**Descripción**

Al finalizar su viaje por cuba, Edgardo se puso a pensar acerca de problemas mas interesantes que sus alumnos podrian resolver.

En esta ocasión tu trabajo es el siguiente:

Dadas 2 cadenas A y B, debes de encontrar la subsecuencia común mas larga entre ambas cadenas.

**Entrada**

La primera linea contendra la cadena A. En la segunda linea vendra la cadena B.

**Salida**

La longitud de la subsecuencia común mas larga.

**Ejemplo**

| **Entrada** | **Salida** | **Descripción** |
| --- | --- | --- |
| AGCT  AMGXTP | 3 | La subsecuencia comun mas larga es: AGTAGT |

**Límites**

* 1≤|A|≤1031≤|A|≤103
* 1≤|B|≤103

1. #include <bits/stdc++.h>
2. **using** **namespace** std;
4. //DECLARAMOS LAS CADENAS QUE VAMOS A USAR
5. string X, Y;

8. **int** lcs()
9. {
10. **int** m = X.length();
11. **int** n = Y.length();
12. **int** L[m+1][n+1];
13. **int** i, j;
15. /\* siguiendo los paso construimos la matriz  L[m+1][n+1] del modo bottom up  \*/
17. **for** (i=0; i<=m; i++)
18. {
19. **for** (j=0; j<=n; j++)
20. {
22. **if** (i == 0 || j == 0)
23. L[i][j] = 0;
24. //Nos ayuda a validar que no se repitan los calculos
25. **else** **if** (X[i-1] == Y[j-1])
26. L[i][j] = L[i-1][j-1] + 1;
27. //Calculamos el valor maximo
28. **else**
29. L[i][j] = max(L[i-1][j], L[i][j-1]);
30. }
31. }
33. **return** L[m][n];
34. }
36. **int** main()
37. {
39. ios\_base::sync\_with\_stdio(**false**);
40. cin.tie(NULL);
42. //LEEMOS LAS CADENAS
44. getline(cin, X);
45. getline(cin, Y);

48. cout << lcs() << '\n';
50. **return** 0;
51. }

