UNIT-2 MODULE-5 ASSIGNMENT

1. a) i) Encryption of message using Caesar cipher in java

//importing the all defined packages under java.util.\*

import java.util.\*;

class Caesar\_cipher\_encrypt {

public static String encrypt\_string(String plain\_text, int key) {

//checking length bound for the key

if(key<0){

key = (key%26)+26;

}

else if(key>26){

key = key%26;

}

//System.out.println("text----->" + key);

String cipher\_text="";

int string\_length = plain\_text.length();

for(int i=0; i<string\_length; i++){

//Getting each element of the plain text

char ch = plain\_text.charAt(i);

if(java.lang.Character.isLetter(ch)){

//checking whether the given element is a lowercase character

if(java.lang.Character.isLowerCase(ch)){

//computation of cipher text for each element of the string

char c = (char)(ch+key);

if(c < 'z'){

cipher\_text += c;

}

else{

cipher\_text += (char)(ch - (26 -key));

}

}

//checking whether the given element is a lowercase character

else if(java.lang.Character.isUpperCase(ch)){

//computation of cipher text for each element of the string

char c = (char)(ch+key);

if(c < 'z'){

cipher\_text += c;

}

else{

cipher\_text += (char)(ch - (26 -key));

}

}

}

else{

cipher\_text += ch;

}

}

return cipher\_text;

}

public static void main(String args[]){

//System.out.println("Hello Java");

//initalizing the parameters and declaration required for encryption

String text = "This is a private, encrypted message";

//System.out.println("enter the string: + %s", text);

//System.out.println("text----->" + text);

//calling the encryption function in main function

String cipher = encrypt\_string(text,15);

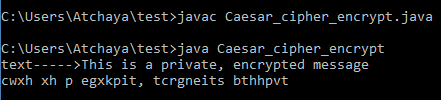
//printing the resulting cipher text

System.out.println(cipher);

}

}

Output:



ii) Encryption of message using Transposition cipher in java

import java.awt.event.\*;

import java.util.\*;

public class Transposition\_cipher\_encrypt

{

public static void main(String args[])

{

String trans\_key;

String trans\_message;

String encrypted\_Message;

// Letters in the x-axis

int x=0;

// Letters in the y-axis

int y=0;

trans\_key = "tape";

trans\_message = "This is a private, encrypted message";

encrypted\_Message = "";

System.out.println(trans\_key + trans\_message + encrypted\_Message);

// To set the temp as [x][y]

char temp[][]=new char [trans\_key.length()][trans\_message.length()];

char msg[] = trans\_message.toCharArray();

// To populate the array

x=0;

y=0;

// To convert the message into an array of char

int i=0;

while (i< msg.length)

{

temp[x][y]=msg[i];

if (x==(trans\_key.length()-1))

{

x=0;

y=y+1;

} // Close if

else

{

x++;

}

i++;

} // Close for loop

// To sort the key

char t[]=new char [trans\_key.length()];

t=trans\_key.toCharArray();

Arrays.sort(t);

for (int j=0;j<y;j++)

{

for (i=0;i<trans\_key.length();i++)

{

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

}

System.out.println();

}

System.out.println();

// To print out row by row (i.e. y)

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

// To compare the the sorted Key with the key

// For char in the key

for ( i=0;i<trans\_key.length();i++){

int pos=0;

// To get the position of key.charAt(i) from sorted key

for (pos=0;pos<t.length;pos++){

if (trans\_key.charAt(i)==t[pos]){

// To break the for loop once the key is found

break;

}

}

System.out.print(temp[pos][j]);

encrypted\_Message+=temp[pos][j];

}

System.out.println();

}

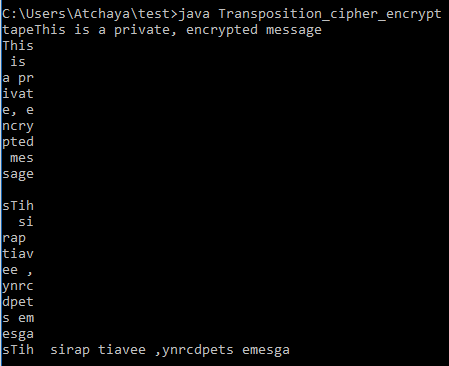
System.out.println(encrypted\_Message);

System.exit(0);

}

}

Output:



iii) Encryption of message using Vignere cipher in java

public class Vignere\_cipher\_encrypt {

public static void main(String[] args) {

//initalizing the parameters and declaration required for encryption

String vignere\_key = "SECURITY";

String vignere\_ori = "This is a private, encrypted message";

System.out.println(vignere\_ori);

//calling the encryption function in main function

String encrypt = encrypt\_string(vignere\_ori, vignere\_key);

//printing the resultant cipher text

System.out.println(encrypt);

}

static String encrypt\_string(String text, final String key) {

String cipher = "";

//conversion of text to upper case

text = text.toUpperCase();

int i = 0, j = 0;

while (i < text.length()) {

char c = text.charAt(i);

if (c < 'A' || c > 'Z') continue;

//computation of cipher text of each character in the string

cipher += (char)((c + key.charAt(j) - 2 \* 'A') % 26 + 'A');

j = ++j % key.length();

i += 1;

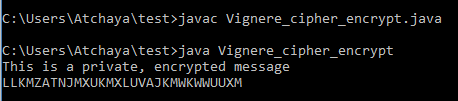
}

return cipher;

