

The intcalc package

Heiko Oberdiek*

2019/12/15 v1.3

Abstract

This package provides expandable arithmetic operations with integers.

Contents

1	Documentation	2
1.1	Introduction	2
1.2	Conditions	2
1.2.1	Preconditions	2
1.2.2	Postconditions	3
1.3	Error handling	3
1.4	Operations	3
1.4.1	Num	3
1.4.2	Inv, Abs, Sgn	4
1.4.3	Min, Max, Cmp	4
1.4.4	Inc, Dec, Add, Sub	5
1.4.5	Shl, Shr	5
1.4.6	Mul, Sqr, Fac, Pow	6
1.4.7	Div, Mod	6
1.5	Interface for programmer	7
2	Implementation	8
2.1	Reload check and package identification	8
2.2	Catcodes	9
2.3	Macros independent of ϵ -TeX	10
2.3.1	Abs, Sgn	10
2.3.2	Min, Max, Cmp	10
2.3.3	Fac	11
2.4	Implementation based on ϵ -TeX	11
2.4.1	Num	11
2.4.2	Inv, Abs, Sgn	12
2.4.3	Min, Max, Cmp	12
2.4.4	Inc, Dec	12
2.4.5	Add, Sub	13
2.4.6	Shl, Shr	13
2.4.7	Mul, Sqr, Fac	14
2.4.8	Pow	14
2.4.9	Div, Mod	15
2.5	Implementation without ϵ -TeX	18

*Please report any issues at <https://github.com/ho-tex/intcalc/issues>

2.5.1	Num	18
2.5.2	Inv, Abs, Sgn	18
2.5.3	Min, Max, Cmp	19
2.5.4	Inc, Dec	19
2.5.5	Add, Sub	21
2.5.6	Shl, Shr	29
2.5.7	\InCa@Tim	31
2.5.8	Mul	34
2.5.9	Sqr, Fac	36
2.5.10	Pow	36
2.5.11	Div	38
2.5.12	Mod	42
2.5.13	Help macros	44
3	Installation	44
3.1	Download	44
3.2	Bundle installation	44
3.3	Package installation	44
3.4	Refresh file name databases	45
3.5	Some details for the interested	45
4	History	45
[2007/09/09 v1.0]		45
[2007/09/27 v1.1]		46
[2016/05/16 v1.2]		46
[2019/12/15 v1.3]		46
5	Index	46

1 Documentation

1.1 Introduction

Package `intcalc` defines arithmetic operations that deal with integers. Integers mean numbers in $\text{T}_{\text{E}}\text{X}$. The same restrictions apply, the range is limited to $[-2147483647, 2147483647]$.

The operations have the form of macros that take one or two integers as parameter and return the integer result. The macro name is a three letter operation name prefixed by the package name, e.g. `\intcalcAdd{10}{43}` returns 53.

The macros are fully expandable, exactly two expansion steps generate the result. Therefore the operations may be used nearly everywhere in $\text{T}_{\text{E}}\text{X}$, even inside `\number`, `\csname`, file names, or other expandable contexts.

The package contains two implementations of the operations. If $\varepsilon\text{-T}_{\text{E}}\text{X}$ is detected then the macros are implemented using its features (`\numexpr`). Otherwise the slower implementation without $\varepsilon\text{-T}_{\text{E}}\text{X}$'s help is chosen.

1.2 Conditions

1.2.1 Preconditions

- Arguments can be anything that $\text{T}_{\text{E}}\text{X}$ interprets as “number”. Examples: plain numbers, count or length register, macros that expands to a number.

- The arguments are limited to the range -2147483647 until 2147483647. These numbers belong to the range. Note that some operations have additional restrictions to the range.
- The argument may be expressions that `\numexpr` understands if ε -TeX is available.
- The resulting number must fit in the allowed range.

1.2.2 Postconditions

Additional properties of the macros apart from calculating a correct result (of course ☺):

- The macros are fully expandable. Thus they can be used inside `\edef`, `\csname`, after `\number`, for example.
- Furthermore exactly two expansion steps calculate the result.
- The number consists of one optional minus sign and one to ten digits. The first digit is larger than zero for numbers that consists of more than one digit.

In short, the number format is exactly the same as `\number` generates. And the tokens (minus sign, digits) have catcode 12 (other).

- Call by value is simulated. First the arguments are converted to numbers. Then these numbers are used in the calculations.

Remember that arguments may contain expensive macros or ε -TeX expressions. This strategy avoids multiple evaluations of such arguments.

1.3 Error handling

There are two kinds of errors if a precondition is violated: Some errors are detected by the macros, example: division by zero. In this cases an undefined control sequence is called and causes a TeX error message, example: `\IntCalcError:DivisionByZero`. The name of the control sequence contains the reason for the error. The TeX error may be ignored. Then the operation returns zero as result. Because the macros are supposed to work in expandible contexts. An traditional error message, however, is not expandable and would break these contexts.

If a number exceeds the range of -2147483647 until 2147483647, then TeX throws an error “Number too big” and recovers by using biggest allowed value. Example for the negative number -3000000000 is replaced by -2147483647.

1.4 Operations

Some definition equations below use the function `Int` that converts a real number to an integer. The number is truncated that means rounding to zero:

$$\text{Int}(x) := \begin{cases} \lfloor x \rfloor & \text{if } x \geq 0 \\ \lceil x \rceil & \text{otherwise} \end{cases}$$

1.4.1 Num

`\intcalcNum {⟨x⟩}`

Macro `\intcalcNum` converts its argument to a normalized integer number without unnecessary leading zeros or signs. The result matches the regular expression:

`0|-?[1-9][0-9]*`

1.4.2 Inv, Abs, Sgn

`\intcalcInv {⟨x⟩}`

Macro `\intcalcInv` switches the sign.

$\text{Inv}(x) := -x$

`\intcalcAbs {⟨x⟩}`

Macro `\intcalcAbs` returns the absolute value of integer $\langle x \rangle$.

$\text{Abs}(x) := |x|$

`\intcalcSgn {⟨x⟩}`

Macro `\intcalcSgn` encodes the sign of $\langle x \rangle$ as number.

$$\text{Sgn}(x) := \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

These return values can easily be distinguished by `\ifcase`:

```
\ifcase\intcalcSgn{⟨x⟩}
  $x=0$
\or
  $x>0$
\else
  $x<0$
\fi
```

1.4.3 Min, Max, Cmp

`\intcalcMin {⟨x⟩} {⟨y⟩}`

Macro `\intcalcMin` returns the smaller of the two integers.

$$\text{Min}(x, y) := \begin{cases} x & \text{if } x < y \\ y & \text{otherwise} \end{cases}$$

`\intcalcMax {⟨x⟩} {⟨y⟩}`

Macro `\intcalcMax` returns the larger of the two integers.

$$\text{Max}(x, y) := \begin{cases} x & \text{if } x > y \\ y & \text{otherwise} \end{cases}$$

`\intcalcCmp {⟨x⟩} {⟨y⟩}`

Macro `\intcalcCmp` encodes the comparison result as number:

$$\text{Cmp}(x, y) := \begin{cases} -1 & \text{if } x < y \\ 0 & \text{if } x = y \\ 1 & \text{if } x > y \end{cases}$$

These values can be distinguished by `\ifcase`:

```
\ifcase\intcalcCmp{<x>}{<y>}
  $x=y$
\or
  $x>y$
\else
  $x<y$
\fi
```

1.4.4 Inc, Dec, Add, Sub

`\intcalcInc {⟨x⟩}`

Macro `\intcalcInc` increments $\langle x \rangle$ by one.

$$\text{Inc}(x) := x + 1$$

`\intcalcDec {⟨x⟩}`

Macro `\intcalcDec` decrements $\langle x \rangle$ by one.

$$\text{Dec}(x) := x - 1$$

`\intcalcAdd {⟨x⟩} {⟨y⟩}`

Macro `\intcalcAdd` adds the two numbers.

$$\text{Add}(x, y) := x + y$$

`\intcalcSub {⟨x⟩} {⟨y⟩}`

Macro `\intcalcSub` calculates the difference.

$$\text{Sub}(x, y) := x - y$$

1.4.5 Shl, Shr

`\intcalcShl {⟨x⟩}`

Macro `\intcalcShl` implements shifting to the left that means the number is multiplied by two. Overflow is possible. The sign is preserved.

$$\text{Shl}(x) := x * 2$$

`\intcalcShr {⟨x⟩}`

Macro `\intcalcShr` implements shifting to the right. That is equivalent to an integer division by two. The sign is preserved.

$$\text{Shr}(x) := \text{Int}(x/2)$$

1.4.6 Mul, Sqr, Fac, Pow

`\intcalcMul {⟨x⟩} {⟨y⟩}`

Macro `\intcalcMul` calculates the product of $\langle x \rangle$ and $\langle y \rangle$.

$$\text{Mul}(x, y) := x * y$$

`\intcalcSqr {⟨x⟩}`

Macro `\intcalcSqr` returns the square product.

$$\text{Sqr}(x) := x^2$$

`\intcalcFac {⟨x⟩}`

Macro `\intcalcFac` returns the factorial of $\langle x \rangle$. Negative numbers are not permitted.

$$\text{Fac}(x) := x! \quad \text{for } x \geq 0$$

$$(0! = 1)$$

`\intcalcPow {⟨x⟩} {⟨y⟩}`

Macro `\intcalcPow` calculates the value of $\langle x \rangle$ to the power of $\langle y \rangle$. The error “division by zero” is thrown if $\langle x \rangle$ is zero and $\langle y \rangle$ is negative. permitted:

$$\text{Pow}(x, y) := \text{Int}(x^y) \quad \text{for } x \neq 0 \text{ or } y \geq 0$$

$$(0^0 = 1)$$

1.4.7 Div, Mod

`\intcalcDiv {⟨x⟩} {⟨y⟩}`

Macro `\intcalcDiv` performs an integer division. Argument `⟨y⟩` must not be zero.

$$\text{Div}(x, y) := \text{Int}(x/y) \quad \text{for } y \neq 0$$

`\intcalcMod {⟨x⟩} {⟨y⟩}`

Macro `\intcalcMod` gets the remainder of the integer division. The sign follows the divisor `⟨y⟩`. Argument `⟨y⟩` must not be zero.

$$\text{Mod}(x, y) := x \% y \quad \text{for } y \neq 0$$

The result ranges:

$$\begin{aligned} -|y| < \text{Mod}(x, y) \leq 0 & \quad \text{for } y < 0 \\ 0 \leq \text{Mod}(x, y) < y & \quad \text{for } y \geq 0 \end{aligned}$$

1.5 Interface for programmer

If the programmer can ensure some more properties about the arguments of the operations, then the following macros are a little more efficient.

