

Logic Compiler (C) 2023 by Karl Kechele

Logic Compiler ist ein Logikcompiler für die boolschen Funktionen.
Es gibt nur einen Variablentyp: Boolean ('0' oder '1')

Syntax des Compilers:

Variablendeklaration: am Anfang des Programmes können alle benötigten Variablen deklariert werden.

Kommentar: alle Zeichen ab '@' bis Zeilenende gelten als Kommentar.

Zuweisungen: a : <Ausdruck> ;

Operatoren: ! a nicht a (höchste Priorität)
a & b a und verknüpft mit b
a + b a oder verknüpft mit b
a % b a exklusiv oder verknüpft mit b
a > b a impliziert b
a = b a ist equivalent mit b

Klammern: () Klammerung von Ausdrücken

Reservierte Wörter: var Deklaration von Variablen
for Definition einer Schleife, welche für alle möglichen Kombinationen der angegebenen Variablen durchlaufen wird.
do Beginn eines Anweisungsblocks.
end Ende eines Anweisungsblocks.
if Programmverzweigung.
else alternativer Zweig der if-Anweisung.
repeat Beginn einer Schleife
wend Ende der repeat Schleife
break beendet vorzeitig do-while und while-end Schleifen.
continue startet den nächsten Schleifendurchlauf
exit beendet das Programm.
? Eingabe von Tastatur.
Ausgabe an Bildschirm.
" Stringbegrenzer.
' alternativer Stringbegrenzer.
\$ wird in Strings als <CR> interpretiert.
\$\$ wird als \$ interpretiert.

Beispiel:

```
var erg, a, b;           @ Deklaration aller benötigten Variablen
                        @ jeweils immer am Anfang eines Programms

?"a => ",a;             @ Ausgabe von "a => ", Eingabe von a
?"b => ",b;             @      "      b      "      b

erg : a & b;             @ Anweisung
if (erg) do              @ if - Zweig
  #"ergebnis = TRUE$";   @ Ausgabe von "ergebnis = TRUE" + <CR>
else                      @ else - Zweig
  #"ergebnis = FALSE$"; @ Ausgabe von "ergebnis = FALSE"
end                       @ Ende der if - Bedingung
#"ergebnis = ",erg;      @ Ausgabe des Ergebnisses

for a,b do               @ for - Schleife: wird 4 mal mit
```

```

erg : a > (b & a);      @ a:b = 0:0, 0:1, 1:0, 1:1 durchlaufen
#a,b," - ",erg,"$";    @ Berechnung von erg
                        @ Ausgabe der Wahrheitstabelle in der
                        @ Form XX - Y <CR>
end                      @ Ende der for ? Schleife

repeat                  @repeat Schleife
  ?a=>?,a;
  if (a) do break; end @Verlassen der Schleife mit break
wend

```

Beispiele können mit dem Button 'Examples' aufgerufen werden.

Verwendung:

1. Code im Tab "Code" eingeben
2. Button "Complie" klicken
3. Wenn der Compiler keine Fehler findet:
4. Im Tab "Assembly" wird der Pseudo Assembler code ausgegeben
5. Button "Run" klicken
6. Im Tab "Console" wird das Ergebnis von "Run" ausgegeben

Der eingegebene Code wird in eine Assembler Syntax konvertiert.

Bei "Run" werden dann die Assemblerbefehle simuliert.

Der Pseudo Assembler code verwendet zwei Register x und y und einen Stack und ist dem Befehlssatz des Prozessors 6502 nachempfunden. Es werden keine Optimierungen beim Assembler code durchgeführt.