

The bigintcalc package

Heiko Oberdiek*

2016/05/16 v1.4

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	3
1.2.1	Preconditions	3
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	\BIC@Tim	29
2.15	Mul	31

*Please report any issues at <https://github.com/ho-tex/oberdiek/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	Test	48
3.1	Catcode checks for loading	48
3.2	Macro tests	49
3.2.1	Preamble with test macro definitions	49
3.2.2	Time	53
3.2.3	Test sets	54
4	Installation	63
4.1	Download	63
4.2	Bundle installation	64
4.3	Package installation	64
4.4	Refresh file name databases	64
4.5	Some details for the interested	64
5	History	65
[2007/09/27 v1.0]		65
[2007/11/11 v1.1]		65
[2011/01/30 v1.2]		65
[2012/04/08 v1.3]		65
[2016/05/16 v1.4]		65

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 \TeX '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 \TeX numbers like count registers or commands created by `\chardef`. Nevertheless they may be used, if they are prefixed by `\number`.

Also ε - \TeX '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 \TeX , 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 expandable 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>}
\jj$x=0$
\or
\jj$x>0$
\else
\jj$x<0$
\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>}
\case$=y$
\or
\case$x>y$
\else
\case$x<y$
\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 \LaTeX .

```
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 [2016/05/16 v1.4 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 -T_EX 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@AfterFi{%
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	+	$\text{Sub}(y, -x)$
else	$y < 0$	$x > -y$	+	$\text{Sub}(x, -y)$
		$x = -y$		0
		else	-	$\text{Sub}(-y, x)$
	else	$x > y$	+	$\text{Add}(x, y)$
		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@AfterFiFiFi{%
650                 \BIC@DoAdd0!#1#5!#60!%
651             }%
652         \else
653             \BIC@AfterFiFiFi{%
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@AfterFiFiFi{%
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@AfterFiFi{%
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@AfterFiFi{%

```



```

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@ProcessMul0!#4!#4!%
1239     }%
1240   \else %  $x \geq 0$ 
1241     \BIC@AfterFiFi{%
1242       \BIC@ProcessMul0!#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#11%
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
                  \if odd 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@AfterFiFi{%
1373         \expandafter\BIC@@PowRec\romannumeral0%
1374         \BIC@ProcessMul0!#1!#4!%
1375         !#1!#2#3!%
1376       }%
1377     \else
1378       \BIC@AfterFiFiFi{%
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{ #1}%
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!#4!#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@AfterFiFiFiFi{%
1531           \BIC@ProcessDiv{#1#2}#3!#40!#5!%
1532         }%
1533       \fi
1534     \fi
1535   \else % y < #1
1536     \BIC@AfterFi{%
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@AfterFi{%
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	
$y = 2$	$\text{ifodd}(x)$	sign 1
	else	0
$y > 2$	$x < 0$	$z \leftarrow x - (x/y) * y; \quad (z < 0) ? z + y : z$
	$x > 0$	$x - (x/y) * y$

#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 </package>

```

3 Test

3.1 Catcode checks for loading

```
1771 ⟨*test1⟩
1772 \catcode`\{=1 %
1773 \catcode`\}=2 %
1774 \catcode`\#=6 %
1775 \catcode`\@=11 %
1776 \expandafter\ifx\csname count@\endcsname\relax
1777 \countdef\count@=255 %
1778 \fi
1779 \expandafter\ifx\csname @gobble\endcsname\relax
1780 \long\def\@gobble#1{}%
1781 \fi
1782 \expandafter\ifx\csname @firstofone\endcsname\relax
1783 \long\def\@firstofone#1{#1}%
1784 \fi
1785 \expandafter\ifx\csname loop\endcsname\relax
1786 \expandafter\@firstofone
1787 \else
1788 \expandafter\@gobble
1789 \fi
1790 {%
1791 \def\loop#1\repeat{%
1792 \def\body{#1}%
1793 \iterate
1794 }%
1795 \def\iterate{%
1796 \body
1797 \let\next\iterate
1798 \else
1799 \let\next\relax
1800 \fi
1801 \next
1802 }%
1803 \let\repeat=\fi
1804 }%
1805 \def\RestoreCatcodes{}
1806 \count@=0 %
1807 \loop
1808 \edef\RestoreCatcodes{%
1809 \RestoreCatcodes
1810 \catcode\the\count@=\the\catcode\count@\relax
1811 }%
1812 \ifnum\count@<255 %
1813 \advance\count@ 1 %
1814 \repeat
1815
1816 \def\RangeCatcodeInvalid#1#2{%
1817 \count@=#1\relax
1818 \loop
1819 \catcode\count@=15 %
1820 \ifnum\count@<#2\relax
1821 \advance\count@ 1 %
1822 \repeat
1823 }
1824 \def\RangeCatcodeCheck#1#2#3{%
```



```

1825 \count@=#1\relax
1826 \loop
1827   \ifnum#3=\catcode\count@
1828   \else
1829     \errmessage{%
1830       Character \the\count@\space
1831       with wrong catcode \the\catcode\count@\space
1832       instead of \number#3%
1833     }%
1834   \fi
1835 \ifnum\count@<#2\relax
1836   \advance\count@ 1 %
1837 \repeat
1838 }
1839 \def\space{ }
1840 \expandafter\ifx\csname LoadCommand\endcsname\relax
1841   \def\LoadCommand{\input bigintcalc.sty\relax}%
1842 \fi
1843 \def\Test{%
1844   \RangeCatcodeInvalid{0}{47}%
1845   \RangeCatcodeInvalid{58}{64}%
1846   \RangeCatcodeInvalid{91}{96}%
1847   \RangeCatcodeInvalid{123}{255}%
1848   \catcode`\@=12 %
1849   \catcode`\=0 %
1850   \catcode`\%=14 %
1851   \LoadCommand
1852   \RangeCatcodeCheck{0}{36}{15}%
1853   \RangeCatcodeCheck{37}{37}{14}%
1854   \RangeCatcodeCheck{38}{47}{15}%
1855   \RangeCatcodeCheck{48}{57}{12}%
1856   \RangeCatcodeCheck{58}{63}{15}%
1857   \RangeCatcodeCheck{64}{64}{12}%
1858   \RangeCatcodeCheck{65}{90}{11}%
1859   \RangeCatcodeCheck{91}{91}{15}%
1860   \RangeCatcodeCheck{92}{92}{0}%
1861   \RangeCatcodeCheck{93}{96}{15}%
1862   \RangeCatcodeCheck{97}{122}{11}%
1863   \RangeCatcodeCheck{123}{255}{15}%
1864   \RestoreCatcodes
1865 }
1866 \Test
1867 \csname @@end\endcsname
1868 \end
1869 </test1>

```

3.2 Macro tests

3.2.1 Preamble with test macro definitions

```

1870 (*test2)
1871 \NeedsTeXFormat{LaTeX2e}
1872 \nofiles
1873 \documentclass{article}
1874 (noetex)\let\SavedNumexpr\numexpr
1875 (noetex)\let\numexpr\UNDEFINED
1876 \makeatletter
1877 \chardef\BIC@TestMode=1 %
1878 \makeatother