In general numbers must obey the following constraints:

- Plain number: digit tokens only, no command tokens.
- Non-negative. Signs are forbidden.
- Arguments and the result must fit in range `0..2147483647`.
- Delimited by exclamation mark. Curly braces around the number are not allowed and will break the code.

`\IntCalcInc ⟨number⟩ !`

Incrementation, range: `0..2147483646`.

`\IntCalcDec ⟨number⟩ !`

Decrementation, range: `1..2147483647`.

`\IntCalcAdd ⟨number A⟩ ! ⟨number B⟩ !`

Addition, $A \geq B$.

`\IntCalcSub ⟨number A⟩ ! ⟨number B⟩ !`

Subtraction, $A \geq B$.

`\IntCalcShl ⟨number⟩ !`

Left shift (multiplication with two), range: `0..1073741823`.

```
\IntCalcShr <number> !
```

Right shift (integer division by two).

```
\IntCalcMul <number A> ! <number B> !
```

Multiplication, $A \geq B$.

```
\IntCalcDiv <number A> ! <number B> !
```

Division operation.

```
\IntCalcMod <number A> ! <number B> !
```

Modulo operation.

2 Implementation

```
1 (*package)
```

2.1 Reload check and package identification

Reload check, especially if the package is not used with L^AT_EX.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@intcalc.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{intcalc}{The package is already loaded}%
29 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^^M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51   \def\x#1#2#3[#4]{\endgroup
52     \immediate\write-1{Package: #3 #4}%
53     \xdef#1{#4}%
54   }%
55 \else
56   \def\x#1#2[#3]{\endgroup
57     #2[#{#3}]%
58     \ifx#1\@undefined
59       \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62       \xdef#1{#3}%
63     \fi
64   }%
65 \fi
66 \expandafter\x\csname ver@intcalc.sty\endcsname
67 \ProvidesPackage{intcalc}%
68 [2019/12/15 v1.3 Expandable calculations with integers (HO)]%
```

2.2 Catcodes

```
69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname InCa@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
```

```

87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^~M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\InCa@AtEnd{%
96     \InCa@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{33}{12}% !
102 \TMP@EnsureCode{40}{12}% (
103 \TMP@EnsureCode{41}{12}% )
104 \TMP@EnsureCode{42}{12}% *
105 \TMP@EnsureCode{43}{12}% +
106 \TMP@EnsureCode{45}{12}% -
107 \TMP@EnsureCode{47}{12}% /
108 \TMP@EnsureCode{58}{11}% : (letter!)
109 \TMP@EnsureCode{60}{12}% <
110 \TMP@EnsureCode{62}{12}% >
111 \TMP@EnsureCode{63}{14}% ? (comment!)
112 \edef\InCa@AtEnd{\InCa@AtEnd\noexpand\endinput}
113 \begingroup\expandafter\expandafter\expandafter\endgroup
114 \expandafter\ifx\csname InCa@TestMode\endcsname\relax
115 \else
116   \catcode63=9 % ? (ignore)
117 \fi
118 ? \let\InCa@@TestMode\InCa@TestMode

```

2.3 Macros independent of ε -TeX

2.3.1 Abs, Sgn

\InCa@Abs

```

119 \def\InCa@Abs#1#2!{%
120   \ifx#1-%
121     #2%
122   \else
123     #1#2%
124   \fi
125 }

```

\InCa@Sgn

```

126 \def\InCa@Sgn#1#2!{%
127   \ifx#1-%
128     -1%
129   \else
130     \ifx#10%
131       0%
132     \else
133       1%
134     \fi
135   \fi
136 }

```

2.3.2 Min, Max, Cmp

`\InCa@Min`

```
137 \def\InCa@Min#1!#2!{%
138   \ifnum#1<#2 %
139     #1%
140   \else
141     #2%
142   \fi
143 }
```

`\InCa@Max`

```
144 \def\InCa@Max#1!#2!{%
145   \ifnum#1>#2 %
146     #1%
147   \else
148     #2%
149   \fi
150 }
```

`\InCa@Cmp`

```
151 \def\InCa@Cmp#1!#2!{%
152   \ifnum#1=#2 %
153     0%
154   \else
155     \ifnum#1<#2 %
156       -%
157     \fi
158     1%
159   \fi
160 }
```

2.3.3 Fac

`\InCa@Fac` It does not make much sense to calculate the faculty by an general algorithm. The allowed range of arguments is too low because of the limited integer domain.

```
161 \def\InCa@Fac#1!{%
162   \ifcase#1 1% 0!
163   \or 1% 1!
164   \or 2% 2!
165   \or 6% 3!
166   \or 24% 4!
167   \or 120% 5!
168   \or 720% 6!
169   \or 5040% 7!
170   \or 40320% 8!
171   \or 362880% 9!
172   \or 3628800% 10!
173   \or 39916800% 11!
174   \or 479001600% 12!
175   \else
176     \ifnum#1<\z@
177       0\IntCalcError:FacNegative%
178     \else
179       0\IntCalcError:FacOverflow%
180     \fi
181   \fi
182 }
```

2.4 Implementation based on ε -TeX

Only `\numexpr` is used from ε -TeX.

```
183 \begingroup\expandafter\expandafter\expandafter\endgroup
184 \expandafter\ifx\csname numexpr\endcsname\relax
185 \else
```

2.4.1 Num

```
\intcalcNum
186 \def\intcalcNum#1{%
187   \the\numexpr#1\relax
188 }%
```

2.4.2 Inv, Abs, Sgn

```
\intcalcInv
189 \def\intcalcInv#1{%
190   \number-\intcalcNum{#1} %
191 }%
```

```
\intcalcAbs
192 \def\intcalcAbs#1{%
193   \number\expandafter\InCa@Abs\the\numexpr#1! %
194 }%
```

```
\intcalcSgn
195 \def\intcalcSgn#1{%
196   \number\expandafter\InCa@Sgn\the\numexpr#1! %
197 }%
```

2.4.3 Min, Max, Cmp

```
\intcalcMin
198 \def\intcalcMin#1#2{%
199   \number\expandafter\InCa@Min
200   \the\numexpr#1\expandafter!%
201   \the\numexpr#2! %
202 }%
```

```
\intcalcMax
203 \def\intcalcMax#1#2{%
204   \number\expandafter\InCa@Max
205   \the\numexpr#1\expandafter!%
206   \the\numexpr#2! %
207 }%
```

```
\intcalcCmp
208 \def\intcalcCmp#1#2{%
209   \number\expandafter\InCa@Cmp
210   \the\numexpr#1\expandafter!\the\numexpr#2! %
211 }%
```

2.4.4 Inc, Dec

```
\intcalcInc
212 \def\intcalcInc#1{%
213   \the\numexpr#1+1\relax
214 }%
```

```
\intcalcDec
215 \def\intcalcDec#1{%
216   \the\numexpr#1-1\relax
217 }%
```

```
\IntCalcInc
218 \def\IntCalcInc#1!{%
219   \the\numexpr#1+1\relax
220 }%
```

```
\IntCalcDec
221 \def\IntCalcDec#1!{%
222   \the\numexpr#1-1\relax
223 }%
```

2.4.5 Add, Sub

```
\intcalcAdd
224 \def\intcalcAdd#1#2{%
225   \the\numexpr#1+(#2)\relax
226 }%
```

```
\intcalcSub
227 \def\intcalcSub#1#2{%
228   \the\numexpr#1-(#2)\relax
229 }%
```

```
\IntCalcAdd
230 \def\IntCalcAdd#1!#2!{%
231   \the\numexpr#1+#2\relax
232 }%
```

```
\IntCalcSub
233 \def\IntCalcSub#1!#2!{%
234   \the\numexpr#1-#2\relax
235 }%
```

2.4.6 Shl, Shr

```
\intcalcShl
236 \def\intcalcShl#1{%
237   \the\numexpr(#1)*2\relax
238 }%
```

```
\intcalcShr
239 \def\intcalcShr#1{%
240   \number\expandafter\InCa@Shr\the\numexpr#1! %
241 }%
```

```

\IntCalcShl
242 \def\IntCalcShl#1!{%
243 \the\numexpr#1*2\relax
244 }%

\IntCalcShr
245 \def\IntCalcShr#1!{%
246 \the\numexpr\ifodd#1 (#1-1)\else#1\fi/2\relax
247 }%

\InCa@Shr
248 \def\InCa@Shr#1#2!{%
249 \ifx#1-%
250 -\InCa@Shr#2!%
251 \else
252 \ifodd#1#2 %
253 \the\numexpr(#1#2-1)/2\relax
254 \else
255 \the\numexpr#1#2/2\relax
256 \fi
257 \fi
258 }%

```

2.4.7 Mul, Sqr, Fac

```

\intcalcMul
259 \def\intcalcMul#1#2{%
260 \the\numexpr(#1)*(#2)\relax
261 }%

\IntCalcMul
262 \def\IntCalcMul#1!#2!{%
263 \the\numexpr#1*#2\relax
264 }%

\intcalcSqr
265 \def\intcalcSqr#1{%
266 \number\expandafter\InCa@Sqr\the\numexpr#1! %
267 }%

\InCa@Sqr
268 \def\InCa@Sqr#1!{%
269 \the\numexpr#1*#1\relax
270 }%

\intcalcFac
271 \def\intcalcFac#1{%
272 \number\expandafter\InCa@Fac\the\numexpr#1! %
273 }%

```

2.4.8 Pow

```

\intcalcPow
274 \def\intcalcPow#1#2{%
275 \number\expandafter\InCa@Pow
276 \the\numexpr#1\expandafter!%
277 \the\numexpr#2! %
278 }%

```

\InCa@Pow

