wdv-notes 217
Stand: 5.APR.1987 (16.)
Wiss.Datenverarbeitung
FREIE UNIVERSITÄT BERLIN

FORTRAN: Die Standard-Funktionen.

© 1962–1994 *Karl-Heinz Dittberner*

Pro

Zur Programmiersprache FORTRAN gehört immer eine Sammlung von eingebauten Standard-Funktionen (Intrinsic Functions), die hier zusammenfassend dargestellt sind.

Alle Angaben beruhen nunmehr auf dem Stand des FORTRAN-77, wie er 1980 in der ISO-Norm 1539 (E) festgelegt wurde. Da das FORTRAN-77 heute auf allen Computern zur Standardausrüstung gehört, wurde auf die Angabe der Änderungen gegenüber FORTRAN-II bzw. FORTRAN-IV verzichtet.

Alle *kursiv gedruckten Funktionen* sind spezielle Erweiterungen von FORTRAN-77, die *nur* auf einem VAX-Computer unter VMS verfügbar sind, über den das Institut im nächsten Jahr verfügen wird. Für die Portierbarkeit von Programmen ist dieses unbedingt zu beachten!

1. Mathematische Funktionen:

Funktion	Name	Datentyp Argument	Datentyp Ergebnis	Bem.
Absolutbetrag $\lvert x \rvert$	**IABS**	INTEGER*4	INTEGER*4	
	IIABS	*INTEGER*2*	*INTEGER*2*	
	JIABS	INTEGER*4	INTEGER*4	
	ABS	REAL*4	REAL*4	
	DABS	REAL*8	REAL*8	
	QABS	*REAL*16*	*REAL*16*	
	CABS	COMPLEX*8	REAL*4	
	CDABS	*COMPLEX*16*	*REAL*8*	
Quadratwurzel $\sqrt{x}$	**SQRT**	REAL*4	REAL*4	
	DSQRT	REAL*8	REAL*8	
	QSQRT	*REAL*16*	*REAL*16*	
	CSQRT	COMPLEX*8	COMPLEX*8	
	CDSQRT	*COMPLEX*16*	*COMPLEX*16*	
Natürlicher Logarithmus $\ln(x)$	**ALOG**	REAL*4	REAL*4	
	DLOG	REAL*8	REAL*8	
	QLOG	*REAL*16*	*REAL*16*	
	CLOG	COMPLEX*8	COMPLEX*8	
	CDLOG	*COMPLEX*16*	*COMPLEX*16*	
Logarithmus $\lg(x)$	**ALOG10**	REAL*4	REAL*4	
	DLOG10	REAL*8	REAL*8	
	QLOG10	*REAL*16*	*REAL*16*	
Exponential-Funktion e^x	**EXP**	REAL*4	REAL*4	
	DEXP	REAL*8	REAL*8	
	QEXP	*REAL*16*	*REAL*16*	
	CEXP	COMPLEX*8	COMPLEX*8	
	CDEXP	*COMPLEX*16*	*COMPLEX*16*	
Produkt $x \cdot y$	**DPROD**	REAL*4	REAL*8	

2. Trigonometrische Funktionen (Radian):

Die Argumente x der folgenden Funktionen sind in Bogenmaß (Radian) anzugeben. Sie werden modulo 2 π genommen.

