|  |  |
| --- | --- |
| load x | kopiert den Wert aus der Speicherzelle x in den Akkumulator |
| loadi n | lädt die ganze Zahl n in den Akkumulator |
| store x | kopiert den Wert aus dem Akkumulator in die Speicherzelle x |
| add x | addiert den Wert aus der Speicherzelle x zum Wert im Akkumulator |
| sub x | subtrahiert den Wert aus der Speicherzelle x vom Wert im Akkumulator |
| mul x | multipliziert den Wert im Akkumulator mit dem Wert in Speicherzelle x |
| div x | dividiert den Wert im Akkumulator durch den Wert in Speicherzelle x (ganzzahlige Division) |
| mod x | dividiert den Wert im Akkumulator durch den Wert in Speicherzelle x und speichert den ganzzahligen Rest dieser Division im Akkumulator |
| addi n | addiert die ganze Zahl n zum Wert im Akkumulator |
| subi n | subtrahiert die ganze Zahl n vom Wert im Akkumulator |
| muli n | multipliziert den Wert im Akkumulator mit der ganzen Zahl n |
| divi n | dividiert den Wert im Akkumulator durch die ganze Zahl n (ganzzahlige Division) |
| modi n | dividiert den Wert im Akkumulator durch die ganze Zahl n und speichert den ganzzahligen Rest dieser Division im Akkumulator |
| jmp x | springt zum Befehl in der Speicherzelle x |
| jmpn x | springt zum Befehl in der Speicherzelle x, falls der Wert imAkkumulator negativ ist |
| jmpz x | springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator gleich 0 ist |
| jmpp x | springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator positiv ist |
| jmpnn x | springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator nicht negativ ist |
| jmpnz x | springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator ungleich 0 ist |
| jmpnp x | springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator nicht positiv ist |
| hold | beendet die Abarbeitung des Programms |