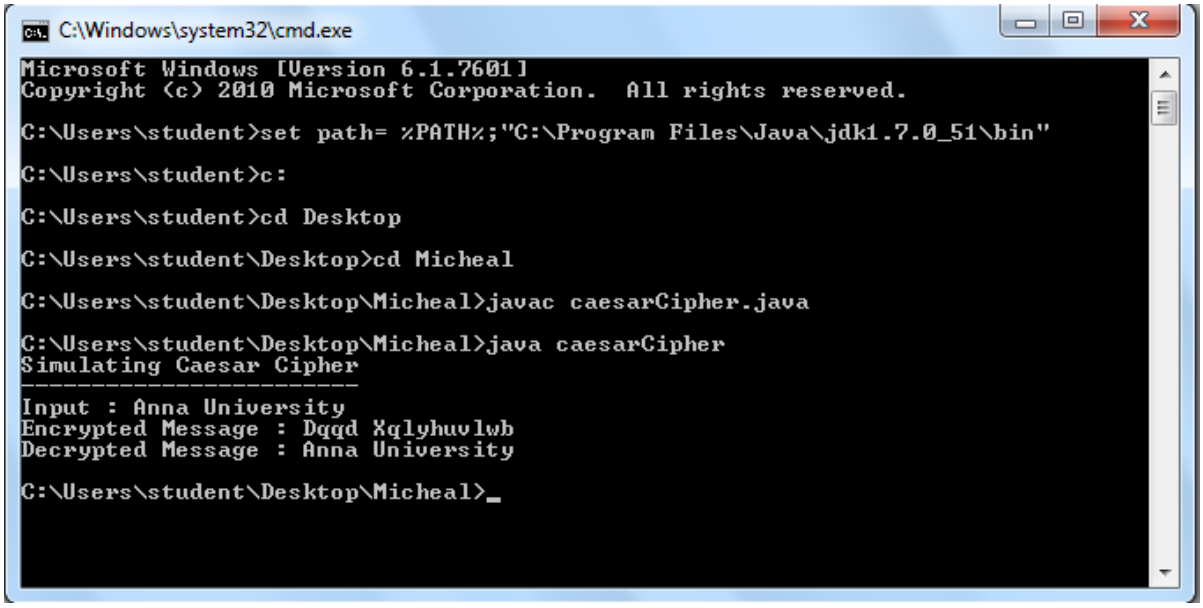
Simulating Caesar Cipher

_____

Input : Anna University

Encrypted Message : Dqqd Xqlyhuvlwb

Decrypted Message : Anna University



PROGRAM:

playfairCipher.java

```
import java.awt.Point;
class playfairCipher {
private static char[][] charTable;
private static Point[] positions;
private static String prepareText(String s, boolean chgJtoI)
{
s = s.toUpperCase().replaceAll("[^A-Z]", "");
return chgJtoI ? s.replace("J", "I") : s.replace("Q", "");
}
private static void createTbl(String key, boolean chgJtoI) {
charTable = new char[5][5]; positions = new Point[26];
String s = prepareText(key + "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
chgJtoI);
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 codec(StringBuilder txt, int dir) {
int len = txt.length();
for (int i = 0; i < len; i += 2)
{
char a = txt.charAt(i);
char b = txt.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 + dir) % 5;
```

```
col2 = (col2 + dir) % 5;
}
else if (col1 == col2) {
row1 = (row1 + dir) % 5;
row2 = (row2 + dir) % 5; }
else { int tmp = col1;
col1 = col2;
col2 = tmp; }
txt.setCharAt(i, charTable[row1][col1]);
txt.setCharAt(i + 1, charTable[row2][col2]); }
return txt.toString(); }
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); }
public static void main(String[] args) throws java.lang.Exception {
String key = "MONARCHY"; String txt = "Balloon";
/* make sure string length is even */
/* change J to I */
boolean chgJtoI = true;
createTbl(key, chgJtoI);
String enc = encode(prepareText(txt, chgJtoI));
System.out.println("Simulating Playfair Cipher\n ----------------------");
System.out.println("Input Message : " + txt); System.out.println("Encrypted
Message : " + enc);
System.out.println("Decrypted Message : " + decode(enc));
}
}
```

OUTPUT:

Simulating Playfair Cipher

_____

Input Message : Balloon

Encrypted Message : IBSUPMNA

Decrypted Message : BALXLOON