columns at right (***y***) plus or minus an object of the *DT* in column 1 (***x***) will return an object of the *DT* at the intersection (thus, wherever a *DT* number is printed at the intersection, the corresponding combination is legal for addition or subtraction[65]).

x *DT* and meaning	***y*** 1	2	3	4	5	6	7	8	9	10
1 ℤ *Long integer*	**1**	2	3	4	5	6	7	-	-	10
2 ℝ *Real number*	2	**2**	3	4	5	6	7	-	-	2
3 ℂ *Complex number*	3	3	**3**	-	-	-	7	-	-	3
4 *Angle* (in various formats)[66]	4	4	-	**4**	-	-	7	-	-	4
5 *Time* interval (in H.MS)	5	5	-	-	**5**	-	7	-	-	-
6 *Date* (in various formats)	6	6	-	-	-	**1**[67]	7	-	-	-
7 *Text string*[68]	-	-	-	-	-	-	**7**	-	-	-
8 *Real* matrix or vector	-	-	-	-	-	-	7	**8**	9	-
9 *Complex* matrix or vector	-	-	-	-	-	-	7	9	**9**	-
10 *Short integer*	1	2	3	4	-	-	7	-	-	**10**[69]

Example:

A *complex number* (*DT* 3) plus or minus a *real number* (*DT* 2) will result in a *complex number*. And a *text string* plus a *time* will result in a *text string*.

---

[65] Else an error `Illegal input data type for this operation` (24) will be thrown.

[66] Angular output is tagged according to the current *angular display mode* chosen.

[67] A *date* minus a *date* returns an integer number of days (there are no other legal arithmetic operations on two *dates*). And a *date* plus a *real number* takes the integer part of that number and adds the respective number of days to said *date*.

[68] In additive operations on *text strings*, such a string must be present in **Y** at the beginning. One character suffices. Adding corresponds to appending ***x*** (converted to a string according to the display format set at execution time, if applicable) to string ***y***. Adding a matrix to a *text string* appends its abbreviation (e.g. `[3×4 ℂ matrix]`, see the chapters about vectors and matrices below in this section). Subtractions from *text strings* are not allowed.

[69] If and when integers of different types (*DT* 1 and 10) or different bases are combined by an arithmetic operation, output will be an integer of the type and base given in **Y**.

The following matrix shows the resulting *data types* of products and ratios in the same way (note that neither *dates* nor *text strings* can be multiplied or divided, and plain numbers cannot be divided by an *angle* or a *time* – cf. footnote 20 on p. 29):

	An object y of *DT* …							
	1	**2**	**3**	**4**	**5**	**8**	**9**	**10**
… **times** an object x of the *DT* below returns a product of the *DT* printed at the intersection.								
1 ℤ *Long integer*	**1**	2	3	4	5	8	9	10
2 ℝ *Real number*	2	**2**	3	4	5	8	9	2
3 ℂ *Complex number*	3	3	**3**	-	-	9	9	3
4 *Angle*	4	4	-	**-**	-	-	-	4
5 *Time* interval	5	5	-	-	**-**	-	-	5
8 *Real* matrix or vector	8	8	9	-	-	**8**	9	8
9 *Complex* matrix or vector	9	9	9	-	-	9	**9**	9
10 *Short integer*	1	2	3	4	5	8	9	**10**[69]
… **divided by** an object x of the *DT* below returns a ratio of the *DT* printed at the intersection.								
1 ℤ *Long integer*[70]	**1/2**	2	3	4	5	8	9	10
2 ℝ *Real number*	2	**2**	3	4	5	8	9	2
3 ℂ *Complex number*	3	3	**3**	-	-	9	9	3
4 *Angle*	-	-	-	**2**	-	-	-	-
5 *Time* interval	-	-	-	-	**2**	-	-	-
8 *Real* matrix[71]	8	8	9	-	-	**8**	9	8
9 *Complex* matrix[71]	9	9	9	-	-	9	**9**	9
10 *Short integer*	1/2	2	3	4	5	8	9	**10**[69]

10 ✗

[70] For example, 15 / 3 returns 5 while 14 / 5 returns 2.8.

[71] The matrix x must be invertible. Dividing by x is equivalent to multiplying times x^{-1}. (see the chapter *Vectors and Matrices: Calculating* below).

