

The `bigintcalc` package

Heiko Oberdiek*

2019/12/15 v1.5

Abstract

This package provides expandable arithmetic operations with big integers that can exceed \TeX 's number limits.

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	Odd	5
1.4.5	Inc, Dec, Add, Sub	5
1.4.6	Shl, Shr	6
1.4.7	Mul, Sqr, Fac, Pow	6
1.4.8	Div, Mul	7
1.5	Interface for programmers	7
2	Implementation	8
2.1	Reload check and package identification	8
2.2	Catcodes	9
2.3	ϵ - \TeX detection	10
2.4	Help macros	10
2.5	Expand number	11
2.6	Normalize expanded number	12
2.7	Num	13
2.8	Inv, Abs, Sgn	13
2.9	Cmp, Min, Max	14
2.10	Odd	16
2.11	Inc, Dec	17
2.12	Add, Sub	20
2.13	Shl, Shr	26
2.14	$\backslash\text{BIC@Tim}$	29
2.15	Mul	31

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

2.16	Sqr	33
2.17	Fac	33
2.18	Pow	34
2.18.1	Help macros	37
2.18.2	Recursive calculation	38
2.19	Div	39
2.20	Mod	45
3	Installation	48
3.1	Download	48
3.2	Bundle installation	48
3.3	Package installation	48
3.4	Refresh file name databases	48
3.5	Some details for the interested	48
4	History	49
[2007/09/27 v1.0]		49
[2007/11/11 v1.1]		49
[2011/01/30 v1.2]		49
[2012/04/08 v1.3]		49
[2016/05/16 v1.4]		49
[2019/12/15 v1.5]		49
5	Index	49

1 Documentation

1.1 Introduction

Package `bigintcalc` defines arithmetic operations that deal with big integers. Big integers can be given either as explicit integer number or as macro code that expands to an explicit number. *Big* means that there is no limit on the size of the number. Big integers may exceed $\text{T}_{\text{E}}\text{X}$'s range limitation of -2147483647 and 2147483647. Only memory issues will limit the usable range.

In opposite to package `intcalc` unexpandable command tokens are not supported, even if they are valid $\text{T}_{\text{E}}\text{X}$ numbers like count registers or commands created by `\chardef`. Nevertheless they may be used, if they are prefixed by `\number`.

Also ε - $\text{T}_{\text{E}}\text{X}$'s `\numexpr` expressions are not supported directly in the manner of package `intcalc`. However they can be given if `\the\numexpr` or `\number\numexpr` are used.

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. `\bigintcalcAdd{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 `\csname`, file names, or other expandable contexts.

1.2 Conditions

1.2.1 Preconditions

- Arguments can be anything that expands to a number that consists of optional signs and digits.

- The arguments and return values must be sound. Zero as divisor or factorials of negative numbers will cause errors.

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`, for example.
- Furthermore exactly two expansion steps calculate the result.
- The number consists of one optional minus sign and one or more 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, but without its range limitation. And the tokens (minus sign, digits) have cat-code 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

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: `\BigIntCalcError: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.

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

`\bigintcalcNum {<x>}`

Macro `\bigintcalcNum` 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

`\bigintcalcInv {⟨x⟩}`

Macro `\bigintcalcInv` switches the sign.

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

`\bigintcalcAbs {⟨x⟩}`

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

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

`\bigintcalcSgn {⟨x⟩}`

Macro `\bigintcalcSgn` 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\bigintcalcSgn{⟨x⟩}
  $x=0$
\or
  $x>0$
\else
  $x<0$
\fi
```

1.4.3 Min, Max, Cmp

`\bigintcalcMin {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcMax {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcCmp {⟨x⟩} {⟨y⟩}`

Macro `\bigintcalcCmp` 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\bigintcalcCmp{<x>}{<y>}
  $x=y$
\or
  $x>y$
\else
  $x<y$
\fi
```

1.4.4 Odd

`\bigintcalcOdd {⟨x⟩}`

$$\text{Odd}(x) := \begin{cases} 1 & \text{if } x \text{ is odd} \\ 0 & \text{if } x \text{ is even} \end{cases}$$

1.4.5 Inc, Dec, Add, Sub

`\bigintcalcInc {⟨x⟩}`

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

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

`\bigintcalcDec {⟨x⟩}`

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

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

`\bigintcalcAdd {⟨x⟩} {⟨y⟩}`

Macro `\bigintcalcAdd` adds the two numbers.

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

`\bigintcalcSub {⟨x⟩} {⟨y⟩}`

Macro `\bigintcalcSub` calculates the difference.

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

1.4.6 Shl, Shr

`\bigintcalcShl {⟨x⟩}`

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

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

`\bigintcalcShr {⟨x⟩}`

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

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

1.4.7 Mul, Sqr, Fac, Pow

`\bigintcalcMul {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcSqr {⟨x⟩}`

Macro `\bigintcalcSqr` returns the square product.

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

`\bigintcalcFac {⟨x⟩}`

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

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

$$(0! = 1)$$

`\bigintcalcPow Mx My`

Macro `\bigintcalcPow` 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:

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

$$(0^0 = 1)$$

1.4.8 Div, Mul

`\bigintcalcDiv {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcMod {⟨x⟩} {⟨y⟩}`

Macro `\bigintcalcMod` 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 programmers

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.
- Delimited by exclamation mark. Curly braces around the number are not allowed and will break the code.

`\BigIntCalcOdd ⟨number⟩ !`

1/0 is returned if `⟨number⟩` is odd/even.

`\BigIntCalcInc ⟨number⟩ !`

Incrementation.

`\BigIntCalcDec ⟨number⟩ !`

Decrementation, positive number without zero.

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

Addition, $A \geq B$.

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

Subtraction, $A \geq B$.

`\BigIntCalcShl $\langle number \rangle$!`

Left shift (multiplication with two).

`\BigIntCalcShr $\langle number \rangle$!`

Right shift (integer division by two).

`\BigIntCalcMul $\langle number A \rangle$! $\langle number B \rangle$!`

Multiplication, $A \geq B$.

`\BigIntCalcDiv $\langle number A \rangle$! $\langle number B \rangle$!`

Division operation.

`\BigIntCalcMod $\langle number A \rangle$! $\langle number B \rangle$!`

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@bigintcalc.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{bigintcalc}{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@bigintcalc.sty\endcsname
67 \ProvidesPackage{bigintcalc}%
68 [2019/12/15 v1.5 Expandable calculations on big 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 BIC@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\BIC@AtEnd{%
96     \BIC@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{33}{12}% !
102 \TMP@EnsureCode{36}{14}% $ (comment!)
103 \TMP@EnsureCode{38}{14}% & (comment!)
104 \TMP@EnsureCode{40}{12}% (
105 \TMP@EnsureCode{41}{12}% )
106 \TMP@EnsureCode{42}{12}% *
107 \TMP@EnsureCode{43}{12}% +
108 \TMP@EnsureCode{45}{12}% -
109 \TMP@EnsureCode{46}{12}% .
110 \TMP@EnsureCode{47}{12}% /
111 \TMP@EnsureCode{58}{11}% : (letter!)
112 \TMP@EnsureCode{60}{12}% <
113 \TMP@EnsureCode{62}{12}% >
114 \TMP@EnsureCode{63}{14}% ? (comment!)
115 \TMP@EnsureCode{91}{12}% [
116 \TMP@EnsureCode{93}{12}% ]
117 \edef\BIC@AtEnd{\BIC@AtEnd\noexpand\endinput}
118 \begingroup\expandafter\expandafter\expandafter\endgroup
119 \expandafter\ifx\csname BIC@TestMode\endcsname\relax
120 \else
121   \catcode63=9 % ? (ignore)
122 \fi
123 ? \let\BIC@@TestMode\BIC@TestMode

```

2.3 ϵ -TeX detection

```

124 \begingroup\expandafter\expandafter\expandafter\endgroup
125 \expandafter\ifx\csname numexpr\endcsname\relax
126   \catcode36=9 % $ (ignore)
127 \else
128   \catcode38=9 % & (ignore)
129 \fi

```

2.4 Help macros

```

\BIC@Fi
130 \let\BIC@Fi\fi

\BIC@AfterFi
131 \def\BIC@AfterFi#1#2\BIC@Fi{\fi#1}%

```

\BIC@AfterFiFi

```
132 \def\BIC@AfterFiFi#1#2\BIC@Fi{\fi\fi#1}%
```

\BIC@AfterFiFiFi

```
133 \def\BIC@AfterFiFiFi#1#2\BIC@Fi{\fi\fi\fi#1}%
```

\BIC@Space

```
134 \begingroup
135   \def\x#1{\endgroup
136     \let\BIC@Space= #1%
137   }%
138 \x{ }
```

2.5 Expand number

```
139 \begingroup\expandafter\expandafter\expandafter\endgroup
140 \expandafter\ifx\csname RequirePackage\endcsname\relax
141   \def\TMP@RequirePackage#1[#2]{%
142     \begingroup\expandafter\expandafter\expandafter\endgroup
143     \expandafter\ifx\csname ver@#1.sty\endcsname\relax
144       \input #1.sty\relax
145     \fi
146   }%
147   \TMP@RequirePackage{pdftexcmds}[2007/11/11]%
148 \else
149   \RequirePackage{pdftexcmds}[2007/11/11]%
150 \fi

151 \begingroup\expandafter\expandafter\expandafter\endgroup
152 \expandafter\ifx\csname pdf@escapehex\endcsname\relax
```

\BIC@Expand

```
153 \def\BIC@Expand#1{%
154   \romannumeral0%
155   \BIC@@Expand#1!\@nil{}%
156 }%
```

\BIC@@Expand

```
157 \def\BIC@@Expand#1#2\@nil#3{%
158   \expandafter\ifcat\noexpand#1\relax
159     \expandafter\@firstoftwo
160   \else
161     \expandafter\@secondoftwo
162   \fi
163   {%
164     \expandafter\BIC@@Expand#1#2\@nil{#3}%
165   }{%
166     \ifx#1!%
167       \expandafter\@firstoftwo
168     \else
169       \expandafter\@secondoftwo
170     \fi
171     { #3}{%
172       \BIC@@Expand#2\@nil{#3#1}%
173     }%
174   }%
175 }%
```

```

\@firstoftwo
176 \expandafter\ifx\csname @firstoftwo\endcsname\relax
177   \long\def\@firstoftwo#1#2{#1}%
178 \fi

\@secondoftwo
179 \expandafter\ifx\csname @secondoftwo\endcsname\relax
180   \long\def\@secondoftwo#1#2{#2}%
181 \fi

182 \else

\BIC@Expand
183 \def\BIC@Expand#1{%
184   \romannumeral0\expandafter\expandafter\expandafter\BIC@Space
185   \pdf@unescapehex{%
186     \expandafter\expandafter\expandafter
187     \BIC@StripHexSpace\pdf@escapehex{#1}20\@nil
188   }%
189 }%

\BIC@StripHexSpace
190 \def\BIC@StripHexSpace#120#2\@nil{%
191   #1%
192   \ifx\#2\%
193   \else
194     \BIC@AfterFi{%
195       \BIC@StripHexSpace#2\@nil
196     }%
197   \BIC@Fi
198   }%

199 \fi

```

2.6 Normalize expanded number

```

\BIC@Normalize #1: result sign
#2: first token of number
200 \def\BIC@Normalize#1#2{%
201   \ifx#2-%
202     \ifx\#1\%
203       \BIC@AfterFiFi{%
204         \BIC@Normalize-%
205       }%
206   \else
207     \BIC@AfterFiFi{%
208       \BIC@Normalize{}%
209     }%
210   \fi
211 \else
212   \ifx#2+%
213     \BIC@AfterFiFi{%
214       \BIC@Normalize{#1}%
215     }%
216   \else
217     \ifx#20%
218       \BIC@AfterFiFiFi{%
219         \BIC@NormalizeZero{#1}%

```

```

220     }%
221     \else
222     \BIC@AfterFiFiFi{%
223     \BIC@NormalizeDigits#1#2%
224     }%
225     \fi
226     \fi
227 \BIC@Fi
228 }

```

\BIC@NormalizeZero

