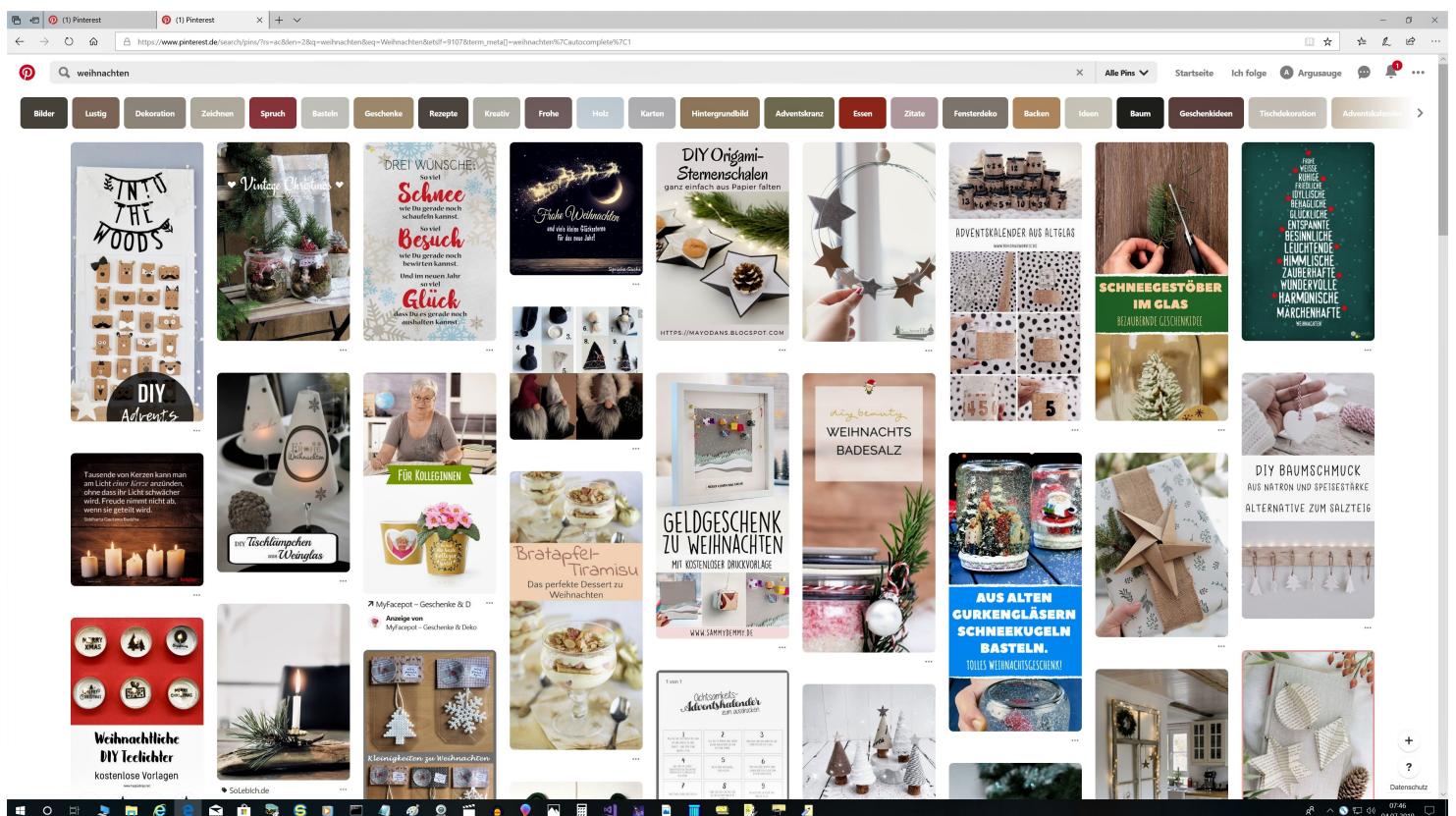


Weihnachten 2019a.pdf

Diese Datei ist eine Pseudo-Pinnwand zum Thema Weihnachtsbilder aus Pinterest
Außerdem dient sie für Formatierungsstudien für L^AT_EX-Dateien mit Quellcode

In der Datei "GenialesPortal2019a.pdf" gibt es eine umfangreiche Erläuterung über das Bildarchiv Pinterest. Dabei wird auch erklärt warum ich eigene Pseudo-Pinnwände zusammengestellt und in meine Homepage eingebunden habe.



Die PDF-Dateien in meiner Homepage wurden alle mit pdf L^AT_EX produziert. Das hyperref Paket ermöglicht die Nutzung von WEB-Verlinkungen. Klickt man die folgenden Links an, dann wird diese Datei geladen.

<http://www.institut-wolfgang-renner.de/HauptSeite2019a.pdf>
<http://www.institut-wolfgang-renner.de/SchlossFuenfeckInhalte2019a.pdf>

Rechtliche Aspekte sind auch in "GenialesPortal2019a.pdf" diskutiert !

Inhaltsverzeichnis

1	Erster Formatierungstest	3
1.1	Sektion 1.1	3
1.2	Sektion 1.2	3
1.2.1	UnterSektion 1.2.1	3
1.2.2	UnterSektion 1.2.2	3
1.2.3	UnterSektion 1.2.3	3
1.2.3.1	UnterUnterSektion 1.2.3.1	3
1.2.3.2	UnterUnterSektion 1.2.3.2	3
1.2.3.3	UnterUnterSektion 1.2.3.3	3
1.2.3.4	UnterUnterSektion 1.2.3.4	3
1.2.4	UnterSektion 1.2.4	3
1.2.5	UnterSektion 1.2.5	3
2	Zweiter Formatierungstest	4
2.1	Sektion 2.1	4
2.2	Sektion 2.2	4
2.2.1	UnterSektion 2.2.1	4
2.2.2	UnterSektion 2.2.2	4
2.2.3	UnterSektion 2.2.3	4
2.2.3.1	UnterUnterSektion 2.2.3.1	4
2.2.3.2	UnterUnterSektion 2.2.3.2	4
2.2.3.3	UnterUnterSektion 2.2.3.3	4
2.2.3.4	UnterUnterSektion 2.2.3.4	4
2.2.4	UnterSektion 2.2.4	4
2.2.5	UnterSektion 2.2.5	4
3	Bilder aus Pinterest	5
4	Formating studies	13
4.1	Parallel environment	13
4.2	Assembler listing with 112 characters width	14
4.3	DspLib-Data files with 124 characters width	15
4.4	DspLib-Data files with 136 characters width	16
4.5	Fortran listing with 148 characters width	17
4.6	C/CPP listing with 160 characters width	18
4.7	C/CPP listing with 172 characters width	19
4.8	DspLib-Data files with 208 characters width	20
4.9	Experimente mit Typewriter Fonts	21
4.9.0.1	MATLIB MATh LIBrary main computation function	23
4.9.0.2	MLWRIT Math Library WRITe function	27
4.9.1	Input Output Functions	38
4.9.1.1	GETMAT GET MATh library constansts	39
4.9.1.2	FLWRIT FiLe WRITe function	43

Kapitel 1

Erster Formatierungstest

1.1 Sektion 1.1

1.2 Sektion 1.2

1.2.1 UnterSektion 1.2.1

1.2.2 UnterSektion 1.2.2

1.2.3 UnterSektion 1.2.3

1.2.3.1 UnterUnterSektion 1.2.3.1

1.2.3.2 UnterUnterSektion 1.2.3.2

1.2.3.3 UnterUnterSektion 1.2.3.3

1.2.3.4 UnterUnterSektion 1.2.3.4

1.2.4 UnterSektion 1.2.4

1.2.5 UnterSektion 1.2.5

Kapitel 2

Zweiter Formatierungstest

2.1 Sektion 2.1

2.2 Sektion 2.2

2.2.1 UnterSektion 2.2.1

2.2.2 UnterSektion 2.2.2

2.2.3 UnterSektion 2.2.3

2.2.3.1 UnterUnterSektion 2.2.3.1

2.2.3.2 UnterUnterSektion 2.2.3.2

2.2.3.3 UnterUnterSektion 2.2.3.3

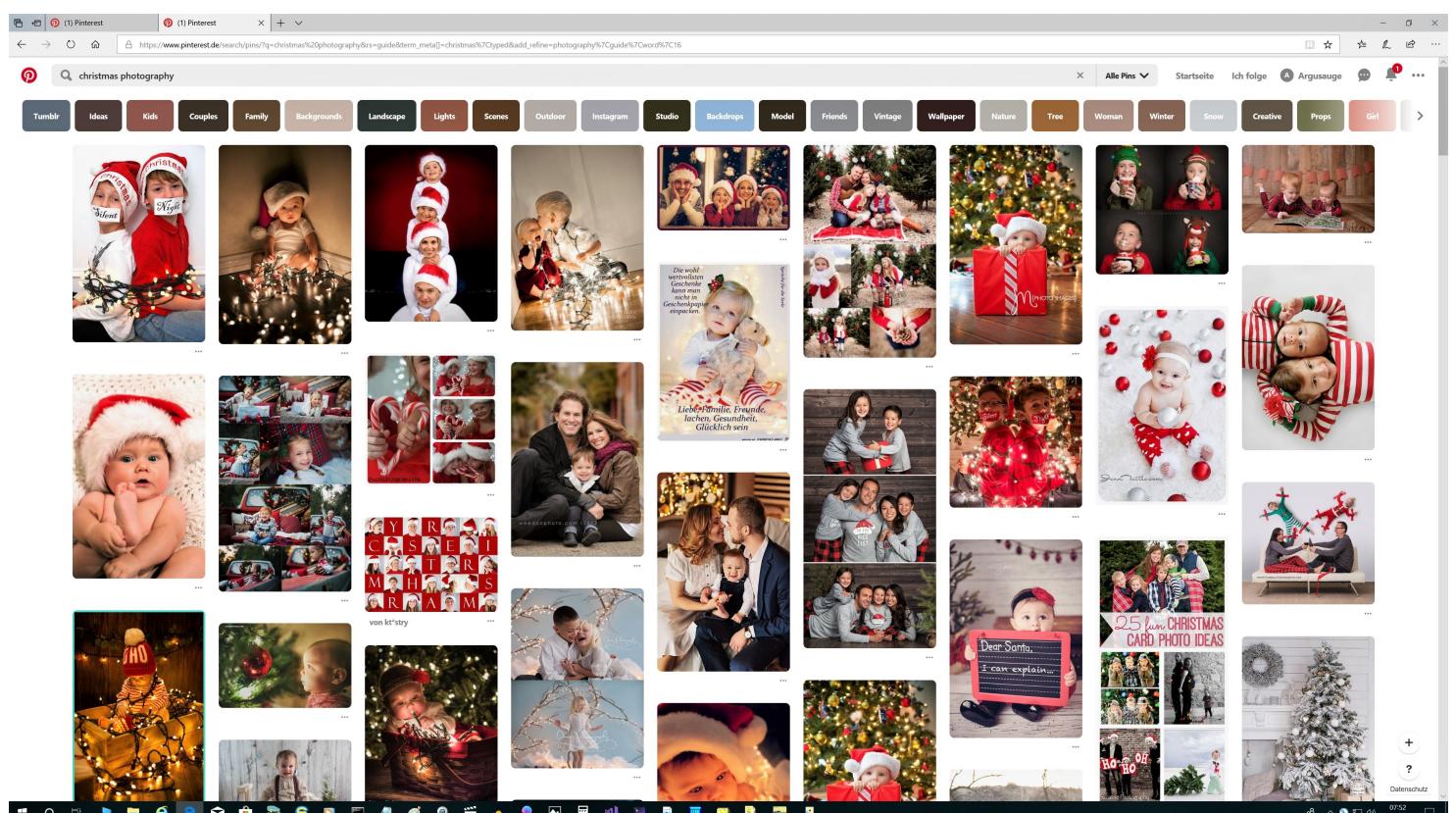
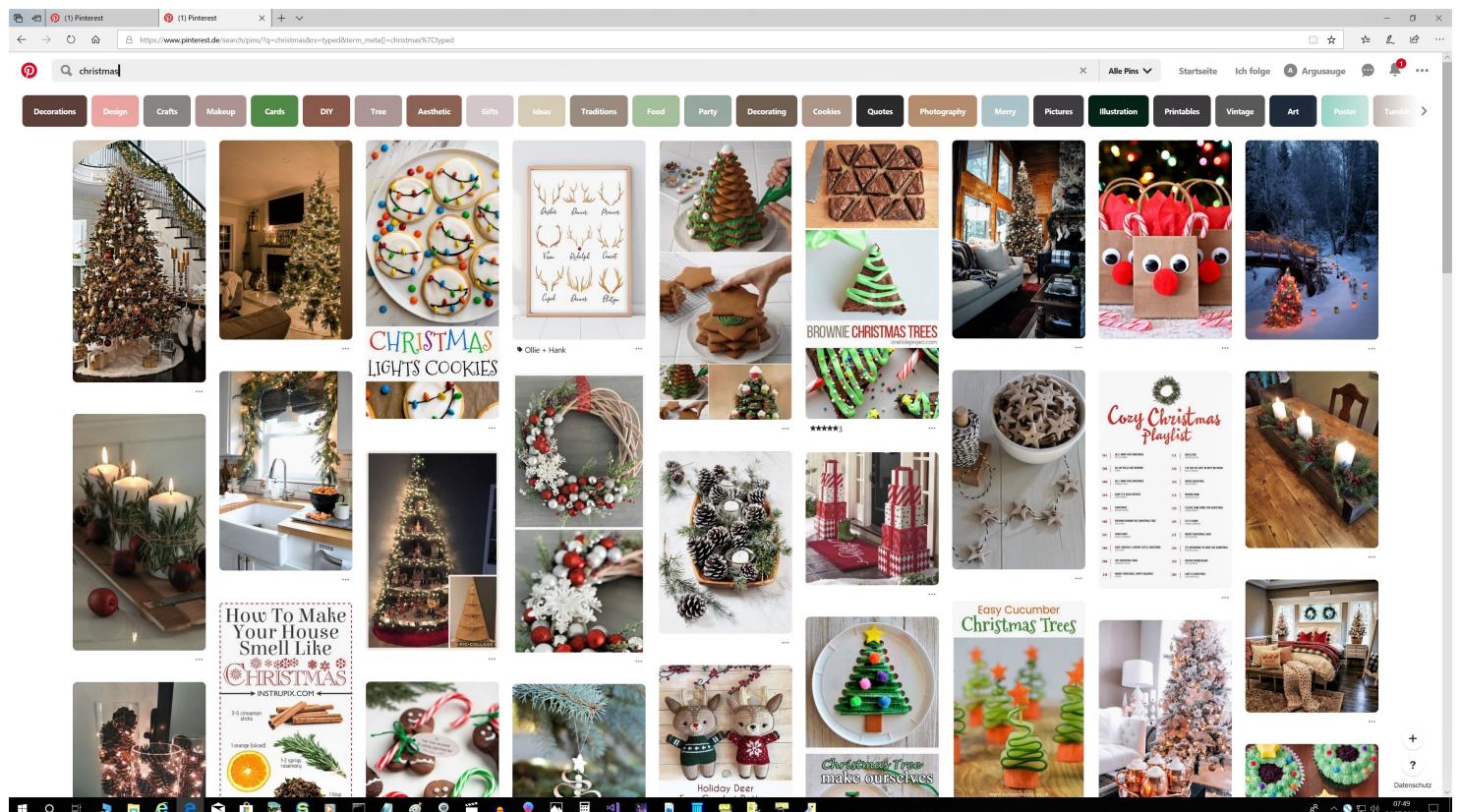
2.2.3.4 UnterUnterSektion 2.2.3.4

