

```

PLSSC-ProgDyn(A, B)
  n ← longueur (A)      //longueur 1ere suite
  m ← longueur (B)      // longueur 2eme chaîne
  pour i ← 1 à n faire
    l[i, 0] ← 0          // initialisation avec des 0 de la première colonne
  fin pour
  pour j ← 1 à m faire   // initialisation avec des 0 de la première ligne
    l[0, j] ← 0
  fpour
  pour i ← 1 à n faire
    pour j ← 1 à m faire // remplissage de la matrice ligne après ligne
      si A[i] = B[j]
        alors l[i, j] ← 1 + l[i - 1, j - 1]
        s[i, j] ← « = »
      sinon si l[i - 1, j] > l[i, j - 1])
        alors l[i, j] ← l[i - 1, j]
        s[i, j] ← « ← »
      sinon l[i, j] ← l[i, j - 1]
        s[i, j] ← « → »
      fin si
    renvoyer l[n, m] et s
  fin

```

```

Algo Affichage-PLSC(s, A, i, j)
si i = 0 ou j = 0 alors renvoyer
si s[i, j] = « = »
alors Affichage-PLSC(s, A, i - 1, j - 1)
affiche A[i]
sinon si s[i, j] = « ← » alors Affichage-PLSC(s, A, i - 1, j)
si s[i, j] = « → » alors Affichage-PLSC(s, A, i, j - 1)

```