```

```

1879 \usepackage{bigintcalc}[2016/05/16]
1880 <noetex>\let\numexpr\SavedNumexpr
1881 \usepackage{qstest}
1882 \IncludeTests{*}
1883 \LogTests{log}{*}{*}
1884 \newcommand*{\TestSpaceAtEnd}[1]{%
1885 <noetex> \let\SavedNumexpr\numexpr
1886 <noetex> \let\numexpr\UNDEFINED
1887 \edef\resultA{#1}%
1888 \edef\resultB{#1 }%
1889 <noetex> \let\numexpr\SavedNumexpr
1890 \Expect*{\resultA\space}*{\resultB}%
1891 }
1892 \newcommand*{\TestResult}[2]{%
1893 <noetex> \let\SavedNumexpr\numexpr
1894 <noetex> \let\numexpr\UNDEFINED
1895 \edef\result{#1}%
1896 <noetex> \let\numexpr\SavedNumexpr
1897 \Expect*{\result}{#2}%
1898 }
1899 \newcommand*{\TestResultTwoExpansions}[2]{%
1900 <noetex>
1901 \begin{group}
1902 \let\numexpr\UNDEFINED
1903 \expandafter\expandafter\expandafter
1904 \end{group}
1905 /<noetex>
1906 \expandafter\expandafter\expandafter\Expect
1907 \expandafter\expandafter\expandafter{#1}{#2}%
1908 }
1909 \newcount\TestCount
1910 <etex>\newcommand*{\TestArg}[1]{\numexpr#1\relax}
1911 <noetex>\newcommand*{\TestArg}[1]{#1}
1912 \newcommand*{\TestTeXDivide}[2]{%
1913 \TestCount=\TestArg{#1}\relax
1914 \divide\TestCount by \TestArg{#2}\relax
1915 \Expect*{\bigintcalcDiv{#1}{#2}}*{\the\TestCount}%
1916 }
1917 \newcommand*{\Test}[2]{%
1918 \TestResult{#1}{#2}%
1919 \TestResultTwoExpansions{#1}{#2}%
1920 \TestSpaceAtEnd{#1}%
1921 }
1922 \newcommand*{\TestExch}[2]{\Test{#2}{#1}}
1923 \newcommand*{\TestInv}[2]{%
1924 \Test{\bigintcalcInv{#1}}{#2}%
1925 }
1926 \newcommand*{\TestAbs}[2]{%
1927 \Test{\bigintcalcAbs{#1}}{#2}%
1928 }
1929 \newcommand*{\TestSgn}[2]{%
1930 \Test{\bigintcalcSgn{#1}}{#2}%
1931 }
1932 \newcommand*{\TestMin}[3]{%
1933 \Test{\bigintcalcMin{#1}{#2}}{#3}%
1934 }
1935 \newcommand*{\TestMax}[3]{%
1936 \Test{\bigintcalcMax{#1}{#2}}{#3}%

```

```

1937 }
1938 \newcommand*\TestCmp}[3]{%
1939   \Test{\bigintcalcCmp{#1}{#2}}{#3}%
1940 }
1941 \newcommand*\TestOdd}[2]{%
1942   \Test{\bigintcalcOdd{#1}}{#2}%
1943   \edef\x{%
1944     \noexpand\Test{%
1945       \noexpand\BigIntCalcOdd
1946       \bigintcalcAbs{#1}!%
1947     }{#2}%
1948   }%
1949   \x
1950 }
1951 \newcommand*\TestInc}[2]{%
1952   \Test{\bigintcalcInc{#1}}{#2}%
1953   \ifnum\bigintcalcSgn{#1}>-1 %
1954     \edef\x{%
1955       \noexpand\Test{%
1956         \noexpand\BigIntCalcInc\bigintcalcNum{#1}!%
1957       }{#2}%
1958     }%
1959     \x
1960   \fi
1961 }
1962 \newcommand*\TestDec}[2]{%
1963   \Test{\bigintcalcDec{#1}}{#2}%
1964   \ifnum\bigintcalcSgn{#1}>0 %
1965     \edef\x{%
1966       \noexpand\Test{%
1967         \noexpand\BigIntCalcDec\bigintcalcNum{#1}!%
1968       }{#2}%
1969     }%
1970     \x
1971   \fi
1972 }
1973 \newcommand*\TestAdd}[3]{%
1974   \Test{\bigintcalcAdd{#1}{#2}}{#3}%
1975   \ifnum\bigintcalcSgn{#1}>0 %
1976     \ifnum\bigintcalcSgn{#2}> 0 %
1977       \ifnum\bigintcalcCmp{#1}{#2}>0 %
1978         \edef\x{%
1979           \noexpand\Test{%
1980             \noexpand\BigIntCalcAdd
1981             \bigintcalcNum{#1}!\bigintcalcNum{#2}!%
1982           }{#3}%
1983         }%
1984         \x
1985       \else
1986         \edef\x{%
1987           \noexpand\Test{%
1988             \noexpand\BigIntCalcAdd
1989             \bigintcalcNum{#2}!\bigintcalcNum{#1}!%
1990           }{#3}%
1991         }%
1992         \x
1993       \fi
1994     \fi

```

```

1995 \fi
1996 }
1997 \newcommand*\TestSub}[3]{%
1998 \Test{\bigintcalcSub{#1}{#2}}{#3}%
1999 \ifnum\bigintcalcSgn{#1}>0 %
2000 \ifnum\bigintcalcSgn{#2}> 0 %
2001 \ifnum\bigintcalcCmp{#1}{#2}>0 %
2002 \edef\x{%
2003 \noexpand\Test{%
2004 \noexpand\BigIntCalcSub
2005 \bigintcalcNum{#1}!\bigintcalcNum{#2}!%
2006 }{#3}%
2007 }%
2008 \x
2009 \fi
2010 \fi
2011 \fi
2012 }
2013 \newcommand*\TestShl}[2]{%
2014 \Test{\bigintcalcShl{#1}}{#2}%
2015 \edef\x{%
2016 \noexpand\Test{%
2017 \noexpand\BigIntCalcShl\bigintcalcAbs{#1}!%
2018 }{\bigintcalcAbs{#2}}%
2019 }%
2020 \x
2021 }
2022 \newcommand*\TestShr}[2]{%
2023 \Test{\bigintcalcShr{#1}}{#2}%
2024 \edef\x{%
2025 \noexpand\Test{%
2026 \noexpand\BigIntCalcShr\bigintcalcAbs{#1}!%
2027 }{\bigintcalcAbs{#2}}%
2028 }%
2029 \x
2030 }
2031 \newcommand*\TestMul}[3]{%
2032 \Test{\bigintcalcMul{#1}{#2}}{#3}%
2033 \edef\x{%
2034 \noexpand\Test{%
2035 \noexpand\BigIntCalcMul
2036 \bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
2037 }{\bigintcalcAbs{#3}}%
2038 }%
2039 \x
2040 }
2041 \newcommand*\TestSqr}[2]{%
2042 \Test{\bigintcalcSqr{#1}}{#2}%
2043 }
2044 \newcommand*\TestFac}[2]{%
2045 \expandafter\TestExch\expandafter{%
2046 \the\numexpr#2%
2047 }{\bigintcalcFac{#1}}%
2048 }
2049 \newcommand*\TestFacBig}[2]{%
2050 \Test{\bigintcalcFac{#1}}{#2}%
2051 }
2052 \newcommand*\TestPow}[3]{%

```

```

2053 \Test{\bigintcalcPow{#1}{#2}}{#3}%
2054 }
2055 \newcommand*{\TestDiv}[3]{%
2056 \Test{\bigintcalcDiv{#1}{#2}}{#3}%
2057 \TestTeXDivide{#1}{#2}%
2058 }
2059 \newcommand*{\TestDivBig}[3]{%
2060 \Test{\bigintcalcDiv{#1}{#2}}{#3}%
2061 \edef\x{%
2062 \noexpand\Test{%
2063 \noexpand\BigIntCalcDiv\bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
2064 }\bigintcalcAbs{#3}}%
2065 }%
2066 }
2067 \newcommand*{\TestMod}[3]{%
2068 \Test{\bigintcalcMod{#1}{#2}}{#3}%
2069 \ifcase\ifcase\bigintcalcSgn{#1} 0%
2070 \or
2071 \ifcase\bigintcalcSgn{#2} 1%
2072 \or 0%
2073 \else 1%
2074 \fi
2075 \else
2076 \ifcase\bigintcalcSgn{#2} 1%
2077 \or 1%
2078 \else 0%
2079 \fi
2080 \fi\relax
2081 \edef\x{%
2082 \noexpand\Test{%
2083 \noexpand\BigIntCalcMod
2084 \bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
2085 }\bigintcalcAbs{#3}}%
2086 }%
2087 \x
2088 \fi
2089 }