```

229 \def\BIC@NormalizeZero#1#2{%
230 \ifx#2!%
231 \BIC@AfterFi{ 0}%
232 \else
233 \ifx#20%
234 \BIC@AfterFiFi{%
235 \BIC@NormalizeZero{#1}%
236 }%
237 \else
238 \BIC@AfterFiFi{%
239 \BIC@NormalizeDigits#1#2%
240 }%
241 \fi
242 \BIC@Fi
243 }

```

\BIC@NormalizeDigits

```

244 \def\BIC@NormalizeDigits#1!{ #1}

```

2.7 Num

\bigintcalcNum

```

245 \def\bigintcalcNum#1{%
246 \romannumeral0%
247 \expandafter\expandafter\expandafter\BIC@Normalize
248 \expandafter\expandafter\expandafter{%
249 \expandafter\expandafter\expandafter}%
250 \BIC@Expand{#1}!%
251 }

```

2.8 Inv, Abs, Sgn

\bigintcalcInv

```

252 \def\bigintcalcInv#1{%
253 \romannumeral0\expandafter\expandafter\expandafter\BIC@Space
254 \bigintcalcNum{-#1}%
255 }

```

\bigintcalcAbs

```

256 \def\bigintcalcAbs#1{%
257 \romannumeral0%
258 \expandafter\expandafter\expandafter\BIC@Abs
259 \bigintcalcNum{#1}%
260 }

```

\BIC@Abs

```
261 \def\BIC@Abs#1{%
262   \ifx#1-%
263     \expandafter\BIC@Space
264   \else
265     \expandafter\BIC@Space
266     \expandafter#1%
267   \fi
268 }
```

\bigintcalcSgn

```
269 \def\bigintcalcSgn#1{%
270   \number
271   \expandafter\expandafter\expandafter\BIC@Sgn
272   \bigintcalcNum{#1}! %
273 }
```

\BIC@Sgn

```
274 \def\BIC@Sgn#1#2!{%
275   \ifx#1-%
276     -1%
277   \else
278     \ifx#10%
279       0%
280     \else
281       1%
282     \fi
283   \fi
284 }
```

2.9 Cmp, Min, Max

\bigintcalcCmp

```
285 \def\bigintcalcCmp#1#2{%
286   \number
287   \expandafter\expandafter\expandafter\BIC@Cmp
288   \bigintcalcNum{#2}!{#1}%
289 }
```

\BIC@Cmp

```
290 \def\BIC@Cmp#1!#2{%
291   \expandafter\expandafter\expandafter\BIC@@Cmp
292   \bigintcalcNum{#2}!#1!%
293 }
```

\BIC@@Cmp

```
294 \def\BIC@@Cmp#1#2!#3#4!{%
295   \ifx#1-%
296     \ifx#3-%
297       \BIC@AfterFiFi{%
298         \BIC@@Cmp#4!#2!%
299       }%
300   \else
301     \BIC@AfterFiFi{%
302       -1 %
303     }%
304   \fi
```

```

305 \else
306 \ifx#3-%
307 \BIC@AfterFiFi{%
308 1 %
309 }%
310 \else
311 \BIC@AfterFiFi{%
312 \BIC@CmpLength#1#2!#3#4!#1#2!#3#4!%
313 }%
314 \fi
315 \BIC@Fi
316 }

\BIC@PosCmp
317 \def\BIC@PosCmp#1!#2!{%
318 \BIC@CmpLength#1!#2!#1!#2!%
319 }

\BIC@CmpLength
320 \def\BIC@CmpLength#1#2!#3#4!{%
321 \ifx\#2\%
322 \ifx\#4\%
323 \BIC@AfterFiFi\BIC@CmpDiff
324 \else
325 \BIC@AfterFiFi{%
326 \BIC@CmpResult{-1}%
327 }%
328 \fi
329 \else
330 \ifx\#4\%
331 \BIC@AfterFiFi{%
332 \BIC@CmpResult1%
333 }%
334 \else
335 \BIC@AfterFiFi{%
336 \BIC@CmpLength#2!#4!%
337 }%
338 \fi
339 \BIC@Fi
340 }

\BIC@CmpResult
341 \def\BIC@CmpResult#1#2!#3!#1 }

\BIC@CmpDiff
342 \def\BIC@CmpDiff#1#2!#3#4!{%
343 \ifnum#1<#3 %
344 \BIC@AfterFiFi{%
345 -1 %
346 }%
347 \else
348 \ifnum#1>#3 %
349 \BIC@AfterFiFi{%
350 1 %
351 }%
352 \else
353 \ifx\#2\%
354 \BIC@AfterFiFiFi{%

```

```

355         0 %
356     }%
357     \else
358         \BIC@AfterFiFiFi{%
359         \BIC@CmpDiff#2!#4!%
360     }%
361     \fi
362     \fi
363     \BIC@Fi
364 }

\bigintcalcMin
365 \def\bigintcalcMin#1{%
366     \romannumeral0%
367     \expandafter\expandafter\expandafter\BIC@MinMax
368     \bigintcalcNum{#1}!-%
369 }

\bigintcalcMax
370 \def\bigintcalcMax#1{%
371     \romannumeral0%
372     \expandafter\expandafter\expandafter\BIC@MinMax
373     \bigintcalcNum{#1}!!%
374 }

\BIC@MinMax #1:  $x$ 
#2: sign for comparison
#3:  $y$ 
375 \def\BIC@MinMax#1!#2!#3{%
376     \expandafter\expandafter\expandafter\BIC@MinMax
377     \bigintcalcNum{#3}!#1!#2!%
378 }

\BIC@@MinMax #1:  $y$ 
#2:  $x$ 
#3: sign for comparison
379 \def\BIC@@MinMax#1!#2!#3!{%
380     \ifnum\BIC@@Cmp#1!#2!=#31 %
381         \BIC@AfterFi{ #1}%
382     \else
383         \BIC@AfterFi{ #2}%
384     \BIC@Fi
385 }

```

2.10 Odd

```

\bigintcalcOdd
386 \def\bigintcalcOdd#1{%
387     \romannumeral0%
388     \expandafter\expandafter\expandafter\BIC@Odd
389     \bigintcalcAbs{#1}!%
390 }

\BigIntCalcOdd
391 \def\BigIntCalcOdd#1!{%
392     \romannumeral0%
393     \BIC@Odd#1!%
394 }

```

```

\BIC@Odd #1: x
395 \def\BIC@Odd#1#2{%
396 \ifx#2!%
397 \ifodd#1 %
398 \BIC@AfterFiFi{ 1}%
399 \else
400 \BIC@AfterFiFi{ 0}%
401 \fi
402 \else
403 \expandafter\BIC@Odd\expandafter#2%
404 \BIC@Fi
405 }

```

2.11 Inc, Dec

\bigintcalcInc

```

406 \def\bigintcalcInc#1{%
407 \romannumeral0%
408 \expandafter\expandafter\expandafter\BIC@IncSwitch
409 \bigintcalcNum{#1}!%
410 }

```

\BIC@IncSwitch

```

411 \def\BIC@IncSwitch#1#2!{%
412 \ifcase\BIC@Cmp#1#2!-1!%
413 \BIC@AfterFi{ 0}%
414 \or
415 \BIC@AfterFi{%
416 \BIC@Inc#1#2!}%
417 }%
418 \else
419 \BIC@AfterFi{%
420 \expandafter-\romannumeral0%
421 \BIC@Dec#2!}%
422 }%
423 \BIC@Fi
424 }

```

\bigintcalcDec

```

425 \def\bigintcalcDec#1{%
426 \romannumeral0%
427 \expandafter\expandafter\expandafter\BIC@DecSwitch
428 \bigintcalcNum{#1}!%
429 }

```

\BIC@DecSwitch

```

430 \def\BIC@DecSwitch#1#2!{%
431 \ifcase\BIC@Sgn#1#2! %
432 \BIC@AfterFi{ -1}%
433 \or
434 \BIC@AfterFi{%
435 \BIC@Dec#1#2!}%
436 }%
437 \else
438 \BIC@AfterFi{%
439 \expandafter-\romannumeral0%
440 \BIC@Inc#2!}%

```

```

441     }%
442     \BIC@Fi
443 }

\BigIntCalcInc
444 \def\BigIntCalcInc#1!{%
445     \romannumeral0\BIC@Inc#1!{%}%
446 }

\BigIntCalcDec
447 \def\BigIntCalcDec#1!{%
448     \romannumeral0\BIC@Dec#1!{%}%
449 }

\BIC@Inc
450 \def\BIC@Inc#1#2!#3{%
451     \ifx\#2\%
452         \BIC@AfterFi{%
453             \BIC@@Inc1#1#3!{%}%
454         }%
455     \else
456         \BIC@AfterFi{%
457             \BIC@Inc#2!#1#3}%
458         }%
459     \BIC@Fi
460 }

\BIC@@Inc
461 \def\BIC@@Inc#1#2#3!#4{%
462     \ifcase#1 %
463         \ifx\#3\%
464             \BIC@AfterFiFi{ #2#4}%
465         \else
466             \BIC@AfterFiFi{%
467                 \BIC@@Inc0#3!#2#4}%
468             }%
469         \fi
470     \else
471         \ifnum#2<9 %
472             \BIC@AfterFiFi{%
473 &             \expandafter\BIC@@Inc\the\numexpr#2+1\relax
474 $             \expandafter\expandafter\expandafter\BIC@@Inc
475 $             \ifcase#2 \expandafter1%
476 $             \or\expandafter2%
477 $             \or\expandafter3%
478 $             \or\expandafter4%
479 $             \or\expandafter5%
480 $             \or\expandafter6%
481 $             \or\expandafter7%
482 $             \or\expandafter8%
483 $             \or\expandafter9%
484 $?             \else\BigIntCalcError:ThisCannotHappen%
485 $             \fi
486                 0#3!#4}%
487             }%
488         \else
489             \BIC@AfterFiFi{%
490                 \BIC@@Inc01#3!#4}%

```

```

491     }%
492     \fi
493     \BIC@Fi
494 }

\BIC@@@Inc
495 \def\BIC@@@Inc#1#2#3!#4{%
496   \ifx\#3\%
497     \ifnum#2=1 %
498       \BIC@AfterFiFi{ #1#4}%
499     \else
500       \BIC@AfterFiFi{ #1#4}%
501     \fi
502   \else
503     \BIC@AfterFi{%
504       \BIC@@Inc#2#3!{#1#4}%
505     }%
506   \BIC@Fi
507 }

\BIC@Dec
508 \def\BIC@Dec#1#2!#3{%
509   \ifx\#2\%
510     \BIC@AfterFi{%
511       \BIC@@Dec1#1#3!{ }%
512     }%
513   \else
514     \BIC@AfterFi{%
515       \BIC@Dec#2!{#1#3}%
516     }%
517   \BIC@Fi
518 }

\BIC@@Dec
519 \def\BIC@@Dec#1#2#3!#4{%
520   \ifcase#1 %
521     \ifx\#3\%
522       \BIC@AfterFiFi{ #2#4}%
523     \else
524       \BIC@AfterFiFi{%
525         \BIC@@Dec0#3!{#2#4}%
526       }%
527     \fi
528   \else
529     \ifnum#2>0 %
530       \BIC@AfterFiFi{%
531         & \expandafter\BIC@@@Dec\the\numexpr#2-1\relax
532         $ \expandafter\expandafter\expandafter\BIC@@@Dec
533         $ \ifcase#2
534         $? \BigIntCalcError:ThisCannotHappen%
535         $ \or\expandafter0%
536         $ \or\expandafter1%
537         $ \or\expandafter2%
538         $ \or\expandafter3%
539         $ \or\expandafter4%
540         $ \or\expandafter5%
541         $ \or\expandafter6%
542         $ \or\expandafter7%

```

```

543 $      \or\expandafter8%
544 $?    \else\BigIntCalcError:ThisCannotHappen%
545 $      \fi
546      0#3!{#4}%
547      }%
548      \else
549      \BIC@AfterFiFi{%
550      \BIC@@@Dec91#3!{#4}%
551      }%
552      \fi
553 \BIC@Fi
554 }

```

\BIC@@@Dec