2.2.4 UnterSektion 2.2.4

2.2.5 UnterSektion 2.2.5

Kapitel 3

Bilder aus Pinterest

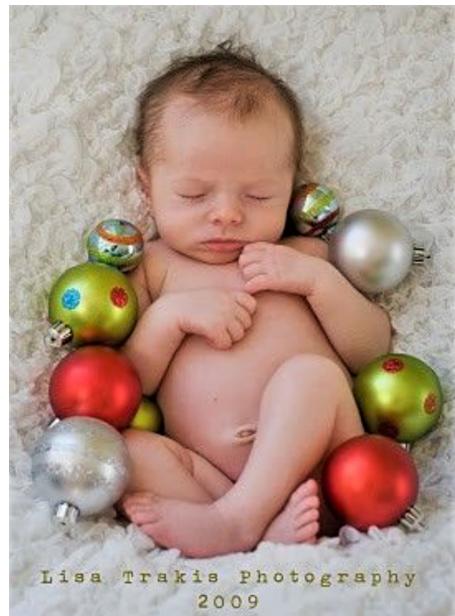


The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.



The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console.





The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.



The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.



The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console.



The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.



The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

Kapitel 4

Formatting studies

4.1 Parallel environment

Die vorliegende Datei zeigt einige Bilder, welche ich gezeichnet habe. Am Ende sind zwei Bücher von meinem Großonkel dazu gefügt. Es gibt architektonische Zeichnungen. Sowie körperliche Darstellungen. Ich habe mich von klein auf für das Bauwesen interessiert. Meine guten Schulfächer waren die MINT-Fächer (Mathematik, Informatik, Naturwissenschaften und Technik), sowie graphische (Malen) und bildende Kunst (Töpfen usw.). Ein Musikinstrument habe ich nie gelernt. Meine Generation hat sich vorwiegend mit den Themen beschäftigt, denen ich auch zugeneigt war. Aktiver Musikunterricht fand oftmals überhaupt nicht statt. Heute ist das Musizieren bei Jugendlichen wesentlich populärer als in meiner Generation. Dafür ist beispielsweise Elektronik basteln und allgemeines Werken ziemlich out.

Die vorliegende Datei zeigt einige Bilder, welche ich gezeichnet habe. Am Ende sind zwei Bücher von meinem Großonkel dazu gefügt. Es gibt architektonische Zeichnungen. Sowie körperliche Darstellungen. Ich habe mich von klein auf sehr für das Bauwesen interessiert. Meine guten Schulfächer waren die MINT-Fächer (Mathematik, Informatik, Naturwissenschaften und Technik), sowie graphische (Malen) und bildende Kunst (Töpfen usw.). Ein Musikinstrument habe ich nie gelernt. Meine Generation hat sich vorwiegend mit den Themen beschäftigt, denen ich auch zugeneigt war. Aktiver Musikunterricht fand oftmals überhaupt nicht statt. Heute ist das Musizieren bei Jugendlichen wesentlich populärer als in meiner Generation. Dafür ist beispielsweise Elektronik basteln und allgemeines Werken ziemlich out.

Die vorliegende Datei zeigt einige Bilder, welche ich gezeichnet habe. Am Ende sind zwei Bücher von meinem Großonkel dazu gefügt. Es gibt architektonische Zeichnungen. Sowie körperliche Darstellungen. Ich habe mich von klein auf sehr für das Bauwesen interessiert. Meine guten Schulfächer waren die MINT-Fächer (Mathematik, Informatik, Naturwissenschaften und Technik), sowie graphische (Malen) und bildende Kunst (Töpfen usw.). Ein Musikinstrument habe ich nie gelernt. Meine Generation hat sich vorwiegend mit den Themen beschäftigt, denen ich auch zugeneigt war. Aktiver Musikunterricht fand oftmals überhaupt nicht statt. Heute ist das Musizieren bei Jugendlichen wesentlich populärer als in meiner Generation. Dafür ist beispielsweise Elektronik basteln und allgemeines Werken ziemlich out.

Die vorliegende Datei zeigt einige Bilder, welche ich gezeichnet habe. Am Ende sind zwei Bücher von meinem Großonkel dazu gefügt. Es gibt architektonische Zeichnungen. Sowie körperliche Darstellungen. Ich habe mich von klein auf sehr für das Bauwesen interessiert. Meine guten Schulfächer waren die MINT-Fächer (Mathematik, Informatik, Naturwissenschaften und Technik), sowie graphische (Malen) und bildende Kunst (Töpfen usw.). Ein Musikinstrument habe ich nie gelernt. Meine Generation hat sich vorwiegend mit den Themen beschäftigt, denen ich auch zugeneigt war. Aktiver Musikunterricht fand oftmals überhaupt nicht statt. Heute ist das Musizieren bei Jugendlichen wesentlich populärer als in meiner Generation. Dafür ist beispielsweise Elektronik basteln und allgemeines Werken ziemlich out.

4.2 Assembler listing with 112 characters width

The given L^AT_EX-File tells something about Pinterest and serves for formatingstudies on typewriter fonts inside "verbatim" environments. For the documentation of my DspLib-Project and my X86-128 architecture, I will need "verbatim" environments with linelength of 112 characters (Assembly), 148 characters (Fortran) and 160 characters (C/CPP). Furthermore linelength of 100,112,124,136,148,160,172,184,196,208 may be used for DspLib data files.

Die vorliegende L^AT_EX-Datei erzählt etwas über Pinterest und dient für Formatierungsstudien von Schreibmaschinen Zeichensätze innerhalb der "verbatim" Umgebung. Für die Dokumentation von meinem DspLib-Projekt und meiner X86-128 Architektur werde ich "verbatim" Umgebungen mit Zeilenlängen von 112 Zeichen (Assembler), 148 Zeichen (Fortran) und 160 Zeichen (C/CPP) brauchen. Zudem 100,112,124,136,148,160,172,184,196,208 für Daten.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

4.3 DspLib-Data files with 124 characters width

The given L^AT_EX-File tells something about Pinterest and serves for formatingstudies on typewriter fonts inside "verbatim" environments. For the documentation of my DspLib-Project and my X86-128 architecture, I will need "verbatim" environments with linelength of 112 characters (Assembly), 148 characters (Fortran) and 160 characters (C/CPP). Furthermore linelength of 100,112,124,136,148,160,172,184,196,208 may be used for DspLib data files.

Die vorliegende L^AT_EX-Datei erzählt etwas über Pinterest und dient für Formatierungsstudien von Schreibmaschinen Zeichensätze innerhalb der "verbatim" Umgebung. Für die Dokumentation von meinem DspLib-Projekt und meiner X86-128 Architektur werde ich "verbatim" Umgebungen mit Zeilenlängen von 112 Zeichen (Assembler), 148 Zeichen (Fortran) und 160 Zeichen (C/CPP) brauchen. Zudem 100,112,124,136,148,160,172,184,196,208 für Daten.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

4.4 DspLib-Data files with 136 characters width

The given L^AT_EX-File tells something about Pinterest and serves for formatingstudies on typewriter fonts inside "verbatim" environments. For the documentation of my DspLib-Project and my X86-128 architecture, I will need "verbatim" environments with linelength of 112 characters (Assembly), 148 characters (Fortran) and 160 characters (C/CPP). Furthermore linelength of 100,112,124,136,148,160,172,184,196,208 may be used for DspLib data files.

Die vorliegende L^AT_EX-Datei erzählt etwas über Pinterest und dient für Formatierungsstudien von Schreibmaschinen Zeichensätze innerhalb der "verbatim" Umgebung. Für die Dokumentation von meinem DspLib-Projekt und meiner X86-128 Architektur werde ich "verbatim" Umgebungen mit Zeilenlängen von 112 Zeichen (Assembler), 148 Zeichen (Fortran) und 160 Zeichen (C/CPP) brauchen. Zudem 100,112,124,136,148,160,172,184,196,208 für Daten.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

4.5 Fortran listing with 148 characters width

The given L^AT_EX-File tells something about Pinterest and serves for formatingstudies on typewriter fonts inside "verbatim" environments. For the documentation of my DspLib-Project and my X86-128 architecture, I will need "verbatim" environments with linelength of 112 characters (Assembly), 148 characters (Fortran) and 160 characters (C/CPP). Furthermore linelength of 100,112,124,136,148,160,172,184,196,208 may be used for DspLib data files.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

Die vorliegende L^AT_EX-Datei erzählt etwas über Pinterest und dient für Formatierungsstudien von Schreibmaschinen Zeichensätze innerhalb der "verbatim" Umgebung. Für die Dokumentation von meinem DspLib-Projekt und meiner X86-128 Architektur werde ich "verbatim" Umgebungen mit Zeilenlängen von 112 Zeichen (Assembler), 148 Zeichen (Fortran) und 160 Zeichen (C/CPP) brauchen. Zudem 100,112,124,136,148,160,172,184,196,208 für Daten.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

4.6 C/CPP listing with 160 characters width

The given L^AT_EX-File tells something about Pinterest and serves for formatingstudies on typewriter fonts inside "verbatim" environments. For the documentation of my DspLib-Project and my X86-128 architecture, I will need "verbatim" environments with linelength of 112 characters (Assembly), 148 characters (Fortran) and 160 characters (C/CPP). Furthermore linelength of 100,112,124,136,148,160,172,184,196,208 may be used for DspLib data files.