```
279 \def\InCa@Pow#1#2!#3#4!{%
280   \ifcase#3#4 % power = 0
281     1%
282   \or % power = 1
283     #1#2%
284   \or % power = 2
285     \the\numexpr#1#2*#1#2\relax
286   \else
287     \ifcase#1#2 % basis = 0, power <> 0
288       0%
289       \ifx#3-% power < 0
290         0\IntCalcError:DivisionByZero%
291       \fi
292     \or
293       1% basis = 1
294     \else
295       \ifnum#1#2=\m@ne % basis = -1
296         \ifodd#3#4 %
297           -%
298         \fi
299         1%
300       \else % |basis| > 1
301         \ifx#3-% power < 0
302           0%
303         \else % power > 2
304           \InCa@PowRec#1#2!#3#4!1!%
305         \fi
306       \fi
307     \fi
308   \fi
309 }%
```

\InCa@PowRec

```
Pow(b, p) {
  PowRec(b, p, 1)
}
PowRec(b, p, r) {
  if p == 1 then
    return r*b
  else
    ifodd p then
      return PowRec(b*b, (p-1)/2, r*b) % p div 2 = (p-1)/2
    else
      return PowRec(b*b, (p-1)/2, r)
    fi
  fi
}
310 \def\InCa@PowRec#1!#2!#3!{%
311   \ifnum#2=\@ne
312     \the\numexpr#1**3\relax
313   \else
314     \ifodd#2 %
315       \expandafter\InCa@PowRec
316       \the\numexpr#1*#1\expandafter!%
317       \the\numexpr(#2-1)/2\expandafter!%
318       \the\numexpr#1**3\expandafter\expandafter\expandafter!%
319     \else
320       \expandafter\InCa@PowRec
```

```

321     \the\numexpr#1*#1\expandafter!%
322     \the\numexpr(#2-1)/2\expandafter!%
323     \number#3\expandafter\expandafter\expandafter!%
324     \fi
325     \fi
326 }%

```

2.4.9 Div, Mod

TeX's `\divide` truncates, ϵ -TeX's `\numexpr` rounds the result of a division. The rounding method is called “Symmetric Arithmetic Rounding” or “Round-Half-Up” (“Kaufmännisches Runden” in German):

$$\begin{aligned}
 1 &= 3 \text{ divide } 2 = 1.5 = \text{numexpr } 3/2 = 2 \\
 -1 &= -3 \text{ divide } 2 = -1.5 = \text{numexpr } -3/2 = -2
 \end{aligned}$$

Macro `\intcalcDiv` follows TeX and truncates. The calculation is done by the following formula:

$$\text{Div}(X, Y) = (X - (Y - 1)/2)/Y \quad \text{for } X, Y > 0 \tag{1}$$

The operator ‘/’ is `\numexpr`'s division.

```

\intcalcDiv
327 \def\intcalcDiv#1#2{%
328   \number\expandafter\InCa@Div
329   \the\numexpr#1\expandafter!%
330   \the\numexpr#2! %
331 }%

\InCa@Div
332 \def\InCa@Div#1!#2!{%
333   \ifcase#2 %
334     0\IntCalcError:DivisionByZero%
335   \else
336     \ifcase#1 %
337       0%
338     \else
339       \expandafter\InCa@@Div
340       \romannumeral 0%
341       \ifnum#1<\z@
342         \expandafter-\number-#1%
343       \else
344         \expandafter+\number#1%
345       \fi
346       \expandafter!%
347       \romannumeral 0%
348       \ifnum#2<\z@
349         \expandafter-\number-#2%
350       \else
351         \expandafter+\number#2%
352       \fi
353     !%
354   \fi
355   \fi
356 }%

\IntCalcDiv

```

```

357 \def\InCa@Temp#1{%
358   \def\IntCalcDiv##1!##2!{%
359     \number
360     \ifcase##2 %
361       0\IntCalcError:DivisionByZero%
362     \else
363       \ifcase##1 %
364         0%
365       \else
366         \the\numexpr(##1-(##2-1)/2)/##2\relax
367       \fi
368     \fi
369     #1%
370   }%
371 }%
372 \InCa@Temp{ }%

\InCa@@Div
373 \def\InCa@@Div#1#2!#3#4!{%
374   #1#3%
375   \the\numexpr(#2-(#4-1)/2)/#4\relax
376 }%

\intcalcMod
377 \def\intcalcMod#1#2{%
378   \number\expandafter\InCa@Mod
379   \the\numexpr#1\expandafter!%
380   \the\numexpr#2! %
381 }%

\InCa@Mod
382 \def\InCa@Mod#1!#2!{%
383   \ifcase#2 %
384     0\IntCalcError:DivisionByZero%
385   \else
386     \ifcase#1 %
387       0%
388     \else
389       \expandafter\InCa@@Mod
390       \romannumeral 0%
391       \ifnum#1<\z@
392         \expandafter-\number-#1%
393       \else
394         \expandafter+\number#1%
395       \fi
396       \expandafter!%
397       \romannumeral 0%
398       \ifnum#2<\z@
399         \expandafter-\number-#2%
400       \else
401         \expandafter+\number#2%
402       \fi
403     !%
404   \fi
405   \fi
406 }%

\IntCalcMod

```

```

407 \def\InCa@Temp#1{%
408   \def\IntCalcMod##1!##2!{%
409     \number
410     \ifcase##2 %
411       0\IntCalcError:DivisionByZero%
412     \else
413       \ifcase##1 %
414         0%
415       \else
416         \the\numexpr##1-(##1-(##2-1)/2)/##2*##2\relax
417       \fi
418     \fi
419     #1%
420   }%
421 }%
422 \InCa@Temp{ }%

```

\InCa@@Mod

```

423 \def\InCa@@Mod#1#2!#3#4!{%
424   \if#3+%
425     \if#1+%
426       \the\numexpr#2-\InCa@@Div+#2!+#4!*#4\relax
427     \else
428       \expandafter\InCa@ModX
429       \the\numexpr-#2+\InCa@@Div+#2!+#4!*#4!#4!%
430     \fi
431   \else
432     -%
433     \if#1+%
434       \expandafter\InCa@ModX
435       \the\numexpr-#2+\InCa@@Div+#2!+#4!*#4!#4!%
436     \else
437       \the\numexpr#2-\InCa@@Div+#2!+#4!*#4\relax
438     \fi
439   \fi
440 }%

```

\InCa@ModX

```

441 \def\InCa@ModX#1!#2!{%
442   \ifcase#1 %
443     0%
444   \else
445     \the\numexpr#1+#2\relax
446   \fi
447 }%
448 \expandafter\InCa@AtEnd
449 \fi%

```

2.5 Implementation without ϵ -TeX

2.5.1 Num

\intcalcNum

```

450 \def\intcalcNum#1{%
451   \number\expandafter\InCa@FirstOfOne\number#1! %
452 }

```

2.5.2 Inv, Abs, Sgn

```
\intcalcInv
453 \def\intcalcInv#1{%
454   \number\expandafter\InCa@FirstOfOne\number-#1! %
455 }

\InCa@FirstOfOne
456 \def\InCa@FirstOfOne#1!{#1}

\intcalcAbs
457 \def\intcalcAbs#1{%
458   \number\expandafter\InCa@Abs\number#1! %
459 }

\intcalcSgn
460 \def\intcalcSgn#1{%
461   \number\expandafter\InCa@Sgn\number#1! %
462 }
```

2.5.3 Min, Max, Cmp

```
\intcalcMin
463 \def\intcalcMin#1#2{%
464   \number\expandafter\InCa@Min
465   \number\number#1\expandafter!\number#2! %
466 }

\intcalcMax
467 \def\intcalcMax#1#2{%
468   \number\expandafter\InCa@Max
469   \number\number#1\expandafter!\number#2! %
470 }

\intcalcCmp
471 \def\intcalcCmp#1#2{%
472   \number\expandafter\InCa@Cmp
473   \number\number#1\expandafter!\number#2! %
474 }%
```

2.5.4 Inc, Dec

```
\intcalcInc
475 \def\intcalcInc#1{%
476   \number\expandafter\InCa@IncSwitch\number#1! %
477 }

\InCa@IncSwitch
478 \def\InCa@IncSwitch#1#2!{%
479   \ifx#1-%
480     -%
481     \csname InCa@Empty%
482     \InCa@Dec#2!%
483   \else
484     \csname InCa@Empty%
485     \InCa@Inc#1#2!%
486   \fi
487 }
```

```

\intcalcDec
488 \def\intcalcDec#1{%
489 \number\expandafter\InCa@DecSwitch\number#1! %
490 }

\InCa@DecSwitch
491 \def\InCa@DecSwitch#1#2!{%
492 \ifx#1-%
493 -%
494 \csname InCa@Empty%
495 \expandafter\InCa@Inc#2!%
496 \else
497 \ifx#10%
498 -1%
499 \else
500 \csname InCa@Empty%
501 \InCa@Dec#1#2!%
502 \fi
503 \fi
504 }

\IntCalcInc
505 \def\IntCalcInc#1!{%
506 \number\csname InCa@Empty\InCa@Inc#1! %
507 }

\IntCalcDec
508 \def\IntCalcDec#1!{%
509 \number\csname InCa@Empty\InCa@Dec#1! %
510 }

\InCa@Inc
511 \def\InCa@Inc#1#2{%
512 \ifx#2!%
513 \csname InCa@IncDigit#1\endcsname1%
514 \else
515 \csname InCa@IncDigit#1%
516 \expandafter\InCa@Inc\expandafter#2%
517 \fi
518 }

\InCa@IncDigit[0-8]
519 \def\InCa@Temp#1#2{%
520 \expandafter\def\csname InCa@IncDigit#1\endcsname##1{%
521 \endcsname
522 0%
523 \ifcase##1 %
524 #1%
525 \else
526 #2%
527 \fi
528 }%
529 }
530 \InCa@Temp 01
531 \InCa@Temp 12
532 \InCa@Temp 23
533 \InCa@Temp 34

```

```
534 \InCa@Temp 45
535 \InCa@Temp 56
536 \InCa@Temp 67
537 \InCa@Temp 78
538 \InCa@Temp 89
```

\InCa@IncDigit9

```
539 \expandafter\def\csname InCa@IncDigit9\endcsname#1{%
540 \expandafter\endcsname
541 \ifcase#1 %
542 09%
543 \else
544 10%
545 \fi
546 }
```

\InCa@Dec