```

3.2.2 Time

```

2090 \begingroup\expandafter\expandafter\expandafter\endgroup
2091 \expandafter\ifx\csname pdfresettimer\endcsname\relax
2092 \else
2093 \makeatletter
2094 \newcount\SummaryTime
2095 \newcount\TestTime
2096 \SummaryTime=\z@
2097 \newcommand*{\PrintTime}[2]{%
2098 \typeout{%
2099 [Time #1: \strip@pt\dimexpr\number#2sp\relax\space s}%
2100 }%
2101 }%
2102 \newcommand*{\StartTime}[1]{%
2103 \renewcommand*{\TimeDescription}{#1}%
2104 \pdfresettimer
2105 }%
2106 \newcommand*{\TimeDescription}{}%
2107 \newcommand*{\StopTime}{%
2108 \TestTime=\pdfelapseddtime

```

```

2109 \global\advance\SummaryTime\TestTime
2110 \PrintTime\TimeDescription\TestTime
2111 }%
2112 \let\saved@qstest\qstest
2113 \let\saved@endqstest\endqstest
2114 \def\qstest#1#2{%
2115 \saved@qstest{#1}{#2}%
2116 \StartTime{#1}%
2117 }%
2118 \def\endqstest{%
2119 \StopTime
2120 \saved@endqstest
2121 }%
2122 \AtEndDocument{%
2123 \PrintTime{summary}\SummaryTime
2124 }%
2125 \makeatother
2126 \fi

```

3.2.3 Test sets

```

2127 \makeatletter
2128
2129 \begin{qstest}{inv}{inv}%
2130 \TestInv{0}{0}%
2131 \TestInv{1}{-1}%
2132 \TestInv{-1}{1}%
2133 \TestInv{10}{-10}%
2134 \TestInv{-10}{10}%
2135 \TestInv{2147483647}{-2147483647}%
2136 \TestInv{-2147483647}{2147483647}%
2137 \TestInv{12345678901234567890}{-12345678901234567890}%
2138 \TestInv{-12345678901234567890}{12345678901234567890}%
2139 \TestInv{ 0 }{0}%
2140 \TestInv{ 1 }{-1}%
2141 \TestInv{--1}{-1}%
2142 \TestInv{\number\z0}{0}%
2143 \TestInv{\ifx\relax\relax1\fi}{-1}%
2144 \TestInv{\ifx\relax\relax-\fi\ifx234\else1\fi}{1}%
2145 \end{qstest}
2146
2147 \begin{qstest}{abs}{abs}%
2148 \TestAbs{0}{0}%
2149 \TestAbs{1}{1}%
2150 \TestAbs{-1}{1}%
2151 \TestAbs{10}{10}%
2152 \TestAbs{-10}{10}%
2153 \TestAbs{2147483647}{2147483647}%
2154 \TestAbs{-2147483647}{2147483647}%
2155 \TestAbs{12345678901234567890}{12345678901234567890}%
2156 \TestAbs{-12345678901234567890}{12345678901234567890}%
2157 \TestAbs{ 0 }{0}%
2158 \TestAbs{ 1 }{1}%
2159 \TestAbs{--1}{1}%
2160 \TestAbs{--++1}{1}%
2161 \TestAbs{00000000000}{0}%
2162 \TestAbs{00000001000}{1000}%
2163 \TestAbs{\ifx\relax\relax 0\else 1\fi}{0}%
2164 \end{qstest}

```

```

2165
2166 \begin{qstest}{sign}{sign}%
2167   \TestSgn{0}{0}%
2168   \TestSgn{1}{1}%
2169   \TestSgn{-1}{-1}%
2170   \TestSgn{10}{1}%
2171   \TestSgn{-10}{-1}%
2172   \TestSgn{2147483647}{1}%
2173   \TestSgn{-2147483647}{-1}%
2174   \TestSgn{12345678901234567890}{1}%
2175   \TestSgn{-12345678901234567890}{-1}%
2176   \TestSgn{ 0 }{0}%
2177   \TestSgn{ 2 }{1}%
2178   \TestSgn{ -2 }{-1}%
2179   \TestSgn{-2}{1}%
2180   \TestSgn{\number\z@}{0}%
2181   \TestSgn{\number\@ne}{1}%
2182   \TestSgn{\number\m@ne}{-1}%
2183   \TestSgn{%
2184     -+-+\number\z@\number\z@
2185     \iftrue1\fi\iftrue2\fi\iftrue3\fi
2186   }{1}%
2187 \end{qstest}
2188
2189 \begin{qstest}{min}{min}%
2190   \TestMin{0}{1}{0}%
2191   \TestMin{1}{0}{0}%
2192   \TestMin{-10}{-20}{-20}%
2193   \TestMin{ 1 }{ 2 }{1}%
2194   \TestMin{ 2 }{ 1 }{1}%
2195   \TestMin{1}{1}{1}%
2196   \TestMin{\number\z@}{\number\@ne}{0}%
2197   \TestMin{\number\@ne}{\number\m@ne}{-1}%
2198 \end{qstest}
2199
2200 \begin{qstest}{max}{max}%
2201   \TestMax{0}{1}{1}%
2202   \TestMax{1}{0}{1}%
2203   \TestMax{-10}{-20}{-10}%
2204   \TestMax{ 1 }{ 2 }{2}%
2205   \TestMax{ 2 }{ 1 }{2}%
2206   \TestMax{1}{1}{1}%
2207   \TestMax{\number\z@}{\number\@ne}{1}%
2208   \TestMax{\number\@ne}{\number\m@ne}{1}%
2209 \end{qstest}
2210
2211 \begin{qstest}{cmp}{cmp}%
2212   \TestCmp{0}{0}{0}%
2213   \TestCmp{-21}{17}{-1}%
2214   \TestCmp{3}{4}{-1}%
2215   \TestCmp{-10}{-10}{0}%
2216   \TestCmp{-10}{-11}{1}%
2217   \TestCmp{100}{5}{1}%
2218   \TestCmp{9}{10}{-1}%
2219   \TestCmp{10}{9}{1}%
2220   \TestCmp{ 3 }{ 3 }{0}%
2221   \TestCmp{-9}{-10}{1}%
2222   \TestCmp{-10}{-9}{-1}%

```

```

2223 \TestCmp{-3}{-3}{0}%
2224 \TestCmp{0}{-2}{1}%
2225 \TestCmp{0}{2}{-1}%
2226 \TestCmp{2}{0}{1}%
2227 \TestCmp{-2}{0}{-1}%
2228 \TestCmp{12}{11}{1}%
2229 \TestCmp{11}{12}{-1}%
2230 \TestCmp{2147483647}{-2147483647}{1}%
2231 \TestCmp{-2147483647}{2147483647}{-1}%
2232 \TestCmp{2147483647}{2147483647}{0}%
2233 \TestCmp{\number\z@}{\number\@ne}{-1}%
2234 \TestCmp{\number\@ne}{\number\m@ne}{1}%
2235 \TestCmp{ 4 }{ 5 }{-1}%
2236 \TestCmp{ -3 }{ -7 }{1}%
2237 \end{qstest}
2238
2239 \begin{qstest}{odd}{odd}
2240 \tracingmacros=1
2241 \TestOdd{0}{0}%
2242 \TestOdd{1}{1}%
2243 \TestOdd{2}{0}%
2244 \TestOdd{3}{1}%
2245 \TestOdd{14}{0}%
2246 \TestOdd{15}{1}%
2247 \TestOdd{12345678901234567896}{0}%
2248 \TestOdd{12345678901234567897}{1}%
2249 \end{qstest}
2250
2251 \begin{qstest}{inc}{inc}%
2252 \TestInc{0}{1}%
2253 \TestInc{1}{2}%
2254 \TestInc{-1}{0}%
2255 \TestInc{10}{11}%
2256 \TestInc{-10}{-9}%
2257 \TestInc{ 3 }{4}%
2258 \TestInc{999}{1000}%
2259 \TestInc{-1000}{-999}%
2260 \TestInc{129}{130}%
2261 \TestInc{2147483646}{2147483647}%
2262 \TestInc{-2147483647}{-2147483646}%
2263 \TestInc{12345678901234567890}{12345678901234567891}%
2264 \TestInc{9999999999999999999}{10000000000000000000}%
2265 \TestInc{-12345678901234567891}{-12345678901234567890}%
2266 \TestInc{-10000000000000000000}{-9999999999999999999}%
2267 \end{qstest}
2268
2269 \begin{qstest}{dec}{dec}%
2270 \TestDec{0}{-1}%
2271 \TestDec{1}{0}%
2272 \TestDec{-1}{-2}%
2273 \TestDec{10}{9}%
2274 \TestDec{-10}{-11}%
2275 \TestDec{1000}{999}%
2276 \TestDec{-999}{-1000}%
2277 \TestDec{130}{129}%
2278 \TestDec{2147483647}{2147483646}%
2279 \TestDec{-2147483646}{-2147483647}%
2280 \TestDec{12345678901234567891}{12345678901234567890}%