Die vorliegende L^AT_EX-Datei erzählt etwas über Pinterest und dient für Formatierungsstudien von Schreibmaschinen Zeichensätze innerhalb der "verbatim" Umgebung. Für die Dokumentation von meinem DspLib-Projekt und meiner X86-128 Architektur werde ich "verbatim" Umgebungen mit Zeilenlängen von 112 Zeichen (Assembler), 148 Zeichen (Fortran) und 160 Zeichen (C/CPP) brauchen. Zudem 100,112,124,136,148,160,172,184,196,208 für Daten.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

4.7 C/CPP listing with 172 characters width

The given L^AT_EX-File tells something about Pinterest and serves for formatingstudies on typewriter fonts inside "verbatim" environments. For the documentation of my DspLib-Project and my X86-128 architecture, I will need "verbatim" environments with linelength of 112 characters (Assembly), 148 characters (Fortran) and 160 characters (C/CPP). Furthermore linelength of 100,112,124,136,148,160,172,184,196,208 may be used for DspLib data files.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

Die vorliegende L^AT_EX-Datei erzählt etwas über Pinterest und dient für Formatierungsstudien von Schreibmaschinen Zeichensätze innerhalb der "verbatim" Umgebung. Für die Dokumentation von meinem DspLib-Projekt und meiner X86-128 Architektur werde ich "verbatim" Umgebungen mit Zeilenlängen von 112 Zeichen (Assembler), 148 Zeichen (Fortran) und 160 Zeichen (C/CPP) brauchen. Zudem 100,112,124,136,148,160,172,184,196,208 für Daten.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

4.8 DspLib-Data files with 208 characters width

The given L^AT_EX-File tells something about Pinterest and serves for formatingstudies on typewriter fonts inside "verbatim" environments. For the documentation of my DspLib-Project and my X86-128 architecture, I will need "verbatim" environments with linelength of 112 characters (Assembly), 148 characters (Fortran) and 160 characters (C/CPP). Furthermore linelength of 100,112,124,136,148,160,172,184,196,208 may be used for DspLib data files.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

2

4.9 Experimente mit Typewriter Fonts

Die vorliegende Datei zeigt einige Bilder, welche ich gezeichnet habe. Am Ende sind zwei Bücher von meinem Großonkel dazu gefügt. Es gibt architektonische Zeichnungen. Sowie körperliche Darstellungen. Ich habe mich von klein auf für das Bauwesen interessiert. Meine guten Schulfächer waren die MINT-Fächer (Mathematik, Informatik, Naturwissenschaften und Technik), sowie graphische (Malen) und bildende Kunst (Töpfern usw.). Ein Musikinstrument habe ich nie gelernt. Meine Generation hat sich vorwiegend mit den Themen beschäftigt, denen ich auch zugeneigt war. Aktiver Musikunterricht fand oftmals überhaupt nicht statt. Heute ist das Musizieren bei Jugendlichen wesentlich populärer als in meiner Generation. Dafür ist beispielsweise Elektronik basteln und allgemeines Werken ziemlich out.

\fontsize{5pt}{5pt}

The documentation from DspLib and X86-128 architecture must be suited for international usage. This means english will be the mainly used language. As DspLib is designed to support multilanguage menu structures inside of the console as well inside of graphical oriented Displays, this document contains a dual language design with english as main language and german as addition. This english part spans the full textwidth to show it against the outcome of "verbatim" sections.

Footnotesize 124 Zeichen breit

Normal Size

4.9.0.1 MATLAB MATH LIBRARY main computation function

The function MATLIB reads unicode character string tokens from a diskfile or memoryarea. The diskfile or memory must be opened with the OPNFIL function. One data-token consists from an optional name string and an following optional data string. The name and the data part are separated with an equal character. Each token is ended with an token end character. This is either a comma or a semicolon.

Die Function MATLAB liest unicode charakter String Tokens von einer Diskdatei oder Speicherbereich. Die Diskdatei oder der Speicherbereich muss vor mit der OPNFIL Funktion geöffnet worden sein. Ein Daten-Token besteht aus einem optionalen Namen und einem optionalen Datenteil, getrennt mit einem Gleichheitszeichen. Jeder Token wird entweder mit einem Komma oder Semikolon beendet.

```

;* DL1(MATLIB,1IARXYPr16,IA3ANG,P64RES,P64AR1,P64AR2) !! P64RES = ARTAN2(P64AR1/P64AR2;IA3ANG) M1AR2TAN PPPI *
;* DL1(MATLIB,1IARXYPr17,IA3ANG,P64RES,P64AR1,P64AR2) !! P64RES = ARKOT2(P64AR1/P64AR2;IA3ANG) M1AR2KOT PPPI *
;* DL1(MATLIB,1IARXYPr18,UNUSED,P64RES,P64AR1,UNUSED) !! P64RES = EXPDUA[2.000]^P64AR1 M1EXPDUA PPUU *
;* DL1(MATLIB,1IARXYPr19,UNUSED,P64RES,P64AR1,UNUSED) !! P64RES = LOGDUA[00002](P64AR1) M1LOGDUA PPUU *

;* DL1(MATLIB,1IARXYPr20,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = P64AR1 + D08AR2 M1MATADD PPDU *
;* DL1(MATLIB,1IARXYPr21,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = P64AR1 - D08AR2 M1MATSUB PPDU *
;* DL1(MATLIB,1IARXYPr22,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = P64AR1 * D08AR2 M1MATMUL PPDU *
;* DL1(MATLIB,1IARXYPr23,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = P64AR1 / D08AR2 M1MATDIV PPDU *
;* DL1(MATLIB,1IARXYPr24,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = D08AR2 - P64AR1 M1REVSUB PPDU *
;* DL1(MATLIB,1IARXYPr25,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = D08AR2 / P64AR1 M1REVDIV PPDU *
;* DL1(MATLIB,1IARXYPr26,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = P64AR1 ^ D08AR2 M1MATPWR PPDU *
;* DL1(MATLIB,1IARXYPr27,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = MAROOT[D8AR2](P64AR1) M1MAROOT PPDU *
;* DL1(MATLIB,1IARXYPr28,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = D08AR2 ^ P64AR1 M1MATEXP PPDU *
;* DL1(MATLIB,1IARXYPr29,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = MATLOG[D8AR2](P64AR1) M1MATLOG PPDU *
;* DL1(MATLIB,1IARXYPr30,UNUSED,UNUSED,P64AR1,D08AR2) !! UNUSED = MATCMP(P64AR1,D08AR2) M1MATCMP UPDU *
;* DL1(MATLIB,1IARXYPr31,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = MATMIN(P64AR1,D08AR2) M1MATMIN PPDU *
;* DL1(MATLIB,1IARXYPr32,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = MATMAX(P64AR1,D08AR2) M1MATMAX PPDU *
;* DL1(MATLIB,1IARXYPr33,UNUSED,P64RES,P64AR1,D08AR2) !! P64RES = MATSGN(P64AR1,D08AR2) M1MATSGN PPDU *
;* DL1(MATLIB,1IARXYPr34,IA3ANG,P64RES,P64AR1,D08AR2) !! P64RES = ARSIN2(P64AR1/D08AR2;IA3ANG) M1AR2SIN PPDI *
;* DL1(MATLIB,1IARXYPr35,IA3ANG,P64RES,P64AR1,D08AR2) !! P64RES = ARCCOS2(P64AR1/D08AR2;IA3ANG) M1AR2COS PPDI *
;* DL1(MATLIB,1IARXYPr36,IA3ANG,P64RES,P64AR1,D08AR2) !! P64RES = ARTAN2(P64AR1/D08AR2;IA3ANG) M1AR2TAN PPDI *
;* DL1(MATLIB,1IARXYPr37,IA3ANG,P64RES,P64AR1,D08AR2) !! P64RES = ARKOT2(P64AR1/D08AR2;IA3ANG) M1AR2KOT PPDI *
;* DL1(MATLIB,1IARXYPr38,UNUSED,P64RES,P64AR1,UNUSED) !! P64RES = EXPDUA[2.000]^P64AR1 M1EXPDUA PPUU *
;* DL1(MATLIB,1IARXYPr39,UNUSED,P64RES,P64AR1,UNUSED) !! P64RES = LOGDUA[00002](P64AR1) M1LOGDUA PPUU *

;* DL1(MATLIB,1IARXYPr40,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = P64AR1 + I04AR2 M1MATADD PPIU *
;* DL1(MATLIB,1IARXYPr41,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = P64AR1 - I04AR2 M1MATSUB PPIU *
;* DL1(MATLIB,1IARXYPr42,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = P64AR1 * I04AR2 M1MATMUL PPIU *
;* DL1(MATLIB,1IARXYPr43,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = P64AR1 / I04AR2 M1MATDIV PPIU *
;* DL1(MATLIB,1IARXYPr44,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = I04AR2 - P64AR1 M1REVSUB PPIU *
;* DL1(MATLIB,1IARXYPr45,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = I04AR2 / P64AR1 M1REVDIV PPIU *
;* DL1(MATLIB,1IARXYPr46,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = P64AR1 ^ I04AR2 M1MATPWR PPIU *
;* DL1(MATLIB,1IARXYPr47,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = MAROOT[I4AR2](P64AR1) M1MAROOT PPDU *
;* DL1(MATLIB,1IARXYPr48,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = I04AR2 ^ P64AR1 M1MATEXP PPIU *
;* DL1(MATLIB,1IARXYPr49,UNUSED,P64RES,P64AR1,I04AR2) !! P64RES = MATLOG[I4AR2](P64AR1) M1MATLOG PPIU *

;* DL1(MATLIB,1IARXYPr50,UNUSED,P64RES,P64AR1,UNUSED) !! P64RES = P64AR1 ^ 000002 M1SQUARE PPUU *
;* DL1(MATLIB,1IARXYPr51,UNUSED,P64RES,P64AR1,UNUSED) !! P64RES = P64AR1 ^ 000003 M1MACUBE PPUU *
;* DL1(MATLIB,1IARXYPr52,UNUSED,P64RES,P64AR1,UNUSED) !! P64RES = SQROOT[00002](P64AR1) M1SQROOT PPUU *
;* DL1(MATLIB,1IARXYPr53,UNUSED,P64RES,P64AR1,UNUSED) !! P64RES = CBROOT[00003](P64AR1) M1CBROOT PPUU *
;* DL1(MATLIB,1IARXYPr54,IA3BIN,P64RES,P64AR1,IA2BIN) !! P64RES = MATBIN(P64AR1,IA2BIN;IA3BIN) M1MATBIN PPII *
;* DL1(MATLIB,1IARXYPr55,IA3GAM,P64RES,P64AR1,IA2GAM) !! P64RES = GAMFAC(P64AR1,IA2GAM;IA3GAM) M1GAMFAC PPII *
;* DL1(MATLIB,1IARXYPr56,IA3STO,P64RES,P64AR1,UNUSED) !! P64RES = ADDSTO(P64AR1,UNUSED;IA3STO) M1ADDSTO PPUI *
;* DL1(MATLIB,1IARXYPr57,IA3STO,P64RES,P64AR1,UNUSED) !! P64RES = SUBSTO(P64AR1,UNUSED;IA3STO) M1SUBSTO PPUI *
;* DL1(MATLIB,1IARXYPr58,IA3STO,P64RES,P64AR1,UNUSED) !! P64RES = MATSTO(P64AR1,UNUSED;IA3STO) M1MATSTO PPUI *
;* DL1(MATLIB,1IARXYPr59,IA3STO,P64RES,UNUSED,UNUSED) !! P64RES = RECALL(UNUSED,UNUSED;IA3STO) M1RECALL PUUI *

;* DL1(MATLIB,1IARXYPr60,IA3ANG,P64RES,P64AR1,IA2TAY) !! P64RES = TAYSIN(P64AR1,IA2TAY;IA3ANG) M1TAYSIN PPII *
;* DL1(MATLIB,1IARXYPr61,IA3ANG,P64RES,P64AR1,IA2TAY) !! P64RES = TAYCOS(P64AR1,IA2TAY;IA3ANG) M1TAYCOS PPII *
;* DL1(MATLIB,1IARXYPr62,IA3ANG,P64RES,P64AR1,IA2TAY) !! P64RES = TAYTAN(P64AR1,IA2TAY;IA3ANG) M1TAYTAN PPII *
;* DL1(MATLIB,1IARXYPr63,IA3ANG,P64RES,P64AR1,IA2TAY) !! P64RES = TAYKOT(P64AR1,IA2TAY;IA3ANG) M1TAYKOT PPII *
;* DL1(MATLIB,1IARXYPr64,IA3ANG,P64RES,P64AR1,IA2TAY) !! P64RES = TYASIN(P64AR1,IA2TAY;IA3ANG) M1TYASIN PPII *
;* DL1(MATLIB,1IARXYPr65,IA3ANG,P64RES,P64AR1,IA2TAY) !! P64RES = TYACOS(P64AR1,IA2TAY;IA3ANG) M1TYACOS PPII *
;* DL1(MATLIB,1IARXYPr66,IA3ANG,P64RES,P64AR1,IA2TAY) !! P64RES = TYATAN(P64AR1,IA2TAY;IA3ANG) M1TYATAN PPII *
;* DL1(MATLIB,1IARXYPr67,IA3ANG,P64RES,P64AR1,IA2TAY) !! P64RES = TYAKOT(P64AR1,IA2TAY;IA3ANG) M1TYAKOT PPII *
;* DL1(MATLIB,1IARXYPr68,UNUSED,P64RES,P64AR1,IA2TAY) !! P64RES = TAYEXP(P64AR1,IA2TAY;IA3BAS) M1TAYEXP PPII *
;* DL1(MATLIB,1IARXYPr69,UNUSED,P64RES,P64AR1,IA2TAY) !! P64RES = TAYLOG(P64AR1,IA2TAY;IA3BAS) M1TAYLOG PPII *

;* DL1(MATLIB,1IARXYPr70,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = MATSIN(P64AR1,IA2TAY;IA3ANG) M1MATTSIN PPUI *
;* DL1(MATLIB,1IARXYPr71,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = MATCOS(P64AR1,IA2TAY;IA3ANG) M1MATCOS PPUI *
;* DL1(MATLIB,1IARXYPr72,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = MATTAN(P64AR1,IA2TAY;IA3ANG) M1MATTAN PPUI *
;* DL1(MATLIB,1IARXYPr73,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = MATKOT(P64AR1,IA2TAY;IA3ANG) M1MATKOT PPUI *
;* DL1(MATLIB,1IARXYPr74,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = ARCSIN(P64AR1,IA2TAY;IA3ANG) M1ARCSIN PPUI *
;* DL1(MATLIB,1IARXYPr75,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = ARCCOS(P64AR1,IA2TAY;IA3ANG) M1ARCCOS PPUI *
;* DL1(MATLIB,1IARXYPr76,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = ARCTAN(P64AR1,IA2TAY;IA3ANG) M1ARCTAN PPUI *
;* DL1(MATLIB,1IARXYPr77,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = ARCKOT(P64AR1,IA2TAY;IA3ANG) M1ARCKOT PPUI *
;* DL1(MATLIB,1IARXYPr78,IA3ANG,P64RES,P64AR1,UNUSED) !! P64RES = CVTANG(P64AR1,IA2TAY;IA3ANG) M1CVTANG PPUI *
;* DL1(MATLIB,1IARXYPr79,IA3CST,P64RES,P64AR1,UNUSED) !! P64RES = GETCST(P64AR1,IA2TAY;IA3CST) M1GETCST PPUI *

;* DL1(MATLIB,1IARXYPr80,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = MATCOP(C64AR1,UNUSED;IA3CVT) M1MATCOP PCUI *
;* DL1(MATLIB,1IARXYPr81,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = CHGSGN(C64AR1,UNUSED;IA3CVT) M1CHGSGN PCUI *
;* DL1(MATLIB,1IARXYPr82,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = MATINT(C64AR1,UNUSED;IA3CVT) M1MATINT PCUI *
;* DL1(MATLIB,1IARXYPr83,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = MANINT(C64AR1,UNUSED;IA3CVT) M1MANINT PCUI *
;* DL1(MATLIB,1IARXYPr84,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = MFLOOR(C64AR1,UNUSED;IA3CVT) M1MFLOOR PCUI *
;* DL1(MATLIB,1IARXYPr85,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = MACEIL(C64AR1,UNUSED;IA3CVT) M1MACEIL PCUI *
;* DL1(MATLIB,1IARXYPr86,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = MANTIS(C64AR1,IA2TAY;IA3CVT) M1MANTIS PCUI *
;* DL1(MATLIB,1IARXYPr87,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = FLOLOG(C64AR1,IA2TAY;IA3CVT) M1FLOLOG PCUI *
;* DL1(MATLIB,1IARXYPr88,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = EXPDEC(C64AR1,IA2TAY;IA3CVT) M1EXPDEC PCUI *
;* DL1(MATLIB,1IARXYPr89,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = LOGDEC(C64AR1,IA2TAY;IA3CVT) M1LOGDEC PCUI *

;* DL1(MATLIB,1IARXYPr90,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = SINHYP(C64AR1,IA2TAY;IA3CVT) M1SINHYP PCUI *
;* DL1(MATLIB,1IARXYPr91,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = COSHYP(C64AR1,IA2TAY;IA3CVT) M1COSHYP PCUI *
;* DL1(MATLIB,1IARXYPr92,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = TANHYP(C64AR1,IA2TAY;IA3CVT) M1TANHYP PCUI *
;* DL1(MATLIB,1IARXYPr93,IA3CVT,P64RES,C64AR1,UNUSED) !! P64RES = KOTHYP(C64AR1,IA2TAY;IA3CVT) M1KOTHYP PCUI *