Funktion	Name	Datentyp Argument	Datentyp Ergebnis	Bem.
Sinus $\sin(x)$	**SIN**	REAL*4	REAL*4	
	DSIN	REAL*8	REAL*8	
	QSIN	*REAL*16*	*REAL*16*	
	CSIN	COMPLEX*8	COMPLEX*8	
	CDSIN	*COMPLEX*16*	*COMPLEX*16*	
Arcussinus $\arcsin(x)$	**ASIN**	REAL*4	REAL*4	
	DASIN	REAL*8	REAL*8	
	QASIN	*REAL*16*	*REAL*16*	
Cosinus $\cos(x)$	**COS**	REAL*4	REAL*4	
	DCOS	REAL*8	REAL*8	
	QCOS	*REAL*16*	*REAL*16*	
	CCOS	COMPLEX*8	COMPLEX*8	
	CDCOS	*COMPLEX*16*	*COMPLEX*16*	
Arcuscosinus $\arccos(x)$	**ACOS**	REAL*4	REAL*4	
	DACOS	REAL*8	REAL*8	
	QACOS	*REAL*16*	*REAL*16*	
Tangens $\tan(x)$	**TAN**	REAL*4	REAL*4	
	DTAN	REAL*8	REAL*8	
	QTAN	*REAL*16*	*REAL*16*	
Arcustangens $\arctan(x)$	**ATAN**	REAL*4	REAL*4	
	DATAN	REAL*8	REAL*8	
	QATAN	*REAL*16*	*REAL*16*	
Arcustangens $\arctan(x/y)$	**ATAN2**	REAL*4	REAL*4	1)
	DATAN2	REAL*8	REAL*8	1)
	QATAN2	*REAL*16*	*REAL*16*	1)

1) Das Ergebnis z liegt im Bereich: $-\pi < z < \pi$.

3. Trigonometrische Funktionen (Grad):

Die Argumente x der folgenden Funktionen sind in Grad (0 – 360°) anzugeben. Sie werden modulo 360° genommen. Diese Funktionen sind eine Spezialität des FORTRAN-77 auf einem VAX-Computer, also nicht portierbar! Die Funktionsnamen sind durch das Anhängen eines „D“ (Degree) an die Funktionsnamen von Punkt 2 gebildet.

Funktion	Name	Datentyp Argument	Datentyp Ergebnis	Bem.
Sinus $\sin(x)$	***SIND***	*REAL*4*	*REAL*4*	
	DSIND	*REAL*8*	*REAL*8*	
	QSIND	*REAL*16*	*REAL*16*	
Arcussinus $\arcsin(x)$	***ASIND***	*REAL*4*	*REAL*4*	
	DASIND	*REAL*8*	*REAL*8*	
	QASIND	*REAL*16*	*REAL*16*	
Cosinus $\cos(x)$	***COSD***	*REAL*4*	*REAL*4*	
	DCOSD	*REAL*8*	*REAL*8*	
	QCOSD	*REAL*16*	*REAL*16*	
Arcuscosinus $\arccos(x)$	***ACOSD***	*REAL*4*	*REAL*4*	
	DACOSD	*REAL*8*	*REAL*8*	
	QACOSD	*REAL*16*	*REAL*16*	
Tangens $\tan(x)$	***TAND***	*REAL*4*	*REAL*4*	
	DTAND	*REAL*8*	*REAL*8*	
	QTAND	*REAL*16*	*REAL*16*	
Arcustangens $\arctan(x)$	***ATAND***	*REAL*4*	*REAL*4*	
	DATAND	*REAL*8*	*REAL*8*	
	QATAND	*REAL*16*	*REAL*16*	
Arcustangens $\arctan(x/y)$	***ATAND2***	*REAL*4*	*REAL*4*	1)
	DATAND2	*REAL*8*	*REAL*8*	1)
	QATAND2	*REAL*16*	*REAL*16*	1)

1) Das Ergebnis z liegt im Bereich: $-180° < z < 180°$.

4. Hyperbolische Funktionen:

Funktion	Name	Datentyp Argument	Datentyp Ergebnis	Bem.
Hyperb. Sinus $\sinh(x)$	**SINH**	REAL*4	REAL*4	
	DSINH	REAL*8	REAL*8	
	QSINH	*REAL*16*	*REAL*16*	
Hyperb. Cosinus $\cosh(x)$	**COSH**	REAL*4	REAL*4	
	DCOSH	REAL*8	REAL*8	
	QCOSH	*REAL*16*	*REAL*16*	
Hyperb. Tangens $\tanh(x)$	**TANH**	REAL*4	REAL*4	
	DTANH	REAL*8	REAL*8	
	QTANH	*REAL*16*	*REAL*16*	

5. Funktionen für Zeichenketten (Strings):

Die Ausstattung mit String-Funktionen, verglichen mit moderneren Programmiersprachen, ist auch unter FORTRAN-77 nur äußerst spartanisch. „Leute“ das ist schlichtweg zu wenig!