```
547 \def\InCa@Dec#1#2{%
548 \ifx#2!%
549 \csname InCa@DecDigit#1\endcsname1%
550 \else
551 \csname InCa@DecDigit#1%
552 \expandafter\InCa@Dec\expandafter#2%
553 \fi
554 }
```

\InCa@DecDigit[1-9]

```
555 \def\InCa@Temp#1#2{%
556 \expandafter\def\csname InCa@DecDigit#1\endcsname##1{%
557 \endcsname
558 0%
559 \ifcase##1 %
560 #1%
561 \else
562 #2%
563 \fi
564 }%
565 }
566 \InCa@Temp 98
567 \InCa@Temp 87
568 \InCa@Temp 76
569 \InCa@Temp 65
570 \InCa@Temp 54
571 \InCa@Temp 43
572 \InCa@Temp 32
573 \InCa@Temp 21
574 \InCa@Temp 10
```

\InCa@DecDigit0

```
575 \expandafter\def\csname InCa@DecDigit0\endcsname#1{%
576 \expandafter\endcsname
577 \ifcase#1 %
578 00%
579 \else
580 19%
581 \fi
582 }
```

2.5.5 Add, Sub

`\intcalcAdd`

```
583 \def\intcalcAdd#1#2{%
584   \number
585   \expandafter\InCa@AddSwitch
586   \number\number#1\expandafter!%
587   \number#2! %
588 }
```

`\intcalcSub`

```
589 \def\intcalcSub#1#2{%
590   \number
591   \expandafter\InCa@AddSwitch
592   \number\number#1\expandafter!%
593   \number-\number#2! %
594 }
```

`\InCa@AddSwitch` Decision table for `\InCa@AddSwitch`. The sign of negative numbers can be removed by a simple `\@gobble` instead of the more expensive `\number-`.

$x < 0$	$y < 0$	$x < y$	–	$\text{Add}(-x, -y)$
		else		$\text{Add}(-y, -x)$
	else	$-x > y$	–	$\text{Sub}(-x, y)$
			+	$\text{Sub}(y, -x)$
else	$y < 0$	$x > -y$	+	$\text{Sub}(x, -y)$
		else	–	$\text{Sub}(-y, x)$
	else	$x > y$	+	$\text{Add}(x, y)$
				$\text{Add}(y, x)$

```
595 \def\InCa@AddSwitch#1!#2!{%
596   \ifnum#1<\z@
597     \ifnum#2<\z@
598       -%
599       \ifnum#1<#2 %
600         \expandafter\InCa@Add\number-#1\expandafter!%
601         \@gobble#2!%
602     \else
603       \expandafter\InCa@Add\number-#2\expandafter!%
604       \@gobble#1!%
605     \fi
606   \else
607     \ifnum-#1>#2 %
608       -%
609       \expandafter\InCa@Sub\@gobble#1!#2!%
610     \else
611       \expandafter\InCa@Sub\number#2\expandafter!%
612       \@gobble#1!%
613     \fi
614   \fi
615 \else
616   \ifnum#2<\z@
617     \ifnum#1>-#2 %
618       \expandafter\InCa@Sub\number#1\expandafter!%
619       \@gobble#2!%
620     \else
621       -%
```

```

622     \expandafter\InCa@Sub\@gobble#2!#1!%
623     \fi
624   \else
625     \ifnum#1>#2 %
626       \InCa@Add#1!#2!%
627     \else
628       \InCa@Add#2!#1!%
629     \fi
630   \fi
631 \fi
632 }

\IntCalcAdd
633 \def\IntCalcAdd#1!#2!{%
634   \number\InCa@Add#1!#2! %
635 }

\IntCalcSub
636 \def\IntCalcSub#1!#2!{%
637   \number\InCa@Sub#1!#2! %
638 }

\InCa@Space
639 \begingroup
640   \def\x#1{\endgroup
641     \let\InCa@Space= #1%
642   }%
643 \x{ }

\InCa@Add
644 \def\InCa@Add#1!#2!{%
645   \ifcase#2 %
646     #1%
647   \else
648     \InCa@@Add#1!#2!00000000\InCa@Space
649   \fi
650 }

\InCa@Sub
651 \def\InCa@Sub#1!#2!{%
652   \ifnum#1=#2 %
653     0%
654   \else
655     \InCa@@Sub#1!#2!00000000\InCa@Space
656   \fi
657 }

\InCa@@Add
658 \def\InCa@@Add#1!#2#3!{%
659   \ifx\InCa@Empty#3\InCa@Empty
660     \@ReturnAfterElseFi{%
661       \InCa@@@Add!!#1!#2%
662     }%
663   \else
664     \@ReturnAfterFi{%
665       \InCa@@@Add#1!#3!#2%
666     }%
667   \fi
668 }

```

```

\InCa@@Sub
669 \def\InCa@@Sub#1!#2#3!{%
670   \ifx\InCa@Empty#3\InCa@Empty
671     \@ReturnAfterElseFi{%
672       \InCa@@Sub!!#1!#2%
673     }%
674   \else
675     \@ReturnAfterFi{%
676       \InCa@@Sub#1!#3!#2%
677     }%
678   \fi
679 }

\InCa@@@Add
680 \def\InCa@@@Add#1!#2!#3#4!#5{%
681   \ifx\InCa@Empty#4\InCa@Empty
682     \csname InCa@Empty%
683     \@ReturnAfterElseFi{%
684       \InCa@ProcessAdd#1#3!#5#2%
685     }%
686   \else
687     \@ReturnAfterFi{%
688       \InCa@@@Add#1#3!#5#2!#4!%
689     }%
690   \fi
691 }

\InCa@@@Sub
692 \def\InCa@@@Sub#1!#2!#3#4!#5{%
693   \ifx\InCa@Empty#4\InCa@Empty
694     \csname @gobble%
695     \@ReturnAfterElseFi{%
696       \InCa@ProcessSub#1#3!#5#2%
697     }%
698   \else
699     \@ReturnAfterFi{%
700       \InCa@@@Sub#1#3!#5#2!#4!%
701     }%
702   \fi
703 }

\InCa@ProcessAdd
704 \def\InCa@ProcessAdd#1#2!#3#4{%
705   \ifx\InCa@Empty#2\InCa@Empty
706     \csname InCa@AddDigit#1\endcsname#3%
707     \romannumeral0#4%
708   \else
709     \csname InCa@AddDigit#1\csname InCa@DigitCarry#3%
710     \@ReturnAfterFi{%
711       \InCa@ProcessAdd#2!#4%
712     }%
713   \fi
714 }

\InCa@ProcessSub
715 \def\InCa@ProcessSub#1#2!#3#4{%
716   \ifx\InCa@Empty#2\InCa@Empty
717     \csname InCa@SubDigit#1\endcsname#3%

```

```

718   \romannumeral0#4%
719   \else
720   \csname InCa@SubDigit#1\csname InCa@DigitCarry#3%
721   \@ReturnAfterFi{%
722   \InCa@ProcessSub#2!#4%
723   }%
724   \fi
725 }

```

\InCa@DigitCarry[0-9]

```

726 \def\InCa@Temp#1#2{%
727   \expandafter\def\csname InCa@DigitCarry#1\endcsname##1{%
728     \ifcase##1 %
729       \endcsname#1%
730     \else
731       \endcsname#2%
732     \fi
733   }%
734 }
735 \InCa@Temp 01
736 \InCa@Temp 12
737 \InCa@Temp 23
738 \InCa@Temp 34
739 \InCa@Temp 45
740 \InCa@Temp 56
741 \InCa@Temp 67
742 \InCa@Temp 78
743 \InCa@Temp 89
744 \InCa@Temp 9{{10}}

```

\InCa@AddDigit0

```

745 \expandafter\def\csname InCa@AddDigit0\endcsname#1{%
746   \ifnum#1>9 %
747     \endcsname10%
748   \else
749     \endcsname0#1%
750   \fi
751 }

```

\InCa@AddDigit[1-9]

```

752 \def\InCa@Temp#1#2#3{%
753   \expandafter\def\csname InCa@AddDigit#1\endcsname##1{%
754     \ifnum##1>#2 %
755       \endcsname 1%
756     \else
757       \endcsname 0%
758     \fi
759     \ifcase##1 #1% 0
760     #3%
761     \else #1% 10
762     \fi
763   }%
764 }
765 \InCa@Temp 18{%
766   \or 2% 1
767   \or 3% 2
768   \or 4% 3
769   \or 5% 4

```

770 \or 6% 5
771 \or 7% 6
772 \or 8% 7
773 \or 9% 8
774 \or 0% 9
775 }%
776 \InCa@Temp 27{ %
777 \or 3% 1
778 \or 4% 2
779 \or 5% 3
780 \or 6% 4
781 \or 7% 5
782 \or 8% 6
783 \or 9% 7
784 \or 0% 8
785 \or 1% 9
786 }%
787 \InCa@Temp 36{ %
788 \or 4% 1
789 \or 5% 2
790 \or 6% 3
791 \or 7% 4
792 \or 8% 5
793 \or 9% 6
794 \or 0% 7
795 \or 1% 8
796 \or 2% 9
797 }%
798 \InCa@Temp 45{ %
799 \or 5% 1
800 \or 6% 2
801 \or 7% 3
802 \or 8% 4
803 \or 9% 5
804 \or 0% 6
805 \or 1% 7
806 \or 2% 8
807 \or 3% 9
808 }%
809 \InCa@Temp 54{ %
810 \or 6% 1
811 \or 7% 2
812 \or 8% 3
813 \or 9% 4
814 \or 0% 5
815 \or 1% 6
816 \or 2% 7
817 \or 3% 8
818 \or 4% 9
819 }%
820 \InCa@Temp 63{ %
821 \or 7% 1
822 \or 8% 2
823 \or 9% 3
824 \or 0% 4
825 \or 1% 5
826 \or 2% 6
827 \or 3% 7

```

828 \or 4% 8
829 \or 5% 9
830 }%
831 \InCa@Temp 72{%
832 \or 8% 1
833 \or 9% 2
834 \or 0% 3
835 \or 1% 4
836 \or 2% 5
837 \or 3% 6
838 \or 4% 7
839 \or 5% 8
840 \or 6% 9
841 }%
842 \InCa@Temp 81{%
843 \or 9% 1
844 \or 0% 2
845 \or 1% 3
846 \or 2% 4
847 \or 3% 5
848 \or 4% 6
849 \or 5% 7
850 \or 6% 8
851 \or 7% 9
852 }%
853 \InCa@Temp 90{%
854 \or 0% 1
855 \or 1% 2
856 \or 2% 3
857 \or 3% 4
858 \or 4% 5
859 \or 5% 6
860 \or 6% 7
861 \or 7% 8
862 \or 8% 9
863 }%

```

\InCa@SubDigit[0-9]