```



```

2281 \TestDec{10000000000000000000}{999999999999999999}%
2282 \TestDec{-12345678901234567890}{-12345678901234567891}%
2283 \TestDec{-999999999999999999}{-1000000000000000000}%
2284 \end{qstest}
2285
2286 \begin{qstest}{add}{add}%
2287 \TestAdd{0}{0}{0}%
2288 \TestAdd{1}{0}{1}%
2289 \TestAdd{0}{1}{1}%
2290 \TestAdd{1}{2}{3}%
2291 \TestAdd{-1}{-1}{-2}%
2292 \TestAdd{2147483646}{1}{2147483647}%
2293 \TestAdd{-2147483647}{2147483647}{0}%
2294 \TestAdd{20}{-5}{15}%
2295 \TestAdd{-4}{-1}{-5}%
2296 \TestAdd{-1}{-4}{-5}%
2297 \TestAdd{-4}{1}{-3}%
2298 \TestAdd{-1}{4}{3}%
2299 \TestAdd{4}{-1}{3}%
2300 \TestAdd{1}{-4}{-3}%
2301 \TestAdd{-4}{-1}{-5}%
2302 \TestAdd{-1}{-4}{-5}%
2303 \TestAdd{ -4 }{ -1 }{-5}%
2304 \TestAdd{ -1 }{ -4 }{-5}%
2305 \TestAdd{ -4 }{ 1 }{-3}%
2306 \TestAdd{ -1 }{ 4 }{3}%
2307 \TestAdd{ 4 }{ -1 }{3}%
2308 \TestAdd{ 1 }{ -4 }{-3}%
2309 \TestAdd{ -4 }{ -1 }{-5}%
2310 \TestAdd{ -1 }{ -4 }{-5}%
2311 \TestAdd{876543210}{111111111}{987654321}%
2312 \TestAdd{999999999}{2}{1000000001}%
2313 \end{qstest}
2314
2315 \begin{qstest}{sub}{sub}
2316 \TestSub{0}{0}{0}%
2317 \TestSub{1}{0}{1}%
2318 \TestSub{1}{2}{-1}%
2319 \TestSub{-1}{-1}{0}%
2320 \TestSub{2147483646}{-1}{2147483647}%
2321 \TestSub{-2147483647}{-2147483647}{0}%
2322 \TestSub{-4}{-1}{-3}%
2323 \TestSub{-1}{-4}{3}%
2324 \TestSub{-4}{1}{-5}%
2325 \TestSub{-1}{4}{-5}%
2326 \TestSub{4}{-1}{5}%
2327 \TestSub{1}{-4}{5}%
2328 \TestSub{-4}{-1}{-3}%
2329 \TestSub{-1}{-4}{3}%
2330 \TestSub{ -4 }{ -1 }{-3}%
2331 \TestSub{ -1 }{ -4 }{3}%
2332 \TestSub{ -4 }{ 1 }{-5}%
2333 \TestSub{ -1 }{ 4 }{-5}%
2334 \TestSub{ 4 }{ -1 }{5}%
2335 \TestSub{ 1 }{ -4 }{5}%
2336 \TestSub{ -4 }{ -1 }{-3}%
2337 \TestSub{ -1 }{ -4 }{3}%
2338 \TestSub{1000000000}{2}{999999998}%

```

```

2339 \TestSub{987654321}{111111111}{876543210}%
2340 \end{qstest}
2341
2342 \begin{qstest}{shl}{shl}
2343 \TestShl{0}{0}%
2344 \TestShl{1}{2}%
2345 \TestShl{2}{4}%
2346 \TestShl{5621}{11242}%
2347 \TestShl{1073741823}{2147483646}%
2348 \end{qstest}
2349
2350 \begin{qstest}{shr}{shr}
2351 \TestShr{0}{0}%
2352 \TestShr{1}{0}%
2353 \TestShr{2}{1}%
2354 \TestShr{3}{1}%
2355 \TestShr{4}{2}%
2356 \TestShr{5}{2}%
2357 \TestShr{6}{3}%
2358 \TestShr{7}{3}%
2359 \TestShr{8}{4}%
2360 \TestShr{9}{4}%
2361 \TestShr{10}{5}%
2362 \TestShr{11}{5}%
2363 \TestShr{12}{6}%
2364 \TestShr{13}{6}%
2365 \TestShr{14}{7}%
2366 \TestShr{15}{7}%
2367 \TestShr{16}{8}%
2368 \TestShr{17}{8}%
2369 \TestShr{18}{9}%
2370 \TestShr{19}{9}%
2371 \TestShr{20}{10}%
2372 \TestShr{21}{10}%
2373 \TestShr{22}{11}%
2374 \TestShr{11241}{5620}%
2375 \TestShr{73054202}{36527101}%
2376 \TestShr{2147483646}{1073741823}%
2377 \end{qstest}
2378
2379 \begin{qstest}{mul}{mul}
2380 \TestMul{0}{0}{0}%
2381 \TestMul{1}{0}{0}%
2382 \TestMul{0}{1}{0}%
2383 \TestMul{1}{1}{1}%
2384 \TestMul{3}{1}{3}%
2385 \TestMul{1}{-3}{-3}%
2386 \TestMul{-4}{-5}{20}%
2387 \TestMul{3}{7}{21}%
2388 \TestMul{7}{3}{21}%
2389 \TestMul{3}{-7}{-21}%
2390 \TestMul{7}{-3}{-21}%
2391 \TestMul{-3}{7}{-21}%
2392 \TestMul{-7}{3}{-21}%
2393 \TestMul{-3}{-7}{21}%
2394 \TestMul{-7}{-3}{21}%
2395 \TestMul{12}{11}{132}%
2396 \TestMul{999}{333}{332667}%