```


4.9.0.2 MLWRIT Math Library WRITe function

The function MLWRIT reads unicode character string tokens from a diskfile or memoryarea. The diskfile or memory must be opened with the OPNFIL function. One data-token consists from an optional name string and an following optional data string. The name and the data part are separated with an equal character. Each token is ended with an token end character. This is either a comma or a semicolon.

Die Function MLWRIT liest unicode charakter String Tokens von einer Diskdatei oder Speicherbereich. Die Diskdatei oder der Speicherbereich muss vor mit der OPNFIL Funktion geöffnet worden sein. Ein Daten-Token besteht aus einem optionalen Namen und einem optionalen Datenteil, getrennt mit einem Gleichheitszeichen. Jeder Token wird entweder mit einem Komma oder Semikolon beendet.


```

;* SSS | LAX : L31 | MM00      : MM31      | USigned : Integer | IEEE   [LE] : VarNam | Floats : Temps
;* -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
;* 000 | HAX : H31 | BM00 * 0001 : BM31 * 0001 | U01VAL  : I01VAL  | HALF * 0002 : H02VAL | F02VAL : T10VAL |
;* 001 | BAX : B31 | WM00 * 0002 : WM31 * 0002 | U02VAL  : I02VAL  | SNGL * 0004 : S04VAL | F04VAL : T08VAL |
;* 002 | WAX : W31 | EM00 * 0004 : EM31 * 0004 | U04VAL  : I04VAL  | DBLE * 0008 : D08VAL | F08VAL : T12VAL |
;* 003 | EAX : E31 | RM00 * 0008 : RM31 * 0008 | U08VAL  : I08VAL  | QUAD * 0016 : Q16VAL | F16VAL : T20VAL |
;* -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
;* 004 | RAX : R31 | QM00 * 0016 : QM31 * 0016 | U16VAL  : I16VAL  | ???? * 0032 : ?32VAL | F32VAL : T36VAL |
;* 005 | QAX : Q31 | GM00 * 0032 : GM31 * 0032 | U32VAL  : I32VAL  | ???? * 0064 : ?64VAL | F64VAL : T68VAL |
;* 006 | GAX : G31 | PM00 * 0064 : PM31 * 0064 | U64VAL  : I64VAL  | ???? * 0128 : 128VAL | F128VA : T132VA |
;* 007 | PAX : P31 | XM00 * 0128 : XM31 * 0128 | U128VA : I128VA | ???? * 0256 : 256VAL | F256VA : T260VA |
;* -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
;*
;* ****
;*
;* DABSIZE read|write DAta string Byte SIZE ... ,08,...,2560,... [inp] INT4
;* DABCNT read|write DAta string Begin CouNT ... ,08,...,2560,... [inp] INT4
;* DAECNT read|write DAta string End CouNT ... ,08,...,2560,... [inp] INT4
;* DASOFF read|write DAta string Stop OFFset [out] INT4
;* DASCNT read|write DAta string Stop CouNT [out] INT4
;*
;*
;* CONTRL <1 IA USN PrBF> = CTRFUN + 10*CTRBLK + 100*CTRPRC 009 004 0872
;*           <1 IA USN PrBF> + 1000*CTRNUl + 10000*CTRFLN + 100000*CTRSTP
;* INT*04 <1 IA USN PrBF> + 1000000*CTRUNI + 10000000*CTRAUX + 100000000*CTRINF 0x0000:0047h
;*
;* -----
;** FigValue 000,001,002,003,004,005,006,007,008,009 ; BYTE: 000,001,002,003,004,005,006,007,008,009
;at_FigToChr db '0','1','2','3','4','5','6','7','8','9' ; BYTE: '0','1','2','3','4','5','6','7','8','9'
;* -----
;** FigValue 010,011,012,013,014,015,016,017,018,019 ; BYTE: 010,011,012,013,014,015,016,017,018,019
;at_Fig10Chr db 'A','B','C','D','E','F','G','H','I','J' ; BYTE: 'A','B','C','D','E','F','G','H','I','J'
;* -----
;** FigValue 020,021,022,023,024,025,026,027,028,029 ; BYTE: 020,021,022,023,024,025,026,027,028,029
;at_Fig20Chr db 'K','L','M','N','O','P','Q','R','S','T' ; BYTE: 'K','L','M','N','O','P','Q','R','S','T'
;* -----
;** FigValue 030,031,032,033,034,035,036,037,038,039 ; BYTE: 030,031,032,033,034,035,036,037,038,039
;at_Fig30Chr db 'U','V','W','X','Y','Z','%', '%', '%', '%' ; BYTE: 'U','V','W','X','Y','Z','%', '%', '%', '%'
;* -----
;** FigValue 01234567890123456789012345678901234567890 ; BYTE: 01234567890123456789012345678901234567890
;at_FigToChr db '0123456789ABCDEFGHijklmnOPQRstuVwxyz%/%/%%' ; BYTE: 0123456789ABCDEFGHijklmnOPQRstuVwxyz%/%/%%
;* -----
;*
;* WritData CtrUni Coding Single Character Multiple Characters ByteOrderMark
;*
;* <U> 0 -> Latin1 000000h ... 0000FFh
;*      - 1 -> Latin1 000000h ... 0000FFh - $u00100 ... $U0FFFF - Ignored
;*      2 -> UTF-08 000000h ... 00007Fh 0000080h ... 010FFFFh Ignored
;*      3 -> UTF-08 000000h ... 00007Fh 0000080h ... 010FFFFh - Written
;*      4 -> UTF-16-LE 000000h ... 00FFFFh 0010000h ... 010FFFFh Ignored
;*      5 -> UTF-16-LE 000000h ... 00FFFFh 0010000h ... 010FFFFh - Written
;*      6 -> UTF-32-LE 000000h ... 10FFFFh Unused Ignored
;*      7 -> UTF-32-LE 000000h ... 10FFFFh Unused - Written
;*      8 -> UTF-16-BE 000000h ... 00FFFFh 0010000h ... 10FFFFh Written
;*      9 -> UTF-32-BE 000000h ... 10FFFFh Unused Written
;*
;* WritData CtrStp WiStopChar W1FillStop W1FillRest [AA@BB &z ] Style
;*
;* <S> 0 -> 00,'&z' Fill 000 Fill 000 [AA@BB |o|oooooooo|o] C / C++
;*      1 -> 00,'&z' Fill ',' Fill ',' [AA@BB | | |o] Fortran
;*      2 -> R1StpChar ! Fill '&z' Fill 000 [AA@BB |&z|ooooooo|o] C / C++
;*      3 -> 00,'&z' Fill '&z' Fill ',' [AA@BB |&z| |o] Fortran
;*      4 -> 00,'&z' - Fill '&z' - Fill '..' - [AA@BB |&z|.....|o] - Testing
;*      5 -> 00,'&z','@' Fill 000 Fill 000 [AA|o|oooooooooooo|o] C / C++
;*      6 -> 00,'&z','@' Fill ',' Fill ',' [AA| | |o] Fortran
;*      7 -> 00,'&z','@' Fill '@' Fill 000 [AA|@|oooooooooooo|o] C / C++
;*      8 -> 00,'&z','@' Fill '@' Fill ',' [AA|@| |o] Fortran
;*      9 -> 00,'&z','@' Fill '@' Fill '..' [AA|@|.....|o] Testing
;*
;* WritData FNwLin FreeByteSize W1FillNull W1NewLine [L1 &n L2 &z ] Style
;*
;* <N> 0 -> FullByteSize 00 00 [L1 L2 &z.....] Removed
;*      1 -> FullByteSize 00 01 [L1 13,10 L2 &z.....] Windows
;*      2 -> FullByteSize 00 02 [L1 10 L2 &z.....] Unix
;*      3 -> FullByteSize 00 03 [L1 13 L2 &z.....] Macintosh
;*      4 -> FullByteSize 00 05 [L1 &n L2 &z.....] - Strings
;*      5 -> FBS-NullSize 01 00 [L1 L2 &z.....|o] Removed
;*      6 -> FBS-NullSize 01 01 [L1 13,10 L2 &z...|o] Windows
;*      7 -> FBS-NullSize 01 02 [L1 10 L2 &z.....|o] Unix
;*      8 -> FBS-NullSize 01 03 [L1 13 L2 &z.....|o] Macintosh
;*      9 -> FBS-NullSize 01 05 [L1 &n L2 &z.....|o] Strings
;*
;* =====
;*
;* Return values:
;* =====