```

555 \def\BIC@@@Dec#1#2#3!#4{%
556   \ifx\#3\%
557     \ifcase#1 %
558       \ifx\#4\%
559         \BIC@AfterFiFiFi{ 0}%
560       \else
561         \BIC@AfterFiFiFi{ #4}%
562       \fi
563     \else
564       \BIC@AfterFiFi{ #1#4}%
565     \fi
566   \else
567     \BIC@AfterFi{%
568     \BIC@@@Dec#2#3!{#1#4}%
569     }%
570   \BIC@Fi
571 }

```

2.12 Add, Sub

\bigintcalcAdd

```

572 \def\bigintcalcAdd#1{%
573   \romannumeral0%
574   \expandafter\expandafter\expandafter\BIC@Add
575   \bigintcalcNum{#1}!%
576 }

```

\BIC@Add

```

577 \def\BIC@Add#1!#2{%
578   \expandafter\expandafter\expandafter
579   \BIC@AddSwitch\bigintcalcNum{#2}!#1!%
580 }

```

\bigintcalcSub

```

581 \def\bigintcalcSub#1#2{%
582   \romannumeral0%
583   \expandafter\expandafter\expandafter\BIC@Add
584   \bigintcalcNum{-#2}!{#1}%
585 }

```

\BIC@AddSwitch Decision table for \BIC@AddSwitch.

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

```

586 \def\BIC@AddSwitch#1#2!#3#4!{%
587 \ifx#1-% x < 0
588 \ifx#3-% y < 0
589 \expandafter-\romannumeral0%
590 \ifnum\BIC@PosCmp#2!#4!=1 % -x > -y
591 \BIC@AfterFiFiFi{%
592 \BIC@AddXY#2!#4!!!%
593 }%
594 \else % -x <= -y
595 \BIC@AfterFiFiFi{%
596 \BIC@AddXY#4!#2!!!%
597 }%
598 \fi
599 \else % y >= 0
600 \ifcase\BIC@PosCmp#2!#3#4!% -x = y
601 \BIC@AfterFiFiFi{ 0}%
602 \or % -x > y
603 \expandafter-\romannumeral0%
604 \BIC@AfterFiFiFi{%
605 \BIC@SubXY#2!#3#4!!!%
606 }%
607 \else % -x <= y
608 \BIC@AfterFiFiFi{%
609 \BIC@SubXY#3#4!#2!!!%
610 }%
611 \fi
612 \fi
613 \else % x >= 0
614 \ifx#3-% y < 0
615 \ifcase\BIC@PosCmp#1#2!#4!% x = -y
616 \BIC@AfterFiFiFi{ 0}%
617 \or % x > -y
618 \BIC@AfterFiFiFi{%
619 \BIC@SubXY#1#2!#4!!!%
620 }%
621 \else % x <= -y
622 \expandafter-\romannumeral0%
623 \BIC@AfterFiFiFi{%
624 \BIC@SubXY#4!#1#2!!!%
625 }%
626 \fi
627 \else % y >= 0
628 \ifnum\BIC@PosCmp#1#2!#3#4!=1 % x > y
629 \BIC@AfterFiFiFi{%
630 \BIC@AddXY#1#2!#3#4!!!%
631 }%

```

```

632     \else % x <= y
633         \BIC@AfterFiFiFi{%
634             \BIC@AddXY#3#4!#1#2!!!%
635         }%
636     \fi
637 \fi
638 \BIC@Fi
639 }

\BigIntCalcAdd
640 \def\BigIntCalcAdd#1!#2!{%
641     \romannumeral0\BIC@AddXY#1!#2!!!%
642 }

\BigIntCalcSub
643 \def\BigIntCalcSub#1!#2!{%
644     \romannumeral0\BIC@SubXY#1!#2!!!%
645 }

\BIC@AddXY
646 \def\BIC@AddXY#1#2!#3#4!#5!#6!{%
647     \ifx\#2\%
648         \ifx\#3\%
649             \BIC@AfterFiFi{%
650                 \BIC@DoAdd0!#1#5!#60!%
651             }%
652         \else
653             \BIC@AfterFiFi{%
654                 \BIC@DoAdd0!#1#5!#3#6!%
655             }%
656         \fi
657     \else
658         \ifx\#4\%
659             \ifx\#3\%
660                 \BIC@AfterFiFiFi{%
661                     \BIC@AddXY#2!{}!#1#5!#60!%
662                 }%
663             \else
664                 \BIC@AfterFiFiFi{%
665                     \BIC@AddXY#2!{}!#1#5!#3#6!%
666                 }%
667             \fi
668         \else
669             \BIC@AfterFiFi{%
670                 \BIC@AddXY#2!#4!#1#5!#3#6!%
671             }%
672         \fi
673     \BIC@Fi
674 }

\BIC@DoAdd #1: carry
#2: reverted result
#3#4: reverted  $x$ 
#5#6: reverted  $y$ 
675 \def\BIC@DoAdd#1#2!#3#4!#5#6!{%
676     \ifx\#4\%
677         \BIC@AfterFi{%
678 &         \expandafter\BIC@Space

```

```

679 & \the\numexpr#1+#3+#5\relax#2%
680 $ \expandafter\expandafter\expandafter\BIC@AddResult
681 $ \BIC@AddDigit#1#3#5#2%
682 }%
683 \else
684 \BIC@AfterFi{%
685 \expandafter\expandafter\expandafter\BIC@DoAdd
686 \BIC@AddDigit#1#3#5#2!#4!#6!%
687 }%
688 \BIC@Fi
689 }

\BIC@AddResult
690 $ \def\BIC@AddResult#1{%
691 $ \ifx#10%
692 $ \expandafter\BIC@Space
693 $ \else
694 $ \expandafter\BIC@Space\expandafter#1%
695 $ \fi
696 $ }%

\BIC@AddDigit #1: carry
#2: digit of  $x$ 
#3: digit of  $y$ 
697 \def\BIC@AddDigit#1#2#3{%
698 \romannumeral0%
699 & \expandafter\BIC@@AddDigit\the\numexpr#1+#2+#3!%
700 $ \expandafter\BIC@@AddDigit\number%
701 $ \csname
702 $ BIC@AddCarry%
703 $ \ifcase#1 %
704 $ #2%
705 $ \else
706 $ \ifcase#2 1\or2\or3\or4\or5\or6\or7\or8\or9\or10\fi
707 $ \fi
708 $ \endcsname#3!%
709 }

\BIC@@AddDigit
710 \def\BIC@@AddDigit#1!{%
711 \ifnum#1<10 %
712 \BIC@AfterFi{ 0#1}%
713 \else
714 \BIC@AfterFi{ #1}%
715 \BIC@Fi
716 }

\BIC@AddCarry0
717 $ \expandafter\def\csname BIC@AddCarry0\endcsname#1{#1}%

\BIC@AddCarry10
718 $ \expandafter\def\csname BIC@AddCarry10\endcsname#1{#1}%

\BIC@AddCarry[1-9]
719 $ \def\BIC@Temp#1#2{%
720 $ \expandafter\def\csname BIC@AddCarry#1\endcsname##1{%
721 $ \ifcase##1 #1\or

```

```

722 $ #2%
723 $? \else\BigIntCalcError:ThisCannotHappen%
724 $ \fi
725 $ }%
726 $ }%
727 $ \BIC@Temp 0{1\or2\or3\or4\or5\or6\or7\or8\or9}%
728 $ \BIC@Temp 1{2\or3\or4\or5\or6\or7\or8\or9\or10}%
729 $ \BIC@Temp 2{3\or4\or5\or6\or7\or8\or9\or10\or11}%
730 $ \BIC@Temp 3{4\or5\or6\or7\or8\or9\or10\or11\or12}%
731 $ \BIC@Temp 4{5\or6\or7\or8\or9\or10\or11\or12\or13}%
732 $ \BIC@Temp 5{6\or7\or8\or9\or10\or11\or12\or13\or14}%
733 $ \BIC@Temp 6{7\or8\or9\or10\or11\or12\or13\or14\or15}%
734 $ \BIC@Temp 7{8\or9\or10\or11\or12\or13\or14\or15\or16}%
735 $ \BIC@Temp 8{9\or10\or11\or12\or13\or14\or15\or16\or17}%
736 $ \BIC@Temp 9{10\or11\or12\or13\or14\or15\or16\or17\or18}%

```

\BIC@SubXY Preconditions:

- $x > y$, $x \geq 0$, and $y \geq 0$
- $\text{digits}(x) = \text{digits}(y)$

```

737 \def\BIC@SubXY#1#2!#3#4!#5!#6!{%
738 \ifx\#2\%
739 \ifx\#3\%
740 \BIC@AfterFiFi{%
741 \BIC@DoSub0!#1#5!#60!%
742 }%
743 \else
744 \BIC@AfterFiFi{%
745 \BIC@DoSub0!#1#5!#3#6!%
746 }%
747 \fi
748 \else
749 \ifx\#4\%
750 \ifx\#3\%
751 \BIC@AfterFiFiFi{%
752 \BIC@SubXY#2!#1#5!#60!%
753 }%
754 \else
755 \BIC@AfterFiFiFi{%
756 \BIC@SubXY#2!#1#5!#3#6!%
757 }%
758 \fi
759 \else
760 \BIC@AfterFiFi{%
761 \BIC@SubXY#2!#4!#1#5!#3#6!%
762 }%
763 \fi
764 \BIC@Fi
765 }

```

\BIC@DoSub #1: carry
#2: reverted result
#3#4: reverted x
#5#6: reverted y

```

766 \def\BIC@DoSub#1#2!#3#4!#5#6!{%
767 \ifx\#4\%
768 \BIC@AfterFi{%

```

```

769     \expandafter\expandafter\expandafter\BIC@SubResult
770     \BIC@SubDigit#1#3#5#2%
771     }%
772 \else
773     \BIC@AfterFi{%
774     \expandafter\expandafter\expandafter\BIC@DoSub
775     \BIC@SubDigit#1#3#5#2!#4!#6!%
776     }%
777 \BIC@Fi
778 }

\BIC@SubResult
779 \def\BIC@SubResult#1{%
780 \ifx#10%
781 \expandafter\BIC@SubResult
782 \else
783 \expandafter\BIC@Space\expandafter#1%
784 \fi
785 }

\BIC@SubDigit #1: carry
#2: digit of  $x$ 
#3: digit of  $y$ 
786 \def\BIC@SubDigit#1#2#3{%
787 \romannumeral0%
788 & \expandafter\BIC@@SubDigit\the\numexpr#2-#3-#1!%
789 $ \expandafter\BIC@@AddDigit\number
790 $ \csname
791 $ BIC@SubCarry%
792 $ \ifcase#1 %
793 $ #3%
794 $ \else
795 $ \ifcase#3 1\or2\or3\or4\or5\or6\or7\or8\or9\or10\fi
796 $ \fi
797 $ \endcsname#2!%
798 }

\BIC@@SubDigit
799 & \def\BIC@@SubDigit#1!{%
800 & \ifnum#1<0 %
801 & \BIC@AfterFi{%
802 & \expandafter\BIC@Space
803 & \expandafter1\the\numexpr#1+10\relax
804 & }%
805 & \else
806 & \BIC@AfterFi{ 0#1}%
807 & \BIC@Fi
808 & }%

\BIC@SubCarry0
809 $ \expandafter\def\csname BIC@SubCarry0\endcsname#1{#1}%

\BIC@SubCarry10
810 $ \expandafter\def\csname BIC@SubCarry10\endcsname#1{#1}%

\BIC@SubCarry[1-9]
811 $ \def\BIC@Temp#1#2{%

```

```

812 $ \expandafter\def\csname BIC@SubCarry#1\endcsname##1{%
813 $ \ifcase##1 #2%
814 $? \else\BigIntCalcError:ThisCannotHappen%
815 $ \fi
816 $ }%
817 $ }%
818 $ \BIC@Temp 1{19\or0\or1\or2\or3\or4\or5\or6\or7\or8}%
819 $ \BIC@Temp 2{18\or19\or0\or1\or2\or3\or4\or5\or6\or7}%
820 $ \BIC@Temp 3{17\or18\or19\or0\or1\or2\or3\or4\or5\or6}%
821 $ \BIC@Temp 4{16\or17\or18\or19\or0\or1\or2\or3\or4\or5}%
822 $ \BIC@Temp 5{15\or16\or17\or18\or19\or0\or1\or2\or3\or4}%
823 $ \BIC@Temp 6{14\or15\or16\or17\or18\or19\or0\or1\or2\or3}%
824 $ \BIC@Temp 7{13\or14\or15\or16\or17\or18\or19\or0\or1\or2}%
825 $ \BIC@Temp 8{12\or13\or14\or15\or16\or17\or18\or19\or0\or1}%
826 $ \BIC@Temp 9{11\or12\or13\or14\or15\or16\or17\or18\or19\or0}%

```

2.13 Shl, Shr

\bigintcalcShl