The following matrices are for powers ($|x| < 1$ may correspond to roots):

… **raised to a power of $x > 0$** of the *DT* below returns a result of the *DT* printed at the intersection.	A number $y > 0$ of *DT* … **1**	**2**	**10**
1 ℤ *Long integer*	**1**	2	10
2 ℝ *Real number*	2 [72]	**2**	10 [73]
10 *Short integer*	1	2	**10** [69]
… **raised to a power of $x < 0$** of the *DT* below …			
1 ℤ, **2** ℝ, and **10**	2		-

… **raised to a power of $x > 0$** of the *DT* below returns a result of the *DT* printed at the intersection.		A number $y < 0$ of *DT* … **1**	**2**	**10**
1 ℤ *Long integer*		**1**	2	10
2 ℝ *Real number*	FP(x) = 0	2	**2**	10 [73]
	else	3	3	-
10 *Short integer*		1	2	**10** [69]
… **raised to a power of $x < 0$** of the *DT* below …				
1 ℤ *Long integer* and **10** *short integer*		2		-
2 ℝ *Real number*	FP(x) = 0	2		-
	else	3		-

Any number of *DT* 1 or 2 raised to *complex* power will return a *complex number*, as well as any *complex number* raised to arbitrary power. Raising a *short integer* to *complex* power is not supported.

Other powers – involving *DTs* 4, 5, 6, 7, 8, or 9 – are not supported.

---

[72] For x < 1, results will be *long integer* if possible (e.g. **625** ENTER↑ **.25** y^x returns 5).

[73] The result will be the (*short*) *integer* part of y^x here.

And this is for logarithms:

… **combined in** $\log_x y$ **with a number** $x > 0$ of the *DT* below results in a *DT* printed at the intersection.	A number $y > 0$ of *DT* … **1**	**2**	**10**
1 ℤ *Long integer*	**1**[72]	2	10[74]
2 ℝ *Real number*	2[72]	**2**	10[74]
10 *Short integer*	1[72]	2	**10**[69]

And for $y < 0$ of *DT* 1 or 2, $\log_x y$ will return a *complex* result if such results are allowed (see the chapters about *complex numbers* below); $\log_x y$ of negative *short integers* is not supported.

As the *DT*s for results of $\sqrt{x}$, $\sqrt[3]{x}$, and $\sqrt[x]{y}$ can be derived easily from the generic power table on previous page, also the *DT*s for results of lg, lb, and ln can be derived from the table for $\log_x y$ above.

This is for integer divisions and remainders:

… **IDIVR-divided by** an object x of the *DT* below returns an integer ratio in **X** and a remainder in **Y** of the *data types* printed at the intersection.	An object y of *DT* … **1**	**2**	**10**
1 ℤ *Long integer*	**1**; **1**	1; 2	10; 10
2 ℝ *Real number*	1; 2	**1; 2**	10; 2
10 *Short integer*	1; 1	1; 2	**10; 10**[69]

Additionally, explicit *DT* conversions are available where necessary:

[74] The result will be the (*short*) *integer* part of $\log_x y$ here.

A closed object x of *DT* …						
1	**2**	**4** *angle*	**5** *time*	**6** *date*	**10**	… will be converted in an object x of the *DT* below by the command printed at the intersection.
IP		-	-	-	IP	**1** ℤ *Long integer*
→REAL (press .d)						**2** ℝ *Real number*
...			-	-	-	**4** *Angle*
→INT (press #)		-	-	-	→INT	**10** *Short integer* (of the base specified)

Recognizing Calculator Settings and Status

Some settings are obvious: as seen above, radix marks and gap settings are recognized in the numeric display immediately; so are date and time display modes (Y.MD / D.MY / M.DY and CLK24 / CLK12) in the time string top left in the *status bar*. Also *program-entry mode* (*PEM*) is easily recognized (see pp. 206ff).

Further modes and system states as well as many settings for specific *data types* are indicated in the *status bar*: The following specific characters may appear trailing the date and time string there, listed below from left to right in various sets – indicators shown in *startup default* are printed in a light blue field again: [75]

Indicator	Set by	Deleted by	Explanation, remarks
ℂ	CPXRES	¬ CPXRES	With CPXRES set, *complex* results of *real number* calculations are allowed, like $\sqrt{-1}$. Else a domain error would be thrown in such a case (see the *ReM*, *App. C*).
ℝ	¬ CPXRES	CPXRES	

[75] The symbol ¬ means "not" (i.e. the trailing *system flag* cleared), while "&" denotes a logical "and" and a comma a logical "or" in this table.