```

```

;*
;* ERROR1 not yet defined
;* MyMlWrit My own MatLib Write function
;*
;* =====#
;* | +12345678901234567_10E+10 : 00 | 1234567890123456 | 1234567890123456 | 1234567890123456 | 1234567890123456 | *
;* +-----+-----+-----+-----+-----+
;* | Exponent Scientific : A2 | ES(+16|+A2|B3) | ES(+16|-A2|B3) | ES(-16|+A2|B3) | ES(-16|-A2|B3) | *
;* +-----+-----+-----+-----+-----+
;* | +1.234567890123456_10E+00 : 00 | 1.E+00 | 1.E+00 | 1.E+00 | 1.E+00 | *
;* | -1.234500000000000_10E+00 : 01 | -1.2E+00 | -1.2E+00 | -1.2E+00 | -1.2E+00 | *
;* | +1.234500000000000_10E+00 : 02 | 1.23E+00 | +1.23E+00 | 1.23E+00 | +1.23E+00 | *
;* | -1.234500000000000_10E+00 : 03 | -1.234E+00 | -1.234E+00 | -1.234E+00 | -1.234E+00 | *
;* | +1.234500000000000_10E+00 : 04 | 1.234 5E+00 | +1.234 5E+00 | 1.234 5E+00 | +1.234 5E+00 | *
;* | -1.234500000000000_10E+00 : 05 | -1.234 50E+00 | -1.234 50E+00 | -1.234 50E+00 | -1.234 50E+00 | *
;* | +1.234500000000000_10E+00 : 06 | 1.234 500E+00 | +1.234 500E+00 | 1.234 500E+00 | +1.234 500E+00 | *
;* | -1.234500000000000_10E+00 : 07 - 16 | -1.234 500 OE+00 | -1.234 500 OE+00 | -1.234 500 OE+00 | -1.234 500 OE+00 | *
;* | : | | | | |
;* | +1.234500000000000_10E+00 : 17 . . | 1.234 5E+00 | +1.234 5E+00 | 1.234 5E+00 | +1.234 5E+00 | *
;* +-----+-----+-----+-----+-----+
;* | +1.234567890123456_12E+00 : 00 | 1._12E+00 | 1._12E+00 | 1._12E+00 | 1._12E+00 | *
;* | -1.234500000000000_12E+00 : 01 | -1.2_12E+00 | -1.2_12E+00 | -1.2_12E+00 | -1.2_12E+00 | *
;* | +1.234500000000000_12E+00 : 02 | 1.23_12E+00 | +1.23_12E+00 | 1.23_12E+00 | +1.23_12E+00 | *
;* | -1.234500000000000_12E+00 : 03 | -1.234_12E+00 | -1.234_12E+00 | -1.234_12E+00 | -1.234_12E+00 | *
;* | +1.234500000000000_12E+00 : 04 | 1.234 5_12E+00 | +1.234 5_12E+00 | 1.234 5_12E+00 | +1.234 5_12E+00 | *
;* | -1.234500000000000_12E+00 : 05 - 16 | -1.234 50_12E+00 | -1.234 50_12E+00 | -1.234 50_12E+00 | -1.234 50_12E+00 | *
;* | : | | | | |
;* | +1.234500000000000_12E+00 : 17 . . | 1.234 5_12E+00 | +1.234 5_12E+00 | 1.234 5_12E+00 | +1.234 5_12E+00 | *
;* +-----+-----+-----+-----+-----+
;* | Exponent Engineering : A2 | EE(+16|+A2|B3) | EE(+16|-A2|B3) | EE(-16|+A2|B3) | EE(-16|-A2|B3) | *
;* +-----+-----+-----+-----+-----+
;* | +1.234500000000000_10E+01 : 00 | 12.E+00 | 12.E+00 | 12.E+00 | 12.E+00 | *
;* | -1.234500000000000_10E+01 : 01 | -12.3E+00 | -12.3E+00 | -12.3E+00 | -12.3E+00 | *
;* | +1.234500000000000_10E+01 : 02 | 12.34E+00 | +12.34E+00 | 12.34E+00 | +12.34E+00 | *
;* | -1.234500000000000_10E+01 : 03 | -12.345E+00 | -12.345E+00 | -12.345E+00 | -12.345E+00 | *
;* | +1.234500000000000_10E+01 : 04 | 12.345 0E+00 | 12.345 0E+00 | 12.345 0E+00 | +12.345 0E+00 | *
;* | -1.234500000000000_10E+01 : 05 | -12.345 00E+00 | -12.345 00E+00 | -12.345 00E+00 | -12.345 00E+00 | *
;* | +1.234500000000000_10E+01 : 06 - 16 | 12.345 000E+00 | 12.345 000E+00 | 12.345 000E+00 | +12.345 000E+00 | *
;* | : | | | | |
;* | +1.234500000000000_10E+01 : 17 . . | 12.345E+00 | +12.345E+00 | 12.345E+00 | +12.345E+00 | *
;* +-----+-----+-----+-----+-----+
;* | +1.234500000000000_12E+01 : 00 | 12._12E+00 | 12._12E+00 | 12._12E+00 | 12._12E+00 | *
;* | -1.234500000000000_12E+01 : 01 | -12.3_12E+00 | -12.3_12E+00 | -12.3_12E+00 | -12.3_12E+00 | *
;* | +1.234500000000000_12E+01 : 02 | 12.34_12E+00 | +12.34_12E+00 | 12.34_12E+00 | +12.34_12E+00 | *
;* | -1.234500000000000_12E+01 : 03 | -12.345_12E+00 | -12.345_12E+00 | -12.345_12E+00 | -12.345_12E+00 | *
;* | +1.234500000000000_12E+01 : 04 - 16 | 12.345 0_12E+00 | 12.345 0_12E+00 | 12.345 0_12E+00 | +12.345 0_12E+00 | *
;* | : | | | | |
;* | +1.234500000000000_12E+01 : 17 . . | 12.345_12E+00 | +12.345_12E+00 | 12.345_12E+00 | +12.345_12E+00 | *
;* +-----+-----+-----+-----+-----+
;* | Exponent Engineering : A2 | EE(+16|+A2|B3) | EE(+16|-A2|B3) | EE(-16|+A2|B3) | EE(-16|-A2|B3) | *
;* +-----+-----+-----+-----+-----+
;* | +1.234500000000000_10E+02 : 00 | 123.E+00 | 123.E+00 | 123.E+00 | 123.E+00 | *
;* | -1.234500000000000_10E+02 : 01 | -123.4E+00 | -123.4E+00 | -123.4E+00 | -123.4E+00 | *
;* | +1.234500000000000_10E+02 : 02 | 123.45E+00 | +123.45E+00 | 123.45E+00 | +123.45E+00 | *
;* | -1.234500000000000_10E+02 : 03 | -123.450E+00 | -123.450E+00 | -123.450E+00 | -123.450E+00 | *
;* | +1.234500000000000_10E+02 : 04 | 123.450 0E+00 | 123.450 0E+00 | 123.450 0E+00 | +123.450 0E+00 | *
;* | -1.234500000000000_10E+02 : 05 | -123.450 00E+00 | -123.450 00E+00 | -123.450 00E+00 | -123.450 00E+00 | *
;* | +1.234500000000000_10E+02 : 06 - 15 | 123.450 000E+00 | 123.450 000E+00 | 123.450 000E+00 | +123.450 000E+00 | *
;* | : | | | | |
;* | +1.234500000000000_10E+02 : 17 . . | 123.45E+00 | +123.45E+00 | 123.45E+00 | +123.45E+00 | *
;* +-----+-----+-----+-----+-----+
;* | +1.234500000000000_12E+02 : 00 | 123._12E+00 | 123._12E+00 | 123._12E+00 | 123._12E+00 | *
;* | -1.234500000000000_12E+02 : 01 | -123.4_12E+00 | -123.4_12E+00 | -123.4_12E+00 | -123.4_12E+00 | *
;* | +1.234500000000000_12E+02 : 02 | 123.45_12E+00 | +123.45_12E+00 | 123.45_12E+00 | +123.45_12E+00 | *
;* | -1.234500000000000_12E+02 : 03 - 16 | -123.450_12E+00 | -123.450_12E+00 | -123.450_12E+00 | -123.450_12E+00 | *
;* | : | | | | |
;* | +1.234500000000000_12E+02 : 17 . . | 123.45E+00 | +123.45E+00 | 123.45E+00 | +123.45E+00 | *
;* +-----+-----+-----+-----+-----+
;* | General Scientific/Engin : A2 | GS(+16|+A2|B3) | GS(+16|-A2|B3) | GS(-16|+A2|B3) | GS(-16|-A2|B3) | *
;* +-----+-----+-----+-----+-----+
;* | +1.234500000000000_10E+00 : 00 | 1.| | 1.| | 1.| | *
;* | -1.234500000000000_10E+00 : 01 | -1.2| | -1.2| | -1.2| | *
;* | +1.234500000000000_10E+00 : 02 | 1.23| | +1.23| | 1.23| | *
;* | -1.234500000000000_10E+00 : 03 | -1.234| | -1.234| | -1.234| | *
;* | +1.234500000000000_10E+00 : 04 | 1.234 5| | +1.234 5| | 1.234 5| | *
;* | -1.234500000000000_10E+00 : 05 | -1.234 50| | -1.234 50| | -1.234 50| | *
;* | +1.234500000000000_10E+00 : 06 | 1.234 500| | +1.234 500| | 1.234 500| | *
;* | -1.234500000000000_10E+00 : 07 | -1.234 500 0| | -1.234 500 0| | -1.234 500 0| | *
;* | +1.234500000000000_10E+00 : 08 | 1.234 500 00| | -1.234 500 00| | -1.234 500 00| | *
;* | -1.234500000000000_10E+00 : 09 | -1.234 500 000| | -1.234 500 000| | -1.234 500 000| | *
;* | +1.234500000000000_10E+00 : 10 - 16 | 1.234 500 000 0| | -1.234 500 000 0| | -1.234 500 000 0| | *
;* | : | | | | |
;* | +1.234500000000000_10E+00 : 17 . . | 1.234 5| | +1.234 5| | 1.234 5| | *
;* +-----+-----+-----+-----+-----+
;* | +1.234500000000000_12E+00 : 00 | 1._12| | 1._12| | 1._12| | *