}

}

Output:



b) i) Decryption of message using Caesar cipher in java

//importing the all defined packages under java.util.\*

import java.util.\*;

class Caesar\_cipher {

public static String decrypt\_string(String plain\_text, int key) {

//checking length bound for the key

if(key>26){

key = key%26;

}

else if(key<0){

key = (key%26)+26;

}

//System.out.println("text----->" + key);

String cipher\_text="";

int string\_length = plain\_text.length();

for(int i=0; i<string\_length; i++){

//Getting each element of the cipher text

char ch = plain\_text.charAt(i);

if(java.lang.Character.isLetter(ch)){

if(java.lang.Character.isLowerCase(ch)){

//checking whether the given element is a lowercase character

char c = (char)(ch-key);

if(c < 'a'){

cipher\_text += (char)(ch + (26 -key));

}

else{

cipher\_text += c;

}

}

else if(java.lang.Character.isUpperCase(ch)){

//checking whether the given element is a lowercase character

char c = (char)(ch-key);

if(c < 'A'){

cipher\_text += (char)(ch + (26 -key));

}

else{

cipher\_text += c;

}

}

}

else{

cipher\_text += ch;

}

}

return cipher\_text;

}

public static void main(String args[]){

//System.out.println("Hello Java");

//initalizing the parameters and declaration required for decryption

String text = "HDUILPGT HTRJGXIN XH WPGS";

//System.out.println("enter the string: + %s", text);

//System.out.println("text----->" + text);

//calling the decryption function in main function

String cipher = decrypt\_string(text,15);

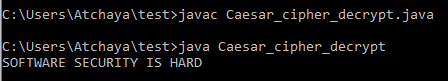
//printing the resulting plain text

System.out.println(cipher);

}

}

Output:



ii) Decryption of message using Transposition cipher in java

import java.util.\*;

import java.io.\*;

import java.lang.\*;

public class Transposition\_cipher\_decrypt {

public static void main(String[] args) {

String encrypted\_text = "IEUBOIPI NSCCLF DTTETISL ERASWMSE";

String key = "DEER";

System.out.println("Enter Encrypted String:");

System.out.println(encrypted\_text);

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

System.out.println(key);

System.out.println(decryptCT(key, encrypted\_text));

public static String decryptCT(String key, String text) {

int[] arrange = arrangeKey(key);

int lenkey = arrange.length;

int lentext = text.length();

int row = (int) Math.ceil((double) lentext / lenkey);

String regex = "(?<=\\G.{" + row + "})";

String[] get = text.split(regex);

char[][] grid = new char[row][lenkey];

for (int x = 0; x < lenkey; x++) {

for (int y = 0; y < lenkey; y++) {

if (arrange[x] == y) {

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

grid[z][y] = get[arrange[y]].charAt(z);

}

}

}

}

String dec = "";

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

for (int y = 0; y < lenkey; y++) {

dec = dec + grid[x][y];

}

}

return dec;

}

public static int[] arrangeKey(String key) {

//arrange position of grid

String[] keys = key.split("");

Arrays.sort(keys);

int[] num = new int[key.length()];

for (int x = 0; x < keys.length; x++) {

for (int y = 0; y < key.length(); y++) {

if (keys[x].equals(key.charAt(y) + "")) {

num[y] = x;

break;

}

}

}

return num;

}

}

iii) Decryption of message using Substitution cipher in java

public class Substitution\_cipher {

final static String key = "ZYXWVUTSRQPONMLKJIHGFEDCAB";

static String text = "SZXPVIH ZIV GVMZXRLFH";

public static void main(String[] args) {

System.out.println("Enter the text");

System.out.println(text);

System.out.println("\nDecoded: " + decode(text));

}

static String decode(String s) {

/\* creation of stringBuilder class which creates string of desired length and adds 16 characters to the trailing sequence \*/

StringBuilder sb = new StringBuilder(s.length());

/\* Conversion of string into array of characters using toCharArray function \*/

for (char c : s.toCharArray())

/\* Appending the characters of encrypted string with the index of key text to produce plain text \*/

sb.append((char) (key.indexOf((int) c) + 32));

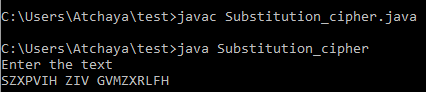
/\* returning the resulting plain text to the main function \*/

return sb.toString();

}

}

Output:



iv) Decryption of message using Vignere cipher in java

public class Vignere\_cipher\_decrypt {

public static void main(String[] args) {

//initalizing the parameters and declaration required for encryption

String vignere\_key = "SECURITY";

String vignere\_enc\_text = "LLGLV BQ QQHVG GF JUTSBLY ";

//System.out.println(vignere\_enc\_text);

//calling the decrypt\_string function inside the print function and printing the resultant plain text

System.out.println(decrypt\_string(vignere\_enc\_text, vignere\_key));

}

static String decrypt\_string(String text, final String key) {

String cipher = "";

//conversion of text to upper case

text = text.toUpperCase();

int i = 0, j = 0;

while ( i < text.length()) {

char c = text.charAt(i);

if (c < 'A' || c > 'Z' || c == '\0') continue;

//computation of cipher text of each character in the string

cipher += (char)((c - key.charAt(j) + 26) % 26 + 'A');

j = ++j % key.length();

i += 1;

}

return cipher;

}

}

Output:

