

Übung zur Zeichenkettenverarbeitung

- Es ist ein Fortran-Programm zu schreiben, in dem ein Character-Feld `Namen` mit `nMax` Elementen und je `nZ` Zeichen zu vereinbaren ist.
Die Konstanten `nMax`, `nZ` sind durch eine `PARAMETER`-Anweisung festzulegen.
- Das Programm soll den Benutzer auffordern, eine Liste von Zeichenketten (Namen, Begriffe, ...) einzugeben.
Zum Einlesen einer Zeichenkette:

```
read(*,'(A)') CharVariable
```

Die Eingabe soll beendet werden,

- a) wenn der Benutzer eine „leere“ Zeichenkette eingibt (nur `Enter`-Taste oder Leerzeichen, erkennbar daran, dass `Len_Trim(str)` gleich Null ist), oder
- b) wenn eine spezielle (vorher vereinbarte) Zeichenkette eingegeben wurde (z. B. `#`) oder
- c) sobald die maximal vorgesehene Feldgröße (`nMax`) überschritten wird oder
- d) wenn das physische Ende der Eingabedatei erreicht wurde (z.B. bei File-Umlenkung).

Nach Ende der Eingabe soll das Programm ausgeben, welches dieser Kriterien zu diesem Ende geführt hat und wieviele Wörter gelesen wurden.

Hinweis: Um das Programm mehrfach mit gleichen Eingabedaten testen zu können, empfiehlt es sich, mit dem Editor eine Eingabedatei vorzubereiten (z. B. unter dem Namen `liste.txt`), die durch File-Umlenkung als Standardeingabedatei verwendet wird:

```
./myprog < liste.txt
```

Erkennung des physischen Endes einer Eingabedatei:

```
READ (*,'(A)',END=999) CharVariable
...                ! Verarbeitung der gelesenen Zeichenkette
999 CONTINUE       ! Reaktion auf Dateiende ab hier
```

- Die eingegebenen Begriffe sollen anschließend in alphabetischer Reihenfolge wieder ausgegeben werden, ohne ihre Position im Speicher zu verändern (Sortieren mit Hilfe eines Index- oder Permutationsvektors).
Das Ordnen der Indizes kann sofort nach dem Einlesen eines jeden Wortes erfolgen, d. h. an welche Position ist das aktuelle Wort in die bisherige (bereits geordnete) Liste einzufügen?
- Zum Testen kann es hilfreich sein, die ungeordnete und die geordnete Wort-Liste nebeneinander mit entsprechenden Indizes anzuzeigen:

```
WRITE (*,*) k, Namen(k), ind(k), Namen(ind(k))
C          Einlesereihenfolge  Sortierreihenfolge
```

- **Zusatz:** Die Ausgabe der alphabetisch geordneten Begriffe soll möglichst **kompakt** erfolgen, d.h. mehrere Begriffe in einer Zeile ohne Leerzeichen, aber durch Komma getrennt. Eine maximale Zeilenlänge (z.B. 80 Zeichen) soll dabei nicht überschritten werden (neue Zeile, wenn das nächste Wort zu lang ist).

Vorschlag: Eine `Character`-Variable `ZEILE` vereinbaren und als „Ausgabepuffer“ verwenden (Ausgabe, falls Restlänge weniger als nächstes Wort – oder Ende der Wörterliste erreicht).

- Weitere zusätzliche Funktionen z.B.: statistische Auswertungen (in wievielen Wörtern kommt eine bestimmte Buchstabengruppe vor, wieviele Wörter beginnen mit 'x' oder enden auf 'y', ...)

- **Modifikation des Programms als „Filter“:**

von eingelesenen Zeilen (z.B. eines Fortran-Programms) werden nach bestimmten Kriterien ausgewählte Zeilen wieder ausgegeben (wie beim `grep`-Kommando von Unix), z.B. alle Zeilen, die eines der Worte `integer` oder `real` enthalten.

(Dabei ist es nicht notwendig, alle eingelesenen Zeilen als Character-Feld zu speichern.)

oder

alle Buchstaben in Großbuchstaben umwandeln, also ein „Übersetzungsprogramm“ (in Unix als `tr`-Kommando bekannt).