```

```

;* | -1.2345000000000000_12E+00 : 01 | -1.2_12 | -1.2_12 | -1.2_12 | -1.2_12 |
;* | +1.2345000000000000_12E+00 : 02 | 1.23_12 | +1.23_12 | 1.23_12 | +1.23_12 |
;* | -1.2345000000000000_12E+00 : 03 | -1.234_12 | -1.234_12 | -1.234_12 | -1.234_12 |
;* | +1.2345000000000000_12E+00 : 04 | 1.234_5_12 | +1.234_5_12 | 1.234_5_12 | +1.234_5_12 |
;* | -1.2345000000000000_12E+00 : 05 | -1.234_50_12 | -1.234_50_12 | -1.234_50_12 | -1.234_50_12 |
;* | +1.2345000000000000_12E+00 : 06 | 1.234_500_12 | +1.234_500_12 | 1.234_500_12 | +1.234_500_12 |
;* | -1.2345000000000000_12E+00 : 07 | -1.234_500_0_12 | -1.234_500_0_12 | -1.234_500_0_12 | -1.234_500_0_12 |
;* | -1.2345000000000000_12E+00 : 08 - 15 | 1.234_500_00_12 | -1.234_500_00_12 | -1.234_500_00_12 | -1.234_500_00_12 |
;* | : | : | : | : |
;* | +1.2345000000000000_10E+00 : 17 . . | 1.234_5_12 | +1.234_5_12 | 1.234_5_12 | +1.234_5_12 |
;* +-----+-----+-----+-----+-----+
;* | General Scientific/Engin : A2 | GS(+16|+A2|B3) | GS(+16|-A2|B3) | GS(-16|+A2|B3) | GS(-16|-A2|B3) |
;* +-----+-----+-----+-----+-----+
;* | Integer +12345_10 : 00 | 12 345 | 12 345 | 12 345 | 12 345 |
;* | -12345_10 : 01 | -12 345 | -12 345 | -12 345 | -12 345 |
;* | +12345_10 : 02 | 12 345 | +12 345 | 12 345 | +12 345 |
;* | -12345_10 : 03 | -12 345 | -12 345 | -12 345 | -12 345 |
;* | +12345_10 : 04 | 12 345 | +12 345 | 12 345 | +12 345 |
;* | -12345_10 : 05 | -12 345 | -12 345 | -12 345 | -12 345 |
;* | +12345_10 : 06 | 012 345 | +012 345 | 012 345 | +012 345 |
;* | -12345_10 : 07 | -0 012 345 | -0 012 345 | -0 012 345 | -0 012 345 |
;* | +12345_10 : 08 | 00 012 345 | +00 012 345 | 00 012 345 | +00 012 345 |
;* | -12345_10 : 09 | -000 012 345 | -000 012 345 | -000 012 345 | -000 012 345 |
;* | +12345_10 : 10 | 0 000 012 345 | +0 000 012 345 | 0 000 012 345 | +0 000 012 345 |
;* | -12345_10 : 11 | -00 000 012 345 | -00 000 012 345 | -00 000 012 345 | -00 000 012 345 |
;* | +12345_10 : 12 - 16 | 000 000 012 345 | +000 000 012 345 | 000 000 012 345 | +000 000 012 345 |
;* | : | : | : | : |
;* | +12345_10 : 17 . . | 12 345 | +12 345 | 12 345 | +12 345 |
;* +-----+-----+-----+-----+-----+
;* | Integer +12345_12 : 00 | 12 345_12 | 12 345_12 | 12 345_12 | 12 345_12 |
;* | -12345_12 : 01 | -12 345_12 | -12 345_12 | -12 345_12 | -12 345_12 |
;* | +12345_12 : 02 | 12 345_12 | +12 345_12 | 12 345_12 | +12 345_12 |
;* | -12345_12 : 03 | -12 345_12 | -12 345_12 | -12 345_12 | -12 345_12 |
;* | +12345_12 : 04 | 12 345_12 | +12 345_12 | 12 345_12 | +12 345_12 |
;* | -12345_12 : 05 | -12 345_12 | -12 345_12 | -12 345_12 | -12 345_12 |
;* | +12345_12 : 06 | 012 345_12 | +012 345_12 | 012 345_12 | +012 345_12 |
;* | -12345_12 : 07 | -0 012 345_12 | -0 012 345_12 | -0 012 345_12 | -0 012 345_12 |
;* | +12345_12 : 08 | 00 012 345_12 | +00 012 345_12 | 00 012 345_12 | +00 012 345_12 |
;* | -12345_12 : 09 | -000 012 345_12 | -000 012 345_12 | -000 012 345_12 | -000 012 345_12 |
;* | : | : | : | : |
;* | +12345_10 : 17 . . | 12 345_12 | +12 345_12 | 12 345_12 | +12 345_12 |
;* #=====#
;* | General Decimal (5) : A2 | Dec(+16|+A2|3) | Dec(+16|-A2|3) | Dec(-16|+A2|3) | Dec(-16|-A2|3) |
;* +-----+-----+-----+-----+-----+
;* | +1.2345000000000000_10E+00 : 00 | 1. | 1. | 1. | 1. |
;* | -1.2345000000000000_10E+00 : 01 | -1.2 | -1.2 | -1.2 | -1.2 |
;* | +1.2345000000000000_10E+00 : 02 | 1.23 | +1.23 | 1.23 | +1.23 |
;* | -1.2345000000000000_10E+00 : 03 | -1.234 | -1.234 | -1.234 | -1.234 |
;* | +1.2345000000000000_10E+00 : 04 | 1.234_5 | +1.234_5 | 1.234_5 | +1.234_5 |
;* | -1.2345000000000000_10E+00 : 05 | -1.234_50 | -1.234_50 | -1.234_50 | -1.234_50 |
;* | +1.2345000000000000_10E+00 : 06 | 1.234_500 | +1.234_500 | 1.234_500 | +1.234_500 |
;* | -1.2345000000000000_10E+00 : 07 | -1.234_500_0 | -1.234_500_0 | -1.234_500_0 | -1.234_500_0 |
;* | +1.2345000000000000_10E+00 : 08 | 1.234_500_00 | +1.234_500_00 | 1.234_500_00 | +1.234_500_00 |
;* | -1.2345000000000000_10E+00 : 09 | -1.234_500_000 | -1.234_500_000 | -1.234_500_000 | -1.234_500_000 |
;* | +1.2345000000000000_10E+00 : 10 - 16 | 1.234_500_000_0 | +1.234_500_000_0 | 1.234_500_000_0 | +1.234_500_000_0 |
;* | : | : | : | : |
;* | +1.2345000000000000_10E+00 : 17 . . | 1.234_5 | +1.234_5 | 1.234_5 | +1.234_5 |
;* +-----+-----+-----+-----+-----+
;* | Integer +12345_10 : 00 | 12 345 | 12 345 | 12 345 | 12 345 |
;* | -12345_10 : 01 | -12 345 | -12 345 | -12 345 | -12 345 |
;* | +12345_10 : 02 | 12 345 | +12 345 | 12 345 | +12 345 |
;* | -12345_10 : 03 | -12 345 | -12 345 | -12 345 | -12 345 |
;* | +12345_10 : 04 | 12 345 | +12 345 | 12 345 | +12 345 |
;* | -12345_10 : 05 | -12 345 | -12 345 | -12 345 | -12 345 |
;* | +12345_10 : 06 | 012 345 | +012 345 | 012 345 | +012 345 |
;* | -12345_10 : 07 | -0 012 345 | -0 012 345 | -0 012 345 | -0 012 345 |
;* | +12345_10 : 08 | 00 012 345 | +00 012 345 | 00 012 345 | +00 012 345 |
;* | -12345_10 : 09 | -000 012 345 | -000 012 345 | -000 012 345 | -000 012 345 |
;* | +12345_10 : 10 | 0 000 012 345 | +0 000 012 345 | 0 000 012 345 | +0 000 012 345 |
;* | -12345_10 : 11 | -00 000 012 345 | -00 000 012 345 | -00 000 012 345 | -00 000 012 345 |
;* | +12345_10 : 12 - 16 | 000 000 012 345 | +000 000 012 345 | 000 000 012 345 | +000 000 012 345 |
;* | : | : | : | : |
;* | +12345_10 : 17 . . | 12 345 | +12 345 | 12 345 | +12 345 |
;* +-----+-----+-----+-----+-----+
;* | General Hexadecimal (6) : A2 | Hex(+16|+A2|3) | Hex(+16|-A2|3) | Hex(-16|+A2|3) | Hex(-16|-A2|3) |
;* +-----+-----+-----+-----+-----+
;* | +1.2345000000000000_16E+00 : 00 | 1.h | 1.h | 1.h | 1.h |
;* | -1.2345000000000000_16E+00 : 01 | -1.2h | -1.2h | -1.2h | -1.2h |
;* | +1.2345000000000000_16E+00 : 02 | 1.23h | +1.23h | 1.23h | +1.23h |
;* | -1.2345000000000000_16E+00 : 03 | -1.234h | -1.234h | -1.234h | -1.234h |
;* | +1.2345000000000000_16E+00 : 04 | 1.234_5h | +1.234_5h | 1.234_5h | +1.234_5h |
;* | -1.2345000000000000_16E+00 : 05 | -1.234_50h | -1.234_50h | -1.234_50h | -1.234_50h |
;* | +1.2345000000000000_16E+00 : 06 | 1.234_500h | +1.234_500h | 1.234_500h | +1.234_500h |
;* | -1.2345000000000000_16E+00 : 07 | -1.234_500_0h | -1.234_500_0h | -1.234_500_0h | -1.234_500_0h |
;* | +1.2345000000000000_16E+00 : 08 | 1.234_500_00h | +1.234_500_00h | 1.234_500_00h | +1.234_500_00h |
;* | -1.2345000000000000_16E+00 : 09 - 16 | -1.234_500_000h | -1.234_500_000h | -1.234_500_000h | -1.234_500_000h |

```

```

;* | +1.2345000000000000_16E+00 : 17 . . | 1.234 5h | +1.234 5h | 1.234 5h | +1.234 5h | * *
;* | Integer +12345_16 : 00 | 12 345h | 12 345h | 12 345h | 12 345h | * *
;* | -12345_16 : 01 | -12 345h | -12 345h | -12 345h | -12 345h | * *
;* | +12345_16 : 02 | 12 345h | +12 345h | 12 345h | +12 345h | * *
;* | -12345_16 : 03 | -12 345h | -12 345h | -12 345h | -12 345h | * *
;* | +12345_16 : 04 | 12 345h | +12 345h | 12 345h | +12 345h | * *
;* | -12345_16 : 05 | -12 345h | -12 345h | -12 345h | -12 345h | * *
;* | +12345_16 : 06 | 012 345h | +012 345h | 012 345h | +012 345h | * *
;* | -12345_16 : 07 | -0 012 345h | -0 012 345h | -0 012 345h | -0 012 345h | * *
;* | +12345_16 : 08 | 00 012 345h | +00 012 345h | 00 012 345h | +00 012 345h | * *
;* | -12345_16 : 09 | -000 012 345h | -000 012 345h | -000 012 345h | -000 012 345h | * *
;* | +12345_16 : 10 | 0 000 012 345h | +0 000 012 345h | 0 000 012 345h | +0 000 012 345h | * *
;* | -12345_16 : 11 - 16 | -00 000 012 345h | * *
;* | +12345_16 : 12 - 16 | 000 000 012 345h | | | | * *
;* | : | | | | | * *
;* | +12345_16 : 17 . . | 12 345h | +12 345h | 12 345h | +12 345h | * *
;* | General Binary (7) : A2 | Bin(+16|+A2|3) | Bin(+16|-A2|3) | Bin(-16|+A2|3) | Bin(-16|-A2|3) | * *
;* | +1.1011000000000000_02E+00 : 00 | 1.b | 1.b | 1.b | 1.b | * *
;* | -1.1011000000000000_02E+00 : 01 | -1.1b | -1.1b | -1.1b | -1.1b | * *
;* | +1.1011000000000000_02E+00 : 02 | 1.10b | +1.10b | 1.10b | +1.10b | * *
;* | -1.1011000000000000_02E+00 : 03 | -1.101b | -1.101b | -1.101b | -1.101b | * *
;* | +1.1011000000000000_02E+00 : 04 | 1.101 1b | +1.101 1b | 1.101 1b | +1.101 1b | * *
;* | -1.1011000000000000_02E+00 : 05 | -1.101 10b | -1.101 10b | -1.101 10b | -1.101 10b | * *
;* | +1.1011000000000000_02E+00 : 06 | 1.101 100b | +1.101 100b | 1.101 100b | +1.101 100b | * *
;* | -1.1011000000000000_02E+00 : 07 | -1.101 100 0b | -1.101 100 0b | -1.101 100 0b | -1.101 100 0b | * *
;* | +1.1011000000000000_02E+00 : 08 | 1.101 100 00b | +1.101 100 00b | 1.101 100 00b | +1.101 100 00b | * *
;* | -1.1011000000000000_02E+00 : 09 - 16 | -1.101 100 000b | -1.101 100 000b | -1.101 100 000b | -1.101 100 000b | * *
;* | : | | | | * *
;* | +1.1011000000000000_02E+00 : 17 . . | 1.101 1b | +1.234 5b | 1.101 1b | +1.101 1b | * *
;* | Integer +11011_02 : 00 | 11 011b | 11 011b | 11 011b | 11 011b | * *
;* | -11011_02 : 01 | -11 011b | -11 011b | -11 011b | -11 011b | * *
;* | +11011_02 : 02 | 11 011b | +11 011b | 11 011b | +11 011b | * *
;* | -11011_02 : 03 | -11 011b | -11 011b | -11 011b | -11 011b | * *
;* | +11011_02 : 04 | 11 011b | +11 011b | 11 011b | +11 011b | * *
;* | -11011_02 : 05 | -11 011b | -11 011b | -11 011b | -11 011b | * *
;* | +11011_02 : 06 | 011 011b | +011 011b | 011 011b | +011 011b | * *
;* | -11011_02 : 07 | -0 011 011b | -0 011 011b | -0 011 011b | -0 011 011b | * *
;* | +11011_02 : 08 | 00 011 011b | +00 011 011b | 00 011 011b | +00 011 011b | * *
;* | -11011_02 : 09 | -000 011 011b | -000 011 011b | -000 011 011b | -000 011 011b | * *
;* | +11011_02 : 10 | 0 000 011 011b | +0 000 011 011b | 0 000 011 011b | +0 000 011 011b | * *
;* | -11011_02 : 11 - 16 | -00 000 011 011b | * *
;* | +11011_02 : 12 - 16 | 000 000 011 011b | | | | * *
;* | : | | | | * *
;* | +11011_02 : 17 . . | 11 011b | +11 011b | 11 011b | +11 011b | * *
;* | General Octal (8) : A2 | Oct(+16|+A2|3) | Oct(+16|-A2|3) | Oct(-16|+A2|3) | Oct(-16|-A2|3) | * *
;* | +1.2345000000000000_08E+00 : 00 | 1.o | 1.o | 1.o | 1.o | * *
;* | -1.2345000000000000_08E+00 : 01 | -1.2o | -1.2o | -1.2o | -1.2o | * *
;* | +1.2345000000000000_08E+00 : 02 | 1.23o | +1.23o | 1.23o | +1.23o | * *
;* | -1.2345000000000000_08E+00 : 03 | -1.234o | -1.234o | -1.234o | -1.234o | * *
;* | +1.2345000000000000_08E+00 : 04 | 1.234 5o | +1.234 5o | 1.234 5o | +1.234 5o | * *
;* | -1.2345000000000000_08E+00 : 05 | -1.234 50o | -1.234 50o | -1.234 50o | -1.234 50o | * *
;* | +1.2345000000000000_08E+00 : 06 | 1.234 500o | +1.234 500o | 1.234 500o | +1.234 500o | * *
;* | -1.2345000000000000_08E+00 : 07 | -1.234 500 0o | -1.234 500 0o | -1.234 500 0o | -1.234 500 0o | * *
;* | -1.2345000000000000_08E+00 : 08 | 1.234 500 00o | -1.234 500 00o | 1.234 500 00o | -1.234 500 00o | * *
;* | -1.2345000000000000_08E+00 : 09 - 16 | -1.234 500 000o | -1.234 500 000o | -1.234 500 000o | -1.234 500 000o | * *
;* | : | | | | * *
;* | +1.2345000000000000_08E+00 : 17 . . | 1.234 5h | +1.234 5o | 1.234 5o | +1.234 5o | * *
;* | Integer +12345_08 : 00 | 12 345o | 12 345o | 12 345o | 12 345o | * *
;* | -12345_08 : 01 | -12 345o | -12 345o | -12 345o | -12 345o | * *
;* | +12345_08 : 02 | 12 345o | +12 345o | 12 345o | +12 345o | * *
;* | -12345_08 : 03 | -12 345o | -12 345o | -12 345o | -12 345o | * *
;* | +12345_08 : 04 | 12 345o | +12 345o | 12 345o | +12 345o | * *
;* | -12345_08 : 05 | -12 345o | -12 345o | -12 345o | -12 345o | * *
;* | +12345_08 : 06 | 012 345o | +012 345o | 012 345o | +012 345o | * *
;* | -12345_08 : 07 | -0 012 345o | -0 012 345o | -0 012 345o | -0 012 345o | * *
;* | +12345_08 : 08 | 00 012 345o | +00 012 345o | 00 012 345o | +00 012 345o | * *
;* | -12345_08 : 09 | -000 012 345o | -000 012 345o | -000 012 345o | -000 012 345o | * *
;* | +12345_08 : 10 | 0 000 012 345o | +0 000 012 345o | 0 000 012 345o | +0 000 012 345o | * *
;* | -12345_08 : 11 - 16 | -00 000 012 345o | * *
;* | +12345_08 : 12 - 16 | 000 000 012 345o | | | | * *
;* | : | | | | * *
;* | +12345_08 : 17 . . | 12 345o | +12 345o | 12 345h | +12 345o | * *
;* | General BaseWished (9) : A2 | Bas(+16|+A2|3) | Bas(+16|-A2|3) | Bas(-16|+A2|3) | Bas(-16|-A2|3) | * *
;* | +1.2345000000000000_12E+00 : 00 | 1._12 | 1._12 | 1._12 | 1._12 | * *
;* | -1.2345000000000000_12E+00 : 01 | -1.2_12 | -1.2_12 | -1.2_12 | -1.2_12 | * *
;* | +1.2345000000000000_12E+00 : 02 | 1.23_12 | +1.23_12 | 1.23_12 | +1.23_12 | * *

