Liebe Seminaristen und Seminaristinnen,

Zum Endspurt am Wochenende noch ein vierter und zugleich letzter Newsletter
bezueglich der Assembler-Seminaraufgabe WS97/98. Aufgrund umfangreicher
eMail-Schriftwechsel und leidenschaftlicher Diskussionen habe ich eine
Neuauflage der Beispielprogramme des 3.Newsletters der
Assembler-Seminaraufgabe WS96/97 erstellt.

* ALPHA.PAS :

```
program alpha;

var abcdef : string;

function abc : string; external;
{$L ABC}

function def(n:integer) : string; external;
{$L DEF}

begin
  writeln(abc);
  writeln(def(3));
  abcdef := abc + def(3);
  writeln(abcdef);
end.
```

* ALPHA.ASM :

```
CODE   SEGMENT PUBLIC
       ASSUME CS:CODE

       EXTRN ABC:NEAR
       EXTRN DEF:NEAR

ALPHA  PROC
; writeln(abc)
       MOV  AX,CS
       MOV  DS,AX
       PUSH DS
       MOV  DX,OFFSET RESULT
       PUSH DX
       CALL ABC
       POP  DX
       POP  DS
       MOV  BX,0
       MOV  BL,RESULT
       INC  BL
       MOV  RESULT[BX],'$'
       INC  DX
       MOV  AH,9h
       INT  21h
       MOV  DX,OFFSET CRLF
       INT  21h
; writeln(def(3))
       MOV  AX,CS
       MOV  DS,AX
       PUSH DS
       MOV  DX,OFFSET RESULT
       PUSH DX
       MOV  AX,3
       PUSH AX
       CALL DEF
       POP  DX
       POP  DS
       MOV  BX,0
       MOV  BL,RESULT
       INC  BL
       MOV  RESULT[BX],'$'
       INC  DX
       MOV  AH,9h
       INT  21h
       MOV  DX,OFFSET CRLF
       INT  21h
       MOV  AX,4C00h
       INT  21h
ALPHA  ENDP

RESULT DB   256 DUP (?)
CRLF   DB   13,10,'$'

CODE   ENDS

SSEG   SEGMENT STACK
       DW   64 DUP (?)
SSEG   ENDS
```

```
       END  ALPHA
```

```
* ABC.ASM

       PUBLIC ABC

CODE   SEGMENT PUBLIC
       ASSUME CS:CODE

ABC    PROC
       PUSH BP
       MOV  BP,SP
       MOV  ES,[BP+6]
       MOV  BX,[BP+4]
       MOV  BYTE PTR ES:[BX],3
       MOV  BYTE PTR ES:[BX+1],'A'
       MOV  BYTE PTR ES:[BX+2],'B'
       MOV  BYTE PTR ES:[BX+3],'C'
       MOV  SP,BP
       POP  BP
       RET
ABC    ENDP

CODE   ENDS
       END
```

```
* DEF.ASM

       PUBLIC DEF

CODE   SEGMENT PUBLIC
       ASSUME CS:CODE

DEF    PROC
       PUSH BP
       MOV  BP,SP
       MOV  ES,[BP+8]
       MOV  BX,[BP+6]
       MOV  SI,0
       MOV  CX,[BP+4]
       MOV  BYTE PTR ES:[BX],CL
DEF1:  JCXZ DEF2
       INC  SI
       MOV  BYTE PTR ES:[BX+SI],' '
       DEC  CX
       JMP  DEF1
DEF2:  ADD  BYTE PTR ES:[BX],3
       MOV  BYTE PTR ES:[BX+SI+1],'D'
       MOV  BYTE PTR ES:[BX+SI+2],'E'
       MOV  BYTE PTR ES:[BX+SI+3],'F'
       MOV  SP,BP
       POP  BP
       RET  2
DEF    ENDP

CODE   ENDS
       END
```

Ferner erreichten mich noch Fragen bezueglich der maximalen Laenge der Zeichenketten fuer die Routine INPUT (Eingabe einer Zeichenkette incl. Editiermoeglichkeiten). Die Beschraenkung auf eine Bildschirmzeile ist meines Erachtens ein sinnvoller Kompromiss zwischen allgemeiner Verwendbarkeit und Programmieraufwand der Routine INPUT. Bitte wertet diesen Hinweis aber nicht als Aufforderung fuer eine nachtraegliche Vereinfachung einer bereits bestehenden komfortableren Routine INPUT.

Und nun noch einmal viel Erfolg beim Endspurt der Assembler-Seminaraufgabe WS97/98. Am Freitag (und am Wochenende) bin ich aufgrund von drei Vorlesungen Programmierung, der Abnahme einer externen Abschlussarbeit in Norderstedt und dem Clubausflug der IGE-Bahntouristik im eigenen Speisewagen von Nuernberg zum Chiemsee nicht erreichbar. Nach der Vorlesung Rechnernetze am Montag (8:00-9:15) koennt Ihr mich wieder mit Fragen loechern.

Mit freundlichen Gruessen

Birger