

Klausur IA12.0/13.0 451 Assembler am 05.08.2002

Dauer : 120 Minuten (+15 Minuten "Lesezeit") **keine externen Hilfsmittel**

Übersetze aus dem Pascal-Programm *FunList*

```
Program FunList;

Type tPItem = ^tItem;
      tItem = Record
                Next : tPItem;
                Content : String
            End;
Var Root : tPItem;
    Item : tPItem;

Procedure InsertItem ( Var List : tPItem; NewItem : tPItem );

Begin
    If (List <> Nil) And (NewItem^.Content > List^.Content) Then
        InsertItem(List^.Next,NewItem)
    Else Begin
        NewItem^.Next := List;
        List := NewItem
    End
End;

Procedure ShowList ( List : tPItem );

Begin
    If List <> Nil Then Begin
        WriteLn(List^.Content);
        ShowList(List^.Next)
    End
End;

Begin
    Root := Nil;
    While Not Eof Do Begin
        Item := New(tPItem);
        ReadLn(Item^.Content);
        InsertItem(Root,Item)
    End;
    ShowList(Root);
    While Root <> Nil Do Begin
        Item := Root;
        Root := Root^.Next;
        Dispose(Item)
    End
End.
```

die Routinen *InsertItem* und *ShowList* möglichst bedeutungstreu in das Assembler-Modul (8086-Prozessor) *FunUtils*. Jenes Assembler-Modul *FunUtils* soll in das wie folgt modifizierte Pascal-Programm *FunList* einbindbar sein :

```
Procedure InsertItem ( Var List : tPItem; NewItem : tPItem );
    Far; External;
Procedure ShowList ( List : tPItem );
    Far; External;
{$L FUNUTILS}
```

Folgende Tipps mögen bei der Bearbeitung der Aufgabenstellung hilfreich sein :

- Für die Eingabedaten *FUNLIST.DTA*

Z
Y
X
C
B
A
ZZ
YY
XX
CC
BB
AA
ZZZ
YYY
XXX
CCC
BBB
AAA

erzeugt der Aufruf

```
FUNLIST <FUNLIST.DTA >FUNLIST.LST
```

die Ausgabedaten *FUNLIST.LST* :

A
AA
AAA
B
BB
BBB
C
CC
CCC
X
XX
XXX
Y
YY
YYY
Z
ZZ
ZZZ

- Bei der Übergabe einer Variablen per Referenzaufruf wird die Adresse der Variablen in der Reihenfolge Segment / Offset - Anteil auf den Stack gepusht.
- Eine Pointer-Variable enthält die Adresse eines Wertes in der Reihenfolge Offset / Segment - Anteil.

- Bei der Übergabe einer Pointer-Variablen per Wertaufruf wird der Inhalt in der Reihenfolge Segment / Offset - Anteil auf den Stack gepusht.
- Bei der Übergabe einer Pointer-Variablen per Referenzaufruf wird quasi die Adresse der Adresse eines Wertes übergeben.
- Für den Vergleich zweier Zeichenketten auf die Eigenschaft "Erste Zeichenkette größer als zweite Zeichenkette" bietet sich ein separates Unterprogramm mit nachfolgend in Pascal-Notation beschriebener Schnittstelle an :

```
Function GreaterString ( Var X,Y : String ) : Boolean
```

- Mittels der zeichenweisen Ausgabe einer Zeichenkette via DOS-Funktion 2 ("Write Character") können auch Dollarzeichen ausgegeben werden.

Und "zu guter letzt" noch ein Hinweis zur Benotung : Bei "ordentlicher" Bearbeitung der Routinen *ShowList* und *GreaterString* wird die Note "gut" gewährt. Für die Note "sehr gut" ist zusätzlich auch die "ordentliche" Bearbeitung der Routine *InsertItem* erforderlich, insbesondere die Behandlung eines Pointers als Referenzparameter. Zur "ordentlichen" Bearbeitung gehört auch die stichwortartige Kommentierung des entstehenden Assembler-Quellcodes, ggfs. ergänzt um "die eine oder andere" Skizze.

Das PTL-Team wünscht viel Erfolg