

Physikalisch-Technische Lehranstalt Wedel

Staatsexamen für Technische Assistenten WS2001/2 schriftliche Prüfung im Fach Prozesstechnik gewählter Vorschlag

Dauer : 180 Minuten

keine externen Hilfsmittel

Aufgabe :

Entwickeln Sie ein 8086-Assembler-Programm zur Messdatenanalyse durch komplette Übersetzung des nachfolgenden Pascal-Programms *WS2001_2*.

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 Integer-Funktionswerts über das AX-Register erfolgen. Bedenke : Referenzparameter bestehen aus einem 16-Bit-Segment- und 16-Bit-Offsetanteil.

Weitere Hinweise zu Borland TurboPascal :

- Die Standardfunktion ORD liefert für Argumente ordinalen Datentyps die zugehörige Ordnungszahl.
- Die Standardfunktion CHR liefert für Argumente des Datentyps Integer das zugehörige Zeichen des Rechnerzeichensatzes (z.B. IBM-ASCII-Code).

Vergessen Sie nicht die hinreichende Kommentierung Ihres Assembler-Programms z.B. durch Zuordnung der Pascal-Befehle zu den Assembler-Befehlen.

Das PTL-Team wünscht viel Erfolg

```

Program WS2001_2;

Const TMax = 100;

Type TMesswerte = Array[0..TMax] Of Integer;

Var Messwerte : TMesswerte;
    Minimum    : Integer;
    Maximum    : Integer;
    Wert       : String;

Procedure Messung(X:TMesswerte); Far; External;
{$L MESSUNG}

Procedure Int2String(I:Integer;Breite:Byte;Linksbuendig:Boolean;
                    Var S:String);

Var Negativ : Boolean;

Begin
    If I < 0 Then Begin
        Negativ := True;
        I := -1 * I
    End Else
        Negativ := False;
    S := '';
    Repeat
        S := Chr(I Mod 10 + Ord('0')) + S;
        I := I Div 10
    Until I = 0;
    If Negativ Then
        S := '-' + S;
    If Not Linksbuendig Then
        While Length(S) < Breite Do
            S := ' ' + S
    End;

Function DivRound(Z:Integer;N:Word):Integer;

Var Negativ : Boolean;
    Result   : Integer;

Begin
    If Z < 0 Then Begin
        Negativ := True;
        Z := -1 * Z
    End Else
        Negativ := False;
    Result := Z Div N;
    If (N Mod 2 = 0) And (Z Mod N >= N Div 2) Or
       (N Mod 2 <> 0) And (Z Mod N > N Div 2) Then
        Result := Result + 1;
    If Negativ Then
        Result := -1 * Result;
    DivRound := Result
End;

Function Summe(Var X:TMesswerte):Integer;

Var I,Result : Integer;

Begin
    Result := 0;
    For I := 1 To X[0] Do
        Result := Result + X[I];
    Summe := Result
End;

```

```
Procedure MiniMax(Var X:TMesswerte; Var Mini,Max:Integer);
```

```
Var I : Integer;
```

```
Begin
```

```
  Mini := X[1];
```

```
  Max := X[1];
```

```
  For I := 1 to X[0] Do
```

```
    If X[I] < Mini Then
```

```
      Mini := X[I]
```

```
    Else
```

```
      If X[I] > Max Then
```

```
        Max := X[I]
```

```
End;
```

```
Begin
```

```
  Messung(Messwerte);
```

```
  Int2String(Summe(Messwerte),7,False,Wert);
```

```
  Writeln('Summe      :',Wert);
```

```
  MiniMax(Messwerte,Minimum,Maximum);
```

```
  Int2String(Minimum,7,False,Wert);
```

```
  Writeln('Minimum   :',Wert);
```

```
  Int2String(Maximum,7,False,Wert);
```

```
  Writeln('Maximum   :',Wert);
```

```
  Int2String(DivRound(Summe(Messwerte),Messwerte[0]),7,False,Wert);
```

```
  Writeln('Mittelwert :',Wert)
```

```
End.
```