```

864 \def\InCa@Temp#1#2{%
865 \expandafter\def\csname InCa@SubDigit#1\endcsname##1{%
866 \ifnum##1>#1 %
867 \endcsname 1%
868 \else
869 \endcsname 0%
870 \fi
871 \ifcase##1 #1% 0
872 #2%
873 \else #1% 10
874 \fi
875 }%
876 }
877 \InCa@Temp 0{%
878 \or 9% 1
879 \or 8% 2
880 \or 7% 3
881 \or 6% 4
882 \or 5% 5
883 \or 4% 6

```

```
884 \or 3% 7
885 \or 2% 8
886 \or 1% 9
887 }
888 \InCa@Temp 1{%
889 \or 0% 1
890 \or 9% 2
891 \or 8% 3
892 \or 7% 4
893 \or 6% 5
894 \or 5% 6
895 \or 4% 7
896 \or 3% 8
897 \or 2% 9
898 }
899 \InCa@Temp 2{%
900 \or 1% 1
901 \or 0% 2
902 \or 9% 3
903 \or 8% 4
904 \or 7% 5
905 \or 6% 6
906 \or 5% 7
907 \or 4% 8
908 \or 3% 9
909 }
910 \InCa@Temp 3{%
911 \or 2% 1
912 \or 1% 2
913 \or 0% 3
914 \or 9% 4
915 \or 8% 5
916 \or 7% 6
917 \or 6% 7
918 \or 5% 8
919 \or 4% 9
920 }
921 \InCa@Temp 4{%
922 \or 3% 1
923 \or 2% 2
924 \or 1% 3
925 \or 0% 4
926 \or 9% 5
927 \or 8% 6
928 \or 7% 7
929 \or 6% 8
930 \or 5% 9
931 }
932 \InCa@Temp 5{%
933 \or 4% 1
934 \or 3% 2
935 \or 2% 3
936 \or 1% 4
937 \or 0% 5
938 \or 9% 6
939 \or 8% 7
940 \or 7% 8
941 \or 6% 9
```

```

942 }
943 \InCa@Temp 6{
944 \or 5% 1
945 \or 4% 2
946 \or 3% 3
947 \or 2% 4
948 \or 1% 5
949 \or 0% 6
950 \or 9% 7
951 \or 8% 8
952 \or 7% 9
953 }
954 \InCa@Temp 7{
955 \or 6% 1
956 \or 5% 2
957 \or 4% 3
958 \or 3% 4
959 \or 2% 5
960 \or 1% 6
961 \or 0% 7
962 \or 9% 8
963 \or 8% 9
964 }
965 \InCa@Temp 8{
966 \or 7% 1
967 \or 6% 2
968 \or 5% 3
969 \or 4% 4
970 \or 3% 5
971 \or 2% 6
972 \or 1% 7
973 \or 0% 8
974 \or 9% 9
975 }
976 \InCa@Temp 9{
977 \or 8% 1
978 \or 7% 2
979 \or 6% 3
980 \or 5% 4
981 \or 4% 5
982 \or 3% 6
983 \or 2% 7
984 \or 1% 8
985 \or 0% 9
986 }

```

2.5.6 Shl, Shr

\intcalcShl

```

987 \def\intcalcShl#1{
988 \number\expandafter\InCa@ShlSwitch\number#1! %
989 }

```

\InCa@ShlSwitch

```

990 \def\InCa@ShlSwitch#1#2!{
991 \ifx#1-%
992 -\csname InCa@Empty%
993 \InCa@Shl#2!%

```

```

994 \else
995 \csname InCa@Empty%
996 \InCa@Shl#1#2!%
997 \fi
998 }

\IntCalcShl
999 \def\IntCalcShl#1!{%
1000 \number
1001 \csname InCa@Empty%
1002 \InCa@Shl#1! %
1003 }

\IntCal@ShlDigit
1004 \def\InCa@Shl#1#2{%
1005 \ifx#2!%
1006 \csname InCa@ShlDigit#1\endcsname0%
1007 \else
1008 \csname InCa@ShlDigit#1%
1009 \@ReturnAfterFi{%
1010 \InCa@Shl#2%
1011 }%
1012 \fi
1013 }

\InCa@ShlDigit0
1014 \expandafter\def\csname InCa@ShlDigit0\endcsname{%
1015 \endcsname0%
1016 }

\InCa@ShlDigit[1-9]
1017 \def\InCa@Temp#1#2#3#4#5{%
1018 \expandafter\def\csname InCa@ShlDigit#1\endcsname##1{%
1019 \expandafter\endcsname
1020 \ifcase##1 %
1021 #2#3%
1022 \else
1023 #4#5%
1024 \fi
1025 }%
1026 }
1027 \InCa@Temp 10203
1028 \InCa@Temp 20405
1029 \InCa@Temp 30607
1030 \InCa@Temp 40809
1031 \InCa@Temp 51011
1032 \InCa@Temp 61213
1033 \InCa@Temp 71415
1034 \InCa@Temp 81617
1035 \InCa@Temp 91819

\intcalcShr
1036 \def\intcalcShr#1{%
1037 \number\expandafter\InCa@ShrSwitch\number#1! %
1038 }

\InCa@ShrSwitch

```

```

1039 \def\InCa@ShrSwitch#1#2!{%
1040   \ifx#1-%
1041     -\InCa@Shr#2!%
1042   \else
1043     \InCa@Shr#1#2!%
1044   \fi
1045 }

\IntCalcShr
1046 \def\IntCalcShr#1!{%
1047   \number\InCa@Shr#1! %
1048 }

\InCa@Shr
1049 \def\InCa@Shr#1#2{%
1050   \InCa@ShrDigit#1!%
1051   \ifx#2!%
1052   \else
1053     \@ReturnAfterFi{%
1054       \ifodd#1 %
1055         \@ReturnAfterElseFi{%
1056           \InCa@Shr{1#2}%
1057         }%
1058       \else
1059         \expandafter\InCa@Shr\expandafter#2%
1060       \fi
1061     }%
1062   \fi
1063 }

1064 \def\InCa@ShrDigit#1!{%
1065   \ifcase#1 0% 0
1066   \or 0% 1
1067   \or 1% 2
1068   \or 1% 3
1069   \or 2% 4
1070   \or 2% 5
1071   \or 3% 6
1072   \or 3% 7
1073   \or 4% 8
1074   \or 4% 9
1075   \or 5% 10
1076   \or 5% 11
1077   \or 6% 12
1078   \or 6% 13
1079   \or 7% 14
1080   \or 7% 15
1081   \or 8% 16
1082   \or 8% 17
1083   \or 9% 18
1084   \or 9% 19
1085   \fi
1086 }

```

2.5.7 \InCa@Tim

\InCa@Tim Macro \InCa@Tim implements “Number *times* digit”.

```

1087 \def\InCa@Temp#1{%

```

```

1088 \def\InCa@Tim##1##2{%
1089   \number
1090   \ifcase##2 % 0
1091     0%
1092   \or % 1
1093     ##1%
1094   \else % 2-9
1095     \csname InCa@Empty%
1096       \InCa@ProcessTim##2##1!%
1097     \fi
1098   #1%
1099 }%
1100 }
1101 \InCa@Temp{ }

```

\InCa@ProcessTim

```

1102 \def\InCa@ProcessTim#1#2#3{%
1103   \ifx#3!%
1104     \csname InCa@TimDigit#2\endcsname#10%
1105   \else
1106     \csname InCa@TimDigit#2\csname InCa@Param#1%
1107       \@ReturnAfterFi{%
1108         \InCa@ProcessTim#1#3%
1109       }%
1110   \fi
1111 }

```

\InCa@Param[0-9]

```

1112 \def\InCa@Temp#1{%
1113   \expandafter\def\csname InCa@Param#1\endcsname{%
1114     \endcsname#1%
1115   }%
1116 }
1117 \InCa@Temp 0%
1118 \InCa@Temp 1%
1119 \InCa@Temp 2%
1120 \InCa@Temp 3%
1121 \InCa@Temp 4%
1122 \InCa@Temp 5%
1123 \InCa@Temp 6%
1124 \InCa@Temp 7%
1125 \InCa@Temp 8%
1126 \InCa@Temp 9%

```

\InCa@TimDigit0

```

1127 \expandafter\def\csname InCa@TimDigit0\endcsname#1#2{%
1128   \endcsname
1129   0#2%
1130 }

```

\InCa@TimDigit1

```

1131 \expandafter\def\csname InCa@TimDigit1\endcsname#1#2{%
1132   \ifcase#2 %
1133     \endcsname 0#1%
1134   \else
1135     \csname InCa@AddDigit#1\endcsname #2%
1136   \fi
1137 }

```

\InCa@TimDigit[2-9]

```
1138 \def\InCa@Temp#1#2{%
1139   \expandafter\def\csname InCa@TimDigit#1\endcsname##1{%
1140     \expandafter\InCa@TimDigitCarry
1141     \number
1142     \ifcase##1 0% 0
1143     #2%
1144     \fi
1145     !%
1146   }%
1147 }
1148 \InCa@Temp 2{%
1149   \or 2% 1
1150   \or 4% 2
1151   \or 6% 3
1152   \or 8% 4
1153   \or 10% 5
1154   \or 12% 6
1155   \or 14% 7
1156   \or 16% 8
1157   \or 18% 9
1158 }
1159 \InCa@Temp 3{%
1160   \or 3% 1
1161   \or 6% 2
1162   \or 9% 3
1163   \or 12% 4
1164   \or 15% 5
1165   \or 18% 6
1166   \or 21% 7
1167   \or 24% 8
1168   \or 27% 9
1169 }
1170 \InCa@Temp 4{%
1171   \or 4% 1
1172   \or 8% 2
1173   \or 12% 3
1174   \or 16% 4
1175   \or 20% 5
1176   \or 24% 6
1177   \or 28% 7
1178   \or 32% 8
1179   \or 36% 9
1180 }
1181 \InCa@Temp 5{%
1182   \or 5% 1
1183   \or 10% 2
1184   \or 15% 3
1185   \or 20% 4
1186   \or 25% 5
1187   \or 30% 6
1188   \or 35% 7
1189   \or 40% 8
1190   \or 45% 9
1191 }
1192 \InCa@Temp 6{%
1193   \or 6% 1
1194   \or 12% 2
```