```

```

2397 \TestMul{1000}{4321}{4321000}%
2398 \TestMul{12345}{173955}{2147474475}%
2399 \TestMul{1073741823}{2}{2147483646}%
2400 \TestMul{2}{1073741823}{2147483646}%
2401 \TestMul{-1073741823}{2}{-2147483646}%
2402 \TestMul{2}{-1073741823}{-2147483646}%
2403 \TestMul{6706022400}{13}{87178291200}%
2404 \end{qstest}
2405
2406 \begin{qstest}{sqr}{sqr}
2407 \TestSqr{0}{0}%
2408 \TestSqr{1}{1}%
2409 \TestSqr{2}{4}%
2410 \TestSqr{3}{9}%
2411 \TestSqr{4}{16}%
2412 \TestSqr{9}{81}%
2413 \TestSqr{10}{100}%
2414 \TestSqr{46340}{2147395600}%
2415 \TestSqr{-1}{1}%
2416 \TestSqr{-2}{4}%
2417 \TestSqr{-46340}{2147395600}%
2418 \end{qstest}
2419
2420 \begin{qstest}{fac}{fac}
2421 \TestFac{0}{1}%
2422 \TestFac{1}{1}%
2423 \TestFac{2}{2}%
2424 \TestFac{3}{2*3}%
2425 \TestFac{4}{2*3*4}%
2426 \TestFac{5}{2*3*4*5}%
2427 \TestFac{6}{2*3*4*5*6}%
2428 \TestFac{7}{2*3*4*5*6*7}%
2429 \TestFac{8}{2*3*4*5*6*7*8}%
2430 \TestFac{9}{2*3*4*5*6*7*8*9}%
2431 \TestFac{10}{2*3*4*5*6*7*8*9*10}%
2432 \TestFac{11}{2*3*4*5*6*7*8*9*10*11}%
2433 \TestFac{12}{2*3*4*5*6*7*8*9*10*11*12}%
2434 \TestFacBig{13}{6227020800}%
2435 \TestFacBig{14}{87178291200}%
2436 \TestFacBig{15}{1307674368000}%
2437 \TestFacBig{16}{20922789888000}%
2438 \TestFacBig{17}{355687428096000}%
2439 \TestFacBig{18}{6402373705728000}%
2440 \TestFacBig{19}{121645100408832000}%
2441 \TestFacBig{20}{2432902008176640000}%
2442 \TestFacBig{21}{51090942171709440000}%
2443 \TestFacBig{22}{112400072777607680000}%
2444 \end{qstest}
2445
2446 \begin{qstest}{pow}{pow}
2447 \TestPow{-2}{0}{1}%
2448 \TestPow{-1}{0}{1}%
2449 \TestPow{0}{0}{1}%
2450 \TestPow{1}{0}{1}%
2451 \TestPow{2}{0}{1}%
2452 \TestPow{3}{0}{1}%
2453 \TestPow{-2}{1}{-2}%
2454 \TestPow{-1}{1}{-1}%

```

```

2455 \TestPow{1}{1}{1}%
2456 \TestPow{2}{1}{2}%
2457 \TestPow{3}{1}{3}%
2458 \TestPow{-2}{2}{4}%
2459 \TestPow{-1}{2}{1}%
2460 \TestPow{0}{2}{0}%
2461 \TestPow{1}{2}{1}%
2462 \TestPow{2}{2}{4}%
2463 \TestPow{3}{2}{9}%
2464 \TestPow{0}{1}{0}%
2465 \TestPow{1}{-2}{1}%
2466 \TestPow{1}{-1}{1}%
2467 \TestPow{-1}{-2}{1}%
2468 \TestPow{-1}{-1}{-1}%
2469 \TestPow{-1}{3}{-1}%
2470 \TestPow{-1}{4}{1}%
2471 \TestPow{-2}{-1}{0}%
2472 \TestPow{-2}{-2}{0}%
2473 \TestPow{2}{3}{8}%
2474 \TestPow{2}{4}{16}%
2475 \TestPow{2}{5}{32}%
2476 \TestPow{2}{6}{64}%
2477 \TestPow{2}{7}{128}%
2478 \TestPow{2}{8}{256}%
2479 \TestPow{2}{9}{512}%
2480 \TestPow{2}{10}{1024}%
2481 \TestPow{-2}{3}{-8}%
2482 \TestPow{-2}{4}{16}%
2483 \TestPow{-2}{5}{-32}%
2484 \TestPow{-2}{6}{64}%
2485 \TestPow{-2}{7}{-128}%
2486 \TestPow{-2}{8}{256}%
2487 \TestPow{-2}{9}{-512}%
2488 \TestPow{-2}{10}{1024}%
2489 \TestPow{3}{3}{27}%
2490 \TestPow{3}{4}{81}%
2491 \TestPow{3}{5}{243}%
2492 \TestPow{-3}{3}{-27}%
2493 \TestPow{-3}{4}{81}%
2494 \TestPow{-3}{5}{-243}%
2495 \TestPow{2}{30}{1073741824}%
2496 \TestPow{-3}{19}{-1162261467}%
2497 \TestPow{5}{13}{1220703125}%
2498 \TestPow{-7}{11}{-1977326743}%
2499 \end{qstest}
2500
2501 \begin{qstest}{div}{div}
2502 \TestDiv{1}{1}{1}%
2503 \TestDiv{2}{1}{2}%
2504 \TestDiv{-2}{1}{-2}%
2505 \TestDiv{2}{-1}{-2}%
2506 \TestDiv{-2}{-1}{2}%
2507 \TestDiv{15}{2}{7}%
2508 \TestDiv{-16}{2}{-8}%
2509 \TestDiv{1}{2}{0}%
2510 \TestDiv{1}{3}{0}%
2511 \TestDiv{2}{3}{0}%
2512 \TestDiv{-2}{3}{0}%

```

2513 \TestDiv{2}{-3}{0}%
 2514 \TestDiv{-2}{-3}{0}%
 2515 \TestDiv{13}{3}{4}%
 2516 \TestDiv{-13}{-3}{4}%
 2517 \TestDiv{-13}{3}{-4}%
 2518 \TestDiv{-6}{5}{-1}%
 2519 \TestDiv{-5}{5}{-1}%
 2520 \TestDiv{-4}{5}{0}%
 2521 \TestDiv{-3}{5}{0}%
 2522 \TestDiv{-2}{5}{0}%
 2523 \TestDiv{-1}{5}{0}%
 2524 \TestDiv{0}{5}{0}%
 2525 \TestDiv{1}{5}{0}%
 2526 \TestDiv{2}{5}{0}%
 2527 \TestDiv{3}{5}{0}%
 2528 \TestDiv{4}{5}{0}%
 2529 \TestDiv{5}{5}{1}%
 2530 \TestDiv{6}{5}{1}%
 2531 \TestDiv{-5}{4}{-1}%
 2532 \TestDiv{-4}{4}{-1}%
 2533 \TestDiv{-3}{4}{0}%
 2534 \TestDiv{-2}{4}{0}%
 2535 \TestDiv{-1}{4}{0}%
 2536 \TestDiv{0}{4}{0}%
 2537 \TestDiv{1}{4}{0}%
 2538 \TestDiv{2}{4}{0}%
 2539 \TestDiv{3}{4}{0}%
 2540 \TestDiv{4}{4}{1}%
 2541 \TestDiv{5}{4}{1}%
 2542 \TestDiv{12345}{678}{18}%
 2543 \TestDiv{32372}{5952}{5}%
 2544 \TestDiv{284271294}{18162}{15651}%
 2545 \TestDiv{217652429}{12561}{17327}%
 2546 \TestDiv{462028434}{5439}{84947}%
 2547 \TestDiv{2147483647}{1000}{2147483}%
 2548 \TestDiv{2147483647}{-1000}{-2147483}%
 2549 \TestDiv{-2147483647}{1000}{-2147483}%
 2550 \TestDiv{-2147483647}{-1000}{2147483}%
 2551 \TestDiv{0}{3}{0}%
 2552 \TestDiv{1}{3}{0}%
 2553 \TestDiv{2}{3}{0}%
 2554 \TestDiv{3}{3}{1}%
 2555 \TestDiv{4}{3}{1}%
 2556 \TestDiv{5}{3}{1}%
 2557 \TestDiv{6}{3}{2}%
 2558 \TestDiv{7}{3}{2}%
 2559 \TestDiv{8}{3}{2}%
 2560 \TestDiv{9}{3}{3}%
 2561 \TestDiv{10}{3}{3}%
 2562 \TestDiv{11}{3}{3}%
 2563 \TestDiv{12}{3}{4}%
 2564 \TestDiv{13}{3}{4}%
 2565 \TestDiv{14}{3}{4}%
 2566 \TestDiv{15}{3}{5}%
 2567 \TestDiv{16}{3}{5}%
 2568 \TestDiv{17}{3}{5}%
 2569 \TestDiv{18}{3}{6}%
 2570 \TestDiv{19}{3}{6}%

```

2571 \TestDiv{20}{3}{6}%
2572 \TestDiv{21}{3}{7}%
2573 \TestDiv{22}{3}{7}%
2574 \TestDiv{23}{3}{7}%
2575 \TestDiv{24}{3}{8}%
2576 \TestDiv{25}{3}{8}%
2577 \TestDiv{26}{3}{8}%
2578 \TestDiv{27}{3}{9}%
2579 \TestDiv{28}{3}{9}%
2580 \TestDiv{29}{3}{9}%
2581 \TestDiv{30}{3}{10}%
2582 \TestDiv{31}{3}{10}%
2583 \TestDivBig{17363436332507}{24702}{702916214}%
2584 \end{qstest}
2585
2586 \begin{qstest}{mod}{mod}
2587 \TestMod{-6}{5}{4}%
2588 \TestMod{-5}{5}{0}%
2589 \TestMod{-4}{5}{1}%
2590 \TestMod{-3}{5}{2}%
2591 \TestMod{-2}{5}{3}%
2592 \TestMod{-1}{5}{4}%
2593 \TestMod{0}{5}{0}%
2594 \TestMod{1}{5}{1}%
2595 \TestMod{2}{5}{2}%
2596 \TestMod{3}{5}{3}%
2597 \TestMod{4}{5}{4}%
2598 \TestMod{5}{5}{0}%
2599 \TestMod{6}{5}{1}%
2600 \TestMod{-5}{4}{3}%
2601 \TestMod{-4}{4}{0}%
2602 \TestMod{-3}{4}{1}%
2603 \TestMod{-2}{4}{2}%
2604 \TestMod{-1}{4}{3}%
2605 \TestMod{0}{4}{0}%
2606 \TestMod{1}{4}{1}%
2607 \TestMod{2}{4}{2}%
2608 \TestMod{3}{4}{3}%
2609 \TestMod{4}{4}{0}%
2610 \TestMod{5}{4}{1}%
2611 \TestMod{-6}{-5}{-1}%
2612 \TestMod{-5}{-5}{0}%
2613 \TestMod{-4}{-5}{-4}%
2614 \TestMod{-3}{-5}{-3}%
2615 \TestMod{-2}{-5}{-2}%
2616 \TestMod{-1}{-5}{-1}%
2617 \TestMod{0}{-5}{0}%
2618 \TestMod{1}{-5}{-4}%
2619 \TestMod{2}{-5}{-3}%
2620 \TestMod{3}{-5}{-2}%
2621 \TestMod{4}{-5}{-1}%
2622 \TestMod{5}{-5}{0}%
2623 \TestMod{6}{-5}{-4}%
2624 \TestMod{-5}{-4}{-1}%
2625 \TestMod{-4}{-4}{0}%
2626 \TestMod{-3}{-4}{-3}%
2627 \TestMod{-2}{-4}{-2}%
2628 \TestMod{-1}{-4}{-1}%