```

827 \def\bigintcalcShl#1{%
828 \romannumeral0%
829 \expandafter\expandafter\expandafter\BIC@Shl
830 \bigintcalcNum{#1}!%
831 }

```

\BIC@Shl

```

832 \def\BIC@Shl#1#2!{%
833 \ifx#1-%
834 \BIC@AfterFi{%
835 \expandafter-\romannumeral0%
836 & \BIC@@Shl#2!!%
837 $ \BIC@AddXY#2!#2!!!%
838 }%
839 \else
840 \BIC@AfterFi{%
841 & \BIC@@Shl#1#2!!%
842 $ \BIC@AddXY#1#2!#1#2!!!%
843 }%
844 \BIC@Fi
845 }

```

\BigIntCalcShl

```

846 \def\BigIntCalcShl#1!{%
847 \romannumeral0%
848 & \BIC@@Shl#1!!%
849 $ \BIC@AddXY#1!#1!!!%
850 }

```

\BIC@@Shl

```

851 & \def\BIC@@Shl#1#2!{%
852 & \ifx\#2\%
853 & \BIC@AfterFi{%
854 & \BIC@@@Shl0!#1%
855 & }%
856 & \else
857 & \BIC@AfterFi{%
858 & \BIC@@Shl#2!#1%

```

```

859 &    }%
860 &    \BIC@Fi
861 & }%

\BIC@@@Shl #1: carry
#2: result
#3#4: reverted number
862 & \def\BIC@@@Shl#1#2!#3#4!{%
863 &   \ifx\#4\%
864 &     \BIC@AfterFi{%
865 &       \expandafter\BIC@Space
866 &       \the\numexpr#3*2+#1\relax#2%
867 &     }%
868 &   \else
869 &     \BIC@AfterFi{%
870 &       \expandafter\BIC@@@Shl\the\numexpr#3*2+#1!#2!#4!%
871 &     }%
872 &   \BIC@Fi
873 & }%

\BIC@@@Shl
874 & \def\BIC@@@Shl#1!{%
875 &   \ifnum#1<10 %
876 &     \BIC@AfterFi{%
877 &       \BIC@@@Shl0#1%
878 &     }%
879 &   \else
880 &     \BIC@AfterFi{%
881 &       \BIC@@@Shl#1%
882 &     }%
883 &   \BIC@Fi
884 & }%

\bigintcalcShr
885 \def\bigintcalcShr#1{%
886   \romannumeral0%
887   \expandafter\expandafter\expandafter\BIC@Shr
888   \bigintcalcNum{#1}!%
889 }

\BIC@Shr
890 \def\BIC@Shr#1#2!{%
891   \ifx#1-%
892     \expandafter-\romannumeral0%
893     \BIC@AfterFi{%
894       \BIC@@Shr#2!%
895     }%
896   \else
897     \BIC@AfterFi{%
898       \BIC@@Shr#1#2!%
899     }%
900   \BIC@Fi
901 }

\BigIntCalcShr
902 \def\BigIntCalcShr#1!{%
903   \romannumeral0%
904   \BIC@@Shr#1!%
905 }

```

\BIC@@Shr

```
906 \def\BIC@@Shr#1#2!{%
907   \ifcase#1 %
908     \BIC@AfterFi{ 0}%
909   \or
910     \ifx\#2\%
911       \BIC@AfterFiFi{ 0}%
912     \else
913       \BIC@AfterFiFi{%
914         \BIC@@@Shr#1#2!!%
915       }%
916     \fi
917   \else
918     \BIC@AfterFi{%
919       \BIC@@@Shr0#1#2!!%
920     }%
921   \BIC@Fi
922 }
```

\BIC@@@Shr #1: carry
#2#3: number
#4: result

```
923 \def\BIC@@@Shr#1#2#3!#4!{%
924   \ifx\#3\%
925     \ifodd#1#2 %
926       \BIC@AfterFiFi{%
927         \expandafter\BIC@ShrResult\the\numexpr(#1#2-1)/2\relax
928       $ \expandafter\expandafter\expandafter\BIC@ShrResult
929       $ \csname BIC@ShrDigit#1#2\endcsname
930         #4!%
931       }%
932     \else
933       \BIC@AfterFiFi{%
934         \expandafter\BIC@ShrResult\the\numexpr#1#2/2\relax
935       $ \expandafter\expandafter\expandafter\BIC@ShrResult
936       $ \csname BIC@ShrDigit#1#2\endcsname
937         #4!%
938       }%
939     \fi
940   \else
941     \ifodd#1#2 %
942       \BIC@AfterFiFi{%
943         \expandafter\BIC@@@Shr\the\numexpr(#1#2-1)/2\relax1%
944       $ \expandafter\expandafter\expandafter\BIC@@@Shr
945       $ \csname BIC@ShrDigit#1#2\endcsname
946         #3!#4!%
947       }%
948     \else
949       \BIC@AfterFiFi{%
950         \expandafter\BIC@@@Shr\the\numexpr#1#2/2\relax0%
951       $ \expandafter\expandafter\expandafter\BIC@@@Shr
952       $ \csname BIC@ShrDigit#1#2\endcsname
953         #3!#4!%
954       }%
955     \fi
956   \BIC@Fi
957 }
```

```

\BIC@ShrResult
958 & \def\BIC@ShrResult#1#2!{ #2#1}%
959 $ \def\BIC@ShrResult#1#2#3!{ #3#1}%

```

```

\BIC@@@Shr #1: new digit
#2: carry
#3: remaining number
#4: result
960 \def\BIC@@@Shr#1#2#3!#4!{%
961 \BIC@@@Shr#2#3!#4#1!%
962 }

```

```

\BIC@ShrDigit[00-19]
963 $ \def\BIC@Temp#1#2#3#4{%
964 $ \expandafter\def\csname BIC@ShrDigit#1#2\endcsname{#3#4}%
965 $ }%
966 $ \BIC@Temp 0000%
967 $ \BIC@Temp 0101%
968 $ \BIC@Temp 0210%
969 $ \BIC@Temp 0311%
970 $ \BIC@Temp 0420%
971 $ \BIC@Temp 0521%
972 $ \BIC@Temp 0630%
973 $ \BIC@Temp 0731%
974 $ \BIC@Temp 0840%
975 $ \BIC@Temp 0941%
976 $ \BIC@Temp 1050%
977 $ \BIC@Temp 1151%
978 $ \BIC@Temp 1260%
979 $ \BIC@Temp 1361%
980 $ \BIC@Temp 1470%
981 $ \BIC@Temp 1571%
982 $ \BIC@Temp 1680%
983 $ \BIC@Temp 1781%
984 $ \BIC@Temp 1890%
985 $ \BIC@Temp 1991%

```

2.14 \BIC@Tim

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

#1: plain number without sign
#2: digit

```

\BIC@@Tim #1#2: number
#3: reverted number
986 \def\BIC@@Tim#1#2!{%
987 \ifx\#2\%
988 \BIC@AfterFi{%
989 \BIC@ProcessTim0!#1%
990 }%
991 \else
992 \BIC@AfterFi{%
993 \BIC@@Tim#2!#1%
994 }%
995 \BIC@Fi
996 }

```

```

\BIC@ProcessTim #1: carry
#2: result
#3#4: reverted number
#5: digit
997 \def\BIC@ProcessTim#1#2!#3#4!#5{%
998   \ifx\#4\%
999     \BIC@AfterFi{%
1000       \expandafter\BIC@Space
1001 &     \the\numexpr#3*#5+#1\relax
1002 $     \romannumeral0\BIC@TimDigit#3#5#1%
1003       #2%
1004     }%
1005   \else
1006     \BIC@AfterFi{%
1007       \expandafter\BIC@@ProcessTim
1008 &     \the\numexpr#3*#5+#1%
1009 $     \romannumeral0\BIC@TimDigit#3#5#1%
1010       !#2!#4!#5%
1011     }%
1012   \BIC@Fi
1013 }

\BIC@@ProcessTim #1#2: carry?, new digit
#3: new number
#4: old number
#5: digit
1014 \def\BIC@@ProcessTim#1#2!{%
1015   \ifx\#2\%
1016     \BIC@AfterFi{%
1017       \BIC@ProcessTim0#1%
1018     }%
1019   \else
1020     \BIC@AfterFi{%
1021       \BIC@ProcessTim#1#2%
1022     }%
1023   \BIC@Fi
1024 }

\BIC@TimDigit #1: digit 0–9
#2: digit 3–9
#3: carry 0–9
1025 $ \def\BIC@TimDigit#1#2#3{%
1026 $   \ifcase#1 % 0
1027 $     \BIC@AfterFi{ #3}%
1028 $   \or % 1
1029 $     \BIC@AfterFi{%
1030 $       \expandafter\BIC@Space
1031 $       \number\csname BIC@AddCarry#2\endcsname#3 %
1032 $     }%
1033 $   \else
1034 $     \ifcase#3 %
1035 $       \BIC@AfterFiFi{%
1036 $         \expandafter\BIC@Space
1037 $         \number\csname BIC@MulDigit#2\endcsname#1 %
1038 $       }%
1039 $     \else
1040 $       \BIC@AfterFiFi{%

```

```

1041 $ \expandafter\BIC@Space
1042 $ \romannumeral0%
1043 $ \expandafter\BIC@AddXY
1044 $ \number\csname BIC@MulDigit#2\endcsname#1!%
1045 $ #3!!!%
1046 $ }%
1047 $ \fi
1048 $ \BIC@Fi
1049 $ }%

```

\BIC@MulDigit[3-9]

```

1050 $ \def\BIC@Temp#1#2{%
1051 $ \expandafter\def\csname BIC@MulDigit#1\endcsname##1{%
1052 $ \ifcase##1 0%
1053 $ \or ##1%
1054 $ \or #2%
1055 $? \else\BigIntCalcError:ThisCannotHappen%
1056 $ \fi
1057 $ }%
1058 $ }%
1059 $ \BIC@Temp 3{6\or9\or12\or15\or18\or21\or24\or27}%
1060 $ \BIC@Temp 4{8\or12\or16\or20\or24\or28\or32\or36}%
1061 $ \BIC@Temp 5{10\or15\or20\or25\or30\or35\or40\or45}%
1062 $ \BIC@Temp 6{12\or18\or24\or30\or36\or42\or48\or54}%
1063 $ \BIC@Temp 7{14\or21\or28\or35\or42\or49\or56\or63}%
1064 $ \BIC@Temp 8{16\or24\or32\or40\or48\or56\or64\or72}%
1065 $ \BIC@Temp 9{18\or27\or36\or45\or54\or63\or72\or81}%

```

2.15 Mul

\bigintcalcMul

```

1066 \def\bigintcalcMul#1#2{%
1067 \romannumeral0%
1068 \expandafter\expandafter\expandafter\BIC@Mul
1069 \bigintcalcNum{#1}!{#2}%
1070 }

```

\BIC@Mul

```

1071 \def\BIC@Mul#1!#2{%
1072 \expandafter\expandafter\expandafter\BIC@MulSwitch
1073 \bigintcalcNum{#2}!#1!%
1074 }

```

\BIC@MulSwitch Decision table for \BIC@MulSwitch.

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

