

Übung 3 - Musterlösung

1. Algorithmen im Alltag

Germ, Mehl, Wasser, Salz und Olivenöl in eine Schüssel geben
Die Mischung so lange kneten bis sie einem Teig entspricht
Schüssel an einen warmen Ort stellen
Timer auf 1 Stunde stellen
Timer starten
Tomaten in Scheiben schneiden
Tomatenscheiben mit Olivenöl in der Pfanne am Herd zu einer Sauce verarbeiten
Warten bis der Timer abgelaufen ist
Auf der Arbeitsfläche Mehl verteilen
Den Teig auf der Arbeitsfläche mit den Händen ausbreiten
Ein Blatt Backpapier auf ein Backblech legen
Den ausgebreiteten Teig auf das Backpapier legen
Tomatensauce auf dem Teig verteilen
Käse auf dem Teig verteilen
Ofen auf 200°C vorheizen
Backblech in den Ofen schieben
Timer auf 15 Minuten stellen
Timer starten
Warten bis der Timer abgelaufen ist
Ofen ausschalten
Gericht aus dem Ofen nehmen
Basilikum und Oregano auf dem Gericht verteilen
Gericht in 8 Stücke schneiden
Einen Teller nehmen
Ein Stück auf den Teller legen

2. Syntax/Semantik

```
1 READ S10
2 READ S11
3 LOAD S10, R1
4 LOAD S11, R2
5 MULT R1, R2, R3
6 ADD R3, R3, R3
7 STORE R3, S12
8 WRITE S12
9 STOP
```

```
1 READ S10
2 READ S11
3 LOAD S10, R1
4 LOAD S11, R2
5 MULT R1, R2, R3
6 ADD R3, R3, R3
7 STORE R3, S12
8 WRITE R3, S12
```

In **Zeile 8** wird der Befehl „WRITE“ mit einem Argument zu viel aufgerufen. Es handelt sich dabei um einen Syntax-Fehler.

In Zeile 9 wurde der Befehl „STOP“ vergessen. Dieser Befehl beendet das Assemblerprogramm und gibt die Kontrolle zurück an das Betriebssystem. Es handelt sich dabei um einen Syntax-Fehler.

3. Sprachen

Der Vorteil einer Maschinensprache liegt in der ...

- a.) Möglichkeit zu beschreiben, wie ein Problem gelöst wird.
- b.) Lesbarkeit der Programme.
- c.) **Möglichkeit, die Hardware eines Computers optimal anzusprechen.**
- d.) Veranschaulichung von Programmabläufen auf Maschinenebene.

4. Sortieralgorithmen

Gegeben:

Lineares Zahlenfeld $z = [15, 2, 42, 17, 4, 9, 47]$

Die Elemente im Zahlenfeld werden folgendermaßen angesprochen:

$z[0] = 15, z[1] = 2, \dots, z[6] = 47$

Bubblesort:

- Das lineare Zahlenfeld z wird durchlaufen, beginnend bei $z[0]$
- Es werden je zwei benachbarte Elemente betrachtet: $z[i]$ und $z[i + 1]$
- Ist $z[i] > z[i + 1]$, tauschen die Elemente $z[i]$ und $z[i + 1]$ ihre Position im linearen Feld
- Nach dem ersten Durchlauf ($i = 1$) „steigt“ das größte Element nach „oben“ (Bubble-Sort); d.h. das größte Element befindet sich bereits an der richtigen Position im Feld.

$i = 1$	$z[0]$	$z[1]$	$z[2]$	$z[3]$	$z[4]$	$z[5]$	$z[6]$
	15	2	42	17	4	9	47
	2	15					
	2	15	42				
	2	15	17	42			
	2	15	17	4	42		
	2	15	17	4	9	42	
	2	15	17	4	9	42	47

- Lineares Zahlenfeld nach dem zweiten Durchlauf ($i = 2$); beim zweiten Durchlauf muss das Element an der letzten Position nicht mehr betrachtet werden.

$i = 2$	$z[0]$	$z[1]$	$z[2]$	$z[3]$	$z[4]$	$z[5]$	$z[6]$
	2	15	17	4	9	42	47
	2	15					
	2	15	17				
	2	15	4	17			
	2	15	4	9	17		
	2	15	4	9	17	42	
	2	15	4	9	17	42	47

- Lineares Zahlenfeld nach dem dritten Durchlauf ($i = 3$)

$i = 3$	$z[0]$	$z[1]$	$z[2]$	$z[3]$	$z[4]$	$z[5]$	$z[6]$
	2	15	4	9	17	42	47
	2	15					
	2	4	15				
	2	4	9	15			
	2	4	9	15	17		
	2	4	9	15	17	42	47

- Beim vierten Durchlaufe passieren keine Vertauschungen mehr; das lineare Zahlenfeld ist somit (aufsteigend) sortiert.

Insertion-Sort („Sortieren durch Einfügen“):

- Das **sortierte Teilfeld** besteht zu Beginn aus dem ersten Element des linearen Zahlenfeldes ($z[0]$).
- Das **unsortierte Teilfeld** besteht zu Beginn aus den restlichen Elementen des Feldes ($z[1], \dots, z[6]$).
- SOLANGE das **unsortierte Teilfeld** Elemente hat:
Entferne das erste Element aus dem **unsortierten Teilfeld** und füge es an der richtigen Stelle im **sortierten Teilfeld** ein

$z[0]$	$z[1]$	$z[2]$	$z[3]$	$z[4]$	$z[5]$	$z[6]$
15	2	42	17	4	9	47
2	15	42	17	4	9	47
2	15	42	17	4	9	47
2	15	17	42	4	9	47
2	4	15	17	42	9	47
2	4	9	15	17	42	47
2	4	9	15	17	42	47

Vorteil von „Insertion Sort“ im Vergleich zu „Bubblesort“:

„Insertion Sort“ arbeitet (im Durchschnitt) etwas schneller als „Bubblesort“; weniger Vertauschungen sind notwendig

Jedoch sind beide Sortieralgorithmen **nicht** für größere Datenmengen geeignet.