

1. Schreiben Sie eine Java-Applikation, welche die folgende Problemstellungen löst und auf dem Bildschirm ausgibt. **Überprüfen Sie zu Ihrer eigenen Sicherheit, ob eine Teilaufgabe richtig gelöst ist, bevor Sie mit der nächsten Teilaufgabe fortfahren.**

1. Lesen Sie vom Benutzer zwei beliebige Strings S_1 und S_2 ein.
2. Bestimmen Sie die Länge n von S_1 und m von S_2 .
3. Legen Sie eine Matrix *Needleman* mit $(n + 1)$ Zeilen und $(m + 1)$ Spalten an.
4. Schreiben Sie eine Methode *int cost(char a, char b)*, die -1 zurückgibt, wenn $a \neq b$ ist und 1 zurückgibt wenn $a = b$ ist.
5. Schreiben Sie eine Methode *int max(int a, int b, int c)*, die das Maximum dreier Zahlen bestimmt.
6. Schreiben Sie eine Methode *String printMatrix(int [][] matrix)*, die eine Matrix auf dem Bildschirm ausgibt.
7. Initialisieren Sie

- (a) die Matrixposition *Needleman*(0,0) mit 0.
- (b) *Needleman*($i, 0$) mit $-i$ für $i = 1 \dots n$ mit i .
- (c) *Needleman*(0, j) mit $-j$ für $j = 1 \dots m$ mit j .

8. Berechnen Sie alle anderen Matrixpositionen *Needleman*(i, j) wie folgt

$$Needleman(i, j) = \max \begin{cases} Needleman(i-1, j) - 1 \\ Needleman(i, j-1) - 1 \\ Needleman(i-1, j-1) + cost(S_1[i-1], S_2[j-1]) \end{cases}$$

9. Geben Sie den Wert *Needleman*(n, m) aus.
10. Geben Sie die Matrix *Needleman* aus.
(für eine 4,0 muss diese Aufgabe und alle vorherigen Teilaufgaben gelöst werden.)
11. Erweitern Sie die Methode *String printMatrix(int [][] matrix)* derart, dass nun auch das i -te Zeichen von S_1 vor der i -ten Zeile und das j -te Zeichen von S_2 vor der j -ten Spalte steht.
(für eine 3,0 muss diese Aufgabe und alle vorherigen Teilaufgaben gelöst werden.)
12. Bestimmen Sie einen möglichen maximalen Pfad von *Needleman*(0,0) nach *Needleman*(n, m). Der Pfad soll dabei jeweils der Maximumauswahl des 8. Schrittes folgen. Orientieren Sie sich dabei an folgenden Tipps.
 - (a) Fangen Sie am Ende der Matrix an und bestimmen Sie das Maximum der davon links, darüber und diagonal links darüber liegenden Zelle.
 - (b) Das Maximum liegt auf dem Pfad und kann nun als neuer Ausgangspunkt verwendet werden.
 - (c) Suchen Sie solange neue Maxima bis Sie *Needleman*(0,0) als Maximum identifiziert haben.
 - (d) Speichern Sie den Pfad in einer Matrix der gleichen Größe wie *Needleman*. Verwenden Sie String als Datentyp der Matrix. Alle Positionen sind von Anfang an auf ein Leerzeichen initialisiert. Die jeweils gefunden Maxima werden in dieser Matrix an der entsprechenden Position gespeichert.

(für eine 1,0 muss diese Aufgabe und alle vorherigen Teilaufgaben gelöst werden.)

	A	G	T	C
0	-1	-2	-3	-4
-1	1	0	-1	-2
-2	0	0	-1	0
-3	-1	1	0	-1
-4	-2	0	2	1
-5	-3	-1	1	3

Ausgabe 10.)

		A	G	T	C
	0	-1	-2	-3	-4
A	-1	1	0	-1	-2
C	-2	0	0	-1	0
G	-3	-1	1	0	-1
T	-4	-2	0	2	1
C	-5	-3	-1	1	3

Ausgabe 11.)

	A	G	T	C
0				
A	1			
C	0			
G		1		
T			2	
C				3

Ausgabe 12.)