```

1075 \def\BIC@MulSwitch#1#2!#3#4!{%
1076   \ifcase\BIC@Sgn#1#2! % x = 0
1077     \BIC@AfterFi{ 0}%
1078   \or % x > 0
1079     \ifcase\BIC@Sgn#3#4! % y = 0
1080       \BIC@AfterFiFi{ 0}%
1081     \or % y > 0
1082       \ifnum\BIC@PosCmp#1#2!#3#4!=1 % x > y
1083         \BIC@AfterFiFiFi{%
1084           \BIC@ProcessMul0!#1#2!#3#4!%
1085         }%
1086       \else % x <= y
1087         \BIC@AfterFiFiFi{%
1088           \BIC@ProcessMul0!#3#4!#1#2!%
1089         }%
1090       \fi
1091     \else % y < 0
1092       \expandafter-\romannumeral0%
1093       \ifnum\BIC@PosCmp#1#2!#4!=1 % x > -y
1094         \BIC@AfterFiFiFi{%
1095           \BIC@ProcessMul0!#1#2!#4!%
1096         }%
1097       \else % x <= -y
1098         \BIC@AfterFiFiFi{%
1099           \BIC@ProcessMul0!#4!#1#2!%
1100         }%
1101       \fi
1102     \fi
1103   \else % x < 0
1104     \ifcase\BIC@Sgn#3#4! % y = 0
1105       \BIC@AfterFiFi{ 0}%
1106     \or % y > 0
1107       \expandafter-\romannumeral0%
1108       \ifnum\BIC@PosCmp#2!#3#4!=1 % -x > y
1109         \BIC@AfterFiFiFi{%
1110           \BIC@ProcessMul0!#2!#3#4!%
1111         }%
1112       \else % -x <= y
1113         \BIC@AfterFiFiFi{%
1114           \BIC@ProcessMul0!#3#4!#2!%
1115         }%
1116       \fi
1117     \else % y < 0
1118       \ifnum\BIC@PosCmp#2!#4!=1 % -x > -y
1119         \BIC@AfterFiFiFi{%
1120           \BIC@ProcessMul0!#2!#4!%
1121         }%
1122       \else % -x <= -y
1123         \BIC@AfterFiFiFi{%
1124           \BIC@ProcessMul0!#4!#2!%
1125         }%
1126       \fi
1127     \fi
1128   \BIC@Fi
1129 }

```

\BigIntCalcMul

```

1130 \def\BigIntCalcMul#1!#2!{%

```

```

1131 \romannumeral0%
1132 \BIC@ProcessMul0!#1!#2!%
1133 }

```

```

\BIC@ProcessMul #1: result
#2: number  $x$ 
#3#4: number  $y$ 
1134 \def\BIC@ProcessMul#1!#2!#3#4!{%
1135 \ifx\#4\%
1136 \BIC@AfterFi{%
1137 \expandafter\expandafter\expandafter\BIC@Space
1138 \bigintcalcAdd{\BIC@Tim#2!#3}{#10}%
1139 }%
1140 \else
1141 \BIC@AfterFi{%
1142 \expandafter\expandafter\expandafter\BIC@ProcessMul
1143 \bigintcalcAdd{\BIC@Tim#2!#3}{#10}!#2!#4!%
1144 }%
1145 \BIC@Fi
1146 }

```

2.16 Sqr

```

\bigintcalcSqr
1147 \def\bigintcalcSqr#1{%
1148 \romannumeral0%
1149 \expandafter\expandafter\expandafter\BIC@Sqr
1150 \bigintcalcNum{#1}!%
1151 }

\BIC@Sqr
1152 \def\BIC@Sqr#1{%
1153 \ifx#1-%
1154 \expandafter\BIC@@Sqr
1155 \else
1156 \expandafter\BIC@@Sqr\expandafter#1%
1157 \fi
1158 }

\BIC@@Sqr
1159 \def\BIC@@Sqr#1!{%
1160 \BIC@ProcessMul0!#1!#1!%
1161 }

```

2.17 Fac

```

\bigintcalcFac
1162 \def\bigintcalcFac#1{%
1163 \romannumeral0%
1164 \expandafter\expandafter\expandafter\BIC@Fac
1165 \bigintcalcNum{#1}!%
1166 }

\BIC@Fac
1167 \def\BIC@Fac#1#2!{%
1168 \ifx#1-%
1169 \BIC@AfterFi{ 0\BigIntCalcError:FacNegative}%

```

```

1170 \else
1171 \ifnum\BIC@PosCmp#1#2!13!<0 %
1172 \ifcase#1#2 %
1173 \BIC@AfterFiFiFi{ 1}% 0!
1174 \or\BIC@AfterFiFiFi{ 1}% 1!
1175 \or\BIC@AfterFiFiFi{ 2}% 2!
1176 \or\BIC@AfterFiFiFi{ 6}% 3!
1177 \or\BIC@AfterFiFiFi{ 24}% 4!
1178 \or\BIC@AfterFiFiFi{ 120}% 5!
1179 \or\BIC@AfterFiFiFi{ 720}% 6!
1180 \or\BIC@AfterFiFiFi{ 5040}% 7!
1181 \or\BIC@AfterFiFiFi{ 40320}% 8!
1182 \or\BIC@AfterFiFiFi{ 362880}% 9!
1183 \or\BIC@AfterFiFiFi{ 3628800}% 10!
1184 \or\BIC@AfterFiFiFi{ 39916800}% 11!
1185 \or\BIC@AfterFiFiFi{ 479001600}% 12!
1186 ? \else\BigIntCalcError:ThisCannotHappen%
1187 \fi
1188 \else
1189 \BIC@AfterFiFi{%
1190 \BIC@ProcessFac#1#2!479001600!%
1191 }%
1192 \fi
1193 \BIC@Fi
1194 }

```

```

\BIC@ProcessFac #1: n
#2: result
1195 \def\BIC@ProcessFac#1!#2!{%
1196 \ifnum\BIC@PosCmp#1!12!=0 %
1197 \BIC@AfterFi{ #2}%
1198 \else
1199 \BIC@AfterFi{%
1200 \expandafter\BIC@@ProcessFac
1201 \romannumeral0\BIC@ProcessMul0!#2!#1!%
1202 !#1!%
1203 }%
1204 \BIC@Fi
1205 }

```

```

\BIC@@ProcessFac #1: result
#2: n
1206 \def\BIC@@ProcessFac#1!#2!{%
1207 \expandafter\BIC@ProcessFac
1208 \romannumeral0\BIC@Dec#2!}%
1209 !#1!%
1210 }

```

2.18 Pow

```

\bigintcalcPow #1: basis
#2: power
1211 \def\bigintcalcPow#1{%
1212 \romannumeral0%
1213 \expandafter\expandafter\expandafter\BIC@Pow
1214 \bigintcalcNum{#1}!%
1215 }

```

```

\BIC@Pow #1: basis
#2: power
1216 \def\BIC@Pow#1!#2{%
1217 \expandafter\expandafter\expandafter\BIC@PowSwitch
1218 \bigintcalcNum{#2}!#1!%
1219 }

\BIC@PowSwitch #1#2: power  $y$ 
#3#4: basis  $x$ 
Decision table for \BIC@PowSwitch.

```

$y = 0$	1		
$y = 1$	x		
$y = 2$	$x < 0$	$\text{Mul}(-x, -x)$	
	else	$\text{Mul}(x, x)$	
$y < 0$	$x = 0$	DivisionByZero	
	$x = 1$	1	
	$x = -1$	ifodd(y)	-1
		else	1
	else ($ x > 1$)	0	
$y > 2$	$x = 0$	0	
	$x = 1$	1	
	$x = -1$	ifodd(y)	-1
		else	1
	$x < -1$ ($x < 0$)	ifodd(y)	$-\text{Pow}(-x, y)$
		else	$\text{Pow}(-x, y)$
	else ($x > 1$)	$\text{Pow}(x, y)$	

```

1220 \def\BIC@PowSwitch#1#2!#3#4!{%
1221 \ifcase\ifx\#2\%
1222 \ifx#100 %  $y = 0$ 
1223 \else\ifx#111 %  $y = 1$ 
1224 \else\ifx#122 %  $y = 2$ 
1225 \else4 %  $y > 2$ 
1226 \fi\fi\fi
1227 \else
1228 \ifx#1-3 %  $y < 0$ 
1229 \else4 %  $y > 2$ 
1230 \fi
1231 \fi
1232 \BIC@AfterFi{ 1}%  $y = 0$ 
1233 \or %  $y = 1$ 
1234 \BIC@AfterFi{ #3#4}%
1235 \or %  $y = 2$ 
1236 \ifx#3-%  $x < 0$ 
1237 \BIC@AfterFiFi{%
1238 \BIC@ProcessMulo!#4!#4!%
1239 }%
1240 \else %  $x \geq 0$ 
1241 \BIC@AfterFiFi{%
1242 \BIC@ProcessMulo!#3#4!#3#4!%
1243 }%
1244 \fi
1245 \or %  $y < 0$ 
1246 \ifcase\ifx\#4\%
1247 \ifx#300 %  $x = 0$ 

```

```

1248         \else\ifx#311 % x = 1
1249         \else3 % x > 1
1250         \fi\fi
1251     \else
1252         \ifcase\BIC@MinusOne#3#4! %
1253             3 % |x| > 1
1254         \or
1255             2 % x = -1
1256 ?         \else\BigIntCalcError:ThisCannotHappen%
1257         \fi
1258     \fi
1259     \BIC@AfterFiFi{ 0\BigIntCalcError:DivisionByZero}% x = 0
1260 \or % x = 1
1261     \BIC@AfterFiFi{ 1}% x = 1
1262 \or % x = -1
1263     \ifcase\BIC@ModTwo#2! % even(y)
1264         \BIC@AfterFiFiFi{ 1}%
1265     \or % odd(y)
1266         \BIC@AfterFiFiFi{ -1}%
1267 ?     \else\BigIntCalcError:ThisCannotHappen%
1268     \fi
1269 \or % |x| > 1
1270     \BIC@AfterFiFi{ 0}%
1271 ?     \else\BigIntCalcError:ThisCannotHappen%
1272     \fi
1273 \or % y > 2
1274     \ifcase\ifx\#4\%
1275         \ifx#300 % x = 0
1276         \else\ifx#311 % x = 1
1277         \else4 % x > 1
1278         \fi\fi
1279     \else
1280         \ifx#3-%
1281             \ifcase\BIC@MinusOne#3#4! %
1282                 3 % x < -1
1283             \else
1284                 2 % x = -1
1285             \fi
1286         \else
1287             4 % x > 1
1288         \fi
1289     \fi
1290     \BIC@AfterFiFi{ 0}% x = 0
1291 \or % x = 1
1292     \BIC@AfterFiFi{ 1}% x = 1
1293 \or % x = -1
1294     \ifcase\BIC@ModTwo#1#2! % even(y)
1295         \BIC@AfterFiFiFi{ 1}%
1296     \or % odd(y)
1297         \BIC@AfterFiFiFi{ -1}%
1298 ?     \else\BigIntCalcError:ThisCannotHappen%
1299     \fi
1300 \or % x < -1
1301     \ifcase\BIC@ModTwo#1#2! % even(y)
1302         \BIC@AfterFiFiFi{%
1303             \BIC@PowRec#4!#1#2!1!%
1304         }%
1305 \or % odd(y)

```

```

1306     \expandafter-\romannumeral0%
1307     \BIC@AfterFiFiFi{%
1308         \BIC@PowRec#4!#1#2!1!%
1309     }%
1310 ?   \else\BigIntCalcError:ThisCannotHappen%
1311     \fi
1312     \or % x > 1
1313     \BIC@AfterFiFi{%
1314         \BIC@PowRec#3#4!#1#2!1!%
1315     }%
1316 ?   \else\BigIntCalcError:ThisCannotHappen%
1317     \fi
1318 ? \else\BigIntCalcError:ThisCannotHappen%
1319     \BIC@Fi
1320 }

```

2.18.1 Help macros

`\BIC@ModTwo` Macro `\BIC@ModTwo` expects a number without sign and returns digit 1 or 0 if the number is odd or even.

```

1321 \def\BIC@ModTwo#1#2!{%
1322     \ifx\#2\%
1323         \ifodd#1 %
1324             \BIC@AfterFiFi1%
1325         \else
1326             \BIC@AfterFiFi0%
1327         \fi
1328     \else
1329         \BIC@AfterFi{%
1330             \BIC@ModTwo#2!%
1331         }%
1332     \BIC@Fi
1333 }

```

`\BIC@MinusOne` Macro `\BIC@MinusOne` expects a number and returns digit 1 if the number equals minus one and returns 0 otherwise.

