

### 3.3.1 Einfache lineare Programme

Der erste Grundbaustein jedes Algorithmus ist die Sequenz. Daher beschäftigen wir uns zuallererst mit Programmen, die aus einer Sequenz von Anweisungen besteht. Wir benötigen dabei meist die Befehle zum Laden von Adressen, zum Speichern des Inhalts des Akkumulators und die arithmetischen Befehle, um eine Berechnung durchzuführen.

Beispiel: Berechnung einer Summe

```

Start:   LOAD      100   #Lädt Inhalt der Speicherzelle 100
         ADDI      5     #Addiert Wert "5" zum Akkumulator
         STORE     110   #Speichert Akkumulator in Zelle 110
         HOLD      #Hält den Prozessor an
  
```

Nach dem Assemblieren sieht man im Speicher die in Maschinensprache übersetzte Befehlskette:

Speicheranzeige									
	0	1	2	3	4	5	6	7	
0	276	100	522	5	277	110	99	0	
10	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	
30	0	0	0	0	0	0	0	0	
40	0	0	0	0	0	0	0	0	

Ähnlich laufen Programme, die mehrere Zellen oder Zahlen Addieren, Subtrahieren, Multiplizieren ...

**Aufgaben:**

Erstellen Sie je ein Programm, das die Formel berechnet ([Zahl] steht für eine Speicheradresse)

- a)  $[100] + [101] \cdot [102] = [104]$
- b)  $[100] + [101] + [102] + [103] = [104]$
- c) Geben Sie den Term an, den folgendes Programm berechnet. Beschreiben Sie welche Zahl am Ende in Zelle 103 steht.

```
LOAD 100  
DIV 101  
MUL 101  
STORE 103  
LOAD 100  
SUB 103  
STORE 103
```

- d) Erstellen Sie eine Term-Kurzschreibweise für Aufgabe c). Implementieren Sie ein Programm, welches diesem Term umsetzt.
- e) Erstellen Sie ein Programm, das den Inhalt der Zelle 100 durch die Zelle 101 teilt und das ganzzahlige Ergebnis in Zelle 110 und den Rest der Division in Zelle 111 ablegt.
- f) Erstellen Sie ein Programm, das die Werte der Speicherzellen 100 und 101 vertauscht.
- g) Erstellen Sie ein Programm, das die Zellen wie bei Aufgabe f) vertauscht, aber benutzen Sie dabei keine weiteren Zellen als Zwischenspeicher.