**Experiment** **6**

**Aim**: Write a program to implement left recursion removal algorithm.

**Tools**: java compiler, Text editor.

**Java code: -**

import java.util.\*;

public class lf3

{

static void plcr(char c,String X, String Y, int m, int n)

{

int[][] LS = new int[m + 1][n + 1];

String X1 = "";

String Y1 = "";

int len = 0;

int row = 0, col = 0;

int q = 0;

int i,j;

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

for (j = 0; j <= n; j++) {

if (i == 0 || j == 0)

LS[i][j] = 0;

else if (X.charAt(i - 1) == Y.charAt(j - 1)) {

LS[i][j] = LS[i - 1][j - 1] + 1;

if (len < LS[i][j]) {

len = LS[i][j];

if(i != row+1)

break;

row = i;

col = j;

}

}

else{

LS[i][j] = 0;

}

}

}

//System.out.print(i+" "+m);

for (int k = i; k<m; k++)

System.out.print(X.charAt(k));

System.out.println();

if (len == 0) {

System.out.println(c+"->"+X+"|"+Y);

return;

}

String rS = "";

while (LS[row][col] != 0) {

rS = X.charAt(row - 1) + rS;

--len;

row--;

col--;

}

int len1 = rS.length();

//System.out.println(len1);

System.out.println(c+"->"+rS+c+"'");

for(int k = m; k>len1; k--)

X1= X.charAt(k - 1) + X1;

for(int k = n; k>len1; k--)

Y1= Y.charAt(k - 1) + Y1;

System.out.println(c+"'->"+X1+"|"+Y1);

}

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Non-Terminal: ");

char c = sc.next().charAt(0);

System.out.print("Enter the number of Strings: ");

String[] string = new String [sc.nextInt()];

sc.nextLine();

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

{

string[i] = sc.nextLine();

}

String X,Y;

X = string[0];

Y = string[1];

int m = X.length();

int n = Y.length();

plcr(c,X, Y, m, n);

}

}

**Output: -**