```

1195 \or 18% 3
1196 \or 24% 4
1197 \or 30% 5
1198 \or 36% 6
1199 \or 42% 7
1200 \or 48% 8
1201 \or 54% 9
1202 }
1203 \InCa@Temp 7{%
1204 \or 7% 1
1205 \or 14% 2
1206 \or 21% 3
1207 \or 28% 4
1208 \or 35% 5
1209 \or 42% 6
1210 \or 49% 7
1211 \or 56% 8
1212 \or 63% 9
1213 }
1214 \InCa@Temp 8{%
1215 \or 8% 1
1216 \or 16% 2
1217 \or 24% 3
1218 \or 32% 4
1219 \or 40% 5
1220 \or 48% 6
1221 \or 56% 7
1222 \or 64% 8
1223 \or 72% 9
1224 }
1225 \InCa@Temp 9{%
1226 \or 9% 1
1227 \or 18% 2
1228 \or 27% 3
1229 \or 36% 4
1230 \or 45% 5
1231 \or 54% 6
1232 \or 63% 7
1233 \or 72% 8
1234 \or 81% 9
1235 }

```

\InCa@TimDigitCarry

```

1236 \def\InCa@TimDigitCarry#1!{%
1237 \ifnum#1<10 %
1238 \csname InCa@AddDigit#1\expandafter\endcsname
1239 \else
1240 \@ReturnAfterFi{%
1241 \InCa@@TimDigitCarry#1!%
1242 }%
1243 \fi
1244 }

```

\InCa@@TimDigitCarry

```

1245 \def\InCa@@TimDigitCarry#1#2!#3{%
1246 \csname InCa@DigitCarry#1%
1247 \csname InCa@AddDigit#2\endcsname #3%
1248 }

```

2.5.8 Mul

`\intcalcMul`

```

1249 \def\intcalcMul#1#2{%
1250   \number
1251   \expandafter\InCa@MulSwitch
1252   \number\number#1\expandafter!%
1253   \number#2! %
1254 }

```

`\InCa@MulSwitch` Decision table for `\InCa@MulSwitch`.

$x < 0$	$y < 0$	$x < y$	+	$\text{Mul}(-x, -y)$
		else		$\text{Mul}(-y, -x)$
	else	$-x > y$	-	$\text{Mul}(-x, y)$
		else		$\text{Mul}(y, -x)$
else	$y < 0$	$x > -y$	-	$\text{Mul}(x, -y)$
		else		$\text{Mul}(-y, x)$
	else	$x > y$	+	$\text{Mul}(x, y)$
		else		$\text{Mul}(y, x)$

```

1255 \def\InCa@MulSwitch#1!#2!{%
1256   \ifnum#1<\z@
1257     \ifnum#2<\z@
1258       \ifnum#1<#2 %
1259         \expandafter\InCa@Mul\number-#1\expandafter!%
1260         \@gobble#2!%
1261       \else
1262         \expandafter\InCa@Mul\number-#2\expandafter!%
1263         \@gobble#1!%
1264       \fi
1265     \else
1266       -%
1267       \ifnum-#1>#2 %
1268         \expandafter\InCa@Mul\@gobble#1!#2!%
1269       \else
1270         \expandafter\InCa@Mul\number#2\expandafter!%
1271         \@gobble#1!%
1272       \fi
1273     \fi
1274   \else
1275     \ifnum#2<\z@
1276       -%
1277       \ifnum#1>-#2 %
1278         \expandafter\InCa@Mul\number#1\expandafter!%
1279         \@gobble#2!%
1280       \else
1281         \expandafter\InCa@Mul\@gobble#2!#1!%
1282       \fi
1283     \else
1284       \ifnum#1>#2 %
1285         \InCa@Mul#1!#2!%
1286       \else
1287         \InCa@Mul#2!#1!%
1288       \fi
1289     \fi
1290   \fi
1291 }

```

\IntCalcMul

```
1292 \def\IntCalcMul#1!#2!{%
1293   \number\InCa@Mul#1!#2! %
1294 }
```

\InCa@Mul

```
1295 \def\InCa@Mul#1!#2!{%
1296   \ifcase#2 %
1297     0%
1298   \or
1299     #1%
1300   \or
1301     \csname InCa@Empty%
1302       \expandafter\InCa@Shl#1!%
1303   \else
1304     \ifnum#2<10 %
1305       \InCa@Tim{#1}#2%
1306     \else
1307       \InCa@ProcessMul!#2!#1!%
1308     \fi
1309   \fi
1310 }
```

\InCa@Mul

```
1311 \def\InCa@ProcessMul#1!#2#3!#4!{%
1312   \ifx\InCa@Empty#3\InCa@Empty
1313     \expandafter\InCa@Add\number
1314     #10\expandafter\expandafter\expandafter!%
1315     \InCa@Tim{#4}#2!%
1316   \else
1317     \ifx\InCa@Empty#1\InCa@Empty
1318       \expandafter\expandafter\expandafter\InCa@ProcessMul
1319       \InCa@Tim{#4}#2!%
1320       #3!#4!%
1321     \else
1322       \expandafter\InCa@ProcessMul\number
1323       \expandafter\InCa@Add\number%
1324       #10\expandafter\expandafter\expandafter!%
1325       \InCa@Tim{#4}#2!!%
1326       #3!#4!%
1327     \fi
1328   \fi
1329 }
```

2.5.9 Sqr, Fac

\intcalcSqr

```
1330 \def\intcalcSqr#1{%
1331   \number\expandafter\InCa@Sqr\number#1! %
1332 }
```

\InCa@Sqr

```
1333 \def\InCa@Sqr#1#2!{%
1334   \ifx#1-%
1335     \InCa@Mul#2!#2!%
1336   \else
1337     \InCa@Mul#1#2!#1#2!%

```

```

1338 \fi
1339 }

```

```
\intcalcFac
```

```

1340 \def\intcalcFac#1{%
1341 \number\expandafter\InCa@Fac\number#1! %
1342 }

```

2.5.10 Pow

```
\intcalcPow
```

```

1343 \def\intcalcPow#1#2{%
1344 \number\expandafter\InCa@Pow
1345 \number\number#1\expandafter!%
1346 \number#2! %
1347 }

```

```
\InCa@Pow
```

```

1348 \def\InCa@Pow#1#2!#3#4!{%
1349 \ifcase#3#4 % power = 0
1350 1%
1351 \or % power = 1
1352 #1#2%
1353 \or % power = 2
1354 \ifx#1-%
1355 \InCa@Mul#2!#2!%
1356 \else
1357 \InCa@Mul#1#2!#1#2!%
1358 \fi
1359 \else
1360 \ifcase#1#2 % basis = 0, power <> 0
1361 0%
1362 \ifx#3-% power < 0
1363 0\IntCalcError:DivisionByZero%
1364 \fi
1365 \or
1366 1% basis = 1
1367 \else
1368 \ifnum#1#2=\m@ne % basis = -1
1369 \ifodd#3#4 %
1370 -%
1371 \fi
1372 1%
1373 \else % |basis| > 1
1374 \ifx#3-% power < 0
1375 0%
1376 \else % power > 2
1377 \ifx#1-% basis < 0
1378 \ifodd#3#4 %
1379 -%
1380 \fi
1381 \InCa@PowRec#2!#3#4!1!%
1382 \else
1383 \InCa@PowRec#1#2!#3#4!1!%
1384 \fi
1385 \fi
1386 \fi
1387 \fi

```

```

1388 \fi
1389 }

\InCa@PowRec Pow(b, p) {
    PowRec(b, p, 1)
}
PowRec(b, p, r) {
    if p == 1 then
        return r
    else
        ifodd p then
            return PowRec(b*b, p div 2, r*b) % p div 2 = (p-1)/2
        else
            return PowRec(b*b, p div 2, r)
        fi
    fi
}

1390 \def\InCa@PowRec#1!#2!#3!{%
1391 \ifnum#2=\@ne
1392 \ifnum#1>#3 %
1393 \InCa@Mul#1!#3!%
1394 \else
1395 \InCa@Mul#3!#1!%
1396 \fi
1397 \else
1398 \expandafter\InCa@PowRec
1399 \number\InCa@Mul#1!#1!\expandafter!%
1400 \number\intcalcShr{#2}\expandafter!%
1401 \number
1402 \ifodd#2 %
1403 \ifnum#1>#3 %
1404 \InCa@Mul#1!#3!%
1405 \else
1406 \InCa@Mul#3!#1!%
1407 \fi
1408 \else
1409 #3%
1410 \fi
1411 \expandafter!%
1412 \fi
1413 }

```

2.5.11 Div

```

\intcalcDiv
1414 \def\intcalcDiv#1#2{%
1415 \number\expandafter\InCa@Div
1416 \number\number#1\expandafter!%
1417 \number#2! %
1418 }

\InCa@Div
1419 \def\InCa@Div#1!#2!{%
1420 \ifcase#2 %
1421 0\IntCalcError:DivisionByZero%
1422 \else
1423 \ifcase#1 %
1424 0%

```

```

1425 \else
1426 \expandafter\InCa@DivSwitch
1427 \number#1\expandafter!%
1428 \number#2!%
1429 \fi
1430 \fi
1431 }

```

\IntCalcDiv

```

1432 \def\InCa@Temp#1{%
1433 \def\IntCalcDiv##1!##2!{%
1434 \number
1435 \ifcase##2 %
1436 0\IntCalcError:DivisionByZero%
1437 \else
1438 \ifcase##1 %
1439 0%
1440 \else
1441 \InCa@@Div##1!##2!%
1442 \fi
1443 \fi
1444 #1%
1445 }%
1446 }
1447 \InCa@Temp{ }%

```

\InCa@DivSwitch Decision table for \InCa@DivSwitch.

$x < 0$	$y < 0$	+	$\text{Div}(-x, -y)$
	else	-	$\text{Div}(-x, y)$
else	$y < 0$	-	$\text{Div}(x, -y)$
	else	+	$\text{Div}(x, y)$