```

```

;* | -1.2345000000000000_12E+00 : 03 | -1.234_12| -1.234_12 | -1.234_12 | *  

;* | +1.2345000000000000_12E+00 : 04 | 1.234_5_12| +1.234_5_12 | 1.234_5_12 | *  

;* | -1.2345000000000000_12E+00 : 05 | -1.234_50_12| -1.234_50_12 | -1.234_50_12 | *  

;* | +1.2345000000000000_12E+00 : 06 | 1.234_500_12| +1.234_500_12 | 1.234_500_12 | *  

;* | -1.2345000000000000_12E+00 : 07 | -1.234_500_0_12| -1.234_500_0_12 | -1.234_500_0_12 | *  

;* | -1.2345000000000000_12E+00 : 08 - 15 | 1.234_500_00_12| -1.234_500_00_12 | -1.234_500_00_12 | -1.234_500_00_12 | *  

;* | : | | | | *  

;* | +1.2345000000000000_12E+00 : 16 . . | 1.234_5_12| +1.234_5_12 | 1.234_5_12 | *  

;* | - - - - + - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - *  

;* | Integer +12345_12 : 00 | 12_345_12| 12_345_12 | 12_345_12 | *  

;* | -12345_12 : 01 | -12_345_12| -12_345_12 | -12_345_12 | *  

;* | +12345_12 : 02 | 12_345_12| +12_345_12 | 12_345_12 | *  

;* | -12345_12 : 03 | -12_345_12| -12_345_12 | -12_345_12 | *  

;* | +12345_12 : 04 | 12_345_12| +12_345_12 | 12_345_12 | *  

;* | -12345_12 : 05 | -12_345_12| -12_345_12 | -12_345_12 | *  

;* | +12345_12 : 06 | 012_345_12| +012_345_12 | 012_345_12 | *  

;* | -12345_12 : 07 | -0_012_345_12| -0_012_345_12 | -0_012_345_12 | *  

;* | +12345_12 : 08 | 00_012_345_12| +00_012_345_12 | 00_012_345_12 | *  

;* | -12345_12 : 09 - 15 | -000_012_345_12| -000_012_345_12 | -000_012_345_12 | -000_012_345_12 | *  

;* | : | | | | *  

;* | -12345_12 : 16 . . | -12_345_12| -12_345_12 | -12_345_12 | *  

;* #=====#=====#=====#=====#=====#=====#=====#=====#=====#  

;* | +1.23456789012345678_10E+10 | 123456789012345678 | 123456789012345678 | 123456789012345678 | *  

;* | - - - - + - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - *  

;* | Exponent Scientific | ES(+18|+06|B03) | ES(+18|-06|B03) | ES(-18|+06|B03) | ES(-18|-06|B03) | *  

;* | + - - - - + - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - *  

;* | +1.23456789012345678_10E-02 | 1.234_568E-02| +1.234_568E-02 | 1.234_568E-02 | +1.234_568E-02 | *  

;* | -1.23456789012345678_10E-01 | -1.234_568E-01| -1.234_568E-01 | -1.234_568E-01 | -1.234_568E-01 | *  

;* | +1.23456789012345678_10E+00 | 1.234_568E+00| +1.234_568E+00 | 1.234_568E+00 | +1.234_568E+00 | *  

;* | -1.23456789012345678_10E+01 | -1.234_568E+01| -1.234_568E+01 | -1.234_568E+01 | -1.234_568E+01 | *  

;* | +1.23456789012345678_10E+02 | 1.234_568E+02| +1.234_568E+02 | 1.234_568E+02 | +1.234_568E+02 | *  

;* | - - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - *  

;* | Exponent Engineering | ES(+18|+06|B03) | ES(+18|-06|B03) | ES(-18|+06|B03) | ES(-18|-06|B03) | *  

;* | + - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - - - - + - *  

;* | +1.23456789012345678_10E-06 | 1.234_568E-06| +1.234_568E-06 | 1.234_568E-06 | +1.234_568E-06 | *  

;* | -1.23456789012345678_10E-05 | -12.345_679E-06| -12.345_679E-06 | -12.345_679E-06 | -12.345_679E-06 | *  

;* | +1.23456789012345678_10E-04 | 123.456_789E-06| +123.456_789E-06 | 123.456_789E-06 | +123.456_789E-06 | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | +1.23456789012345678_10E-03 | 1.234_568E-03| +1.234_568E-03 | 1.234_568E-03 | +1.234_568E-03 | *  

;* | -1.23456789012345678_10E-02 | -12.345_679E-03| -12.345_679E-03 | -12.345_679E-03 | -12.345_679E-03 | *  

;* | +1.23456789012345678_10E-01 | 123.456_789E-03| +123.456_789E-03 | 123.456_789E-03 | +123.456_789E-03 | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | +1.23456789012345678_10E+00 | 1.234_568E+00| +1.234_568E+00 | 1.234_568E+00 | +1.234_568E+00 | *  

;* | -1.23456789012345678_10E+01 | -12.345_679E+00| -12.345_679E+00 | -12.345_679E+00 | -12.345_679E+00 | *  

;* | +1.23456789012345678_10E+02 | 123.456_789E+00| +123.456_789E+00 | 123.456_789E+00 | +123.456_789E+00 | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | +1.23456789012345678_10E+03 | 1.234_568E+03| +1.234_568E+03 | 1.234_568E+03 | +1.234_568E+03 | *  

;* | -1.23456789012345678_10E+04 | -12.345_679E+03| -12.345_679E+03 | -12.345_679E+03 | -12.345_679E+03 | *  

;* | +1.23456789012345678_10E+05 | 123.456_789E+03| +123.456_789E+03 | 123.456_789E+03 | +123.456_789E+03 | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | Exponent Engineering | ES(+18|+17|B03) | ES(+18|-17|B03) | ES(-18|+17|B03) | ES(-18|-17|B03) | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | +1.23456789012345678_10E-03 | 1.234_567_9E-03| +1.234_567_9E-03 | 1.234_567_9E-03 | +1.234_567_9E-03 | *  

;* | -1.23456789012345678_10E-02 | -12.345_678_9E-03| -12.345_678_9E-03 | -12.345_678_9E-03 | -12.345_678_9E-03 | *  

;* | +1.23456789012345678_10E-01 | 123.456_789_0E-03| +123.456_789_0E-03 | 123.456_789_0E-03 | +123.456_789_0E-03 | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | +1.23456789012345678_10E+00 | 1.234_567_9E+00| +1.234_567_9E+00 | 1.234_567_9E+00 | +1.234_567_9E+00 | *  

;* | -1.23456789012345678_10E+01 | -12.345_678_9E+00| -12.345_678_9E+00 | -12.345_678_9E+00 | -12.345_678_9E+00 | *  

;* | +1.23456789012345678_10E+02 | 123.456_789_0E+00| +123.456_789_0E+00 | 123.456_789_0E+00 | +123.456_789_0E+00 | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | Hexadecimal | Hex(+18|+04|B02) | Hex(+18|-04|B02) | Hex(-18|+04|B02) | Hex(-18|-04|B02) | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | +1.23456789012345678_16E-02 | 0.00_00h| -0.00_00h | 0.00_00h | -0.00_00h | *  

;* | +1.23456789012345678_16E-02 | 0.00_01h| +0.00_01h | 0.00_01h | +0.00_01h | *  

;* | +1.23456789012345678_16E-02 | 0.00_12h| -0.00_12h | 0.00_12h | -0.00_12h | *  

;* | +1.23456789012345678_16E-02 | 0.01_23h| +0.01_23h | 0.01_23h | +0.01_23h | *  

;* | +1.23456789012345678_16E-02 | 0.12_34h| -0.12_34h | 0.12_34h | -0.12_34h | *  

;* | +1.23456789012345678_16E+00 | 1.23_45h| +1.23_45h | 1.23_45h | +1.23_45h | *  

;* | +1.23456789012345678_16E+01 | 12.34_56h| -12.34_56h | 12.34_56h | -12.34_56h | *  

;* | +1.23456789012345678_16E+02 | 123.45_67h| +123.45_67h | 123.45_67h | +123.45_67h | *  

;* | +1.23456789012345678_16E+03 | 1234.56_78h| -1234.56_78h | 1234.56_78h | -1234.56_78h | *  

;* | +1.23456789012345678_16E+04 | 12345.67_89h| +12345.67_89h | 12345.67_89h | +12345.67_89h | *  

;* | +1.23456789012345678_16E+05 | 123456.78_90h| -123456.78_90h | 123456.78_90h | -123456.78_90h | *  

;* | +1.23456789012345678_16E+06 | 1234567.89_01h| +1234567.89_01h | 1234567.89_01h | +1234567.89_01h | *  

;* | +1.23456789012345678_16E+07 | -1.23456789_16E+07| -1.23456789_16E+07 | -1.23456789_16E+07 | -1.23456789_16E+07 | *  

;* | Hexadecimal | Hex(+18|+08|B02) | Hex(+18|-08|B02) | Hex(-18|+08|B02) | Hex(-18|-08) | *  

;* | + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - - - - - + - *  

;* | Integer Input 00000000_16 | 00_00_00_00h| +00_00_00_00h | 00_00_00_00h | +00_00_00_00h | *  

;* | 00000001_16 | 00_00_00_01h| -00_00_00_01h | 00_00_00_01h | -00_00_00_01h | *  

;* | 00000012_16 | 00_00_00_12h| +00_00_00_12h | 00_00_00_12h | +00_00_00_12h | *

```


4.9.1 Input Output Functions

4.9.1.1 GETMAT GET MATH library constants

The function FLREAD reads unicode character string tokens from a diskfile or memoryarea. The diskfile or memory must be opened with the OPNFIL function. One data-token consists from an optional name string and an following optional data string. The name and the data part are separated with an equal character. Each token is ended with an token end character. This is either a comma or a semicolon.

