

```
import java.util.*;
```

```
public class LongestCommonSubsequence {
```

```
    public static int lcs(String str1, String str2) {
        int m = str1.length();
        int n = str2.length();

        int[][] dp = new int[m + 1][n + 1];
        for (int i = 0; i <= m; i++) {
            for (int j = 0; j <= n; j++) {
                if (i == 0 || j == 0) {
                    dp[i][j] = 0;
                } else if (str1.charAt(i - 1) == str2.charAt(j - 1)) {
                    dp[i][j] = dp[i - 1][j - 1] + 1;
                } else {
                    dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
                }
            }
        }
        return dp[m][n];
    }
}
```

```
    public static String getLCS(String str1, String str2) {
        int m = str1.length();
        int n = str2.length();

        int[][] dp = new int[m + 1][n + 1];
        for (int i = 0; i <= m; i++) {
            for (int j = 0; j <= n; j++) {
                if (i == 0 || j == 0) {
                    dp[i][j] = 0;
                } else if (str1.charAt(i - 1) == str2.charAt(j - 1)) {
                    dp[i][j] = dp[i - 1][j - 1] + 1;
                } else {
                    dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
                }
            }
        }
    }
```

```
    int i = m, j = n;
    StringBuilder lcs = new StringBuilder();
    while (i > 0 && j > 0) {
        if (str1.charAt(i - 1) == str2.charAt(j - 1)) {
            lcs.append(str1.charAt(i - 1));
        }
    }
```

```

        i--;
        j--;
    } else if (dp[i - 1][j] > dp[i][j - 1]) {
        i--;
    } else {
        j--;
    }
}
return lcs.reverse().toString();
}

```

```

public static void countFrequency(String lcs, char[] letters) {
    for (char letter : letters) {
        int count = 0;
        for (char c : lcs.toCharArray()) {
            if (c == letter) {
                count++;
            }
        }
        System.out.println("Frequency of '" + letter + "': " + count);
    }
}

```

```

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

    System.out.print("Enter your full name (without spaces): ");
    String str1 = scanner.nextLine();

    System.out.print("Enter your friend's name (without spaces): ");
    String str2 = scanner.nextLine();

    String lcsString = getLCS(str1, str2);
    System.out.println("LCS: " + lcsString);

    countFrequency(lcsString, new char[]{'A', 'B'});

    scanner.close();
}
}

```

٣٠:٣٤ م الأربعاء ١٨ سبتمبر

onecompiler.com

ENGLISH العربية

OneCompiler

EDITOR CHALLENGES ORGS COMPANY & MORE

LOGIN

LongestCommonSubsequence.java

42sjcsuem

NEW JAVA RUN

```
1 import java.util.*;
2
3
4 public class LongestCommonSubsequence {
5
6     public static int lcs(String str1, String str2) {
7         int m = str1.length();
8         int n = str2.length();
9
10        int[][] dp = new int[m + 1][n + 1];
11        for (int i = 0; i <= m; i++) {
12            for (int j = 0; j <= n; j++) {
13                if (i == 0 || j == 0) {
14                    dp[i][j] = 0;
15                } else if (str1.charAt(i - 1) == str2.charAt(j - 1)) {
16                    dp[i][j] = dp[i - 1][j - 1] + 1;
17                } else {
18                    dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
19                }
20            }
21        }
22        return dp[m][n];
23    }
24
25    public static String getLCS(String str1, String str2) {
26        int m = str1.length();
27        int n = str2.length();
28
29        int[][] dp = new int[m + 1][n + 1];
```

STDIN

SHIMAALafi  
SARAHMohammed

Output:  
Enter your full name (without spaces): Enter  
Frequency of 'A': 2  
Frequency of 'B': 0