A paper with text and numbers

Description automatically generated

A screenshot of a math test

Description automatically generated

A screenshot of a computer exam

Description automatically generated

import java.util.Vector;

public class LongestCommonSubsequence {

    public Vector<String> LCSLength(Vector<String> X, Vector<String> Y) {

        int m = X.size();

        int n = Y.size();

        int[][] b = new int[m][n];

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

        for (int i = 1; i <= m; i++) {

            c[i][0] = 0;

        }

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

            c[0][j] = 0;

        }

        for (int i = 1; i <= m; i++) {

            for (int j = 1; j <= n; j++) {

                if (X.get(i - 1).equals(Y.get(j - 1))) {

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

                    b[i - 1][j - 1] = 1;

                } else if (c[i - 1][j] >= c[i][j - 1]) {

                    c[i][j] = c[i - 1][j];

                    b[i - 1][j - 1] = 2;

                } else {

                    c[i][j] = c[i][j - 1];

                    b[i - 1][j - 1] = 3;

                }

            }

        }

        Vector<String> result = new Vector<>();

        PrintLCS(b, X, m, n, result);

        return result;

    }

    public void PrintLCS(int[][] b, Vector<String> X, int i, int j, Vector<String> result) {

        if (i == 0 || j == 0) {

            return;

        }

        if (b[i - 1][j - 1] == 1) {

            PrintLCS(b, X, i - 1, j - 1, result);

            result.add(X.get(i - 1)); // Add the common element to the result vector

        } else if (b[i - 1][j - 1] == 2) {

            PrintLCS(b, X, i - 1, j, result);

        } else {

            PrintLCS(b, X, i, j - 1, result);

        }

    }

    public static void main(String[] args) {

        // Example vectors

        Vector<String> vectorX = new Vector<>();

        vectorX.add("A");

        vectorX.add("B");

        vectorX.add("C");

        vectorX.add("B");

        vectorX.add("D");

        vectorX.add("A");

        vectorX.add("B");

        Vector<String> vectorY = new Vector<>();

        vectorY.add("B");

        vectorY.add("D");

        vectorY.add("C");

        vectorY.add("A");

        vectorY.add("B");

        vectorY.add("A");

        LongestCommonSubsequence lcs = new LongestCommonSubsequence();

        Vector<String> result = lcs.LCSLength(vectorX, vectorY);

        // Print the original vectors

        System.out.println("Vector 1: " + vectorX);

        System.out.println("Vector 2: " + vectorY);

        // Print the Longest Common Subsequence

        System.out.println("Longest Common Subsequence:");

        for (String element : result) {

            System.out.print(element + " ");

        }

    }

}

CODE OUTPUT:

A screen shot of a computer

Description automatically generated

OPTIMIZATION:

import java.util.Vector;

public class optimization {

    public static void print\_LCS(Vector<Vector<Integer>> cInput,String xInput, String yInput, int i, int j ){

        if(i==0 || j == 0){

            System.out.println();

            return;

        }

        String direction ="";

        int cCurrent = cInput.get(i).get(j);

        int dValue = cInput.get(i-1).get(j-1);

        int uValue = cInput.get(i-1).get(j);

        int lValue = cInput.get(i).get(j-1);

        Character myXinputChar = xInput.charAt(i-1);

        Character myYinputChar = yInput.charAt(j-1);

        // default: the element to the left

        direction = "l";

        if(myXinputChar.equals(myYinputChar)){

                direction = "d";

            }

        else if(uValue >= lValue){

            direction = "u";

        }

            System.out.println("The direction is: " + direction);

            if(direction.equals("d")){

                print\_LCS(cInput, xInput, yInput, i-1, j-1);

                System.out.print(myXinputChar);

            }

            else if(direction.equals("u")){

                print\_LCS(cInput, xInput, yInput, i-1, j);

            }

            else if(direction.equals("l")){

                print\_LCS(cInput, xInput, yInput, i, j-1);

            }

        }

        public static void main(String[] args) {

            String xInput = "ABCBDAB";

            String yInput = "BDCABA";

            Vector<Vector<Integer>> cTable = new Vector<Vector<Integer>>();

            // initiliaze the tables

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

                Vector<Integer> cRow = new Vector<Integer>();

                for(int j = 0; j <= yInput.length(); j++){

                    cRow.add(0);

                }

                cTable.add(cRow);

            }

            // consider only C table

            for(int i = 1; i <= xInput.length(); i++){

                for(int j = 1; j <= yInput.length(); j++){

                    Character myXinputChar = xInput.charAt(i-1);

                    Character myYinputChar = yInput.charAt(j-1);

                    if(myXinputChar.equals(myYinputChar)){

                        cTable.get(i).set(j, cTable.get(i-1).get(j-1) + 1);

                    }

                    else if(cTable.get(i-1).get(j) >= cTable.get(i).get(j-1)){

                        cTable.get(i).set(j, cTable.get(i-1).get(j));

                    }

                    else{

                        cTable.get(i).set(j, cTable.get(i).get(j-1));

                    }

                }

            }

            print\_LCS(cTable, xInput, yInput, 0, 0);

        }

    }

ALL CODE WORKS AS INTENDED !