

## Lab – 1

### Subject : NIS

**Aim: Write a Program to implement following: 1. Extended Euclidean algorithm to find multiplicative inverse 2. Multiplicative cipher 3. Affine Cipher**

#### 1. Multiplicative cipher

Multiplicative cipher is same as shift cipher, the difference is addition and multiplication. In shift cipher key is added with plain text and in multiplicative cipher key is multiplied with the plain text. So it is about encryption.

$$CT = PT * \text{key} \bmod n$$

For decryption, multiplication is done with the inverse of the key with respect to mod n.

$$PT = CT * \text{key}^{-1} \bmod n$$

**Program: -**

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

public class MultiplicativeCipher{

    public static void encryption(String text,int key,int n){

        System.out.println("Encryption is :");
        for(int i=0;i<text.length();i++)
        {
            char c = text.charAt(i);
            int encInt = (int)c;
            // System.out.println(encInt);
            encInt = (((encInt-97) * key) % n)+97;
            char e = (char)encInt;
            // System.out.println(encInt);
            System.out.print(e);
        }

    }

    public static int[] extendedEuclidian(int a,int n){
        int[] arr = new int[2];

        int r1=n,r2=a,r,t,t1=0,t2=1,gcd,inverse,q;

        while(r2 > 0){
```

```

        q=r1/r2;
        r=r1-q*r2;
        r1=r2;
        r2=r;

        t=t1-q*t2;
        t1=t2;
        t2=t;
    }
    gcd=r1;
    inverse=t1;
    if(inverse < 0){
        inverse = positiveInvers(inverse,n);
    }
    arr[0]=inverse;
    arr[1]=gcd;

    return arr;
}

public static int positiveInvers(int inverse,int n){

    while(inverse < 0){
        inverse = inverse + n;
    }
    return inverse;
}

public static void decreption(String text,int key,int n){
    System.out.println("\nDecryption is :");
    int inverse[] = extendedEuclidian(key,n);

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

        char c = text.charAt(i);
        int encInt = (int)c;
        int ctIntoInvers = (((encInt-97) * inverse[0]) % n)+97;

        if(ctIntoInvers < 0){
            ctIntoInvers = positiveInvers(ctIntoInvers,n);
        }

        char e = (char)ctIntoInvers;
        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 the key :");
int key = sc.nextInt();
System.out.println("Enter n : ");
int n = sc.nextInt();

encryption(text,key,n);
decreption(text,key,n);
}
}

```

**Output: -**

```

D:\DDIT\sem6\NIS\LAB\lab2>java MultiplicativeCipher
Enter the text :
chetanmaher
Enter the key :
5
Enter n :
26
Encryption is :
kjuraniajuh
Decryption is :
qrgjansargt

```

```

D:\DDIT\sem6\NIS\LAB\lab2>java MultiplicativeCipher
Enter the text :
kjuraniajuh
Enter the key :
5
Enter n :
26
Encryption is :
ytwhanoatwj
Decryption is :
chetanmaher

```

## 2. Affine Cipher

Affine Cipher is combination of the multiplicative and additive Cipher. In encryption of the affine cipher is done with two keys (key1 and key2). key1 is first multiplied with the plain text and then key2 is added into the plain text. then take modulo with respect to mod n where n is positive integer.

$$CT = (PT * key1) + key2 \bmod n$$

In Decryption first subtraction of the key2 is done with cipher text, then multiply key1's inverse with the (CT-key2) and take mod of whole with respect to n where n is positive integer.

$$PT = (CT - key2) * key1^{-1} \bmod n$$

### Program: -

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

public class AffineCipher{

    public static void encryption(String text,int k1,int k2,int n){
        System.out.println("Encryption is :");

        for(int i=0;i<text.length();i++){
            char c = text.charAt(i);
            int encInt =(int)c;
            encInt = ( ( (encInt-97) * k1 + k2 ) % n )+97;
            char e = (char)encInt;
            System.out.print(e);
        }
    }

    public static void decryption(String text,int k1,int k2,int n){
        System.out.println("\nDescription is :");

        int inverse[] = extendedEuclidian(k1,n);
        //System.out.println("\ninverse is : " + inverse[0]);

        for(int i=0;i<text.length();i++){
            char c = text.charAt(i);
            //System.out.println("\nchar c is : " + c);
            int encInt = (int)c;
            //System.out.println("\nencInt is : " + encInt);
            encInt = encInt - k2-97;
            //System.out.println("\nCT-k2 : " + encInt);
            int ctIntoInvers = (((encInt) * inverse[0]) % n)+97;
            //System.out.println("\n(CT-k2)*invers % 26 is :"+ ctIntoInvers);
            if(ctIntoInvers < 0){
                ctIntoInvers = positiveInvers(ctIntoInvers,n);
            }
            char e = (char)ctIntoInvers;
            System.out.print(e);
        }
    }

    //Extended Euclidian algorithm
```

```

public static int[] extendedEuclidian(int a,int n){

    int[] arr = new int[2];

    int r1=n,r2=a,r,t,t1=0,t2=1,gcd,inverse,q;

    while(r2 > 0){
        q=r1/r2;
        r=r1-q*r2;
        r1=r2;
        r2=r;

        t=t1-q*t2;
        t1=t2;
        t2=t;
    }
    gcd=r1;
    inverse=t1;
    if(inverse < 0){
        inverse = positiveInvers(inverse,n);
    }
    arr[0]=inverse;
    arr[1]=gcd;

    return arr;
}

public static int positiveInvers(int inverse,int n){

    while(inverse < 0){
        inverse = inverse + n;
    }
    return inverse;
}

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 the key1 :");
    int k1 = sc.nextInt();
    System.out.println("Enter the key2 :");
    int k2 = sc.nextInt();

    System.out.println("Enter n : ");
    int n = sc.nextInt();
}

```

```
        encryption(text,k1,k2,n);  
        decreption(text,k1,k2,n);  
    }  
}
```

### Output: -

```
D:\DDIT\sem6\NIS\LAB\lab2>java AffineCipher  
Enter the text :  
hiiii  
Enter the key1 :  
5  
Enter the key2 :  
8  
Enter n :  
26  
Encryption is :  
rwww  
Description is :  
Laaa
```

```
D:\DDIT\sem6\NIS\LAB\lab2>java AffineCipher  
Enter the text :  
rwww  
Enter the key1 :  
5  
Enter the key2 :  
8  
Enter n :  
26  
Encryption is :  
pooo  
Description is :  
hiiii
```