Physikalisch-Technische Lehranstalt Wedel

Staatsexamen für Technische Assistenten SoSe2001
schriftliche Prüfung im Fach Prozesstechnik
gewählter Vorschlag

Dauer : 180 Minuten **keine externen Hilfsmittel**

Aufgabe :

Die Übertragung von eMails im Internet läßt nur Texte in 7Bit-Zeichensätzen zu. Für eine Übertragung von Texten in 8Bit-Zeichensätzen ist beim Sender eine Codierung in ein 7Bit-Zeichensatz und beim Empfänger eine entsprechende Dekodierung notwendig.

Entwickeln Sie daher das Unterprogramm *Bit7And8* zur Umwandlung einer maximal 180 Zeichen langen IBM-ASCII-Codierten Zeichenkette in eine maximal 240 Zeichen lange Base64-Codierte Zeichenkette und umgekehrt durch komplette Übersetzung des nachfolgenden Pascal-Programms in ein 8086-Assembler-Programm. Zwecks Vereinfachung der Aufgabenstellung nimmt die Prozedur *Bit7And8* für die IBM-ASCII-Codierte Zeichenkette eine ganzzahlig durch drei ohne Rest teilbare und für die Base64-Codierte Zeichenkette eine ganzzahlig durch vier ohne Rest teilbare aktuelle Länge an.

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

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).
- Die Standardfunktion LONGINT wandelt Werte eines numerischen ordinalen Datentyps in den Datentyp *LongInt* um.

Anmerkung zum Ablauf des Programms *SoSe2001* : Der erste Aufruf der Prozedur *Show* bewirkt die Ausgabe

```
Moin Moin
TW9pbiBNb2lu
Moin Moin
```

Abschließend noch ein Tipp für Ihre Bearbeitungsreihenfolge (und meine spätere Beurteilung Ihrer Lösung) : Gesamtstruktur, Hauptprogramm, Prozedur *Show*, Prozedur *Bit7And8* Then-Zweig, Prozedur *Bit7And8* Else-Zweig.

Das PTL-Team wünscht viel Erfolg

```
Program SoSe2001;

Uses Crt;

Type String180 = String[180];
     String240 = String[240];

Var S : String180;
    C : Char;

Procedure Bit7And8(Var IbmAscii : String180;
                   Var Base64   : String240;
                       Encode   : Boolean);

Const Base64Code : Array[0..63] Of Char =
        ('A','B','C','D','E','F','G','H','I','J','K','L','M',
         'N','O','P','Q','R','S','T','U','V','W','X','Y','Z',
         'a','b','c','d','e','f','g','h','i','j','k','l','m',
         'n','o','p','q','r','s','t','u','v','w','x','y','z',
         '0','1','2','3','4','5','6','7','8','9','+','/');

Var Chars    : LongInt;
    CharBits : Array [1..4] Of Byte Absolute Chars;
    GZaehler : Byte;
    Zaehler  : Byte;
    Position : Byte;

Begin
  If Encode Then Begin
    GZaehler := 0;
    Base64 := '';
    Chars := 0;
    While GZaehler < Ord(IbmAscii[0]) Do Begin
      For Zaehler := 1 To 3 Do
        CharBits[4-Zaehler] := Ord(IbmAscii[GZaehler+Zaehler]);
      For Zaehler := 1 To 4 Do
        Base64 := Base64 +
                  (Base64Code[(Chars Shr ((4-Zaehler)*6)) And $3F]);
      GZaehler := GZaehler+3
    End
  End Else Begin
    GZaehler := 0;
    IbmAscii := '';
    While GZaehler < Ord(Base64[0]) Do Begin
      Chars := 0;
      For Zaehler := 1 To 4 Do Begin
        Position := 0;
        While Base64[GZaehler+Zaehler] <> Base64Code[Position] Do
          Position := Position+1;
        Chars := Chars + LongInt(Position) Shl ((4-Zaehler)*6)
      End;
      For Zaehler := 3 DownTo 1 Do
        IbmAscii := IbmAscii + (Chr(CharBits[Zaehler]));
      GZaehler := GZaehler+4
    End
  End
End;

Procedure Show(Var X:String180);

Var Y : String240;

Begin
  Writeln(X);
  Bit7And8(X,Y,True);
  WriteLn(Y);
  Bit7and8(X,Y,False);
  Writeln(X)
End;
```

```
Begin
  S := 'Moin Moin';
  Show(S);
  S := '';
  Repeat
    C := ReadKey;
    If C <> '$' Then Begin
      Write(C);
      S := S+C
    End
  Until ((C = '$') And (Ord(S[0]) Mod 3 = 0)) Or (Ord(S[0]) = 180);
  WriteLn;
  Show(S)
End.
```