

## danke fuer die Tips und neue Frage ....

*Source:* <http://coding.derkeiler.com/Archive/Cobol/comp.lang.cobol/2004-01/0538.html>

---

*From:* john (cyber100\_at\_gmx.de)

*Date:* 01/20/04

Date: 20 Jan 2004 04:49:10 -0800

Hallo,

erstmal vielen Dank für die Tip's. Ich habe das Problem mit der Inspect-Anweisung gelöst, da der Compiler auch Hexa-Literale versteht. Es handelt sich auch nicht um das Umsetzen von x'c1' sondern x'41. Tipfehler meinerseits.

Jetzt die Neue Frage .... variable Satzlaenge.

Ich habe eine Eingabedatei wo die Saetze die Laenge von 1 – 255 Zeichen haben können. Also habe ich die Eingabe wie folgt definiert:

```
FD EINGABE
  RECORD VARYING FROM 1 TO 255 DEPENDIG ON E-MAX.
01 EINGABESATZ.
  02 E-ZEICHEN PIC X OCCURS 255.
```

Ausgabe die "gleiche" Definition.

```
FD Ausagbe
  RECORD VARYING FROM 1 TO 255 DEPENDIG ON A-MAX.
01 Ausgabesatz.
  02 A-ZEICHEN PIC X OCCURS 255.
```

In der working-storage section die Zaehler wie folgt definiert:

```
77 E-MAX PIC 9(4) COMP VALUE 1.
77 A-MAX PIC 9(4) COMP VALUE 1.
77 D-MAX PIC 9(4) COMP VALUE 1.
```

Dann folgt noch die Definition einer Datenstruktur diese möchte ich auch variable Anlegen. Soll heissen A-max = D-max. Habe das wir folgt probiert.

Funktioniert leider nicht.

```
01 DRUCKSATZ.
  02 GRUNDINFO.
    05 FELD1 PIC X.
```

danke fuer die Tips und neue Frage ....

comp.lang.cobol: danke fuer die Tips und neue Frage ....

```
05 REST OCCURS 1 TO 254 TIMES DEPENDING ON D-MAX.  
10 FELD2 PIC X.
```

Mein PROCEDURE DIVISION sieht wie folgt aus. Die Ausgabedatei besteht dann immer nur aus einem Zeichen pro Satz.

Wenn sich den Bereich fest definiere habe ich keine Probleme ausser das bei kurzen Saetzen der Rest mit blanks gefüllt ist

PROCEDURE DIVISION.

```
    STEUERTEIL.  
00508 *+----- STEUERTEIL  
-----+  
00509 *I STEUERT DAS GANZE PROGRAMM.  
I  
00510 *I SOLANGE DURCHLAUFEN BIS DATEIENDE ERREICHT.  
I  
00511 *+-----+  
00512 PERFORM VORARBEITEN.  
00513 PERFORM LESEN.  
00514 PERFORM SB WITH TEST AFTER UNTIL EOF = "AM ENDE"  
00515 PERFORM PN.  
00516 STOP RUN.  
00517  
00518 VORARBEITEN.  
00519 *+----- VORARBEITEN  
-----+  
00520 *I DATEIEN OEFFNEN UND SYSTEMDATUM UEBERNEHMEN.  
I  
00521 *+-----+  
00522 OPEN INPUT EINGABE,  
00523 OUTPUT AUSGABE, LISTE.  
00524 SET DATEI-OFFEN TO TRUE.  
00525  
00526 LESEN.  
00527 *+----- LESEN  
-----+  
00528 *I DATENFELDER MIT ANFANGSWERTEN VORBESETZEN  
I  
00529 *I EINGABESATZ LESEN. LAENGENFELD EINGABE -->  
AUSGABE, I  
00530 *I EINGABESATZ --> ACCOUNTSATZ UEBERTRAGEN  
I  
00531 *+-----+  
00532 INITIALIZE AUSGABESATZ, EINGABESATZ, DRUCKSATZ.  
00533 READ EINGABE  
00534 AT END MOVE "AM ENDE" TO EOF  
00535 END-READ.  
00536 MOVE E-MAX TO A-MAX, D-MAX.  
00537 MOVE EINGABESATZ TO DRUCKSATZ.  
00538 LESEX.
```

danke fuer die Tips und neue Frage ....

```
00539 EXIT.
00540
00541 SB.
00542 *+----- SATZBEARBEITUNG
-----+
00543 *I AUSWERTUNG DES STEUERZEICHENS UND WENN NOETIG
I
00544 *I EINFUEGEN EINER LEERZEILE
I
00545 *I - DEN SATZZAEHLER UM EINS ERHOEHEN
I
00546 *I - NAECHSTEN SATZ LESEN
I
00547 *+-----+
00548 ADD 1 TO SATZZAEHLER.
00549 INSPECT FELD1 REPLACING ALL X'41' BY "*"
00550 IF FELD1 = '*'
00551 ADD 1 TO CONVZAEHLER
00552 MOVE ' ' TO FELD1
00553 MOVE SPACES TO AUSGABESATZ
00554 WRITE AUSGABESATZ
00555 MOVE DRUCKSATZ TO AUSGABESATZ
00556 WRITE AUSGABESATZ
00557 ELSE
00558 MOVE DRUCKSATZ TO AUSGABESATZ
00559 WRITE AUSGABESATZ
00560 END-IF.
00562 PERFORM LESEN.
00563
00564 PN.
00565 *+----- PROGRAMM-NACHLAUF
-----+
00566 *I AUSGABE DES ENDETEXT.
I
00567 *I SCHLIESSEN DER VERWENDETEN DATEIEN.
I
00568 *+-----+
00569 DISPLAY "***** SAETZE VERARBEITET = " SATZZAEHLER
UPON DSS.
00570 DISPLAY "***** " UPON DSS.
00571 DISPLAY "***** SAETZE CONVERTIERT = " CONVZAEHLER
UPON DSS.
00572 SET DATEI-CLOSE TO TRUE.
00573 CLOSE EINGABE, AUSGABE.
```

Danke für die Hilfe  
John