```
import java.util.Scanner;

class CLCSubseqSubstr
{
  public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

       System.out.print("Input the first string: ");

       String X = input.nextLine();

       System.out.print("Input the second string: ");

       String Y = input.nextLine();

        System.out.print("Input the constrained string: ");

        String P = input.nextLine();

       int m = X.length(), n = Y.length(), r = P.length();

       int[][][] M1 = new int[m + 1][n + 1][r + 1];

       System.out.println();

       System.out.println("A longest string which is a " +
                          "subsequence of " + X +
                          ", a substring of " + Y +
```

```
                ", and has " + P + " as a subsequence is " +
                CLCSS(X, Y, P, m, n, r, M1));

    System.out.println();

    System.out.println("The length of a longest string " +
                "which is a subsequence of " + X +
                ", a substring of " + Y + ", and has " + P +
                " as a subsequence is " +
                CLCSS(X, Y, P, m, n, r, M1).length());
        }

public static String CLCSS(String X, String Y, String P, int m, int n, int r, int[][][]M)
        {
          M = new int[m + 1][n + 1][r + 1];

          for (int i = 1; i <= m; i++)
           {
                for (int j = 1; j <= n; j++)
                        {
                                if (X.charAt(i - 1) == Y.charAt(j - 1))
                                  {
                                        M[i][j][0] = M[i - 1][j - 1][0] + 1;
                                  }
                                else
                                  {
                                        M[i][j][0] = M[i - 1][j][0];
                                  }
```

```
                    }
        }


for (int j = 0; j <= n; j++)
    {
        for (int k = 1; k <= r; k++)
                {
                   M[0][j][k] = -100*(m + n);
                }
    }

 for (int i = 0; i <= m; i++)
    {
        for (int k = 1; k <= r; k++)
                {
                    M[i][0][k] = -100*(m + n);
                }
    }


for (int i = 1; i <= m; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            for (int k = 1; k <= r; k++)
            {
                if (X.charAt(i - 1) == Y.charAt(j - 1))
                  {
```

```
                                    if (X.charAt(i - 1) == P.charAt(k - 1))
                                            M[i][j][k] = M[i - 1][j - 1][k - 1] + 1;
                                    else
                                            M[i][j][k] = M[i - 1][j - 1][k] + 1;
                                }
                                else
                                {
                                        M[i][j][k] = M[i - 1][j][k];
                                }
                        }
                }
        }

 int maxLength = 0;     // keeps the max length of CLCSS

 int lastIndexOnY = n; // keeps the last index of
                                // CLCSS which is a substring in Y

for (int i = 1; i <= m; i++)
        {
                for (int j = 1; j <= n; j++)
                {
                        if (M[i][j][r] > maxLength)
                        {
                                maxLength = M[i][j][r];
                                lastIndexOnY = j;
                        }
                }
        }
```

```
        return Y.substring(lastIndexOnY - maxLength,  lastIndexOnY);
    }
}
```

```
/* A sample run of the program.

C:\Users\raol\OneDrive - University of South Carolina
Aiken\Desktop\LC_Subseq_Substr\CLC_Subseq_Substr>javac CLCSubseqSubstr.java

C:\Users\raol\OneDrive - University of South Carolina
Aiken\Desktop\LC_Subseq_Substr\CLC_Subseq_Substr>java CLCSubseqSubstr

Input the first string: abxycdz

Input the second string: abczw

Input the constrained string: bz

A longest string which is a subsequence of abxycdz, a substring of abczw, and has bz as a subsequence is
abcz

The length of a longest string which is a subsequence of abxycdz, a substring of abczw, and has bz as a
subsequence is 4

C:\Users\raol\OneDrive - University of South Carolina
Aiken\Desktop\LC_Subseq_Substr\CLC_Subseq_Substr>

*/
```