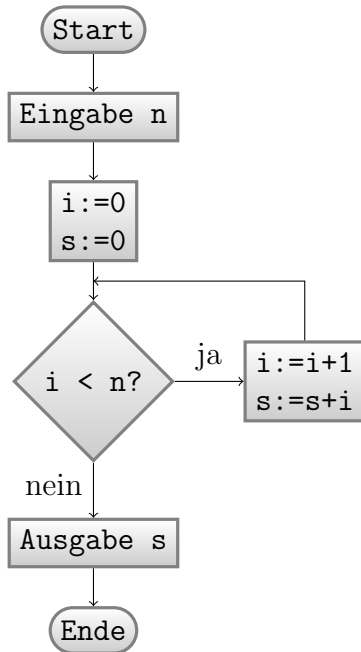


Aufgabe 1

Daten und Programme befinden sich im gleichen Speicher.

Aufgabe 2



Es gibt auch noch andere Lösungsvarianten.

Aufgabe 3

- (a) Es gibt verschiedene Möglichkeiten, den jeweiligen Zustand des Programms darzustellen. Hier ist eine davon:

a	b	r
42	15	?
15=0?		<i>false</i>
15	12	12
12=0?		<i>false</i>
12	3	3
3=0?		<i>false</i>
3	0	0
0=0?		<i>true</i>

Ausgabe: 3

- (b) Der Algorithmus bestimmt den grössten gemeinsamen Teiler von a und b.

Aufgabe 4

Maschinensprache: (Programmiersprache der 1. Generation) Programmbefehle und Daten bestehen aus *Binärzahlen*. Die Anweisungen können direkt vom entsprechenden Mikroprozessor ausgeführt werden.

Assemblersprache: (Programmiersprache der 2. Generation) Programmbefehle bestehen aus Mnemonics (Wörter, die auf die auszuführende Operation schliessen lassen, wie z. B. `add`). Daten lassen sich durch symbolische Adressen („Variablen“) darstellen. Damit Assemblercode ausgeführt werden kann, muss er von einem speziellen Programm, dem *Assembler* (oder *Assemblierer*) in den Maschinencode des entsprechenden Mikroprozessors übersetzt werden.

höhere Programmiersprache: (Programmiersprache der 3. Generation) Anweisungen bestehen aus natürlichsprachigen Wörtern (`print()`, `if`, `while`, `,` ...) und weisen einen noch höheren Abstraktionsgrad als die Assemblersprachen auf. Damit ein Programm in einer höheren Programmiersprache von einem Mikroprozessor ausgeführt werden kann, benötigt man entweder einen *Compiler* (übersetzt das Programm in ein Maschinenprogramm) oder einen *Interpreter* (übersetzt ein Programm und führt es sofort aus).

Aufgabe 5

Adresse	Wert	Kommentar
...	...	
10	104	Lade Wert von Zelle 04 in ACC
11	401	Schiebe Inhalt von ACC um 1 Stelle nach rechts
12	410	Schiebe Inhalt von ACC um 1 Stelle nach links
13	605	Speichere Inhalt von ACC in Zelle 05
14	505	Gib Inhalt Zelle 05 aus
15	910	Stoppe Ausführung und setze PC auf 10
...	...	

Aufgabe 6

Adresse	Wert	Kommentar
...	...	
02	600	Konstante (Opcode für Speichern)
03	000	Konstante (der in Zelle XY zu schreibende Wert)
04	0XY	Adresse der Speicherzelle (vorgegeben)
...	...	
10	104	Lade Wert von Zelle 04 (XY) in ACC
11	202	Addiere den Wert von Zelle 02 (600) zu ACC
12	614	Speichere den Wert von ACC (6XY) in Zelle 14
13	103	Lade Wert von Zelle 03 (000) in ACC
14	6XY	Speichere Wert von ACC in Zelle XY
15	910	Beende Programm und setze PC auf 10
...	...	

Aufgabe 7

Adresse	Wert	Kommentar
...	...	
04	ABC	Summand 1
03	XYZ	Summand 2
...	...	
10	104	Lade Wert von Zelle 4 (ABC) in ACC
11	205	Addiere den Wert von Zelle 5 zu ACC
12	607	Speichere den ACC-Wert (dreistellig) in Zelle 7
13	403	Schiebe Inhalt von ACC um 3 Stellen nach rechts
14	606	Speichere Wert von ACC (0 oder 1) in Zelle 6
15	910	Beende Programm und setze PC auf 10
...	...	

Aufgabe 8

Adresse	Wert	Kommentar
00	001	Konstante
...	...	
04	ABC	Minuend (die zu vermindernde Zahl)
05	XYZ	Subtrahend (die Zahl, mit der vermindert wird)
06	000	Zähler (zu Beginn auf 0)
...	...	
10	104	Lade Wert von Zelle 4 in ACC
11	705	Subtrahiere den Wert von Zelle 5 von ACC
12	320	Teste, ob $ACC < 0$; falls ja, setze $PC=20$
13	604	Schreibe verminderten Wert in Zelle 4 zurück
14	106	Lade den Zähler in ACC
15	200	Erhöhe den Zähler um den Wert von Zelle 0 (also 1)
16	606	Speichere den Wert des Zähler in ACC in Zelle 6 zurück
17	810	Sprung zur Zelle 10
...	...	
20	910	Beende Programm und setze PC auf 10
...	...	

Aufgabe 9

Adresse	Wert	Kommentar
...	...	
10	810	Springe zur Zelle 10
...	...	