Practical No:- 2

Roll No.: 08 Date.:09/08/2022

Aim.: program to implement various classical substitution Cipher, Vigenere Cipher ,and Affine cipher.

**Code(Shift cipher):-**

//Shift Cipher

import java.io.\*;

import java.util.\*;

public class javaapplicatrion9 { //to keep track of index

public static final String ALPHA = "abcdefghijklmnopqrstuvwxyz";

public static String encrypt(String message, int shiftKey) {

message = message.toLowerCase();

String cipherText = "";

for (int ii = 0; ii < message.length(); ii++) {

int charPosition = ALPHA.indexOf(message.charAt(ii));

//System.out.println("\nchar position :"+charPosition);

int keyVal = (shiftKey + charPosition) % 26;

//System.out.println("\nKey Value :"+keyVal);

char replaceVal = ALPHA.charAt(keyVal);

//System.out.println("\nReplace Value :"+replaceVal);

cipherText += replaceVal;

}

return cipherText;

}

public static String decrypt(String message, int shiftKey) {

message = message.toLowerCase();

String plainText = "";

for (int jj = 0; jj < message.length(); jj++) {

int charPosition = ALPHA.indexOf(message.charAt(jj));

int keyVal = ( charPosition - shiftKey) % 26;

if (keyVal<0)

{

keyVal=26-(-keyVal);

}

char replaceVal = ALPHA.charAt(keyVal);

plainText += replaceVal;

}

return plainText;

} public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String message = "";

int key = 0;

System.out.print("Enter the String for Encryption:");

message = sc.next();

System.out.println("\nEnter Shift Key:");

key = sc.nextInt();

String enc\_msg = encrypt(message, key);

System.out.println("\nEncrpyted msg:" + enc\_msg);

String dec\_msg = decrypt(enc\_msg, key);

System.out.println("\nDecrypted msg:" + dec\_msg);

}

}

**Output(Shift cipher):-**

1]

Enter the String for Encryption:shubhamkhedaskarunity

Enter Shift Key:15

Encrpyted msg:hwjqwpbzwtsphzpgjcxin

Decrypted msg:shubhamkhedaskarunity

2]

Enter the String for Encryption:HiteshShubhamRajas

Enter Shift Key:4

Encrpyted msg:lmxiwlwlyfleqvenew

Decrypted msg:hiteshshubhamrajas

**Code(Affine cipher):-**

import java.util.Scanner;

class AffineCipher

{

// Key values of a and b

//can be 1,3,5,7,9,11,15,17,19,21,23,25

static int a = 3;

static int b = 7;

static String encryptMessage(char[] msg,int key1,int key2)

{

a=key1;

b=key2;

/// Cipher Text initially empty

String cipher = "";

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

{

// Avoid space to be encrypted

/\* applying encryption formula ( a x + b ) mod m

{here x is msg[i] and m is 26} and added 'A' to

bring it in range of ascii alphabet[ 65-90 | A-Z ] \*/

if (msg[i] != ' ')

{

cipher = cipher

+ (char) ((((a \* (msg[i] - 'A')) + b) % 26) + 'A');

} else // else simply append space character

{

cipher += msg[i];

}

}

return cipher;

}

static String decryptCipher(String cipher)

{

String msg = "";

int a\_inv = 0;

int flag = 0;

//Find a^-1 (the multiplicative inverse of a

//in the group of integers modulo m.)

for (int i = 0; i < 26; i++)

{flag = (a \* i) % 26;

//System.out.println(i);

// Check if (a\*i)%26 == 1,

// then i will be the multiplicative inverse of a

if (flag == 1)

{

a\_inv = i;

//System.out.println(i+" is selected");

}

}

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

{

/\*Applying decryption formula a^-1 ( x - b ) mod m

{here x is cipher[i] and m is 26} and added 'A'

to bring it in range of ASCII alphabet[ 65-90 | A-Z ] \*/

if (cipher.charAt(i) != ' ')

{

msg = msg + (char) (((a\_inv \*

((cipher.charAt(i) + 'A' - b)) % 26)) + 'A');

}

else //else simply append space character

{

msg += cipher.charAt(i);

}

}

return msg;

}

// Driver code

public static void main(String[] args)

{ Scanner sc = new Scanner(System.in);

String message = "";

int key1 = 0 ,key2=0;

System.out.print("Enter the String for Encryption:");

message = sc.next();

System.out.println("\nEnter first Key:");

key1 = sc.nextInt();

System.out.println("\nEnter second Key:");

key2 = sc.nextInt();

// Calling encryption function

String cipherText = encryptMessage(message.toCharArray(),key1,key2);

System.out.println("Encrypted Message is : " + cipherText);

// Calling Decryption function

System.out.println("Decrypted Message is: " + decryptCipher(cipherText));

}}

**Output:-**

1]

Enter the String for Encryption:hitesh

Enter first Key:3

Enter second Key:7

Encrypted Message is : UXELBU

Decrypted Message is: NOZKYN

2]

Enter the String for Encryption:shubham

Enter first Key:3

Enter second Key:5

Encrypted Message is : ZSFASXH

Decrypted Message is: YNAHNGS

**Code(Vignere):-**

import java.util.Scanner;

public class Vigner {

public static void main(String[] s){

Scanner sc = new Scanner(System.in);

String message = "";

System.out.print("Enter the String for Encryption:");

message = sc.next();

message = message.toLowerCase();

char msg[] = new char[message.length()];

msg = getChars(message);

System.out.println(msg);

String key2 = null;

System.out.print("Enter the key:");

key2 = sc.next();

key2 = key2.toUpperCase();

char key[] = new char[key2.length()];

key = getChars(key2);

System.out.println(key);

int msgLen = msg.length, i, j;

System.out.println("length is "+msgLen);

char newKey[] = new char[msgLen];

char encryptedMsg[] = new char[msgLen];

char decryptedMsg[] = new char[msgLen];

//generate new key in cyclic manner equal to the length of original message

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

if(j == key.length)

j = 0;

newKey[i] = key[j];

}

//encryption

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

encryptedMsg[i] = (char)(((msg[i] + newKey[i]) % 26) + 'A');

}

//decryption

for(i = 0; i < msgLen; ++i)

decryptedMsg[i] = (char)((((encryptedMsg[i] - newKey[i]) + 26) % 26) + 'A');

System.out.println("Original Message: " + String.valueOf(msg)); //String.valueOf() converts character array to String

System.out.println("Key: " + String.valueOf(key));

System.out.println("Generated Key: " + String.valueOf(newKey));

System.out.println("\nEncrypted Message: " + String.valueOf(encryptedMsg));

System.out.println("\nDecrypted Message: " + String.valueOf(decryptedMsg));

}

private static char[] getChars(String message) {

char msg1[] = new char[message.length()];

for (int ii = 0; ii < message.length(); ii++) {

char c = message.charAt(ii);

if(c != ' '){

msg1[ii] = c;

}

}

return msg1;

}

}

**Output:-**

1]

Enter the String for Encryption:SHUBHAM

Enter the key:GRO

Original Message: shubham

Key: GRO

Generated Key: GROGROG

Encrypted Message: EEONEUY

Decrypted Message: YNAHNGS

2]

Enter the String for Encryption:HITESH

Enter the key:fiz

Original Message: hitesh

Key: FIZ

Generated Key: FIZFIZ

Encrypted Message: SWYPGM

Decrypted Message: NOZKYN