```

1448 \def\InCa@DivSwitch#1!#2!{%
1449 \ifnum#1<\z@
1450 \ifnum#2<\z@
1451 \expandafter\InCa@@Div\number-#1\expandafter!%
1452 \@gobble#2!%
1453 \else
1454 -%
1455 \expandafter\InCa@@Div\@gobble#1!#2!%
1456 \fi
1457 \else
1458 \ifnum#2<\z@
1459 -%
1460 \expandafter\InCa@@Div\number#1\expandafter!%
1461 \@gobble#2!%
1462 \else
1463 \InCa@@Div#1!#2!%
1464 \fi
1465 \fi
1466 }

```

\InCa@@Div

```

1467 \def\InCa@@Div#1!#2!{%
1468 \ifnum#1>#2 %
1469 \ifcase#2 % 0 already caught
1470 ? \IntCalcError:ThisCannotHappen%

```

```

1471 \or % 1
1472 #1%
1473 \or % 2
1474 \InCa@Shr#1!%
1475 \else
1476 \InCa@DivStart!#1!#2!#2!%
1477 \fi
1478 \else
1479 \ifnum#1=#2 %
1480 1%
1481 \else
1482 0%
1483 \fi
1484 \fi
1485 }

\InCa@DivStart
1486 \def\InCa@DivStart#1!#2#3!#4#5{%
1487 \ifx#5!%
1488 \@ReturnAfterElseFi{%
1489 \InCa@DivStartI{#1#2}#3=!%
1490 }%
1491 \else
1492 \@ReturnAfterFi{%
1493 \InCa@DivStart{#1#2}!#3!#5%
1494 }%
1495 \fi
1496 }

\InCa@StartI
1497 \def\InCa@DivStartI#1!#2!{%
1498 \expandafter\InCa@DivStartII
1499 \number#2\expandafter\expandafter\expandafter!%
1500 \intcalcShl{#2}!%
1501 #1!%
1502 }

\InCa@StartII
1503 \def\InCa@DivStartII#1!#2!{%
1504 \expandafter\InCa@DivStartIII
1505 \number#1\expandafter!%
1506 \number#2\expandafter\expandafter\expandafter!%
1507 \intcalcShl{#2}!%
1508 }

\InCa@StartIII
1509 \def\InCa@DivStartIII#1!#2!#3!{%
1510 \expandafter\InCa@DivStartIV
1511 \number#1\expandafter!%
1512 \number#2\expandafter!%
1513 \number#3\expandafter!%
1514 \number\InCa@Add#3!#2!\expandafter\expandafter\expandafter!%
1515 \intcalcShl{#3}!%
1516 }

\InCa@StartIV
1517 \def\InCa@DivStartIV#1!#2!#3!#4!#5!#6!{%
1518 \InCa@ProcessDiv#6!#1!#2!#3!#4!#5!/%
1519 }

```

\InCa@ProcessDiv

```
1520 \def\InCa@ProcessDiv#1#2#3!#4!#5!#6!#7!#8!#9/{%
1521 #9%
1522 \ifnum#1<#4 % 0
1523 0%
1524 \ifx#2=%
1525 \else
1526 \InCa@ProcessDiv{#1#2}#3!#4!#5!#6!#7!#8!%
1527 \fi
1528 \else % 1-9
1529 \ifnum#1<#5 % 1
1530 1%
1531 \ifx#2=%
1532 \else
1533 \expandafter\InCa@ProcessDiv\expandafter{%
1534 \number\InCa@Sub#1!#4!%
1535 #2%
1536 }#3!#4!#5!#6!#7!#8!%
1537 \fi
1538 \else % 2-9
1539 \ifnum#1<#7 % 2 3 4 5
1540 \ifnum#1<#6 % 2 3
1541 \@ReturnAfterElseFi{%
1542 \expandafter\InCa@@ProcessDiv
1543 \number\InCa@Sub#1!#5!%!%
1544 23%
1545 }%
1546 \else % 4 5
1547 \@ReturnAfterFi{%
1548 \expandafter\InCa@@ProcessDiv
1549 \number\InCa@Sub#1!#6!%!%
1550 45%
1551 }%
1552 \fi
1553 #2#3!#4!#5!#6!#7!#8!%
1554 \else % 6 7 8 9
1555 \ifnum#1<#8 % 6 7
1556 \@ReturnAfterElseFi{%
1557 \expandafter\InCa@@ProcessDiv
1558 \number\InCa@Sub#1!#7!%!%
1559 67%
1560 }%
1561 \else % 8 9
1562 \@ReturnAfterFi{%
1563 \expandafter\InCa@@ProcessDiv
1564 \number\InCa@Sub#1!#8!%!%
1565 89%
1566 }%
1567 \fi
1568 #2#3!#4!#5!#6!#7!#8!%
1569 \fi
1570 \fi
1571 \fi
1572 \ifx#2=%
1573 \expandafter@gobble
1574 \fi
1575 /%
1576 }
```

\InCa@@ProcessDiv

```
1577 \def\InCa@@ProcessDiv#1!#2#3#4#5!#6!{%
1578   \ifnum#1<#6 %
1579     #2%
1580     \@ReturnAfterElseFi{%
1581       \ifx#4=%
1582         \expandafter\InCa@CleanupIV
1583       \else
1584         \@ReturnAfterFi{%
1585           \InCa@ProcessDiv{#1#4}#5!#6!%
1586         }%
1587       \fi
1588     }%
1589   \else
1590     #3%
1591     \@ReturnAfterFi{%
1592       \ifx#4=%
1593         \expandafter\InCa@CleanupIV
1594       \else
1595         \@ReturnAfterFi{%
1596           \expandafter\InCa@ProcessDiv\expandafter{%
1597             \number\InCa@Sub#1!#6! %
1598             #4%
1599           }#5!#6!%
1600         }%
1601       \fi
1602     }%
1603   \fi
1604 }
```

\InCa@CleanupIV

```
1605 \def\InCa@CleanupIV#1!#2!#3!#4!{}
```

2.5.12 Mod

\intcalcMod

```
1606 \def\intcalcMod#1#2{%
1607   \number\expandafter\InCa@Mod
1608   \number\number#1\expandafter!%
1609   \number#2! %
1610 }
```

\intcalc@Mod Pseudocode/decision table for \intcalc@Mod.

```
if    y = 0   DivisionByZero
elsif y < 0   - Mod(-x, -y)
elsif x = 0   0
elsif y = 1   0
elsif y = 2   ifodd(x) ? 1 : 0
elsif x < 0   z ← x - (x/y) * y; (z < 0) ? z + y : z
else         x - (x/y) * y
```

```
1611 \def\InCa@Mod#1!#2!{%
1612   \ifcase#2 %
1613     0\IntCalcError:DivisionByZero%
1614   \else
1615     \ifnum#2<\z@
```

```

1616     -%
1617     \expandafter\InCa@Mod
1618     \number-#1\expandafter!%
1619     \number-#2!%
1620     \else
1621     \ifcase#1 %
1622     0%
1623     \else
1624     \ifcase#2 % 0 already caught
1625 ?     \IntCalcError:ThisCannotHappen%
1626     \or % 1
1627     0%
1628     \or % 2
1629     \ifodd#1 1\else 0\fi
1630     \else
1631     \ifnum#1<\z@
1632     \expandafter\InCa@ModShift
1633     \number-%
1634     \expandafter\InCa@Sub
1635     \number@gobble#1\expandafter!%
1636     \number\intcalcMul{#2}{%
1637     \expandafter\InCa@Div@gobble#1!#2!%
1638     }!%
1639     !#2!%
1640     \else
1641     \expandafter\InCa@Sub\number#1\expandafter!%
1642     \number\intcalcMul{#2}{\InCa@Div#1!#2!}%
1643     \fi
1644     \fi
1645     \fi
1646     \fi
1647     \fi
1648 }

```

\IntCalcMod

```

1649 \def\InCa@Temp#1{%
1650   \def\IntCalcMod##1!##2!{%
1651     \number
1652     \ifcase##2 %
1653     0\IntCalcError:DivisionByZero%
1654     \else
1655     \ifcase##1 %
1656     0%
1657     \else
1658     \ifcase##2 % 0 already caught
1659 ?     \IntCalcError:ThisCannotHappen
1660     \or % 1
1661     0%
1662     \or % 2
1663     \ifodd ##1 1\else 0\fi
1664     \else
1665     \expandafter\InCa@Sub\number##1\expandafter!%
1666     \number\intcalcMul{##2}{\InCa@Div##1!##2!}%
1667     \fi
1668     \fi
1669     \fi
1670     #1%
1671   }%

```

```

1672 }
1673 \InCa@Temp{ }%

```

`\InCa@ModShift`

```

1674 \def\InCa@ModShift#1!#2!{%
1675   \ifnum#1<\z@
1676     \expandafter\InCa@Sub\number#2\expandafter!%
1677     \@gobble#1!%
1678   \else
1679     #1%
1680   \fi
1681 }

```

2.5.13 Help macros

`\InCa@Empty`

```

1682 \def\InCa@Empty{}

```

`\@gobble`

```

1683 \expandafter\ifx\csname @gobble\endcsname\relax
1684   \long\def\@gobble#1{}%
1685 \fi

```

`\@ReturnAfterFi`

```

1686 \long\def\@ReturnAfterFi#1\fi{\fi#1}%

```

`\@ReturnAfterElseFi`

```

1687 \long\def\@ReturnAfterElseFi#1\else#2\fi{\fi#1}%

```

```

1688 \InCa@AtEnd%

```

```

1689 \endpackage

```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/intcalc/intcalc.dtx](#) The source file.

[CTAN:macros/latex/contrib/intcalc/intcalc.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘intcalc’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/intcalc.tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX Files” ([CTAN:pkg/tds](#)). Directories with `texmf` in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the `intcalc.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip intcalc.tds.zip -d ~/texmf
```

¹[CTAN:pkg/intcalc](#)

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain `TEX`:

```
tex intcalc.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
intcalc.sty → tex/generic/intcalc/intcalc.sty
intcalc.pdf → doc/latex/intcalc/intcalc.pdf
intcalc.dtx → source/latex/intcalc/intcalc.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

3.4 Refresh file name databases

If your `TEX` distribution (`TEX Live`, `MiKTEX`, ...) relies on file name databases, you must refresh these. For example, `TEX Live` users run `texhash` or `mktextlsr`.

3.5 Some details for the interested

Unpacking with L^AT_EX. The `.dtx` chooses its action depending on the format:

plain T_EX: Run `docstrip` and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for `docstrip` (really, `docstrip` does not need L^AT_EX), then inform the `autodetect` routine about your intention:

```
latex \let\install=y\input{intcalc.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdfLATEX`:

```
pdflatex intcalc.dtx
makeindex -s gind.ist intcalc.idx
pdflatex intcalc.dtx
makeindex -s gind.ist intcalc.idx
pdflatex intcalc.dtx
```

4 History

[2007/09/09 v1.0]

- First version.