Funktion	Name	Datentyp Argument	Datentyp Ergebnis	Bem.
ASCII Wert	**ICHAR**	CHARACTER	INTEGER*4	1)
Character	**CHAR**	*LOGICAL*1*	CHARACTER	2)
	CHAR	*INTEGER*2*	CHARACTER	2)
	CHAR	INTEGER*4	CHARACTER	2)
Character Relationals	**LLT**	CHARACTER	LOGICAL*4	<
	LLE	CHARACTER	LOGICAL*4	≤
	LGT	CHARACTER	LOGICAL*4	>
	LGE	CHARACTER	LOGICAL*4	≥
Length	**LEN**	CHARACTER	INTEGER*4	3)
Index	**INDEX**	CHARACTER	INTEGER*4	4)

1) Ermittelt den ASCII-Code zum Argument, das vom Typ CHARACTER der Länge 1 sein muß.
2) Ermittelt das ASCII-Zeichen aufgrund eines gegebenen ASCII-Codes.
3) Ermitteln der Länge einer Zeichenkette (Character Expression).
4) Ermitteln der Position (Index) einer bestimmten Substring in einer Zeichenkette. Beispiel: POSITION = INDEX(STRING, SUBSTRING)

6. Nützliche Funktionen:

Funktion	Name	Datentyp Argument	Datentyp Ergebnis	Bem.
SIGN	**ISIGN**	INTEGER*2	INTEGER*2	
(Signum)	***IISIGN***	*INTEGER*2*	*INTEGER*2*	
	JISIGN	INTEGER*4	INTEGER*4	
$\text{sign}(x)$	**SIGN**	REAL*4	REAL*4	
	DSIGN	REAL*8	REAL*8	
	QSIGN	*REAL*16*	*REAL*16*	
INT	**INT**	REAL*4	INTEGER*2	
(Abschneiden)	***IINT***	*REAL*4*	*INTEGER*2*	
	JINT	REAL*4	INTEGER*4	
	IDINT	REAL*8	INTEGER*2	
$[x]$ 1)	***IIDINT***	*REAL*8*	*INTEGER*2*	
	JIDINT	REAL*8	INTEGER*4	
	IQINT	*REAL*16*	*INTEGER*2*	
	IIQINT	*REAL*16*	*INTEGER*2*	
	JIQINT	*REAL*16*	*INTEGER*4*	
	JIQINT	*COMPLEX*8*	*INTEGER*2*	
	JIQINT	COMPLEX*8	INTEGER*4	
	JIQINT	*COMPLEX*16*	*INTEGER*2*	
	JIQINT	*COMPLEX*16*	*INTEGER*4*	
	AINT	REAL*4	REAL*4	
	DINT	REAL*8	REAL*8	
	QINT	*REAL*16*	*REAL*16*	
Nächste	**NINT**	REAL*4	INTEGER*2	
Integer-Zahl	***ININT***	*REAL*4*	*INTEGER*2*	
	JNINT	REAL*4	INTEGER*4	
	IDNINT	REAL*8	INTEGER*2	
$[x+.5\text{sign}(x)]$	***IIDNNT***	*REAL*8*	*INTEGER*2*	
	JIDNNT	REAL*8	INTEGER*4	
	IQNINT	*REAL*16*	*INTEGER*2*	
	IIQNNT	*REAL*16*	*INTEGER*2*	
	JIQNNT	*REAL*16*	*INTEGER*4*	
	ANINT	REAL*4	REAL*8	
	DNINT	REAL*8	REAL*8	
	QNINT	*REAL*16*	*REAL*16*	
MOD	**MOD**	INTEGER*2	INTEGER*2	
(Modulo)	***IMOD***	*INTEGER*2*	*INTEGER*2*	
	JMOD	INTEGER*4	INTEGER*4	
$x_1 - x_2[x_1/x_2]$	**AMOD**	REAL*4	REAL*4	
	DMOD	REAL*8	REAL*8	
	QMOD	*REAL*16*	*REAL*16*	
