**B03902062 資工二 董文捷**

We can get **dp[ m ][ n ]** by **dp[m – 1][n – 1]**, **dp[m – 1][ n ]**, **dp[ m ][n – 1]**

If the **m**th character of string1 is the same as **n**th character of string2, **dp[ m ][ n ]** should be **dp[m – 1][n – 1] + 1**, else it should be the larger value in **dp[m – 1][ n ]** and **dp[ m ][n – 1]**

**ANS:**

| | a | c | a | b | c | c | a | b | a |
|---|---|---|---|---|---|---|---|---|---|
| b | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| c | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| a | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 |
| a | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 4 |
| b | 1 | 1 | 2 | 3 | 3 | 3 | 3 | 4 | 4 |
| a | 1 | 1 | 2 | 3 | 3 | 3 | 4 | 4 | 5 |
| c | 1 | 2 | 2 | 3 | 4 | 4 | 4 | 4 | 5 |
| c | 1 | 2 | 2 | 3 | 4 | 5 | 5 | 5 | 5 |

## My code in C++

```
#include <stdio.h>
const int length1 = 8;
const int length2 = 9;
int main()
{
    char s1[length1 + 1] = "bcaabacc";
    char s2[length2 + 1] = "acabccaba";
    int dp[length1 + 1][length2 + 1] = {{0}};
    int i, j;
    for(i = 1; i <= length1; i++)
    {
        putchar(s1[i - 1]);
        for(j = 1; j <= length2; j++)
        {
            if(s1[i - 1] == s2[j - 1])
                dp[i][j] = dp[i - 1][j - 1] + 1;
            else if(dp[i][j - 1] > dp[i - 1][j])
                dp[i][j] = dp[i][j - 1];
            else
                dp[i][j] = dp[i - 1][j];
            printf(" %d", dp[i][j]);
        }
        putchar('\n');
    }
    return 0;
}
```