[2007/09/27 v1.1]

- `\intcalcNum` added.
- `\intcalcShl` and `\intcalcShr` allow negative numbers. The sign is preserved.
- Reuse `\@gobble` instead of own macro `\IntCalc@Gobble`.
- Small fixes.
- Shorter internal prefix.
- Some programmer's interface.

[2016/05/16 v1.2]

- Documentation updates.

[2019/12/15 v1.3]

- Documentation updates.

5 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols	
<code>\@ReturnAfterElseFi</code>	706, 709, 717, 720, 727, 745, 753, 865, 992, 995, 1001, 1006, 1008, 1014, 1018, 1095, 1104, 1106, 1113, 1127, 1131, 1135, 1139, 1238, 1246, 1247, 1301, 1683
..... 660, 671, 683, 695, 1055, 1488, 1541, 1556, 1580, <u>1687</u>	
<code>\@ReturnAfterFi</code>	664, 675, 687, 699, 710, 721, 1009, 1053, 1107, 1240, 1492, 1547, 1562, 1584, 1591, 1595, <u>1686</u>
<code>\@gobble</code>	601, 604, 609, 612, 619, 622, 1260, 1263, 1268, 1271, 1279, 1281, 1452, 1455, 1461, 1573, 1635, 1637, 1677, <u>1683</u>
<code>\@ne</code>	311, <u>1391</u>
<code>\@undefined</code>	58
A	
<code>\aftergroup</code>	29
C	
<code>\catcode</code>	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 116
<code>\csname</code>	14, 21, 50, 66, 76, 114, 184, 481, 484, 494, 500, 506, 509, 513, 515, 520, 539, 549, 551, 556, 575, 682, 694,
<code>\empty</code>	17, 18
<code>\endcsname</code>	14, 21, 50, 66, 76, 114, 184, 513, 520, 521, 539, 540, 549, 556, 557, 575, 576, 706, 717, 727, 729, 731, 745, 747, 749, 753, 755, 757, 865, 867, 869, 1006, 1014, 1015, 1018, 1019, 1104, 1113, 1114, 1127, 1128, 1131, 1133, 1135, 1139, 1238, 1247, 1683
<code>\endinput</code>	29, 112
<code>\endlinechar</code>	4, 35, 71, 77, 89
I	
<code>\if</code>	424, 425, 433
<code>\ifcase</code> ...	162, 280, 287, 333, 336, 360, 363, 383, 386, 410, 413, 442, 523, 541, 559, 577, 645, 728, 759, 871, 1020, 1065, 1090, 1132, 1142, 1296, 1349, 1360, 1420, 1423, 1435, 1438, 1469, 1612, 1621, 1624, 1652, 1655, 1658

<code>\ifnum</code>	138, 145, 152, 155, 176, 295, 311, 341, 348, 391, 398, 596, 597, 599, 607, 616, 617, 625, 652, 746, 754, 866, 1237, 1256, 1257, 1258, 1267, 1275, 1277, 1284, 1304, 1368, 1391, 1392, 1403, 1449, 1450, 1458, 1468, 1479, 1522, 1529, 1539, 1540, 1555, 1578, 1615, 1631, 1675	<code>\InCa@Inc</code>	485, 495, 506, <u>511</u>
<code>\ifodd</code>	246, 252, 296, 314, 1054, 1369, 1378, 1402, 1629, 1663	<code>\InCa@IncDigit9</code>	<u>539</u>
<code>\ifx</code>	15, 18, 21, 50, 58, 61, 114, 120, 127, 130, 184, 249, 289, 301, 479, 492, 497, 512, 548, 659, 670, 681, 693, 705, 716, 991, 1005, 1040, 1051, 1103, 1312, 1317, 1334, 1354, 1362, 1374, 1377, 1487, 1524, 1531, 1572, 1581, 1592, 1683	<code>\InCa@IncDigit[0-8]</code>	<u>519</u>
<code>\immediate</code>	23, 52	<code>\InCa@IncSwitch</code>	476, <u>478</u>
<code>\InCa@@@Add</code>	661, <u>680</u>	<code>\InCa@Max</code>	<u>144</u> , 204, 468
<code>\InCa@@@Sub</code>	672, <u>692</u>	<code>\InCa@Min</code>	<u>137</u> , 199, 464
<code>\InCa@@@Add</code>	648, <u>658</u>	<code>\InCa@Mod</code>	378, <u>382</u> , 1607, 1611, 1617
<code>\InCa@@@Div</code>	339, <u>373</u> , 426, 429, 435, 437, 1441, 1451, 1455, 1460, 1463, <u>1467</u>	<code>\InCa@ModShift</code>	1632, <u>1674</u>
<code>\InCa@@@Mod</code>	389, <u>423</u>	<code>\InCa@ModX</code>	428, 434, <u>441</u>
<code>\InCa@@@ProcessDiv</code>	1542, 1548, 1557, 1563, <u>1577</u>	<code>\InCa@Mul</code>	1259, 1262, 1268, 1270, 1278, 1281, 1285, 1287, 1293, <u>1295</u> , <u>1311</u> , 1335, 1337, 1355, 1357, 1393, 1395, 1399, 1404, 1406
<code>\InCa@@@Sub</code>	655, <u>669</u>	<code>\InCa@MulSwitch</code>	1251, <u>1255</u>
<code>\InCa@@@TestMode</code>	118	<code>\InCa@Param[0-9]</code>	<u>1112</u>
<code>\InCa@@@TimDigitCarry</code>	1241, <u>1245</u>	<code>\InCa@Pow</code>	275, <u>279</u> , 1344, <u>1348</u>
<code>\InCa@Abs</code>	119, 193, 458	<code>\InCa@PowRec</code> 304, <u>310</u> , 1381, 1383, <u>1390</u>	
<code>\InCa@Add</code>	600, 603, 626, 628, 634, <u>644</u> , 1313, 1323, 1514	<code>\InCa@ProcessAdd</code>	684, <u>704</u>
<code>\InCa@AddDigit0</code>	<u>745</u>	<code>\InCa@ProcessDiv</code> 1518, <u>1520</u> , 1585, 1596	
<code>\InCa@AddDigit[1-9]</code>	<u>752</u>	<code>\InCa@ProcessMul</code> 1307, 1311, 1318, 1322	
<code>\InCa@AddSwitch</code>	585, 591, <u>595</u>	<code>\InCa@ProcessSub</code>	696, <u>715</u>
<code>\InCa@AtEnd</code>	95, 96, 112, 448, 1688	<code>\InCa@ProcessTim</code>	1096, <u>1102</u>
<code>\InCa@CleanupIV</code>	1582, 1593, <u>1605</u>	<code>\InCa@Sgn</code>	<u>126</u> , 196, 461
<code>\InCa@Cmp</code>	<u>151</u> , 209, 472	<code>\InCa@Shl</code>	993, 996, 1002, 1004, 1010, 1302
<code>\InCa@Dec</code>	482, 501, 509, <u>547</u>	<code>\InCa@ShlDigit0</code>	<u>1014</u>
<code>\InCa@DecDigit0</code>	<u>575</u>	<code>\InCa@ShlDigit[1-9]</code>	<u>1017</u>
<code>\InCa@DecDigit[1-9]</code>	<u>555</u>	<code>\InCa@ShlSwitch</code>	988, <u>990</u>
<code>\InCa@DecSwitch</code>	489, <u>491</u>	<code>\InCa@Shr</code>	240, <u>248</u> , 1041, 1043, 1047, <u>1049</u> , 1474
<code>\InCa@DigitCarry[0-9]</code>	<u>726</u>	<code>\InCa@ShrDigit</code>	1050, 1064
<code>\InCa@Div</code>	328, <u>332</u> , 1415, <u>1419</u> , 1637, 1642, 1666	<code>\InCa@ShrSwitch</code>	1037, <u>1039</u>
<code>\InCa@DivStart</code>	1476, <u>1486</u>	<code>\InCa@Space</code>	639, 648, 655
<code>\InCa@DivStartI</code>	1489, 1497	<code>\InCa@Sqr</code>	266, <u>268</u> , 1331, <u>1333</u>
<code>\InCa@DivStartII</code>	1498, 1503	<code>\InCa@StartI</code>	<u>1497</u>
<code>\InCa@DivStartIII</code>	1504, 1509	<code>\InCa@StartII</code>	<u>1503</u>
<code>\InCa@DivStartIV</code>	1510, 1517	<code>\InCa@StartIII</code>	<u>1509</u>
<code>\InCa@DivSwitch</code>	1426, <u>1448</u>	<code>\InCa@StartIV</code>	<u>1517</u>
<code>\InCa@Empty</code>	659, 670, 681, 693, 705, 716, 1312, 1317, <u>1682</u>	<code>\InCa@Sub</code>	609, 611, 618, 622, 637, <u>651</u> , 1534, 1543, 1549, 1558, 1564, 1597, 1634, 1641, 1665, 1676
<code>\InCa@Fac</code>	<u>161</u> , 272, 1341	<code>\InCa@SubDigit[0-9]</code>	<u>864</u>
<code>\InCa@FirstOfOne</code>	451, 454, <u>456</u>	<code>\InCa@Temp</code>	357, 372, 407, 422, 519, 530, 531, 532, 533, 534, 535, 536, 537, 538, 555, 566, 567, 568, 569, 570, 571, 572, 573, 574, 726, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 752, 765, 776, 787, 798, 809, 820, 831, 842, 853, 864, 877, 888, 899, 910, 921, 932, 943, 954, 965, 976, 1017, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1087, 1101, 1112, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1138, 1148,