```

```

2629 \TestMod{0}{-4}{0}%
2630 \TestMod{1}{-4}{-3}%
2631 \TestMod{2}{-4}{-2}%
2632 \TestMod{3}{-4}{-1}%
2633 \TestMod{4}{-4}{0}%
2634 \TestMod{5}{-4}{-3}%
2635 \TestMod{2147483647}{1000}{647}%
2636 \TestMod{2147483647}{-1000}{-353}%
2637 \TestMod{-2147483647}{1000}{353}%
2638 \TestMod{-2147483647}{-1000}{-647}%
2639 \TestMod{ 0 }{ 4 }{0}%
2640 \TestMod{ 1 }{ 4 }{1}%
2641 \TestMod{ -1 }{ 4 }{3}%
2642 \TestMod{ 0 }{ -4 }{0}%
2643 \TestMod{ 1 }{ -4 }{-3}%
2644 \TestMod{ -1 }{ -4 }{-1}%
2645 \TestMod{18362}{25}{12}%
2646 \end{qstest}
2647
2648 \newcommand*{\TestError}[2]{%
2649   \begingroup
2650     \expandafter\def\csname BigIntCalcError:#1\endcsname{%
2651       \Expect*{#2}{0}%
2652     \expandafter\def\csname BigIntCalcError:#1\endcsname{ERROR}%
2653     \Expect*{#2}{0ERROR}%
2654   \endgroup
2655 }
2656 \begin{qstest}{error}{error}
2657   \TestError{FacNegative}{\bigintcalcFac{-1}}%
2658   \TestError{FacNegative}{\bigintcalcFac{-2147483647}}%
2659   \TestError{DivisionByZero}{\bigintcalcPow{0}{-1}}%
2660   \TestError{DivisionByZero}{\bigintcalcDiv{1}{0}}%
2661   \TestError{DivisionByZero}{\bigintcalcMod{1}{0}}%
2662 \end{qstest}
2663
2664 \begin{document}
2665 \end{document}
2666 \end{test2}

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

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

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

Bundle. All the packages of the bundle ‘oberdiek’ 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/oberdiek.tds.zip](#)

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

¹[CTAN:pkg/bigintcalc](#)

4.2 Bundle installation

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

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

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 Package installation

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