Die Function FLREAD liest unicode charakter String Tokens von einer Diskdatei oder Speicherbereich. Die Diskdatei oder der Speicherbereich muss vor mit der OPNFIL Funktion geöffnet worden sein. Ein Daten-Token besteht aus einem optionalen Namen und einem optionalen Datenteil, getrennt mit einem Gleichheitszeichen. Jeder Token wird entweder mit einem Komma oder Semikolon beendet.

```

;* //----- *  

;* IV | IV IR IR IR | IR // 12345678901234567 *  

;* Lab12345: Error2 = CPP_FLREAD( FilNum, &TokCnt, &PagCnt, &LinCnt, &ColCnt, //  

;* NaCtrl, NaBSiz, NaBCnt, NaECnt, &NaSOff, &NaSCnt, // 01 C/C++ direct *  

;* DaCtrl, DaBSiz, DaBCnt, DaECnt, &DaSOff, &DaSCnt, // 02 *  

;* &DatLen, *P64Res, &BasFnd, &ChrCntr, *DaStrg, *NaStrg); // 03 *  

;* IV IV IV | IV IR IR // 04 *  

;* IV | IV IV IV IR ! IR // *  

;* IR PR IR | IR CR CR // *  

;* // *  

;* Lab12345: Error2 = CPP_FLREAD( 000001, &TokCnt, &PagCnt, &LinCnt, &ColCnt, // 01 C/C++ direct *  

;* 1LAUSNEMCT, 2560, 0000, 2560, &NaSOff, &NaSCnt, // 02 /C## call *  

;* 1IAUSNPrBF, 2560, 0000, 2560, &DaSOff, &DaSCnt, // 03 *  

;* &DatLen, *P64Res, &BasFnd, &ChrCntr, *DaStrg, *NaStrg); // 04 *  

;* // *  

;*****  

;23_12345678 equ nam_Variable+XSIZ ; XVAL: Description [Int32 Amd64 IA128 IA256 IA512]  

;#####  

;## OutSTACK ; LVAL: Internal OutputStack for callings [Int32 Amd64 IA128 IA256 IA512]  

;#####  

ffr_ShArg001 equ 000+00000000+0000 ; LVAL: FastCall Shadow Call Arg(LCX) 001 [00000 00000 00000 00000 00000]  

ffr_ShArg002 equ ffr_ShArg001+LSIZ ; LVAL: FastCall Shadow Call Arg(LDX) 002 [00004 00008 00016 00032 00064]  

;fr_ShArg003 equ ffr_ShArg002+LSIZ ; LVAL: FastCall Shadow Call Arg(L08) 003 [00008 00016 00032 00064 00128]  

;fr_ShArg004 equ ffr_ShArg003+LSIZ ; LVAL: FastCall Shadow Call Arg(L09) 004 [00012 00024 00048 00096 00192]  

;* - - - - -  

;fr_StArg005 equ ffr_ShArg004+LSIZ ; LVAL: FastCall Normal Call Argument 005 [00016 00032 00064 00128 00256]  

;fr_StArg006 equ ffr_StArg005+LSIZ ; LVAL: FastCall Stack! Call Argument 006 [00020 00040 00080 00160 00320]  

;fr_StArg007 equ ffr_StArg006+LSIZ ; LVAL: FastCall Normal Call Argument 007 [00024 00048 00096 00192 00384]  

;fr_StArg008 equ ffr_StArg007+LSIZ ; LVAL: FastCall Stack! Call Argument 008 [00028 00056 00112 00224 00448]  

;* - - - - -  

;fr_StArg009 equ ffr_StArg008+LSIZ ; LVAL: FastCall Normal Call Argument 009 [00032 00064 00128 00256 00512]  

;fr_StArg010 equ ffr_StArg009+LSIZ ; LVAL: FastCall Stack! Call Argument 010 [00036 00072 00144 00288 00576]  

;fr_StArg011 equ ffr_StArg010+LSIZ ; LVAL: FastCall Normal Call Argument 011 [00040 00080 00160 00320 00640]  

;fr_StArg012 equ ffr_StArg011+LSIZ ; LVAL: FastCall Stack! Call Argument 012 [00044 00088 00176 00352 00704]  

;* - - - - -  

;fr_StArg013 equ ffr_StArg012+LSIZ ; LVAL: FastCall Normal Call Argument 013 [00048 00096 00192 00384 00768]  

;fr_StArg014 equ ffr_StArg013+LSIZ ; LVAL: FastCall Stack! Call Argument 014 [00052 00104 00208 00416 00832]  

;fr_StArg015 equ ffr_StArg014+LSIZ ; LVAL: FastCall Normal Call Argument 015 [00056 00112 00224 00448 00896]  

;fr_StArg016 equ ffr_StArg015+LSIZ ; LVAL: FastCall Stack! Call Argument 016 [00060 00120 00240 00480 00960]  

;#####  

;## FIXED ; EVAL: Internal Non scaling Variables [Int32 Amd64 IA128 IA256 IA512]  

;#####  

ffr_BEG equ 000+002*LSIZ+0000 ; PARA: Fortran.File.Read_BEGIN offset [%0000 %0000 %0000 %0000 %0000]  

ffr_LNG equ ffr_BEG+MSIZ+0000 ; PARA: Fortran.File.Read_LoNG offset [+0016 +0016 +0032 +0064 +0128]  

;%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%  

;fr_Valu0000 equ ffr_BEG+0000+000 ; EVAL: Values Part0 Non scaling Values ! [%0000 %0000 %0000 %0000 %0000]  

;fr_Valu0001 equ ffr_Valu0000+04 ; EVAL: Value1 Part0 [+0004 +0004 +0004 +0004 +0004]  

;%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%  

;fr_Valu1024 equ ffr_BEG+1024+000 ; XVAL: Values Part1 Non scaling Values ! [%1024 %1024 %1024 %1024 %1024]  

;%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%  

;fr_Valu2048 equ ffr_BEG+2048+000 ; XVAL: Values Part1 Non scaling Values ! [%2048 %2048 %2048 %2048 %2048]  

;#####  

;## LONG ; LVAL: [Int32 Amd64 IA128 IA256 IA512]  

;#####  

ffr_SFTopPtr equ ffr_LNG+LSIZ*0000 ; PNTR: Local! Stack Frame TopLAX Pointer [#0000 #0000 #0000 #0000 #0000]  

ffr_SFBotPtr equ ffr_SFTopPtr+LSIZ ; PNTR: Local! Stack Frame Bottom Pointer [+0004 +0008 +0016 +0032 +0064]  

;=====  

ffr_STRCSIZE equ ffr_LNG+LSIZ*0002 ; PARA: Internal StackFrame Structure END [+0016 +0032 +0064 +0128 +0256]  

;#####  

;## InpSTACK ; LVAL: External InputStack [called!] TOP [Int32 Amd64 IA128 IA256 IA512]  

;#####  

FFR_StackLAX equ 000+00000000+0000 ; LVAL: External StackFrame Register: LAX [00000 00000 00000 00000 00000]  

FFR_StackLDX equ FFR_StackLAX+LSIZ ; LVAL: External StackFrame Register: LDX [+0004 +0008 +0016 +0032 +0064]  

FFR_StackLCX equ FFR_StackLDX+LSIZ ; LVAL: External StackFrame Register: LCX [+0008 +0016 +0032 +0064 +0128]  

;* - - - - -  

FFR_StackLDI equ FFR_StackLCX+LSIZ ; PNTR: External StackFrame Register: LDI [+0012 +0024 +0048 +0096 +0192]  

FFR_StackLSI equ FFR_StackLDI+LSIZ ; PNTR: External StackFrame Register: LSI [+0016 +0032 +0064 +0128 +0256]  

;* - - - - -  

FFR_StackLBX equ FFR_StackLSI+LSIZ ; PNTR: External StackFrame Register: LBX [+0020 +0040 +0080 +0160 +0320]  

FFR_StackLBP equ FFR_StackLBX+LSIZ ; PNTR: External StackFrame Register: LBP [+0024 +0048 +0096 +0192 +0384]  

;* - - - - -  

;FR_StackL12 equ FFR_StackLBP+LSIZ ; PNTR: External StackFrame Register: L12 [+0028 +0056 +0112 +0224 +0448]  

;FR_StackL13 equ FFR_StackL12+LSIZ ; PNTR: External StackFrame Register: L13 [+0032 +0064 +0128 +0256 +0512]  

;FR_StackL14 equ FFR_StackL13+LSIZ ; PNTR: External StackFrame Register: L14 [+0036 +0072 +0144 +0288 +0576]  

;FR_StackL15 equ FFR_StackL14+LSIZ ; PNTR: External StackFrame Register: L15 [+0040 +0080 +0160 +0320 +0640]  

;* - - - - -  

FFR_StackLIP equ FFR_StackLBP+LSIZ ; PNTR: External StackFrame Register: LIP [+0044 +0088 +0176 +0352 +0704]  

;%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/%  

FFR_ExFILNUM equ FFR_StackLIP+LSIZ ; PNTR: FastCall Normal InputArgument 001 [#0048 #0096 #0192 #0384 #0768]  

FFR_ExTOKCNT equ FFR_ExFILNUM+LSIZ ; PNTR: FastCall Normal InputArgument 002 [+0004 +0008 +0016 +0032 +0064]  

FFR_ExPAGCNT equ FFR_ExTOKCNT+LSIZ ; PNTR: FastCall Normal InputArgument 003 [+0008 +0016 +0032 +0064 +0128]  

FFR_ExLINCNT equ FFR_ExPAGCNT+LSIZ ; PNTR: FastCall Normal InputArgument 004 [+0012 +0024 +0048 +0096 +0192]  

FFR_ExCOLCNT equ FFR_ExLINCNT+LSIZ ; PNTR: FastCall Normal InputArgument 005 [+0016 +0032 +0064 +0128 +0256]  

;* - - - - -  

FFR_ExNACTRL equ FFR_ExCOLCNT+LSIZ ; PNTR: FastCall Normal InputArgument 006 [+0020 +0040 +0080 +0160 +0320]

```



```

F1FLRea@10: lea    1bp, [lbx+gmm_eml+0000]
              mov    PNTR [1bp+eml_SFBotPtr], lbp
;* - - - - - ; Fetch LBP := GloMaiMem_Err.Mat.Lib Memory Pnter
F1FLRea@16: lea    lsi, [lbx+gmm_iol+0000]
              mov    PNTR [lsi+iol_SFBotPtr], lsi
;* - - - - - ; Store LBP => ExcMatLib_Stack.Frame.Bottom Pnter
;* - - - - - ; Fetch LSI := GloMaiMem_Inp.Out.Lib Memory Pnter
;* LRea@17: lea    ldi, [lbx+gmm_mlx+0000]
;* UnUsed ! mov    PNTR [ldi+mlx_SFBotPtr], ldi
;* - - - - - ; Store LSI => InpOutLib_Stack.Frame.Bottom Pnter
;***** =====
F1FLRea@21: mov    eax, 0000000000000000
              mov    LVAL [1bp+eml_MbContrl], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage <IAUSN????>
;* - - - - - ; Store LAX => ExcMatLib Informations <IAUSN????>
F1FLRea@22: lea    lax, [ljp+dat_F1FLREAD]
              mov    PNTR [1bp+eml_MbBoxTit], lax
;* - - - - - ; Fetch LAX := ErrMsgBox WindowsTitle FOR_FLREAD(
;* - - - - - ; Store LAX => ExcMatLib Informations StrgPointer
F1FLRea@23: lea    lax, [ljp+dat_MbHlgFil]
              mov    PNTR [1bp+eml_MbHlgFil], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage FileName
;* - - - - - ; Store LAX => ExcMatLib Informations StrgPointer
F1FLRea@24: lea    lax, [ljp+dat_MbHlgFun]
              mov    PNTR [1bp+eml_MbHlgFun], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage Function
;* - - - - - ; Store LAX => ExcMatLib Informations StrgPointer
F1FLRea@25: mov    eax, 0000000010220333
              mov    LVAL [1bp+eml_MbHlgLin], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage LineNumber
;* - - - - - ; Store LAX => ExcMatLib Informations IntegerValue
;* - - - - - ;***** =====
F1FLRea@26: mov    eax, 0000000000000000
              mov    LVAL [1bp+eml_MbParent], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage Null Parent
;* - - - - - ; Store LAX => ExcMatLib Informations Wind Handle
;***** =====
F1FLRea@31: mov    eax, 0000000000000000
              mov    LVAL [1bp+eml_JqCONTRL], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage <IAUSN????>
;* - - - - - ; Store LAX => ExcMatLib Informations <IAUSN????>
F1FLRea@32: mov    eax, 0000000000000000
              mov    LVAL [1bp+eml_JqCONNUM], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage <IAUSN????>
;* - - - - - ; Store LAX => ExcMatLib Informations <IAUSN????>
F1FLRea@33: mov    eax, 0000000000000000
              mov    LVAL [1bp+eml_JqCMUTEX], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage <IAUSN????>
;* - - - - - ; Store LAX => ExcMatLib Informations <IAUSN????>
F1FLRea@34: mov    eax, 0000000000000000
              mov    LVAL [1bp+eml_JqWCOHND], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage <IAUSN????>
;* - - - - - ; Store LAX => ExcMatLib Informations <IAUSN????>
F1FLRea@35: mov    eax, 0000000000000000
              mov    LVAL [1bp+eml_JqRCOHND], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage <IAUSN????>
;* - - - - - ; Store LAX => ExcMatLib Informations <IAUSN????>
F1FLRea@36: mov    eax, 0000000000000000
              mov    LVAL [1bp+eml_JqHEXFMT], lax
;* - - - - - ; Fetch LAX := ErrMsgBox HighLanguage <IAUSN????>
;* - - - - - ; Store LAX => ExcMatLib Informations <IAUSN????>
;***** =====
F1FLRead@9:  mov    eax, 'FFR@'
              mov    edx, 00000BEh
              call   EML1QPut
;* Debugs ! jmp    LJMP F1FLReadGT
;* - - - - - ; fetch EAX := FortranFileReads <?> 0000@0
;* - - - - - ; 00000-EAX-na-[NEBT:MSiz:LFun]-ebt-ui04B1Text-ML
;* - - - - - ; Write Out
;* - - - - - ; Goto: End
;* - - - - - ;***** =====

```

4.9.1.2 FLWRIT FiLe WRITe function

The function FLREAD reads unicode character string tokens from a diskfile or memoryarea. The diskfile or memory must be opened with the OPNFIL function. One data-token consists from an optional name string and an following optional data string. The name and the data part are separated with an equal character. Each token is ended with an token end character. This is either a comma or a semicolon.

Die Function FLREAD liest unicode charakter String Tokens von einer Diskdatei oder Speicherbereich. Die Diskdatei oder der Speicherbereich muss vor mit der OPNFIL Funktion geöffnet worden sein. Ein Daten-Token besteht aus einem optionalen Namen und einem optionalen Datenteil, getrennt mit einem Gleichheitszeichen. Jeder Token wird entweder mit einem Komma oder Semikolon beendet.

Normaler Text