```

1334 \def\BIC@MinusOne#1#2!{%
1335     \ifx#1-%
1336         \BIC@MinusOne#2!%
1337     \else
1338         0%
1339     \fi
1340 }

```

`\BIC@@MinusOne`

```

1341 \def\BIC@@MinusOne#1#2!{%
1342     \ifx#1%
1343         \ifx\#2\%
1344             1%
1345         \else
1346             0%
1347         \fi
1348     \else
1349         0%
1350     \fi
1351 }

```

2.18.2 Recursive calculation

```

\BIC@PowRec Pow(x, y) {
  PowRec(x, y, 1)
}
PowRec(x, y, r) {
  if y == 1 then
    return r
  else
    ifodd y then
      return PowRec(x*x, y div 2, r*x) % y div 2 = (y-1)/2
    else
      return PowRec(x*x, y div 2, r)
    fi
  fi
}
#1: x (basis)
#2#3: y (power)
#4: r (result)
1352 \def\BIC@PowRec#1!#2#3!#4!{%
1353   \ifcase\ifx#2!\ifx\#3\0 \else1 \fi\else1 \fi % y = 1
1354   \ifnum\BIC@PosCmp#1!#4!=1 % x > r
1355     \BIC@AfterFiFi{%
1356       \BIC@ProcessMul0!#1!#4!%
1357     }%
1358   \else
1359     \BIC@AfterFiFi{%
1360       \BIC@ProcessMul0!#4!#1!%
1361     }%
1362   \fi
1363 \or
1364   \ifcase\BIC@ModTwo#2#3! % even(y)
1365     \BIC@AfterFiFi{%
1366       \expandafter\BIC@@PowRec\romannumeral0%
1367       \BIC@@Shr#2#3!%
1368       !#1!#4!%
1369     }%
1370   \or % odd(y)
1371     \ifnum\BIC@PosCmp#1!#4!=1 % x > r
1372       \BIC@AfterFiFiFi{%
1373         \expandafter\BIC@@@PowRec\romannumeral0%
1374         \BIC@ProcessMul0!#1!#4!%
1375         !#1!#2#3!%
1376       }%
1377     \else
1378       \BIC@AfterFiFiFiFi{%
1379         \expandafter\BIC@@@PowRec\romannumeral0%
1380         \BIC@ProcessMul0!#1!#4!%
1381         !#1!#2#3!%
1382       }%
1383     \fi
1384 ? \else\BigIntCalcError:ThisCannotHappen%
1385   \fi
1386 ? \else\BigIntCalcError:ThisCannotHappen%
1387   \BIC@Fi
1388 }

\BIC@@PowRec #1: y/2

```

```

#2: x
#3: new r (r or r * x)
1389 \def\BIC@@PowRec#1!#2!#3!{%
1390   \expandafter\BIC@PowRec\romannumeral0%
1391   \BIC@ProcessMul0!#2!#2!%
1392   !#1!#3!%
1393 }

```

```

\BIC@@@PowRec #1: r * x #2: x #3: y
1394 \def\BIC@@@PowRec#1!#2!#3!{%
1395   \expandafter\BIC@@PowRec\romannumeral0%
1396   \BIC@@Shr#3!%
1397   !#2!#1!%
1398 }

```

2.19 Div

```

\bigintcalcDiv #1: x
#2: y (divisor)
1399 \def\bigintcalcDiv#1{%
1400   \romannumeral0%
1401   \expandafter\expandafter\expandafter\BIC@Div
1402   \bigintcalcNum{#1}!%
1403 }

```

```

\BIC@Div #1: x
#2: y
1404 \def\BIC@Div#1!#2!{%
1405   \expandafter\expandafter\expandafter\BIC@DivSwitchSign
1406   \bigintcalcNum{#2}!#1!%
1407 }

```

```

\BigIntCalcDiv
1408 \def\BigIntCalcDiv#1!#2!{%
1409   \romannumeral0%
1410   \BIC@DivSwitchSign#2!#1!%
1411 }

```

\BIC@DivSwitchSign Decision table for \BIC@DivSwitchSign.

$y = 0$	DivisionByZero	
$y > 0$	$x = 0$	0
	$x > 0$	DivSwitch(+, x, y)
	$x < 0$	DivSwitch(-, $-x, y$)
$y < 0$	$x = 0$	0
	$x > 0$	DivSwitch(-, $x, -y$)
	$x < 0$	DivSwitch(+, $-x, -y$)

```

#1: y (divisor)
#2: x
1412 \def\BIC@DivSwitchSign#1#2!#3#4!{%
1413   \ifcase\BIC@Sgn#1#2! % y = 0
1414     \BIC@AfterFi{ 0\BigIntCalcError:DivisionByZero}%
1415   \or % y > 0
1416     \ifcase\BIC@Sgn#3#4! % x = 0
1417       \BIC@AfterFiFi{ 0}%

```

```

1418 \or % x > 0
1419 \BIC@AfterFiFi{%
1420 \BIC@DivSwitch{ }#3#4!#1#2!%
1421 }%
1422 \else % x < 0
1423 \BIC@AfterFiFi{%
1424 \BIC@DivSwitch-#4!#1#2!%
1425 }%
1426 \fi
1427 \else % y < 0
1428 \ifcase\BIC@Sgn#3#4! % x = 0
1429 \BIC@AfterFiFi{ 0}%
1430 \or % x > 0
1431 \BIC@AfterFiFi{%
1432 \BIC@DivSwitch-#3#4!#2!%
1433 }%
1434 \else % x < 0
1435 \BIC@AfterFiFi{%
1436 \BIC@DivSwitch{ }#4!#2!%
1437 }%
1438 \fi
1439 \BIC@Fi
1440 }

```

\BIC@DivSwitch Decision table for \BIC@DivSwitch.

$y = x$	sign 1	
$y > x$	0	
$y < x$	$y = 1$	sign x
	$y = 2$	sign Shr(x)
	$y = 4$	sign Shr(Shr(x))
	else	sign ProcessDiv(x, y)

#1: sign

#2: x

#3#4: y ($y \neq 0$)

```

1441 \def\BIC@DivSwitch#1#2!#3#4!{%
1442 \ifcase\BIC@PosCmp#3#4!#2!% y = x
1443 \BIC@AfterFi{ #11}%
1444 \or % y > x
1445 \BIC@AfterFi{ 0}%
1446 \else % y < x
1447 \ifx\#1\%
1448 \else
1449 \expandafter-\romannumeral0%
1450 \fi
1451 \ifcase\ifx\#4\%
1452 \ifx#310 % y = 1
1453 \else\ifx#321 % y = 2
1454 \else\ifx#342 % y = 4
1455 \else3 % y > 2
1456 \fi\fi\fi
1457 \else
1458 3 % y > 2
1459 \fi
1460 \BIC@AfterFiFi{ #2}% y = 1
1461 \or % y = 2
1462 \BIC@AfterFiFi{%

```

```

1463     \BIC@Shr#2!%
1464     }%
1465     \or % y = 4
1466     \BIC@AfterFiFi{%
1467     \expandafter\BIC@Shr\romannumeral0%
1468     \BIC@Shr#2!!%
1469     }%
1470     \or % y > 2
1471     \BIC@AfterFiFi{%
1472     \BIC@DivStartX#2!#3#4!!!%
1473     }%
1474 ? \else\BigIntCalcError:ThisCannotHappen%
1475     \fi
1476 \BIC@Fi
1477 }

\BIC@ProcessDiv #1#2:  $x$ 
#3#4:  $y$ 
#5: collect first digits of  $x$ 
#6: corresponding digits of  $y$ 
1478 \def\BIC@DivStartX#1#2!#3#4!#5!#6!{%
1479 \ifx\#4\%
1480 \BIC@AfterFi{%
1481 \BIC@DivStartYii#6#3#4!#5#1!#2=!!%
1482 }%
1483 \else
1484 \BIC@AfterFi{%
1485 \BIC@DivStartX#2!#4!#5#1!#6#3!%
1486 }%
1487 \BIC@Fi
1488 }

\BIC@DivStartYii #1:  $y$ 
#2:  $x, =$ 
1489 \def\BIC@DivStartYii#1!{%
1490 \expandafter\BIC@DivStartYiv\romannumeral0%
1491 \BIC@Shl#1!%
1492 !#1!%
1493 }

\BIC@DivStartYiv #1:  $2y$ 
#2:  $y$ 
#3:  $x, =$ 
1494 \def\BIC@DivStartYiv#1!{%
1495 \expandafter\BIC@DivStartYvi\romannumeral0%
1496 \BIC@Shl#1!%
1497 !#1!%
1498 }

\BIC@DivStartYvi #1:  $4y$ 
#2:  $2y$ 
#3:  $y$ 
#4:  $x, =$ 
1499 \def\BIC@DivStartYvi#1!#2!{%
1500 \expandafter\BIC@DivStartYviii\romannumeral0%
1501 \BIC@AddXY#1!#2!!!%
1502 !#1!#2!%
1503 }

```

```

\BIC@DivStartYviii #1: 6y
                   #2: 4y
                   #3: 2y
                   #4: y
                   #5: x, =
1504 \def\BIC@DivStartYviii#1!#2!{%
1505   \expandafter\BIC@DivStart\romannumeral0%
1506   \BIC@Sh1#2!%
1507   !#1!#2!%
1508 }

\BIC@DivStart #1: 8y
              #2: 6y
              #3: 4y
              #4: 2y
              #5: y
              #6: x, =
1509 \def\BIC@DivStart#1!#2!#3!#4!#5!#6!{%
1510   \BIC@ProcessDiv#6!!#5!#4!#3!#2!#1!=%
1511 }

\BIC@ProcessDiv #1#2#3: x, =
                #4: result
                #5: y
                #6: 2y
                #7: 4y
                #8: 6y
                #9: 8y
1512 \def\BIC@ProcessDiv#1#2#3!#4!#5!{%
1513   \ifcase\BIC@PosCmp#5!#1!% y = #1
1514     \ifx#2=%
1515       \BIC@AfterFiFi{\BIC@DivCleanup{#41}}}%
1516     \else
1517       \BIC@AfterFiFi{%
1518         \BIC@ProcessDiv#2#3!#41!#5!%
1519       }%
1520     \fi
1521   \or % y > #1
1522     \ifx#2=%
1523       \BIC@AfterFiFi{\BIC@DivCleanup{#40}}}%
1524     \else
1525       \ifx\#4\%
1526         \BIC@AfterFiFiFi{%
1527           \BIC@ProcessDiv{#1#2}#3!#5!%
1528         }%
1529       \else
1530         \BIC@AfterFiFiFi{%
1531           \BIC@ProcessDiv{#1#2}#3!#40!#5!%
1532         }%
1533       \fi
1534     \fi
1535   \else % y < #1
1536     \BIC@AfterFiFi{%
1537       \BIC@ProcessDiv{#1}#2#3!#4!#5!%
1538     }%
1539   \BIC@Fi
1540 }

```

```

\BIC@DivCleanup #1: result
#2: garbage
1541 \def\BIC@DivCleanup#1#2={ #1}%

\BIC@@ProcessDiv
1542 \def\BIC@@ProcessDiv#1#2#3!#4!#5!#6!#7!{%
1543 \ifcase\BIC@PosCmp#7!#1!% 4y = #1
1544 \ifx#2=%
1545 \BIC@AfterFiFi{\BIC@DivCleanup{#44}}}%
1546 \else
1547 \BIC@AfterFiFi{%
1548 \BIC@ProcessDiv#2#3!#44!#5!#6!#7!%
1549 }%
1550 \fi
1551 \or % 4y > #1
1552 \ifcase\BIC@PosCmp#6!#1!% 2y = #1
1553 \ifx#2=%
1554 \BIC@AfterFiFiFi{\BIC@DivCleanup{#42}}}%
1555 \else
1556 \BIC@AfterFiFiFi{%
1557 \BIC@ProcessDiv#2#3!#42!#5!#6!#7!%
1558 }%
1559 \fi
1560 \or % 2y > #1
1561 \ifx#2=%
1562 \BIC@AfterFiFiFi{\BIC@DivCleanup{#41}}}%
1563 \else
1564 \BIC@AfterFiFiFi{%
1565 \BIC@DivSub#1!#5!#2#3!#41!#5!#6!#7!%
1566 }%
1567 \fi
1568 \else % 2y < #1
1569 \BIC@AfterFiFi{%
1570 \expandafter\BIC@ProcessDivII\romannumeral0%
1571 \BIC@SubXY#1!#6!!!%
1572 !#2#3!#4!#5!23%
1573 #6!#7!%
1574 }%
1575 \fi
1576 \else % 4y < #1
1577 \BIC@AfterFiFi{%
1578 \BIC@@@ProcessDiv{#1}#2#3!#4!#5!#6!#7!%
1579 }%
1580 \BIC@Fi
1581 }

