

PARTICULARS OF THE EXPERIMENTS PERFORMED



Aim:- Write a program in java to implement the following substitution techniques.

- Caesar Cipher Technique
- Monoalphabetic Technique

=> a] Caesar Cipher:-

- Caesar cipher is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed numbers of positions down the alphabet.
- The method is named after its inventor Julius Caesar.
- This type of substitution is represented by a mathematical equation involving modular arithmetic.

Formula for encryption :-

$$C = (P + x) \text{ modulo } 26$$

Formula for decryption :-

$$P = (C - x) \text{ modulo } 26$$

where , C = Cipher text letter

P = Plain text letter

x = number of positions to advance (key).

modulo = remainder of number which is gained when divided by modulus number.

26 = total number of alphabets.

* Example of encryption :-

PlainText :- HELLO Key = 2

Plain Text : A B C D E F G H I J K L M N O
 Encoding : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Plaintext : P Q R S T U V W X Y Z
 Encoding : 15 16 17 18 19 20 21 22 23 24 25

Plain Text	Encryption ($C = (P + X) \text{ modulo } 26$)	Cipher Text
H \rightarrow 7	$C = (7 + 2) \text{ modulo } 26 \rightarrow 9 \text{ modulo } 26 = 9$	9 \rightarrow J
E \rightarrow 4	$C = (4 + 2) \text{ modulo } 26 \rightarrow 6 \text{ modulo } 26 = 6$	6 \rightarrow G
L \rightarrow 11	$C = (11 + 2) \text{ modulo } 26 \rightarrow 13 \text{ modulo } 26 = 13$	13 \rightarrow N
L \rightarrow 11	$C = (11 + 2) \text{ modulo } 26 \rightarrow 13 \text{ modulo } 26 = 13$	13 \rightarrow N
O \rightarrow 14	$C = (14 + 2) \text{ modulo } 26 \rightarrow 16 \text{ modulo } 26 = 16$	16 \rightarrow Q

\therefore Cipher Text :- JGNNQ

* Example of Decryption :-

Cipher Text :- JGNNQ Key = 2

Cipher Text	Decryption [$P = (C - X) \text{ modulo } 26$]	Plain Text
J \rightarrow 9	$P = (9 - 2) \text{ modulo } 26 \rightarrow 7 \text{ modulo } 26 = 7$	7 \rightarrow H
G \rightarrow 6	$P = (6 - 2) \text{ modulo } 26 \rightarrow 4 \text{ modulo } 26 = 4$	4 \rightarrow E
N \rightarrow 13	$P = (13 - 2) \text{ modulo } 26 \rightarrow 11 \text{ modulo } 26 = 11$	11 \rightarrow L
N \rightarrow 13	$P = (13 - 2) \text{ modulo } 26 \rightarrow 11 \text{ modulo } 26 = 11$	11 \rightarrow L
Q \rightarrow 16	$P = (16 - 2) \text{ modulo } 26 \rightarrow 14 \text{ modulo } 26 = 14$	14 \rightarrow O

\therefore Plaintext = HELLO

Code:-

```
import java.util.Scanner;
public class CeaserCipher
{
    public static final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";

    public static String encrypt(String plainText, int shiftKey)
    {
        plainText = plainText.toLowerCase();
        String cipherText = "";
        for (int i = 0; i < plainText.length(); i++)
        {
            int charPosition = ALPHABET.indexOf(plainText.charAt(i));
            int keyVal = (shiftKey + charPosition) % 26;
            char replaceVal = ALPHABET.charAt(keyVal);
            cipherText += replaceVal;
        }
        return cipherText;
    }

    public static String decrypt(String cipherText, int shiftKey)
    {
        cipherText = cipherText.toLowerCase();
        String plainText = "";
        for (int i = 0; i < cipherText.length(); i++)
        {
            int charPosition = ALPHABET.indexOf(cipherText.charAt(i));
            int keyVal = (charPosition - shiftKey) % 26;
            if (keyVal < 0)
            {
                keyVal = ALPHABET.length() + keyVal;
            }
        }
    }
}
```



```
char replaceVal = ALPHABET.charAt(keyVal);
plainText = replaceVal;
}
return plainText;
}

public static void main (String [] args)
{
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter the string for encryption:");
    String message = new String ();
    message = sc.next ();
    System.out.println ("Encrypted text is: " + encrypt (message, 3));
    System.out.println ("Decrypted text is: " + decrypt (
        encrypt (message, 3), 3));
    sc.close ();
}
```

B] Mono-alphabetic Substitution :-

- Mono-alphabetic cipher is a substitution cipher in which for a given key the cipher alphabet for each plain alphabet is fixed throughout the encryption process.

e.g. A B C D E
Y T V G H

then wherever A appears in list of plainText it is replaced by Y.

- Mono-alphabetic substitution is a technique where 'mono' means one i.e. it represents one to one relationship where one alphabet is replaced by another alphabet.

e.g. :- If we want to encrypt the plainText 'NEELIMA PADMAWAR' by substituting A with Z, B with Y likewise, so following is Solution :-

PlainText :- a b c d e f g h i j k l m n
Key :- z y x w v u t s r q p o n m

PlainText :- o p q r s t u v w x y z
Cipher key :- l k j i h g f e d c b a

So, NEELIMA PADMAWAR will become
M V V O R N Z K Z W N Z D Z I i.e. Cipher text

Code:-

```
import java.io.*;
class Monoalphabetic
{
    public static char normalChar[] = {'a','b','c','d','e','f',
    'g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x',
    'y','z'};
    public static char codedChar[] = {'P','Q','R','S','T','A','B','C',
    'D','V','W','X','Y','Z','L','M','N','O','E','F','G','H','I','J',
    'K','U'};
    public static String stringEncryption (String s)
    {
        String encryptedString = "";
        for (int i = 0; i < s.length(); i++)
        {
            for (int j = 0; j < 26; j++)
            {
                if (s.charAt(i) < 'a' || s.charAt(i) > 'z')
                {
                    encryptedString += s.charAt(i);
                    break;
                }
            }
        }
        return encryptedString;
    }
    public static String Decryption (String s)
    {
        String decryptedString = "";
        for (int i = 0; i < s.length(); i++)
        {
```

```
for (int j = 0; j < 26; j++)
```

```
{
```

```
    if (s.charAt(i) == codedchar[j])
```

```
    {
```

```
        decryptedString += normalchar[j];
```

```
        break;
```

```
    }
```

```
    if (s.charAt(i) < 'A' || s.charAt(i) > 'z') {
```

```
        decryptedString += s.charAt(i);
```

```
        break;
```

```
    } }
```

```
    return decryptedString;
```

```
}
```

```
public static void main (String [] args)
```

```
{
```

```
    String str = "Hello";
```

```
    System.out.println ("plain text:" + str)
```

```
    String encryptedString = StringEncryption (str.toLowerCase());
```

```
    String
```

```
    System.out.println ("Encrypted message:" + encryptedString);
```

```
    System.out.println ("Decrypted message:" + StringDecryption  
        (encryptedString));
```

```
}
```