```
tex bigintcalc.dtx
```

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

<code>bigintcalc.sty</code>	→ <code>tex/generic/oberdiek/bigintcalc.sty</code>
<code>bigintcalc.pdf</code>	→ <code>doc/latex/oberdiek/bigintcalc.pdf</code>
<code>test/bigintcalc-test1.tex</code>	→ <code>doc/latex/oberdiek/test/bigintcalc-test1.tex</code>
<code>test/bigintcalc-test2.tex</code>	→ <code>doc/latex/oberdiek/test/bigintcalc-test2.tex</code>
<code>test/bigintcalc-test3.tex</code>	→ <code>doc/latex/oberdiek/test/bigintcalc-test3.tex</code>
<code>bigintcalc.dtx</code>	→ <code>source/latex/oberdiek/bigintcalc.dtx</code>

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`.

4.4 Refresh file name databases

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

4.5 Some details for the interested

Unpacking with \LaTeX . The `.dtx` chooses its action depending on the format:

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

\LaTeX : Generate the documentation.

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 pdfL^AT_EX:

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

5 History

[2007/09/27 v1.0]

- First version.

[2007/11/11 v1.1]

- Use of package `pdfdoccmds` for Lua^T_EX support.

[2011/01/30 v1.2]

- Already loaded package files are not input in plain T_EX.

[2012/04/08 v1.3]

- Fix: `pdfdoccmds` wasn't loaded in case of L^AT_EX.

[2016/05/16 v1.4]

- Documentation updates.

6 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>\#</code>	1774	<code>\%</code>	1850
<code>\%</code>	1850	<code>\@</code>	1775, 1848
<code>\@</code>	1775, 1848	<code>\@firstofone</code>	1783, 1786
<code>\@firstofone</code>	1783, 1786	<code>\@firstoftwo</code>	159, 167, <u>176</u>
<code>\@firstoftwo</code>	159, 167, <u>176</u>	<code>\@gobble</code>	1780, 1788
<code>\@gobble</code>	1780, 1788	<code>\@ne</code>	2181,
<code>\@ne</code>	2181,	<code>\@nil</code> ..	155, 157, 164, 172, 187, 190, 195
<code>\@nil</code> ..	155, 157, 164, 172, 187, 190, 195	<code>\@secondoftwo</code>	161, 169, <u>179</u>
<code>\@secondoftwo</code>	161, 169, <u>179</u>	<code>\@undefined</code>	58
<code>\@undefined</code>	58	<code>\@advance</code>	1813, 1821, 1836, 2109

A

\aftergroup	29	\BIC@AfterFiFi	132, 203, 207, 213, 234, 238, 297, 301, 307, 311, 323, 325, 331, 335, 349, 398, 400, 464, 466, 472, 489, 498, 500, 522, 524, 530, 549, 564, 649, 653, 669, 740, 744, 760, 911, 913, 926, 933, 942, 949, 1035, 1040, 1080, 1105, 1189, 1237, 1241, 1259, 1261, 1270, 1290, 1292, 1313, 1324, 1326, 1355, 1359, 1365, 1417, 1419, 1423, 1429, 1431, 1435, 1460, 1462, 1466, 1471, 1515, 1517, 1523, 1545, 1547, 1569, 1592, 1594, 1600, 1602, 1608, 1610, 1619, 1621, 1642, 1649, 1684, 1686, 1700, 1702, 1706, 1723, 1725, 1735, 1742
\AtEndDocument	2122	\BIC@AfterFiFiFi	133, 218, 222, 354, 358, 559, 561, 591, 595, 601, 604, 608, 616, 618, 623, 629, 633, 660, 664, 751, 755, 1083, 1087, 1094, 1098, 1109, 1113, 1119, 1123, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1264, 1266, 1295, 1297, 1302, 1307, 1372, 1378, 1526, 1530, 1554, 1556, 1562, 1564, 1635, 1637
B		\BIC@AtEnd	95, 96, 117, 1769
\begin	2129, 2147, 2166, 2189, 2200, 2211, 2239, 2251, 2269, 2286, 2315, 2342, 2350, 2379, 2406, 2420, 2446, 2501, 2586, 2656, 2664	\BIC@Cmp	287, 290
\BIC@@@Shl	870, 874	\BIC@CmpDiff	323, 342
\BIC@@@Shr	943, 944, 950, 951, 960	\BIC@CmpLength	312, 318, 320
\BIC@@@Dec	531, 532, 550, 555	\BIC@CmpResult	326, 332, 341
\BIC@@@Inc	473, 474, 490, 495	\BIC@Dec	421, 435, 448, 508, 1208
\BIC@@@PowRec	1373, 1379, 1394	\BIC@DecSwitch	427, 430
\BIC@@@ProcessDiv	1578, 1616	\BIC@Div	1401, 1404
\BIC@@@Shl	854, 862, 877, 881	\BIC@DivCleanup	1515, 1523, 1541, 1545, 1554, 1562, 1592, 1600, 1608, 1619, 1635
\BIC@@@Shr	914, 919, 923, 961	\BIC@DivStart	1505, 1509
\BIC@@AddDigit	699, 700, 710, 789	\BIC@DivStartX	1472, 1478, 1485
\BIC@@Cmp	291, 294, 380, 412	\BIC@DivStartYii	1481, 1489
\BIC@@Dec	511, 519, 568	\BIC@DivStartYiv	1490, 1494
\BIC@@Expand	155, 157	\BIC@DivStartYvi	1495, 1499
\BIC@@Inc	453, 461, 504	\BIC@DivStartYviii	1500, 1504
\BIC@@MinMax	376, 379	\BIC@DivSub	1565, 1582, 1611
\BIC@@MinusOne	1336, 1341	\BIC@DivSwitch	1420, 1424, 1432, 1436, 1441
\BIC@@PowRec	1366, 1389, 1395	\BIC@DivSwitchSign	1405, 1410, 1412
\BIC@@ProcessDiv	1537, 1542	\BIC@DoAdd	650, 654, 675
\BIC@@ProcessFac	1200, 1206	\BIC@DoSub	741, 745, 766
\BIC@@ProcessTim	1007, 1014	\BIC@Expand	153, 183, 250
\BIC@@Shl	836, 841, 848, 851	\BIC@Fac	1164, 1167
\BIC@@Shr	894, 898, 904, 906, 1367, 1396, 1463, 1467, 1468	\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,
\BIC@@Sqr	1154, 1156, 1159		
\BIC@@SubDigit	788, 799		
\BIC@@TestMode	123		
\BIC@@Tim	986		
\BIC@Abs	258, 261		
\BIC@Add	574, 577, 583		
\BIC@AddCarry0	717		
\BIC@AddCarry10	718		
\BIC@AddCarry[1-9]	719		
\BIC@AddDigit	681, 686, 697		
\BIC@AddResult	680, 690		
\BIC@AddSwitch	579, 586		
\BIC@AddXY	592, 596, 630, 634, 641, 646, 837, 842, 849, 1043, 1501		
\BIC@AfterFi	131, 194, 231, 344, 381, 383, 413, 415, 419, 432, 434, 438, 452, 456, 503, 510, 514, 567, 677, 684, 712, 714, 768, 773, 801, 806, 834, 840, 853, 857, 864, 869, 876, 880, 893, 897, 908, 918, 988, 992, 999, 1006, 1016, 1020, 1027, 1029, 1077, 1136, 1141, 1169, 1197, 1199, 1232, 1234, 1329, 1414, 1443, 1445, 1480, 1484, 1536, 1577, 1626, 1681, 1720, 1754, 1756, 1761, 1766		

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, 1877
 \BIC@Tim 986, 1138, 1143
 \BIC@TimDigit 1002, 1009, 1025
 \bigintcalcAbs .. 4, 256, 389, 1927,
 1946, 2017, 2018, 2026, 2027,
 2036, 2037, 2063, 2064, 2084, 2085
 \BigIntCalcAdd 7, 640, 1980, 1988
 \bigintcalcAdd 5, 572, 1138, 1143, 1974
 \bigintcalcCmp 5, 285, 1939, 1977, 2001
 \BigIntCalcDec 7, 447, 1967
 \bigintcalcDec 5, 425, 1963
 \BigIntCalcDiv 8, 1408, 2063
 \bigintcalcDiv 7, 1399,
 1738, 1745, 1915, 2056, 2060, 2660
 \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, 2047, 2050, 2657, 2658
 \BigIntCalcInc 7, 444, 1956
 \bigintcalcInc 5, 406, 1952
 \bigintcalcInv 4, 252, 1924
 \bigintcalcMax 4, 370, 1936
 \bigintcalcMin 4, 365, 1933
 \BigIntCalcMod 8, 1667, 2083
 \bigintcalcMod 7, 1658, 2068, 2661
 \BigIntCalcMul 8, 1130, 2035
 \bigintcalcMul 6, 1066, 1738, 1745, 2032
 \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, 1956, 1967, 1981, 1989, 2005
 \BigIntCalcOdd 7, 391, 1945
 \bigintcalcOdd 5, 386, 1942
 \bigintcalcPow 6, 1211, 2053, 2659
 \bigintcalcSgn
 4, 269, 1930, 1953, 1964, 1975,
 1976, 1999, 2000, 2069, 2071, 2076
 \BigIntCalcShl 8, 846, 2017
 \bigintcalcShl 6, 827, 2014

<code>\BigIntCalcShr</code>	8, 902, 2026	1104, 1172, 1221, 1246, 1252,	
<code>\bigintcalcShr</code>	6, 885, 2023	1263, 1274, 1281, 1294, 1301,	
<code>\bigintcalcSqr</code>	6, 1147, 2042	1353, 1364, 1413, 1416, 1428,	
<code>\BigIntCalcSub</code>	7, 643, 2004	1442, 1451, 1513, 1543, 1552,	
<code>\bigintcalcSub</code>	5, 581, 1737, 1744, 1998	1590, 1617, 1633, 1672, 1683,	
<code>\body</code>	1792, 1796	1691, 1713, 1722, 2069, 2071, 2076	
C			
<code>\catcode</code>	2, 3, 5, 6, 7, 8,	<code>\ifcat</code>	158
	9, 10, 11, 12, 13, 33, 34, 36, 37,	<code>\ifnum</code>	343, 348, 380, 471,