\BIC@DivSub Next token group: #1-#2 and next digit #3.
1582 \def\BIC@DivSub#1!#2!#3{%
1583 \expandafter\BIC@ProcessDiv\expandafter{%
1584 \romannumeral0%
1585 \BIC@SubXY#1!#2!!!%
1586 #3%
1587 }%
1588 }

\BIC@ProcessDivII #1:  $x' - 2y$ 
#2#3: remaining  $x$ , =
#4: result

```

```

#5:  $y$ 
#6: first possible result digit
#7: second possible result digit
1589 \def\BIC@ProcessDivII#1!#2#3!#4!#5!#6#7{%
1590   \ifcase\BIC@PosCmp#5!#1!%  $y = \#1$ 
1591     \ifx#2=%
1592       \BIC@AfterFiFi{\BIC@DivCleanup{#4#7}}%
1593     \else
1594       \BIC@AfterFiFi{%
1595         \BIC@ProcessDiv#2#3!#4#7!#5!%
1596       }%
1597     \fi
1598   \or %  $y > \#1$ 
1599     \ifx#2=%
1600       \BIC@AfterFiFi{\BIC@DivCleanup{#4#6}}%
1601     \else
1602       \BIC@AfterFiFi{%
1603         \BIC@ProcessDiv{#1#2}#3!#4#6!#5!%
1604       }%
1605     \fi
1606   \else %  $y < \#1$ 
1607     \ifx#2=%
1608       \BIC@AfterFiFi{\BIC@DivCleanup{#4#7}}%
1609     \else
1610       \BIC@AfterFiFi{%
1611         \BIC@DivSub#1!#5!#2#3!#4#7!#5!%
1612       }%
1613     \fi
1614   \BIC@Fi
1615 }

```

```

\BIC@ProcessDivIV #1#2#3:  $x, =, x > 4y$ 
#4: result
#5:  $y$ 
#6:  $2y$ 
#7:  $4y$ 
#8:  $6y$ 
#9:  $8y$ 
1616 \def\BIC@@@ProcessDiv#1#2#3!#4!#5!#6!#7!#8!#9!{%
1617   \ifcase\BIC@PosCmp#8!#1!%  $6y = \#1$ 
1618     \ifx#2=%
1619       \BIC@AfterFiFi{\BIC@DivCleanup{#46}}%
1620     \else
1621       \BIC@AfterFiFi{%
1622         \BIC@ProcessDiv#2#3!#46!#5!#6!#7!#8!#9!%
1623       }%
1624     \fi
1625   \or %  $6y > \#1$ 
1626     \BIC@AfterFiFi{%
1627       \expandafter\BIC@ProcessDivII\romannumeral0%
1628       \BIC@SubXY#1!#7!!%
1629       !#2#3!#4!#5!45%
1630       #6!#7!#8!#9!%
1631     }%
1632   \else %  $6y < \#1$ 
1633     \ifcase\BIC@PosCmp#9!#1!%  $8y = \#1$ 
1634       \ifx#2=%

```

```

1635     \BIC@AfterFiFiFi{\BIC@DivCleanup{#48}}%
1636     \else
1637     \BIC@AfterFiFiFi{%
1638     \BIC@ProcessDiv#2#3!#48!#5!#6!#7!#8!#9!%
1639     }%
1640     \fi
1641     \or % 8y > #1
1642     \BIC@AfterFiFi{%
1643     \expandafter\BIC@ProcessDivII\romannumeral0%
1644     \BIC@SubXY#1!#8!!!%
1645     !#2#3!#4!#5!67%
1646     #6!#7!#8!#9!%
1647     }%
1648     \else % 8y < #1
1649     \BIC@AfterFiFi{%
1650     \expandafter\BIC@ProcessDivII\romannumeral0%
1651     \BIC@SubXY#1!#9!!!%
1652     !#2#3!#4!#5!89%
1653     #6!#7!#8!#9!%
1654     }%
1655     \fi
1656     \BIC@Fi
1657 }

```

2.20 Mod

```

\bigintcalcMod #1: x
#2: y
1658 \def\bigintcalcMod#1{%
1659   \romannumeral0%
1660   \expandafter\expandafter\expandafter\BIC@Mod
1661   \bigintcalcNum{#1}!%
1662 }

\BIC@Mod #1: x
#2: y
1663 \def\BIC@Mod#1!#2{%
1664   \expandafter\expandafter\expandafter\BIC@ModSwitchSign
1665   \bigintcalcNum{#2}!#1!%
1666 }

\BigIntCalcMod
1667 \def\BigIntCalcMod#1!#2!{%
1668   \romannumeral0%
1669   \BIC@ModSwitchSign#2!#1!%
1670 }

```

\BIC@ModSwitchSign Decision table for \BIC@ModSwitchSign.

$y = 0$	DivisionByZero	
$y > 0$	$x = 0$	0
	else	ModSwitch(+, x , y)
$y < 0$	ModSwitch(-, $-x$, $-y$)	

```

#1#2: y
#3#4: x
1671 \def\BIC@ModSwitchSign#1#2!#3#4!{%

```

```

1672 \ifcase\ifx\|#2\|%
1673     \ifx#100 % y = 0
1674     \else1 % y > 0
1675     \fi
1676     \else
1677     \ifx#1-2 % y < 0
1678     \else1 % y > 0
1679     \fi
1680     \fi
1681     \BIC@AfterFi{ 0\BigIntCalcError:DivisionByZero}%
1682 \or % y > 0
1683 \ifcase\ifx\|#4\|\ifx#300 \else1 \fi\else1 \fi % x = 0
1684     \BIC@AfterFiFi{ 0}%
1685     \else
1686     \BIC@AfterFiFi{%
1687     \BIC@ModSwitch{ }#3#4!#1#2!%
1688     }%
1689     \fi
1690 \else % y < 0
1691 \ifcase\ifx\|#4\|%
1692     \ifx#300 % x = 0
1693     \else1 % x > 0
1694     \fi
1695     \else
1696     \ifx#3-2 % x < 0
1697     \else1 % x > 0
1698     \fi
1699     \fi
1700     \BIC@AfterFiFi{ 0}%
1701 \or % x > 0
1702     \BIC@AfterFiFi{%
1703     \BIC@ModSwitch--#3#4!#2!%
1704     }%
1705 \else % x < 0
1706     \BIC@AfterFiFi{%
1707     \BIC@ModSwitch-#4!#2!%
1708     }%
1709     \fi
1710 \BIC@Fi
1711 }

```

\BIC@ModSwitch Decision table for \BIC@ModSwitch.

$y = 1$		0
$y = 2$	ifodd(x)	sign 1
	else	0
$y > 2$	$x < 0$	$z \leftarrow x - (x/y) * y; (z < 0) ? z + y : z$
	$x > 0$	$x - (x/y) * y$

#1: sign

#2#3: x

#4#5: y

```

1712 \def\BIC@ModSwitch#1#2#3!#4#5!{%
1713 \ifcase\ifx\|#5\|%
1714     \ifx#410 % y = 1
1715     \else\ifx#421 % y = 2
1716     \else2 % y > 2
1717     \fi\fi

```

```

1718         \else2 % y > 2
1719         \fi
1720     \BIC@AfterFi{ 0}% y = 1
1721 \or % y = 2
1722     \ifcase\BIC@ModTwo#2#3! % even(x)
1723         \BIC@AfterFiFi{ 0}%
1724     \or % odd(x)
1725         \BIC@AfterFiFi{ #11}%
1726 ? \else\BigIntCalcError:ThisCannotHappen%
1727     \fi
1728 \or % y > 2
1729     \ifx\#1\%
1730     \else
1731         \expandafter\BIC@Space\romannumeral0%
1732         \expandafter\BIC@ModMinus\romannumeral0%
1733     \fi
1734     \ifx#2-% x < 0
1735         \BIC@AfterFiFi{%
1736             \expandafter\expandafter\expandafter\BIC@ModX
1737             \bigintcalcSub{#2#3}{%
1738                 \bigintcalcMul{#4#5}{\bigintcalcDiv{#2#3}{#4#5}}%
1739             }!#4#5!%
1740         }%
1741     \else % x > 0
1742         \BIC@AfterFiFi{%
1743             \expandafter\expandafter\expandafter\BIC@Space
1744             \bigintcalcSub{#2#3}{%
1745                 \bigintcalcMul{#4#5}{\bigintcalcDiv{#2#3}{#4#5}}%
1746             }%
1747         }%
1748     \fi
1749 ? \else\BigIntCalcError:ThisCannotHappen%
1750     \BIC@Fi
1751 }

\BIC@ModMinus
1752 \def\BIC@ModMinus#1{%
1753     \ifx#10%
1754         \BIC@AfterFi{ 0}%
1755     \else
1756         \BIC@AfterFi{ -#1}%
1757     \BIC@Fi
1758 }

\BIC@ModX #1#2: z
#3: x
1759 \def\BIC@ModX#1#2!#3!{%
1760     \ifx#1-% z < 0
1761         \BIC@AfterFi{%
1762             \expandafter\BIC@Space\romannumeral0%
1763             \BIC@SubXY#3!#2!!!%
1764         }%
1765     \else % z >= 0
1766         \BIC@AfterFi{ #1#2}%
1767     \BIC@Fi
1768 }

1769 \BIC@AtEnd%
1770 \end{package}

```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

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

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

Bundle. All the packages of the bundle ‘bigintcalc’ 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/bigintcalc.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 `bigintcalc.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

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

3.3 Package installation

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

```
tex bigintcalc.dtx
```

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

```
bigintcalc.sty → tex/generic/bigintcalc/bigintcalc.sty
bigintcalc.pdf → doc/latex/bigintcalc/bigintcalc.pdf
bigintcalc.dtx → source/latex/bigintcalc/bigintcalc.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 T_EX distribution (T_EX Live, MiK_TE_X, ...) relies on file name databases, you must refresh these. For example, T_EX Live users run `texhash` or `mktexlsr`.

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.

