

ConsoleApp_String

Console applications for string and transformation (German):

2DEC

Argument in Dezimalzahl.

Handhabung:

```
2DEC [x]  
[x] ... Argument
```

DEC2

Dezimalzahl Umwandlung.

Handhabung:

```
DEC2 [d]  
[d] ... Dezimalzahl
```

BIN2DEC

Binärzahl in Dezimalzahl.

Handhabung:

```
BIN2DEC [b]  
[b] ... Binärzahl
```

DEC2BIN

Dezimalzahl in Binärzahl.

Handhabung:

```
DEC2BIN [d]  
[d] ... Dezimalzahl
```

ABCD

Ändert oder löscht Strings einer ASCII Datei.

- Übernahme einer ASCII Datei.
- Ausgabe einer modifizierten ASCII Datei.

Handhabung:

```
ABCD [input] [output] [ab] [cd] [sw]
[input] ... Eingabe Datei
[output] .. Ausgabe Datei
[ab] ..... String der zu modifizieren ist (String_alt)
[cd] ..... String der eingefügt wird (String_neu)
[sw] ..... (0):modifiziert (1):löscht String_alt (2):fügt ein Leerzeichen ein
```

Bsp.:

```
ABCD input.txt output.txt .txt .asc 0
ABCD input.txt output.txt REM 0 1
```

AZUBE

Änderung einzelner Zeichen einer ASCII Datei.

Handhabung:

```
AZUBE [input] [output] [a] [b]
[input] ... Eingabe Datei
[output] .. Ausgabe Datei
[a] ..... Zeichen
[b] ..... ändern
```

Bsp.:

```
AZUBE input.txt output.txt , .
```

TEZUTE

Änderung einzelner Strings einer Datei.

Handhabung:

```
TEZUTE [input] [output] [a] [b]
[input] ... Eingabe Datei
[output] .. Ausgabe Datei
[a] ..... String
[b] ..... ändern
```

Bsp.:

```
78  00
67   8
null 00
9  null
```

```
TEZUTE input.txt output.txt null 00
```

```
78
00
67
8
00
00
9
00
```

Zur formatierten Ausgabe Quellcode bearbeiten.

AB_L

Fügt einen String (a) vor und/oder einen String (b) nach jeder Zeile einer Datei ein.

- Übernahme einer einspaltigen ASCII Datei.
- Ausgabe einer (ein-,) zwei- oder dreispaltigen ASCII Datei.

Handhabung:

```
AB_L [input] [output] [a] [sw] [z] [sw]
[input] ... Eingabe Datei
[output] .. Ausgabe Datei
[a]..... String welcher vor jeder Zeile der Inputdatei eingefügt wird
[sw].....(0): nichts vor jeder Zeile einfügen
[z]..... String welcher nach jeder Zeile der Inputdatei eingefügt wird
[sw].....(0): nichts nach jeder Zeile einfügen
```

Bsp.:

```
AB_L in.txt out.txt move 1 verzeichnis 1
AB_L in.txt out.txt del 1 0 0
```

BIIA

Encodiert eine Datei in eine ASCII Datei (Recodierung via AIIB.EXE).

Arbeitsweise:

- Übernahme einer Datei.
- Ausgabe einer encodierten ASCII Datei (*.bii).

Handhabung:

```
BIIA [input] [output] [mode] [form]
[input] .... Eingabe Datei
[output] ... Ausgabe Datei im ASCII Format (*.bii)
[mode] ..... (1):cd01 SCHRAUSSER CODE cd01 Grossbuchstaben-Kombination
               (2) Zeichen, bsp: AA, FD, ..., gute Lesbarkeit - Zeicheneindeutigkeit
zur Archivierung)(2):cd02 SCHRAUSSER CODE cd02 Tastaturzeichen-Kombination (1 oder 2
Zeichen, bsp: a, 1A, ..., geringer Dateiumfang)
               (3):cd03 SCHRAUSSER CODE cd03 Zahlen-Kombination (3 Zeichen, bsp: 001,
123, ..., maximale Lesbarkeit)
               (4):cd04 SCHRAUSSER CODE cd04 ASCII-Kombination (1 oder 2 Zeichen, bsp:
-, #+, ..., minimaler Dateiumfang)
               (5):HEX Hexadezimalwert - Keine AIIB Recodierung
               (6):DEZ Dezimalwert
               (7):OKT Oktalwert - Keine AIIB Recodierung
               (8):ASCII-Zeichen direkt - Keine AIIB Recodierung
[form] ..... (1):1-spaltig ASCII Ausgabe
               (2):1-zeilig ASCII Ausgabe im Block ohne Seitenumbruch
```

Bsp.:

```
BIIA bild.jpg bild.bii 4 2
```

AIIB

Recodiert eine SCHRAUSSER CODE oder dezimal encodierte Datei (Encodierung via BIIA.EXE).

Arbeitsweise:

- Übernahme einer SCHRAUSSER CODE Datei (*.bii).
- Ausgabe der recodierten Ursprungsdatei.

Handhabung:

```
AIIB [input]
[input] ... SCHRAUSSER CODE Datei
```

Bsp.:

```
AIIB bild.bii
```

STGERSETZ

Änderung ein oder mehrerer Substrings (Einzelzeichen oder Zeichenketten) in einem zusammenhängenden String (ohne Leerzeichen), eingelesen als Zeile einer ASCII Datei.

Arbeitsweise:

- Übernahme einer Zeile aus einem ASCII-File.

Bsp.:

```
12345678
abcdefghijk
ABCDEFGG
1234567890
```

- zeilenweise Bearbeitung der Strings und Ausgabe in ein ASCII-File.

Die Bearbeitungsschritte werden in stgersez_log.txt protokolliert.

Handhabung:

```
STGERSETZ [input] [output] [typ] [alt] [neu]
[input] .... Eingabe Datei (Strings zeilenweise)
[output] ... Ausgabe Datei (Strings zeilenweise)
[typ] ..... (0):löschen
              (1):ändern
              (2):ein Leerzeichen einfügen
[alt] ..... zu ändernde Zeichenkette (Substring) in Eingabe Datei
[neu] ..... ändern in Zeichenkette (Substring) in Ausgabe Datei
```

Die Art des Arguments [neu] hat für typ (0) löschen und (2) ein Leerzeichen einfügen keine Auswirkung, muss aber als beliebiges Zeichen angegeben werden (iuf. '-'). Ein Leerzeichen in der Eingabedatei trennt die Zeichenkette und führen zu einer neuen Zeile in der Ausgabedatei. Manche Sonderzeichen bleiben unberücksichtigt.

Bsp.:

```
STGERSETZ input.txt output.txt 0 8 - (löscht 8)

1234567
abcDEFghijk
ABCDEFGG
```

1234567890

STGERSETZ input.txt output.txt 1 def DEF (ändert def in DEF 8)

12345678

abcDEFghijk

ABCDEFG

1234567890

STGERSETZ input.txt output.txt 2 234 - (fügt für 234 ein Leerzeichen ein)

1 5678

abcdefghijk

ABCDEFG

1 567890