MIN	**MIN**	INTEGER*2	INTEGER*2	
	MIN0	INTEGER*2	INTEGER*2	
	IMIN0	*INTEGER*2*	*INTEGER*2*	
$\min(x_1,\ldots,x_n)$	***JMIN0***	INTEGER*4	INTEGER*4	
	AMIN0	INTEGER*2	REAL*4	
	AIMIN0	*INTEGER*2*	*REAL*4*	
	AJMIN0	INTEGER*4	REAL*4	
	MIN1	REAL*4	INTEGER*2	
	IMIN1	*REAL*4*	*INTEGER*2*	
	JMIN1	REAL*4	INTEGER*4	
	AMIN1	REAL*4	REAL*4	
	DMIN1	REAL*8	REAL*8	
	QMIN1	*REAL*16*	*REAL*16*	
MAX	**MAX**	INTEGER*2	INTEGER*2	
	MAX0	INTEGER*2	INTEGER*2	
	IMAX0	*INTEGER*2*	*INTEGER*2*	
$\max(x_1,\ldots,x_n)$	***JMAX0***	INTEGER*4	INTEGER*4	
	AMAX0	INTEGER*2	REAL*4	
	AIMAX0	*INTEGER*2*	*REAL*4*	
	AJMAX0	INTEGER*4	REAL*4	
	MAX1	REAL*4	INTEGER*2	
	IMAX1	*REAL*4*	*INTEGER*2*	
	JMAX1	REAL*4	INTEGER*4	
	AMAX1	REAL*4	REAL*4	
	DMAX1	REAL*8	REAL*8	
	QMAX1	*REAL*16*	*REAL*16*	
DIM	**IDIM**	INTEGER*2	INTEGER*2	
	IIDIM	*INTEGER*2*	*INTEGER*2*	
x_1	***JIDIM***	INTEGER*4	INTEGER*4	
	DIM	REAL*4	REAL*4	
$-(\min(x_1,x_2))$	**DDIM**	REAL*8	REAL*8	
	QDIM	*REAL*16*	*REAL*16*	

1) $[x]$ sei definiert als die größte Integer-Zahl, deren Betrag nicht größer als der Betrag von x ist und deren Vorzeichen gleich dem von x ist.
Ein Beispiel: [3.6] => 3 und [–3.6] => –3.

7. Konvertieren von Datentypen:

Funktion	Name	Datentyp Argument	Datentyp Ergebnis	Bem.
FIX	**IFIX**	REAL*4	INTEGER*2	
	IIFIX	*REAL*4*	*INTEGER*2*	
	JIFIX	REAL*4	INTEGER*4	
FLOAT	**FLOAT**	INTEGER*2	REAL*4	
=> REAL*4	***FLOATI***	*INTEGER*2*	*REAL*4*	
	FLOATJ	INTEGER*4	REAL*4	
	FLOATJ	REAL*4	REAL*4	
	SNGL	REAL*8	REAL*4	
	SNGLQ	*REAL*16*	*REAL*4*	
	SNGLQ	COMPLEX*8	REAL*4	
	SNGLQ	*COMPLEX*16*	*REAL*4*	
FLOAT	***DFLOAT***	*INTEGER*2*	*REAL*8*	
=> REAL*8	***DFLOTI***	*INTEGER*2*	*REAL*8*	
	DFLOTJ	*INTEGER*4*	*REAL*8*	
	DBLE	*INTEGER*2*	*REAL*8*	
	DBLE	INTEGER*4	REAL*8	
	DBLE	REAL*4	REAL*8	
	DBLE	REAL*8	REAL*8	
	DBLEQ	*REAL*16*	*REAL*8*	
	DBLEQ	COMPLEX*8	REAL*8	
	DBLEQ	*COMPLEX*16*	*REAL*8*	
FLOAT	***QFLOAT***	*INTEGER*2*	*REAL*16*	
=> REAL*16	***QFLOAT***	*INTEGER*4*	*REAL*16*	
	QEXT	*INTEGER*2*	*REAL*16*	
	QEXT	*INTEGER*4*	*REAL*16*	
	QEXT	*REAL*4*	*REAL*16*	
	QEXTD	*REAL*8*	*REAL*16*	
	QEXTD	*REAL*16*	*REAL*16*	
	QEXTD	*COMPLEX*8*	*REAL*16*	
	QEXTD	*COMPLEX*16*	*REAL*16*	

