

Klausur IA12.0/13.0 451 Assembler am 27.01.2003

Dauer : 120 Minuten

keine externen Hilfsmittel

Übersetze das nachfolgende (siehe Rückseite) Pascal-Programm *Klausur* in ein äquivalentes (d.h. möglichst bedeutungstreues) Assemblerprogramm (8086, EXE).

Gemäß den Konventionen von Borland TurboPascal soll die Parameterübergabe (sowohl für Wert- als auch Referenzparameter) über den Stack, die Ablage lokaler Variablen auf dem Stack und die Rückgabe eines Word-Funktionswerts über das AX-Register erfolgen. Bedenke : Referenzparameter bestehen aus einem 16-Bit-Segment- und 16-Bit-Offsetanteil.

Im Modul STDIO steht die Routine DEZOUT zur Ausgabe - fünfstellig rechtsbündig mit Unterdrückung führender Nullen - von Integer-Zahlen zur Verfügung. Der Datentransfer erfolgt nicht über den Stack sondern über das DX-Register.

Kommentiere das Assemblerprogramm durch eindeutige Zuordnung der Pascal-Befehle zu den Assembler-Befehlen. Ohne derartige Kommentierung wird die Klausur nicht gewertet !

Falls keine komplette Lösung angeben werden kann, bearbeite Teilaspekte im Sinne der eindeutigen Zuordnung der Pascal-Befehle zu den Assembler-Befehlen.

Zum besseren Verständnis der Klausur ist die Ausgabe des Programms *Klausur* nachfolgend beigefügt.

Ein mal Eins * sniE lam niE

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Das PTL-Team wünscht viel Erfolg

Anlage : Befehlssatz des 8088

```

Program Klausur;

Type cString = Array[0..15] Of Char;

Const S : cString = ('E','i','n',' ','m','a','l',' ','
                    'E','i','n','s',#0,#0,#0,#0);

Function Length(Var S:cString):Word;
Var Z : Word;
Begin
  Z := 0;
  While S[Z] <> #0 Do Inc(Z);
  Length := Z
End;

Procedure Invert(X:Word; Var S:cString);
Var C : Char;
    Z : Word;
Begin
  Z := Length(S);
  If X < Z Div 2 Then Begin
    C := S[X];
    S[X] := S[Z-X-1];
    S[Z-X-1] := C;
    Invert(X+1,S)
  End
End;

Procedure Output(X:Word; Var S:cString);
Begin
  If S[X] <> #0 Then Begin
    Write(S[X]);
    Output(X+1,S)
  End
End;

Procedure xDirection(Z,Y,X:Word);
Begin
  Write(X:5);
  If Z > 1 Then xDirection(Z-1,Y,X+Y)
End;

Procedure yDirection(Z,Y:Word);
Begin
  xDirection(10,Y,Y);
  WriteLn;
  If Z > 1 Then yDirection(Z-1,Y+1)
End;

Begin
  Output(0,S);
  Write(' ','*',' ');
  Invert(0,S);
  Output(0,S);
  WriteLn; WriteLn;
  yDirection(10,1)
End.

```