Lab – 1

Subject : NIS

Aim: Write a program to implement Additive Cipher and Monoalphabetic Substitution Cipher

1. Additive Cipher

Additive cipher is a one of the encryption technique of type substitution cipher.it basically shift each alphabet with key suppose your plaintext's character is 'x' and key is 2 then it will shift two position so your cipher text is 'z'.it also known as Caesar cipher or shift cipher.

Same way decryption takes place each alphabets will shift back as par the key. Cryptanalysis is very much in this kind of encryption technique, only 25 attempt or less and intruder can have most probable valid guess of plaintext.

Program: -

```
import java.util.*;
import java.io.*;
import java.lang.*;

public class Additive{

    public static void cryptAnalisys(String text){
        System.out.println("CryptAnalisys :-");
        for(int j=1;j<=25;j++)
        {
            for(int i=0;i<text.length();i++){

                char c = text.charAt(i);
                int encInt = (((int)c - 97 - j) % 26)+97;
                char e = (char)encInt;
                System.out.print(e);

            }
            System.out.println(" ");
        }

    }

    public static void encryption(String text,int key){
        System.out.println("Encryption :-");
        for(int i=0;i<text.length();i++){

           char c = text.charAt(i);
           int encInt =(((int)c-97 + key) % 26)+97;
           char e = (char)encInt;
           System.out.print(e);
```

```
        }

    }

    public static void decreption(String text,int key){
        System.out.println("Decryption :-");
        for(int i=0;i<text.length();i++){

            char c = text.charAt(i);
            int encInt = (((int)c-97 - key) % 26)+97;
            char e = (char)encInt;
            System.out.print(e);

         }
    }

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the text :");
        String text = sc.nextLine();

        System.out.println("Enter your choice 0 for Encryption and 1 for decry
ption 2 for crypanalisys :");
        int choice = sc.nextInt();

        if(choice==2){
            cryptAnalisys(text);
        }
        if(choice == 1)
        {
            System.out.println("Enter the key :");
            int key = sc.nextInt();
            decreption(text,key);
        }
        else if(choice == 0)
        {
            System.out.println("Enter the key :");
            int key = sc.nextInt();
            encryption(text,key);
        }else
        {
            System.out.println("Invalide choice !");
        }
    }
}
```

Output: -

```
D:\DDIT\sem6\NIC\LAB\lab1>javac Additive.java

D:\DDIT\sem6\NIC\LAB\lab1>java Additive
Enter the text :
hello
Enter your choice 0 for Encryption and 1 for decryption 2 for crypanalisys :
0
Enter the key :
5
Encryption :-
mjqqt
```

```
D:\DDIT\sem6\NIC\LAB\lab1>java Additive
Enter the text :
mjqqt
Enter your choice 0 for Encryption and 1 for decryption 2 for crypanalisys :
1
Enter the key :
5
Decryption :-
hello
D:\DDIT\sem6\NIC\LAB\lab1>
```

```
D:\DDIT\sem6\NIC\LAB\lab1>java Additive
Enter the text :
mjqqt
Enter your choice 0 for Encryption and 1 for decryption 2 for crypanalisys :
2
CryptAnalisys :-
lipps
khoor
jgnnq
ifmmp
hello
gdkkn
fcjjm
ebiil
dahhk
c`ggj
b_ffi
a^eeh
`]ddg
_\ccf
^[bbe
]Zaad
\Y``c
[X__b
```

2. Monoalphabetic Substitution Cipher

In monoalphabetic substitution Cipher, there is one to one mapping of the alphabets. Plaintext will be replaced by its corresponding mapped character and this is how encryption takes place. reverse process is done in decryption. cryptanalysis is not practically possible because there are 26! Possible outcomes and it is hard to find plaintext from this large number of outcames.

Program: -

```
import java.io.*;
import java.util.*;
import java.lang.*;

public class Substitution{

    public static void encryption(String text,HashMap<String,String> map){
        System.out.println("Encryption :-");
        for(int i=0;i<text.length();i++){

            String s = Character.toString(text.charAt(i));
            for(Map.Entry<String,String> entry : map.entrySet()){

                if(entry.getKey().equals(s)){
                    System.out.print(entry.getValue());
                }
            }
        }
    }

    public static void decreption(String text,HashMap<String,String> map){
        System.out.println("\nDecryption :-");
        for(int i=0;i<text.length();i++){

            String s = Character.toString(text.charAt(i));
            for(Map.Entry<String,String> entry : map.entrySet()){

                if(entry.getValue().equals(s)){
                    System.out.print(entry.getKey());
                }
            }
        }
    }

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the text :");
        String text = sc.nextLine();
```

```
        HashMap<String,String> map = new HashMap<>();
        map.put("a","z");
        map.put("b","p");
        map.put("c","o");
        map.put("d","n");
        map.put("e","m");
        map.put("f","l");
        map.put("g","k");
        map.put("h","j");
        map.put("i","i");
        map.put("j","h");
        map.put("k","g");
        map.put("l","f");
        map.put("m","e");
        map.put("n","d");
        map.put("o","c");
        map.put("p","b");
        map.put("q","a");
        map.put("r","q");
        map.put("s","r");
        map.put("t","s");
        map.put("u","t");
        map.put("v","u");
        map.put("w","v");
        map.put("x","y");
        map.put("y","x");
        map.put("z","w");

        encryption(text,map);
        decreption(text,map);

    }
}
```

Output: -

```
D:\DDIT\sem6\NIC\LAB\lab1>java Substitution
Enter the text :
hello
Encryption :-
jmffc
Decryption :-
jmffc
D:\DDIT\sem6\NIC\LAB\lab1>java Substitution
```

```
D:\DDIT\sem6\NIC\LAB\lab1>java Substitution
Enter the text :
jmffc
Encryption :-
hello
Decryption :-
hello
D:\DDIT\sem6\NIC\LAB\lab1>
```