問3　二つの列の最長共通部分列（Longest Common Subsequence）の長さを求めるアルゴリズムに関する次の記述を読んで，設問に答えよ。

列 X＝{x_1,　x_2,　…,　x_n}に対して，順序を保持して要素を抽出した列を部分列という。また，列の長さはその列の要素の個数で定義される。ここでは，各要素が文字である列を考える。例えば，X＝{ “A”，　“B”，　“C”，　“B”，　“D”，　“A”，　“B” }のとき，図 1 に示すように{ “B”，　“C”，　“D”，　“B” }は X の部分列の例であり，その長さは 4 である。

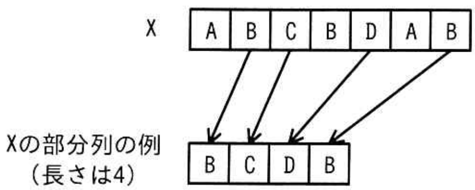

図 1　部分列の例

ある列 Z が二つの列 X，Y 両方の部分列であるとき，Z を X と Y との共通部分列といい，共通部分列のうち長さが最大となるものを最長共通部分列という。最長共通部分列は複数通り存在する場合もあるが，その長さは一意に決まる。X＝{ “A”，“B”，　“C”，　“B”，　“D”，　“A”，　“B” }，Y＝{ “B”，　“D”，　“C”，　“A”，“B”，　“A” }の場合の共通部分列及び最長共通部分列の例を図 2 に示す。

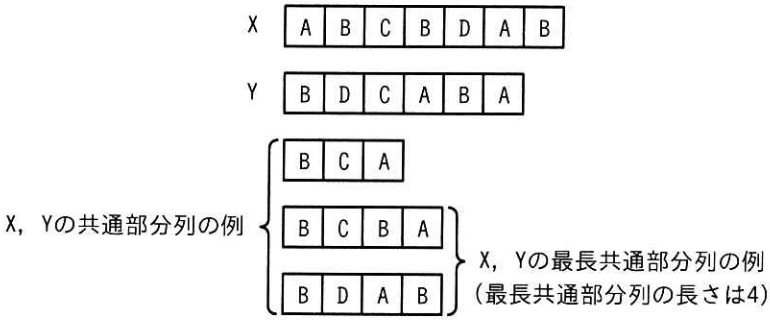

図 2　共通部分列及び最長共通部分列の例

なお，共通部分列が存在しない場合，最長共通部分列は空の列となり，その長さ

は 0 である。

最長共通部分列の長さは，2 本の DNA の塩基配列間の類似度を測る目的などに用いられる。

〔最長共通部分列の長さを求めるアルゴリズム〕

二つの列 X，Y の最長共通部分列の長さを求めるアルゴリズムを考える。列 X，Y それぞれについて，先頭から n 個，k 個の要素を抽出した列を X_n，Y_k と表記し，列 X_n，Y_k それぞれの末尾の要素を x_n，y_k と表記する。例えば，X＝{ “A”，“B”，“C”，“B” }とすると，X_3＝{ “A”，“B”，“C” }，x_3＝“C” である。このとき，列 X_n と列 Y_k との最長共通部分列の中の一つを LCS(n, k)，最長共通部分列の長さを LCSL(n, k)と表記する。なお，X_0 と Y_0 は空の列であり，x_0 と y_0 は存在しない。

ここで，x_n と y_k とが一致しているか否かに着目して次の(1)～(3)に場合分けし，再帰的な関係を用いて LCSL(n, k)を求めることを考える。

(1) $x_n = y_k$ の場合を考える。例えば，$x_n = y_k =$ “A” とする。このとき，LCS(n, k)の末尾の要素は “A” となる。よって，LCS(n, k)は，列 X_{n-1}，Y_{k-1} の最長共通部分列 LCS(n－1, k－1)の末尾に “A” を付加したものと一致する。したがって，LCSL(n, k)＝LCSL(n－1, k－1) ＋ 1 が成り立つ。

(2) $x_n \neq y_k$ の場合を考える。例えば，$x_n =$ “A”，$y_k =$ “B” とする。ここで，LCS(n, k)の末尾の要素は “A” 又は “A” 以外となる。LCS(n, k)の末尾の要素が “A” である場合は，列 Y_k から末尾の “B” を取り除いても最長共通部分列には影響しないので，LCS(n, k)は LCS(n, k－1)と一致する。一方，LCS(n, k)の末尾の要素が “A” でない場合は，列 X_n から末尾の “A” を取り除いても最長共通部分列には影響しないので，LCS(n, k)は LCS(n－1, k)と一致する。よって，LCS(n, k)は LCS(n, k－1)又は LCS(n－1, k)のいずれかと一致する。したがって，LCSL(n, k)は，LCSL(n, k－1)と LCSL(n－1, k)のうちの最大値と一致する。