8. Funktionen für komplexe Zahlen:

Funktion	Name	Datentyp Argument	Datentyp Ergebnis	Bem.
Realteil	**REAL**	COMPLEX*8	REAL*4	
	DREAL	*COMPLEX*16*	*REAL*8*	
Imaginärteil	**AIMAG**	COMPLEX*8	REAL*4	
	DIMAG	*COMPLEX*16*	*REAL*8*	
Konjugiert	**CONJG**	COMPLEX*8	COMPLEX*8	
Komplexe Zahl	***DCONJG***	*COMPLEX*16*	*COMPLEX*16*	
Convert =>	**CMPLX**	*INTEGER*2*	*COMPLEX*8*	1)
COMPLEX*8	**CMPLX**	INTEGER*4	COMPLEX*8	1)
	CMPLX	REAL*4	COMPLEX*8	1)
	CMPLX	REAL*8	COMPLEX*8	1)
	CMPLX	*REAL*16*	*COMPLEX*8*	1)
	CMPLX	COMPLEX*8	COMPLEX*8	1)
	CMPLX	*COMPLEX*16*	*COMPLEX*8*	1)
Convert =>	***DCMPLX***	*INTEGER*2*	*COMPLEX*16*	1)
COMPLEX*16	***DCMPLX***	*INTEGER*4*	*COMPLEX*16*	1)
	DCMPLX	*REAL*4*	*COMPLEX*16*	1)
	DCMPLX	*REAL*8*	*COMPLEX*16*	1)
	DCMPLX	*REAL*16*	*COMPLEX*16*	1)
	DCMPLX	*COMPLEX*8*	*COMPLEX*16*	1)
	DCMPLX	*COMPLEX*16*	*COMPLEX*16*	1)

1) Wird bei den Funktionen CMPLX oder DCMPLX nur ein Argument angegeben, dann wird dieses als Realteil verwendet und der Imaginärteil = 0 gesetzt.

9. Funktionen für Bit-Manipulationen:

Hierzu gehören beispielsweise die Funktionen: ***BTEST***, ***IAND***, ***IBCLR***, ***IBITS***, ***IBSET***, ***IEOR***, ***IOR***, ***ISHFT***, ***ISHFTC***, ***NOT***. Diese Funktionen sind in sehr hohem Maße implementationsabhängig. Deshalb wird hier auf die Wiedergabe verzichtet.

10. Sonstige Hinweise:

■ Jeder Computer-Hersteller liefert zu seinem FORTRAN-Compiler noch weitere – nicht genormte – Bibliotheksroutinen.

■ Diese Aufstellung der FORTRAN-Funktionen zeigt sehr deutlich, daß den „Erfindern“ von FORTRAN nicht der große Wurf gelungen ist. Konzeptionelles Denken mit (etwas) Weitblick ist nirgends zu erkennen. Die meisten Funktionsnamen sind außerdem schlecht gewählt, denn sie klingen „chinesisch“!

Notizen: