```
                                ASSEMBLERLISTING

                            FILE: Comparator_Finish+Comment.txt
================================================================================

VL        001100000
--------------------------------------------------------------------------------
PR B011000001001100   000   INIT1     LDA      STAB      ;
Tabellenpointer initialisieren
PR B011010001001101   001             STA      PTAB      ;                    ~
PR B011000001010100   002             LDA      SSTOR     ;
Storagepointer initialisieren
PR B011010001010101   003             STA      PSTOR     ;                    ~
PR B011000001000111   004             LDA      NULL      ;
Hilfsvariable 0 setzen
PR B011010001010011   005             STA      NUM       ;                    ~
PR B011000001000111   006   INIT2     LDA      NULL      ;                    Zahl 0
laden und an aktuelle adresse im Pointer speichern
PR B011011001010101   007             STA,I    PSTOR     ;                    ~
PR B011000001010101   008             LDA      PSTOR     ;                    Pointer
inkrementieren und rückspeichern
PR B100000001001000   009             ADD      ONE       ;                    ~
PR B011010001010101   010             STA      PSTOR     ;                    ~
PR B100010001010100   011             SUB      SSTOR     ;
Startadresse abziehen
PR B010100001001001   012             CMP      NINE      ;                    Mit 9
vergleichen
PR B001100000000110   013             JLE      INIT2     ;                    Wenn
kleiner gleich 9 muss speicher weiter initialisiert werden.
PR B001110000001111   014             JMP      INPUT     ;                    Sonst
weiter
PA B011100000000000   015   INPUT     RDA               ;
Tastaturpuffer einlesen
PR B010100001001010   016             CMP      ASCOFF    ;                    Mit
ASCII offset von 0 vergleichen.
PR B001000000010110   017             JLT      EXCEPT    ;                    ASCII
codes unter 0 keine Zahlen -> Exception
PR B100010001001010   018             SUB      ASCOFF    ;                    ASCII
offset abziehen
PR B010100001001001   019             CMP      NINE      ;
Exception für input groesser 9
PR B000010000010110   020             JGT      EXCEPT    ;                    ~
PR B001110000011000   021             JMP      SAVE      ;                    Weiter
zu speichern
PR B011000001000111   022   EXCEPT    LDA      NULL      ;                    Bei
Exception eine 0 speichern
PR B001110000011000   023             JMP      SAVE      ;


                                                         ;---Speichern aller Inputs
                                                         in die Tabelle---
PR B011011001001101   024   SAVE      STA,I    PTAB      ;
Wenn exceptionchecks passiert, speichern an aktuellen Tabellenplatz
PR B011000001001101   025             LDA      PTAB      ;
Tabellenpointer inkrementieren und rückspeichern
PR B100000001001000   026             ADD      ONE       ;                              ~
PR B011010001001101   027             STA      PTAB      ;                              ~
PR B100010001001100   028             SUB      STAB      ;
Pointerstart abziehen -> gibt Anzahl der inputs bis jetzt -1
PR B010100001001011   029             CMP      INMAX     ;
Vergleichen mit Maximaler Anzahl an inputs
PR B000110000100000   030             JGE      CLEAN1    ;
Falls gleich, ist die Maximale Anzahl an inputs erreicht -> weiter zum cleanup und
Ausgabe
PR B001110000001111   031             JMP      INPUT     ;
Sonst nächsten input lesen.
PR B011000001001100   032   CLEAN1    LDA      STAB      ;
Tabellenpointer auf Startadresse setzen
PR B011010001001101   033             STA      PTAB      ;
Danach weiter zum Comparator
PR B001110000100011   034             JMP      COMPAR    ;
```

```
48                                                          ;---Zählen, wie oft jede
                                                            Ziffer vorkommt---
49  PR B011001001001101   035   COMPAR    LDA,I   PTAB      ;
    Den Wert an der Adresse im Tabellenpointer laden
50  PR B100000001010100   036             ADD     SSTOR     ;
    Startadresse addieren
51  PR B011010001010101   037             STA     PSTOR     ;                          in
    den pointer speichern
52  PR B011001001010101   038             LDA,I   PSTOR     ;                          Zahl
    an Adresse inkrementieren
53  PR B100000001001000   039             ADD     ONE       ;                              ~
54  PR B011011001010101   040             STA,I   PSTOR     ;
    Inkrementierten wert Rückspeichern
55  PR B011000001001101   041             LDA     PTAB      ;
    Tabellenpointer inkrementieren
56  PR B100000001001000   042             ADD     ONE       ;                              ~
57  PR B011010001001101   043             STA     PTAB      ;
    Rückspeichern
58  PR B100010001001100   044             SUB     STAB      ;
    Adressoffset abziehen und mit der maximalen inputanzahl vergleichen
59  PR B010100001001011   045             CMP     INMAX     ;                          ~
60  PR B000110000110000   046             JGE     OUTPUT    ;                          Wenn
    groesser, output, sonst wiederholen
61  PR B001110000100011   047             JMP     COMPAR    ;
62
63
64                                                          ;---Ausgabe der Ziffern in
                                                            geordneter Reihenfolge---
65  PR B011000001010011   048   OUTPUT    LDA     NUM       ;
    Num dient als momentan betrachtete Zahl, wurde als 0 initialisiert.
