

## 2. Schulaufgabe aus der Informatik

### Definition der Registermaschine für Aufgaben 2 und 3.

LOAD x	kopiert den Wert aus Speicherzelle x in den Akkumulator
DLOAD i	lädt die Ganzzahl i in den Akkumulator
RLOAD	lädt den Wert aus der Speicheradresse in den Akkumulator, die vor Ausführung des Befehls im Akkumulator gespeichert ist. (z.B. Bei Akku = 17 entspricht das dem Befehl LOAD 17)
STORE x	speichert den Wert aus dem Akkumulator in Speicherzelle x
ADD x	addiert den Wert aus Speicherzelle x zum Wert im Akkumulator und legt das Ergebnis dort ab
INC	addiert den Wert 1 zum im Akkumulator gespeicherten Wert dazu und legt das Ergebnis dort ab
SUB x	subtrahiert den Wert aus Speicherzelle x vom Wert im Akkumulator und legt das Ergebnis dort ab
DEC	subtrahiert den Wert 1 vom Wert im Akkumulator und legt das Ergebnis dort ab
DADD i	addiert den Wert i zum Wert im Akkumulator und legt das Ergebnis dort ab
DSUB i	subtrahiert den Wert i vom Wert im Akkumulator und legt das Ergebnis dort ab
DIV x	dividiert ganzzahlig den Wert im Akkumulator durch den Wert in Speicherzelle x und legt das Ergebnis im Akkumulator ab. (z.B. $17:5 \rightarrow 3$ )
DDIV i	dividiert ganzzahlig den Wert im Akkumulator durch die Zahl i und legt das Ergebnis im Akkumulator ab. (z.B. $17:5 \rightarrow 3$ )
MOD x	dividiert den Wert im Akkumulator durch den Wert in Speicherzelle x und legt den Rest der ganzzahligen Division im Akkumulator ab. (z.B. $17:5 \rightarrow 2$ )
DMOD i	dividiert den Wert im Akkumulator durch die Zahl i und legt den Rest der ganzzahligen Division im Akkumulator ab. (z.B. $17:5 \rightarrow 2$ )
JMPP x	springt zum Befehl in Speicheradresse x, falls der Wert im Akkumulator größer als 0 ist.
JMPZ x	springt zum Befehl in Speicheradresse x, falls der Wert im Akkumulator 0 ist.
JMPN x	springt zum Befehl in Speicheradresse x, falls der Wert im Akkumulator kleiner als 0 ist.
JMP x	führt einen unbedingten Sprung zum Befehl in Speicherzelle x aus
HOLD	Beendet die Ausführung des Programms

## 2. Schulaufgabe aus der Informatik

### Definition der Registermaschine für Aufgaben 2 und 3.

LOAD x	kopiert den Wert aus Speicherzelle x in den Akkumulator
DLOAD i	lädt die Ganzzahl i in den Akkumulator
RLOAD	lädt den Wert aus der Speicheradresse in den Akkumulator, die vor Ausführung des Befehls im Akkumulator gespeichert ist. (z.B. Bei Akku = 17 entspricht das dem Befehl LOAD 17)
STORE x	speichert den Wert aus dem Akkumulator in Speicherzelle x
ADD x	addiert den Wert aus Speicherzelle x zum Wert im Akkumulator und legt das Ergebnis dort ab
INC	addiert den Wert 1 zum im Akkumulator gespeicherten Wert dazu und legt das Ergebnis dort ab
SUB x	subtrahiert den Wert aus Speicherzelle x vom Wert im Akkumulator und legt das Ergebnis dort ab
DEC	subtrahiert den Wert 1 vom Wert im Akkumulator und legt das Ergebnis dort ab
DADD i	addiert den Wert i zum Wert im Akkumulator und legt das Ergebnis dort ab
DSUB i	subtrahiert den Wert i vom Wert im Akkumulator und legt das Ergebnis dort ab
DIV x	dividiert ganzzahlig den Wert im Akkumulator durch den Wert in Speicherzelle x und legt das Ergebnis im Akkumulator ab. (z.B. $17:5 \rightarrow 3$ )
DDIV i	dividiert ganzzahlig den Wert im Akkumulator durch die Zahl i und legt das Ergebnis im Akkumulator ab. (z.B. $17:5 \rightarrow 3$ )
MOD x	dividiert den Wert im Akkumulator durch den Wert in Speicherzelle x und legt den Rest der ganzzahligen Division im Akkumulator ab. (z.B. $17:5 \rightarrow 2$ )
DMOD i	dividiert den Wert im Akkumulator durch die Zahl i und legt den Rest der ganzzahligen Division im Akkumulator ab. (z.B. $17:5 \rightarrow 2$ )
JMPP x	springt zum Befehl in Speicheradresse x, falls der Wert im Akkumulator größer als 0 ist.
JMPZ x	springt zum Befehl in Speicheradresse x, falls der Wert im Akkumulator 0 ist.
JMPN x	springt zum Befehl in Speicheradresse x, falls der Wert im Akkumulator kleiner als 0 ist.
JMP x	führt einen unbedingten Sprung zum Befehl in Speicherzelle x aus
HOLD	Beendet die Ausführung des Programms