(3) n＝0 又は k＝0 の場合，最長共通部分列は空の列となり，LCSL(n, k)＝0 である。

〔動的計画法を用いて最長共通部分列の長さを求めるアルゴリズム〕

(1)～(3)の再帰的な関係に従い，LCSL(n, k)を再帰的に計算することによって，列 X，Y の最長共通部分列の長さを求めることができる。しかし，再帰的に計算するアルゴリズムでは，重複して同じ計算をすることによって時間計算量が大きくなり，非効率になる場合がある。そこで，重複して同じ計算をすることを避けるために，LCSL(n, k)の値を動的計画法によって求めることを考える。

二つの列が X={ "A"，"B"，"C"，"B"，"D"，"A"，"B" }，Y={ "B"，"D"，"C"，"A"，"B"，"A" }の場合，0≦n≦7 かつ 0≦k≦6 に対する LCSL(n, k)の値を図 3 に示す。図 3 の左端 2 列は n (0～7) とそれに対応する x_n を，上端 2 行は k (0～6) とそれに対応する y_k を表す。n=0 のときの x_n，k=0 のときの y_k は存在しないので，"−" と表す。各要素は列 X_n と列 Y_k との最長共通部分列の長さ LCSL(n, k)の値を示している。

	k	0	1	2	3	4	5	6
n	x_n \ y_k	−	B	D	C	A	B	A
0	−	0	0	0	0	0	0	0
1	A	0	0	0				
2	B	0	1 (①)	1 (②)				
3	C	0	1	1		ア		
4	B	0	1	1				
5	D	0	1	2				
6	A	0	1	2 (③)				
7	B	0	1				イ	

注記 1　() 内の①～③については，図 3 に続く本文で値の求め方を説明している。
注記 2　LCSL(n, k)の値の一部は，設問のため表示していない。

図 3　LCSL(n, k)の値

図 3 の各要素の値は，〔最長共通部分列の長さを求めるアルゴリズム〕の(1)～(3)の再帰的な関係に従って求められる。

まず，n＝0 又は k＝0 のときは(3)に対応するので，LCSL(n, k)＝0 である。

それ以外の値について，例えば，図 3 の①～③は次のように値が決まる。

①について，$x_2=y_1$ なので(1)に対応し，LCSL(2, 1)＝LCSL(1, 0) ＋ 1 である。LCSL(1, 0)＝0 なので，LCSL(2, 1)＝1 となる。

②について，$x_2 \neq y_2$ なので(2)に対応し，LCSL(2, 1)＝1，LCSL(1, 2)＝0 なので，LCSL(2, 2)＝1 となる。

③について，$x_6 \neq y_2$ なので(2)に対応し，LCSL(6, 1)＝1，LCSL(5, 2)＝2 なので，LCSL(6, 2)＝2 となる。

図 3 の要素の値を全て計算することによって列 X，Y の最長共通部分列の長さが 4 であると分かる。

〔動的計画法を用いて最長共通部分列の長さを求めるプログラム〕

〔動的計画法を用いて最長共通部分列の長さを求めるアルゴリズム〕に基づいて，二つの列の最長共通部分列の長さを求めるプログラムを考える。任意の二つの列をそれぞれ配列 S，T として受け取り，動的計画法を用いて最長共通部分列の長さを求めるプログラムを図 4 に示す。ここで，配列の要素番号は 0 から始まり，整数型の二次元配列 lcsl は，行番号が 0 から配列 S の要素数 s までの(s ＋ 1)行，列番号が 0 から配列 T の要素数 t までの(t ＋ 1)列の大きさをもつ。

```
○整数型: calculate_lcsl(文字型の配列: S, 文字型の配列: T)
  整数型: s ← Sの要素数
  整数型: t ← Tの要素数
  整数型の二次元配列: lcsl ← {(s + 1)行, (t + 1)列の未定義の値}
  整数型: n, k
  for (nを0からsまで1ずつ増やす)
    lcsl[n, 0] ← 0
  endfor
  for (kを0からtまで1ずつ増やす)
    lcsl[0, k] ← 0
  endfor
  for (nを1からsまで1ずつ増やす)
    for (kを1からtまで1ずつ増やす)
      if (S[n - 1]がT[k - 1]と等しい)
        lcsl[n, k] ← [   ウ   ]
      elseif (lcsl[n, k - 1]がlcsl[n - 1, k]より大きい)
        lcsl[n, k] ← [   エ   ]
      else
        lcsl[n, k] ← [   オ   ]
      endif
    endfor
  endfor
  return [   カ   ]
```

図 4　動的計画法を用いて最長共通部分列の長さを求めるプログラム

図 4 のプログラムの時間計算量を，配列 S の要素数 s，配列 T の要素数 t を用いて表すと $O($ [キ] $)$である。

設問 1　列{ “A” ，“C” ，“B” ，“C” ，“D” ，“C” }と列{ “C” ，“D” ，“B” ，“D” ，“C” ，“A” }との最長共通部分列の長さを答えよ。

設問 2　図 3 中の [ア] ，[イ] に入れる適切な数値を答えよ。

設問 3　図 4 中の [ウ] ～ [カ] に入れる適切な字句を答えよ。

設問 4　本文中の [キ] に入れる適切な字句を，s と t を用いて答えよ。