¹[CTAN:pkg/bigintcalc](#)

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

```
latex \let\install=y\input{bigintcalc.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 `pdf \LaTeX` :

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

4 History

[2007/09/27 v1.0]

- First version.

[2007/11/11 v1.1]

- Use of package `pdftexcmds` for `Lua \TeX` support.

[2011/01/30 v1.2]

- Already loaded package files are not input in plain `\TeX` .

[2012/04/08 v1.3]

- Fix: `pdftexcmds` wasn't loaded in case of `\LaTeX` .

[2016/05/16 v1.4]

- Documentation updates.

[2019/12/15 v1.5]

- 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 \@nil . 155, 157, 164, 172, 187, 190, 195
\@firstoftwo . . . . . 159, 167, 176
```


\BIC@Expand [153](#), [183](#), [250](#)
 \BIC@Fac [1164](#), [1167](#)
 \BIC@Fi [130](#), [131](#), [132](#), [133](#), [197](#),
 [227](#), [242](#), [315](#), [339](#), [363](#), [384](#),
 [404](#), [423](#), [442](#), [459](#), [493](#), [506](#),
 [517](#), [553](#), [570](#), [638](#), [673](#), [688](#),
 [715](#), [764](#), [777](#), [807](#), [844](#), [860](#),
 [872](#), [883](#), [900](#), [921](#), [956](#), [995](#),
 [1012](#), [1023](#), [1048](#), [1128](#), [1145](#),
 [1193](#), [1204](#), [1319](#), [1332](#), [1387](#),
 [1439](#), [1476](#), [1487](#), [1539](#), [1580](#),
 [1614](#), [1656](#), [1710](#), [1750](#), [1757](#), [1767](#)
 \BIC@Inc [416](#), [440](#), [445](#), [450](#)
 \BIC@IncSwitch [408](#), [411](#)
 \BIC@MinMax [367](#), [372](#), [375](#)
 \BIC@MinusOne [1252](#), [1281](#), [1334](#)
 \BIC@Mod [1660](#), [1663](#)
 \BIC@ModMinus [1732](#), [1752](#)
 \BIC@ModSwitch [1687](#), [1703](#), [1707](#), [1712](#)
 \BIC@ModSwitchSign [1664](#), [1669](#), [1671](#)
 \BIC@ModTwo
 [1263](#), [1294](#), [1301](#), [1321](#), [1364](#), [1722](#)
 \BIC@ModX [1736](#), [1759](#)
 \BIC@Mul [1068](#), [1071](#)
 \BIC@MulDigit[3-9] [1050](#)
 \BIC@MulSwitch [1072](#), [1075](#)
 \BIC@Normalize [200](#), [247](#)
 \BIC@NormalizeDigits [223](#), [239](#), [244](#)
 \BIC@NormalizeZero [219](#), [229](#)
 \BIC@Odd [388](#), [393](#), [395](#)
 \BIC@PosCmp [317](#), [590](#), [600](#), [615](#),
 [628](#), [1082](#), [1093](#), [1108](#), [1118](#),
 [1171](#), [1196](#), [1354](#), [1371](#), [1442](#),
 [1513](#), [1543](#), [1552](#), [1590](#), [1617](#), [1633](#)
 \BIC@Pow [1213](#), [1216](#)
 \BIC@PowRec [1303](#), [1308](#), [1314](#), [1352](#), [1390](#)
 \BIC@PowSwitch [1217](#), [1220](#)
 \BIC@ProcessDiv
 [1478](#), [1510](#), [1512](#), [1548](#),
 [1557](#), [1583](#), [1595](#), [1603](#), [1622](#), [1638](#)
 \BIC@ProcessDivII
 [1570](#), [1589](#), [1627](#), [1643](#), [1650](#)
 \BIC@ProcessDivIV [1616](#)
 \BIC@ProcessFac [1190](#), [1195](#), [1207](#)
 \BIC@ProcessMul [1084](#), [1088](#), [1095](#),
 [1099](#), [1110](#), [1114](#), [1120](#), [1124](#),
 [1132](#), [1134](#), [1160](#), [1201](#), [1238](#),
 [1242](#), [1356](#), [1360](#), [1374](#), [1380](#), [1391](#)
 \BIC@ProcessTim [989](#), [997](#), [1017](#), [1021](#)
 \BIC@Sgn [271](#), [274](#), [431](#),
 [1076](#), [1079](#), [1104](#), [1413](#), [1416](#), [1428](#)
 \BIC@Shl [829](#), [832](#), [1491](#), [1496](#), [1506](#)
 \BIC@Shr [887](#), [890](#)
 \BIC@ShrDigit[00-19] [963](#)
 \BIC@ShrResult [927](#), [928](#), [934](#), [935](#), [958](#)
 \BIC@Space [134](#),
 [184](#), [253](#), [263](#), [265](#), [678](#), [692](#),
 [694](#), [783](#), [802](#), [865](#), [1000](#), [1030](#),
 [1036](#), [1041](#), [1137](#), [1731](#), [1743](#), [1762](#)
 \BIC@Sqr [1149](#), [1152](#)
 \BIC@StripHexSpace [187](#), [190](#)
 \BIC@SubCarry0 [809](#)
 \BIC@SubCarry10 [810](#)
 \BIC@SubCarry[1-9] [811](#)
 \BIC@SubDigit [770](#), [775](#), [786](#)
 \BIC@SubResult [769](#), [779](#)
 \BIC@SubXY
 [605](#), [609](#), [619](#), [624](#), [644](#), [737](#),
 [1571](#), [1585](#), [1628](#), [1644](#), [1651](#), [1763](#)
 \BIC@Temp [719](#), [727](#),
 [728](#), [729](#), [730](#), [731](#), [732](#), [733](#),
 [734](#), [735](#), [736](#), [811](#), [818](#), [819](#),
 [820](#), [821](#), [822](#), [823](#), [824](#), [825](#),
 [826](#), [963](#), [966](#), [967](#), [968](#), [969](#),
 [970](#), [971](#), [972](#), [973](#), [974](#), [975](#),
 [976](#), [977](#), [978](#), [979](#), [980](#), [981](#),
 [982](#), [983](#), [984](#), [985](#), [1050](#), [1059](#),
 [1060](#), [1061](#), [1062](#), [1063](#), [1064](#), [1065](#)
 \BIC@TestMode [123](#)
 \BIC@Tim [986](#), [1138](#), [1143](#)
 \BIC@TimDigit [1002](#), [1009](#), [1025](#)
 \bigintcalcAbs [4](#), [256](#), [389](#)
 \BigIntCalcAdd [7](#), [640](#)
 \bigintcalcAdd [5](#), [572](#), [1138](#), [1143](#)
 \bigintcalcCmp [5](#), [285](#)
 \BigIntCalcDec [7](#), [447](#)
 \bigintcalcDec [5](#), [425](#)
 \BigIntCalcDiv [8](#), [1408](#)
 \bigintcalcDiv [7](#), [1399](#), [1738](#), [1745](#)
 \BigIntCalcError [484](#),
 [534](#), [544](#), [723](#), [814](#), [1055](#), [1169](#),
 [1186](#), [1256](#), [1259](#), [1267](#), [1271](#),
 [1298](#), [1310](#), [1316](#), [1318](#), [1384](#),
 [1386](#), [1414](#), [1474](#), [1681](#), [1726](#), [1749](#)
 \bigintcalcFac [6](#), [1162](#)
 \BigIntCalcInc [7](#), [444](#)
 \bigintcalcInc [5](#), [406](#)
 \bigintcalcInv [4](#), [252](#)
 \bigintcalcMax [4](#), [370](#)
 \bigintcalcMin [4](#), [365](#)
 \BigIntCalcMod [8](#), [1667](#)
 \bigintcalcMod [7](#), [1658](#)
 \BigIntCalcMul [8](#), [1130](#)
 \bigintcalcMul [6](#), [1066](#), [1738](#), [1745](#)
 \bigintcalcNum [3](#), [245](#), [254](#), [259](#),
 [272](#), [288](#), [292](#), [368](#), [373](#), [377](#),
 [409](#), [428](#), [575](#), [579](#), [584](#), [830](#),
 [888](#), [1069](#), [1073](#), [1150](#), [1165](#),
 [1214](#), [1218](#), [1402](#), [1406](#), [1661](#), [1665](#)
 \BigIntCalcOdd [7](#), [391](#)
 \bigintcalcOdd [5](#), [386](#)
 \bigintcalcPow [6](#), [1211](#)
 \bigintcalcSgn [4](#), [269](#)
 \BigIntCalcShl [8](#), [846](#)
 \bigintcalcShl [6](#), [827](#)

<code>\BigIntCalcShr</code>	8, 902	1224, 1228, 1236, 1246, 1247,
<code>\bigintcalcShr</code>	6, 885	1248, 1274, 1275, 1276, 1280,
<code>\bigintcalcSqr</code>	6, 1147	1322, 1335, 1342, 1343, 1353,
<code>\BigIntCalcSub</code>	7, 643	1447, 1451, 1452, 1453, 1454,
<code>\bigintcalcSub</code>	5, 581 , 1737 , 1744	1479, 1514, 1522, 1525, 1544,
		1553, 1561, 1591, 1599, 1607,
		1618, 1634, 1672, 1673, 1677,
		1683, 1691, 1692, 1696, 1713,
		1714, 1715, 1729, 1734, 1753, 1760
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, 121, 126, 128	
<code>\csname</code>	14, 21, 50, 66,	
	76, 119, 125, 140, 143, 152, 176,	
	179, 701, 717, 718, 720, 790,	
	809, 810, 812, 929, 936, 945,	
	952, 964, 1031, 1037, 1044, 1051	
E		
<code>\empty</code>	17, 18	
<code>\endcsname</code>	14, 21, 50, 66,	
	76, 119, 125, 140, 143, 152, 176,	
	179, 708, 717, 718, 720, 797,	
	809, 810, 812, 929, 936, 945,	
	952, 964, 1031, 1037, 1044, 1051	
<code>\endinput</code>	29, 117	
<code>\endlinechar</code>	4, 35, 71, 77, 89	
I		
<code>\ifcase</code>	412, 431, 462, 475, 520, 533,	
	557, 600, 615, 703, 706, 721,	
	792, 795, 813, 907, 1026, 1034,	
	1052, 1076, 1079, 1104, 1172,	
	1221, 1246, 1252, 1263, 1274,	
	1281, 1294, 1301, 1353, 1364,	
	1413, 1416, 1428, 1442, 1451,	
	1513, 1543, 1552, 1590, 1617,	
	1633, 1672, 1683, 1691, 1713, 1722	
<code>\ifcat</code>	158	
<code>\ifnum</code>	343,	
	348, 380, 471, 497, 529, 590,	
	628, 711, 800, 875, 1082, 1093,	
	1108, 1118, 1171, 1196, 1354, 1371	
<code>\ifodd</code>	397, 925, 941, 1323	
<code>\ifx</code>	15, 18, 21,	
	50, 58, 61, 119, 125, 140, 143,	
	152, 166, 176, 179, 192, 201,	
	202, 212, 217, 230, 233, 262,	
	275, 278, 295, 296, 306, 321,	
	322, 330, 353, 396, 451, 463,	
	496, 509, 521, 556, 558, 587,	
	588, 614, 647, 648, 658, 659,	
	676, 691, 738, 739, 749, 750,	
	767, 780, 833, 852, 863, 891,	
	910, 924, 987, 998, 1015, 1135,	
	1153, 1168, 1221, 1222, 1223,	
		1224, 1228, 1236, 1246, 1247,
		1248, 1274, 1275, 1276, 1280,
		1322, 1335, 1342, 1343, 1353,
		1447, 1451, 1452, 1453, 1454,
		1479, 1514, 1522, 1525, 1544,
		1553, 1561, 1591, 1599, 1607,
		1618, 1634, 1672, 1673, 1677,
		1683, 1691, 1692, 1696, 1713,
		1714, 1715, 1729, 1734, 1753, 1760
<code>\immediate</code>	23, 52	
<code>\input</code>	144	
N		
<code>\number</code>		270, 286, 700, 789, 1031, 1037, 1044
<code>\numexpr</code>	473,	531, 679, 699, 788, 803, 866,
		870, 927, 934, 943, 950, 1001, 1008
P		
<code>\PackageInfo</code>	26	
<code>\pdf@escapehex</code>	187	
<code>\pdf@unescapehex</code>	185	
<code>\ProvidesPackage</code>	19, 67	
R		
<code>\RequirePackage</code>	149	
<code>\romannumeral</code>		154, 184, 246, 253, 257, 366,
		371, 387, 392, 407, 420, 426,
		439, 445, 448, 573, 582, 589,
		603, 622, 641, 644, 698, 787,
		828, 835, 847, 886, 892, 903,
		1002, 1009, 1042, 1067, 1092,
		1107, 1131, 1148, 1163, 1201,
		1208, 1212, 1306, 1366, 1373,
		1379, 1390, 1395, 1400, 1409,
		1449, 1467, 1490, 1495, 1500,
		1505, 1570, 1584, 1627, 1643,
		1650, 1659, 1668, 1731, 1732, 1762
T		
<code>\the</code>	77, 78,	79, 80, 81, 82, 83, 84, 97, 473,
		531, 679, 699, 788, 803, 866,
		870, 927, 934, 943, 950, 1001, 1008
<code>\TMP@EnsureCode</code>	94, 101, 102, 103,	104, 105, 106, 107, 108, 109,
		110, 111, 112, 113, 114, 115, 116
<code>\TMP@RequirePackage</code>	141, 147	
W		
<code>\write</code>	23, 52	
X		
<code>\x</code>	14, 15, 18, 22,	26, 28, 51, 56, 66, 75, 87, 135, 138