	38, 39, 40, 41, 42, 43, 44, 45, 46,		497, 529, 590, 628, 711, 800,
	47, 48, 49, 69, 70, 72, 73, 74, 78,		875, 1082, 1093, 1108, 1118,
	79, 80, 81, 82, 83, 84, 87, 88, 90,		1171, 1196, 1354, 1371, 1812,
	91, 92, 93, 97, 99, 121, 126, 128,		1820, 1827, 1835, 1953, 1964,
	1772, 1773, 1774, 1775, 1810,	<code>\ifodd</code>	397, 925, 941, 1323
	1819, 1827, 1831, 1848, 1849, 1850	<code>\iftrue</code>	2185
<code>\chardef</code>	1877	<code>\ifx</code>	15,
<code>\count@</code>	1777, 1806, 1810, 1812,		18, 21, 50, 58, 61, 119, 125, 140,
	1813, 1817, 1819, 1820, 1821,		143, 152, 166, 176, 179, 192,
	1825, 1827, 1830, 1831, 1835, 1836		201, 202, 212, 217, 230, 233,
<code>\countdef</code>	1777		262, 275, 278, 295, 296, 306,
<code>\csname</code>	14, 21,		321, 322, 330, 353, 396, 451,
	50, 66, 76, 119, 125, 140, 143,		463, 496, 509, 521, 556, 558,
	152, 176, 179, 701, 717, 718,		587, 588, 614, 647, 648, 658,
	720, 790, 809, 810, 812, 929,		659, 676, 691, 738, 739, 749,
	936, 945, 952, 964, 1031, 1037,		750, 767, 780, 833, 852, 863,
	1044, 1051, 1776, 1779, 1782,		891, 910, 924, 987, 998, 1015,
	1785, 1840, 1867, 2091, 2650, 2652		1135, 1153, 1168, 1221, 1222,
D			1223, 1224, 1228, 1236, 1246,
<code>\dimexpr</code>	2099		1247, 1248, 1274, 1275, 1276,
<code>\divide</code>	1914		1280, 1322, 1335, 1342, 1343,
<code>\documentclass</code>	1873		1353, 1447, 1451, 1452, 1453,
E			1454, 1479, 1514, 1522, 1525,
<code>\empty</code>	17, 18		1544, 1553, 1561, 1591, 1599,
<code>\end</code>	1868,		1607, 1618, 1634, 1672, 1673,
	2145, 2164, 2187, 2198, 2209,		1677, 1683, 1691, 1692, 1696,
	2237, 2249, 2267, 2284, 2313,		1713, 1714, 1715, 1729, 1734,
	2340, 2348, 2377, 2404, 2418,		1753, 1760, 1776, 1779, 1782,
	2444, 2499, 2584, 2646, 2662, 2665		1785, 1840, 2091, 2143, 2144, 2163
<code>\endcsname</code>	14, 21,	<code>\immediate</code>	23, 52
	50, 66, 76, 119, 125, 140, 143,	<code>\IncludeTests</code>	1882
	152, 176, 179, 708, 717, 718,	<code>\input</code>	144, 1841
	720, 797, 809, 810, 812, 929,	<code>\iterate</code>	1793, 1795, 1797
	936, 945, 952, 964, 1031, 1037,	L	
	1044, 1051, 1776, 1779, 1782,	<code>\LoadCommand</code>	1841, 1851
	1785, 1840, 1867, 2091, 2650, 2652	<code>\LogTests</code>	1883
<code>\endinput</code>	29, 117	<code>\loop</code>	1791, 1807, 1818, 1826
<code>\endlinechar</code>	4, 35, 71, 77, 89	M	
<code>\endqstest</code>	2113, 2118	<code>\m@ne</code>	2182, 2197, 2208, 2234
<code>\errmessage</code>	1829	<code>\makeatletter</code>	1876, 2093, 2127
<code>\Expect</code>	1890, 1897, 1906, 1915, 2651, 2653	<code>\makeatother</code>	1878, 2125
I			N
<code>\ifcase</code>	412, 431, 462, 475,	<code>\NeedsTeXFormat</code>	1871
	520, 533, 557, 600, 615, 703,	<code>\newcommand</code>	1884, 1892, 1899,
	706, 721, 792, 795, 813, 907,		1910, 1911, 1912, 1917, 1922,
	1026, 1034, 1052, 1076, 1079,		1923, 1926, 1929, 1932, 1935,
			1938, 1941, 1951, 1962, 1973,

1997, 2013, 2022, 2031, 2041, 2044, 2049, 2052, 2055, 2059, 2067, 2097, 2102, 2106, 2107, 2648	\SavedNumexpr 1874, 1880, 1885, 1889, 1893, 1896
\newcount 1909, 2094, 2095	\space 1830, 1831, 1839, 1890, 2099
\next 1797, 1799, 1801	\StartTime 2102, 2116
\nofiles 1872	\StopTime 2107, 2119
\number 270, 286, 700, 789, 1031, 1037, 1044, 1832, 2099, 2142, 2180, 2181, 2182, 2184, 2196, 2197, 2207, 2208, 2233, 2234	\strip@pt 2099
\numexpr 473, 531, 679, 699, 788, 803, 866, 870, 927, 934, 943, 950, 1001, 1008, 1874, 1875, 1880, 1885, 1886, 1889, 1893, 1894, 1896, 1902, 1910, 2046	\SummaryTime 2094, 2096, 2109, 2123
P	
\PackageInfo 26	T
\pdf@escapehex 187	
\pdf@unescapehex 185	\Test 1843, 1866, 1917, 1922, 1924, 1927, 1930, 1933, 1936, 1939, 1942, 1944, 1952, 1955, 1963, 1966, 1974, 1979, 1987, 1998, 2003, 2014, 2016, 2023, 2025, 2032, 2034, 2042, 2050, 2053, 2056, 2060, 2062, 2068, 2082
\pdfelapsedtime 2108	\TestAbs 1926, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163
\pdfresettimer 2104	\TestAdd 1973, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312
\PrintTime 2097, 2110, 2123	\TestArg 1910, 1911, 1913, 1914
\ProvidesPackage 19, 67	\TestCmp 1938, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236
Q	
\qstest 2112, 2114	\TestCount 1909, 1913, 1914, 1915
R	
\RangeCatcodeCheck . . . 1824, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863	\TestDec . . . 1962, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283
\RangeCatcodeInvalid 1816, 1844, 1845, 1846, 1847	\TestDiv 2055, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582
\renewcommand 2103	\TestDivBig 2059, 2583
\repeat . . . 1791, 1803, 1814, 1822, 1837	\TestError 2648, 2657, 2658, 2659, 2660, 2661
\RequirePackage 149	\TestExch 1922, 2045
\RestoreCatcodes 1805, 1808, 1809, 1864	
\result 1895, 1897	
\resultA 1887, 1890	
\resultB 1888, 1890	
\romannumeral 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	
S	
\saved@endqstest 2113, 2120	
\saved@qstest 2112, 2115	

<code>\TestFac</code>	2044, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433	2178, 2179, 2180, 2181, 2182, 2183
<code>\TestFacBig</code> 2049, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443	<code>\TestShl</code>
<code>\TestInc</code> 1951, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266		2013, 2343, 2344, 2345, 2346, 2347
<code>\TestInv</code> 1923, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144		<code>\TestShr</code>
<code>\TestMax</code>	1935, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208	2022, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376
<code>\TestMin</code>	1932, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197	<code>\TestSpaceAtEnd</code>
<code>\TestMod</code> . . . 2067, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645		1884, 1920
<code>\TestMul</code> . . . 2031, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403		<code>\TestSqr</code>
<code>\TestOdd</code>	1941, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248	2041, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417
<code>\TestPow</code>	2052, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498	<code>\TestSub</code> . . . 1997, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339
<code>\TestResult</code>	1892, 1918	<code>\TestTeXDivide</code>
<code>\TestResultTwoExpansions</code> .	1899, 1919	1912, 2057
<code>\TestSgn</code>	1929, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177,	<code>\TestTime</code>
		2095, 2108, 2109, 2110
		<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, 1810, 1830, 1831, 1915, 2046
		<code>\TimeDescription</code> . . . 2103, 2106, 2110
		<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
		<code>\tracingmacros</code>
		2240
		<code>\typeout</code>
		2098
		U
		<code>\UNDEFINED</code> 1875, 1886, 1894, 1902
		<code>\usepackage</code>
		1879, 1881
		W
		<code>\write</code>
		23, 52
		X
		<code>\x</code>
		14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 135, 138, 1943, 1949, 1954, 1959, 1965, 1970, 1978, 1984, 1986, 1992, 2002, 2008, 2015, 2020, 2024, 2029, 2033, 2039, 2061, 2081, 2087
		Z
		<code>\z@</code>
		2096, 2142, 2180, 2184, 2196, 2207, 2233