

Cheat Sheet

- I/O, Verzweigungen, Schleifen

Aufgabe 1:

Schreibt ein Programm, das zunächst "Hello World!" auf der Konsole ausgibt und anschließend nur die ersten 5 Zeichen von "Hello World!".

Tipps:

Formatierung von Ein- und Ausgaben:

Variable Type	Edit Descr.	Example	Comment
string	a	a11	A string is printed into an 11 character wide output field
string	a	a	The length of the output field is the length of the string
integer	i	i4	An integer is printed into a 4 character wide output field
integer	i	i	INVALID! The length has to be specified
real	f	f12.5	A real is printed into a 12 character wide output field. The 5 after the decimal point means that 5 decimal digits of the floating point number are to be printed. No exponent is used with this edit descriptor
real	e	e12.5	A real is printed with an exponent
real	es	es12.5	Same as e, but number does not start with a zero
whitespace	x	x	Add a blank character
whitespace	x	4x	Four whitespace characters are added

Konsolenausgabe:

```
print <format>, <list>
```

→ *list* = ein oder mehrere Werte/Variablen, getrennt durch ein Kommas

Aufgabe 2:

Schreibe ein Programm, das die Variable x 25-mal und die Variable y 20-mal erhöht.

Tipps:

Schleifen in Fortran:

```
do i = loop_start, loop_end, loop_step  
  <code>  
enddo
```

IF in Fortran:

```
if (<logical expression>) <code>
```

Befehle in Schleifen:

```
exit/cycle
```

Modulo in Fortran mit mod(a,b).

Aufgabe 3:

Schreibt ein Programm, das zunächst via Kommandozeile Name und Alter abfragt und anschließend den Input in einer Datei ablegt.

Tipps:

Schreiben:

```
write(<unit>, <format>) <list>
```

→ optional: Zeilenumbruch vermeiden mit *advance='no'*.

Lesen:

```
read (<unit>, <format>) <list>
```

Datei öffnen/schließen:

```
open (unit=10, file='data.txt', iostat=ios)
```

```
...
```

```
close(10)
```

Werte für unit:

6: Konsolenausgabe (alternativ: *)

5: Konsoleneingabe (alternativ: *)

0-255: Kann an Datei gebunden werden (0,5 & 6 bereits vergeben!)

iostat: integer, gibt an ob Dateioperation erfolgreich war (wenn =0, dann erfolgreich)

Aufgabe 4:

Schreibt ein Programm, dass zunächst eine Datei einliest und anschließend auf der Konsole die Anzahl der Zeilen in dieser Datei ausgibt.

Tipps:

Datei öffnen:

```
open(unit, file='bla.txt', iostat=ios, status='old') line
```

→ *status='old'*: setzt voraus, dass es diese Datei gibt, wenn Datei nicht existiert, dann Fehler

Lesen:

```
read(unit, format, iostat=ios) line
```

→ *iostat*: integer, gibt an ob Dateioperation erfolgreich war (wenn =0, dann erfolgreich)

Default: read liest Datei Zeile für Zeile ein. Bei EOF wird ios /= 0.

Aufgabe 5:

Zusatz: Erweitert das Programm der vorherigen Übung (Heron & Pythagoras), sodass Ein- und Ausgabe benutzerfreundlicher sind (z.B. welche Formel soll auf Eingabe angewendet werden, genaue Fehlerangabe beispielsweise "c ist zu klein" o.ä.).

Tipps:

Siehe vorherige Aufgaben. Siehe Lösung zu Übung 1.