66  PR B100000001010100   049             ADD     SSTOR     ;
    Startadresse des Storage addieren und in den Pointer rückspeichern
67  PR B011010001010101   050             STA     PSTOR     ;                          ~
68  PR B011001001010101   051             LDA,I   PSTOR     ;                          Zahl
    im betrachteten Counter als Loopzähler -> die Betrachtete Zahl wird so oft ausgegeben.
69  PR B011010001100000   052             STA     VLOOP     ;                          ~
70  PR B001110000110110   053             JMP     OUTLOOP   ;
    Sprung zum Outloop
71
72  PR B011000001100000   054   OUTLOOP   LDA     VLOOP     ;
73  PR B010100001001000   055             CMP     ONE
    ;                               Vor Ausführung der Ausgabe schauen, wie oft noch
    ausgegeben werden soll
74  PR B000110000111111   056             JGE     NUMOUT    ;
    Wenn eine oder mehr Ausgaben, Sprung zur Ausgabe
75  PR B011000001010011   057             LDA     NUM
    ;                               Wenn Vloop erschöpft, betrachtete Zahl laden und mit
    0 vergleichen
76  PR B010100001001001   058             CMP     NINE      ;                              ~
77  PR B000100001000110   059             JEQ     HALT      ;
    Für 9 als betrachtete Zahl und
78  PR B100000001001000   060             ADD     ONE
    ;                               Sonst incrementieren,
79  PR B011010001010011   061             STA     NUM
    ;                               Rückspeichern
80  PR B001110000110000   062             JMP     OUTPUT    ;
    und zurück zum Output
81  PR B011000001010011   063   NUMOUT    LDA     NUM
    ;                               Betrachtete Zahl laden,
82  PR B100000001001010   064             ADD     ASCOFF    ;
    ASCII-offset addieren
83  PA B011110000000000   065             WRA
    ;                                 und Accumulator ausgeben
84  PR B011000001100000   066             LDA     VLOOP     ;
    VLOOP decrementieren
85  PR B100010001001000   067             SUB     ONE
    ;                               ~
86  PR B011010001100000   068             STA     VLOOP     ;
    und Rückspeichern
87  PR B001110000110110   069             JMP     OUTLOOP   ;
    zurück zu outloop
88  PA B010110000000000   070   HALT      HLT
    ;                                 Programm beenden
```

```
                                                        ;---Storage---

                                                        ;---Konstanten---
PA D0000000000000000   071   NULL      DEC     0         ;
PA D0000000000000001   072   ONE       DEC     1         ;
PA D0000000000001001   073   NINE      DEC     9         ;
PA D0000000000110000   074   ASCOFF    DEC     48
;                                Offset der ASCII 0 zur regulären 0
PA D0000000000000101   075   INMAX     DEC     5
;                                Maximale Anzahl an inputs minus eins (also maximale
speicheradresse)
                                                        ;---Zwischenspeicher der
                                                        Inputs (beliebig zu
                                                        erweitern)---
PR D0000000001001110   076   STAB      DEF     TAB0      ;
Startadresse der Tabelle
PA D0000000000000000   077   PTAB      HEX     0
;                                Tabellenpointer
PA D0000000000000000   078   TAB0      DEC     0         ;
PA D0000000000000000   079   TAB1      DEC     0         ;
PA D0000000000000000   080   TAB2      DEC     0         ;
PA D0000000000000000   081   TAB3      DEC     0         ;
PA D0000000000000000   082   TAB4      DEC     0         ;
                                                        ;---Counter der Einzelnen
                                                        Zahlen---
PA D0000000000000000   083   NUM       DEC     0
;                                Hilfsvariable für die Ausgabe
PR D0000000001010110   084   SSTOR     DEF     C0
;                                Startadresse des Storage
PA D0000000000000000   085   PSTOR     HEX     0
;                                Speicherpointer
PA D0000000000000000   086   C0        DEC     0         ;
PA D0000000000000000   087   C1        DEC     0         ;
PA D0000000000000000   088   C2        DEC     0         ;
PA D0000000000000000   089   C3        DEC     0         ;
PA D0000000000000000   090   C4        DEC     0         ;
PA D0000000000000000   091   C5        DEC     0         ;
PA D0000000000000000   092   C6        DEC     0         ;
PA D0000000000000000   093   C7        DEC     0         ;
PA D0000000000000000   094   C8        DEC     0         ;
PA D0000000000000000   095   C9        DEC     0         ;
                                                        ;---Hilfsvariablen---
PA D0000000000000000   096   VLOOP     DEC     0
;                                Loop-Variable für die Ausgabe

================================================================================

```