

The `bigintcalc` package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/11/11 v1.1

Abstract

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

Contents

1	Documentation	2
1.1	Introduction	2
1.2	Conditions	2
1.2.1	Preconditions	2
1.2.2	Postconditions	3
1.3	Error handling	3
1.4	Operations	3
1.4.1	Num	3
1.4.2	Inv, Abs, Sgn	3
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	5
1.4.7	Mul, Sqr, Fac, Pow	6
1.4.8	Div, Mul	6
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	9
2.4	Help macros	10
2.5	Expand number	10
2.6	Normalize expanded number	11
2.7	Num	12
2.8	Inv, Abs, Sgn	12
2.9	Cmp, Min, Max	13
2.10	Odd	15
2.11	Inc, Dec	15
2.12	Add, Sub	19
2.13	Shl, Shr	24
2.14	<code>\BIC@Tim</code>	27
2.15	Mul	29
2.16	Sqr	30
2.17	Fac	31
2.18	Pow	32
2.18.1	Help macros	34
2.18.2	Recursive calculation	35

2.19 Div	36
2.20 Mod	42
3 Test	44
3.1 Catcode checks for loading	44
3.2 Macro tests	46
3.2.1 Preamble with test macro definitions	46
3.2.2 Time	49
3.2.3 Test sets	50
4 Installation	59
4.1 Download	59
4.2 Bundle installation	59
4.3 Package installation	59
4.4 Refresh file name databases	59
4.5 Some details for the interested	60
5 History	60
[2007/09/27 v1.0]	60
[2007/11/11 v1.1]	60
6 Index	60

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 expandible contexts. An traditional error message, however, is not expandable and would break these contexts.

1.4 Operations

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

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

1.4.1 Num

`\bigintcalcNum {⟨x⟩}`

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

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

1.4.2 Inv, Abs, Sgn

`\bigintcalcInv {⟨x⟩}`

Macro `\bigintcalcInv` switches the sign.

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

`\bigintcalcAbs {⟨x⟩}`

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

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

`\bigintcalcSgn {⟨x⟩}`

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

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

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

```
\ifcase\bigintcalcSgn{<x>}
  $x=0$
\or
  $x>0$
\else
  $x<0$
\fi
```

1.4.3 Min, Max, Cmp

`\bigintcalcMin {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcMax {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcCmp {⟨x⟩} {⟨y⟩}`

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

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

These values can be distinguished by `\ifcase`:

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

1.4.4 Odd

`\bigintcalcOdd {⟨x⟩}`

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

1.4.5 Inc, Dec, Add, Sub

`\bigintcalcInc {⟨x⟩}`

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

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

`\bigintcalcDec {⟨x⟩}`

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

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

`\bigintcalcAdd {⟨x⟩} {⟨y⟩}`

Macro `\bigintcalcAdd` adds the two numbers.

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

`\bigintcalcSub {⟨x⟩} {⟨y⟩}`

Macro `\bigintcalcSub` calculates the difference.

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

1.4.6 Shl, Shr

`\bigintcalcShl {⟨x⟩}`

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

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

`\bigintcalcShr {⟨x⟩}`

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

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

1.4.7 Mul, Sqr, Fac, Pow

`\bigintcalcMul {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcSqr {⟨x⟩}`

Macro `\bigintcalcSqr` returns the square product.

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

`\bigintcalcFac {⟨x⟩}`

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

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

$$(0! = 1)$$

`\bigintcalcPow Mx My`

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

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

$$(0^0 = 1)$$

1.4.8 Div, Mul

`\bigintcalcDiv {⟨x⟩} {⟨y⟩}`

Macro `\bigintcalcDiv` performs an integer division. Argument $\langle y \rangle$ 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 $\langle y \rangle$. Argument $\langle y \rangle$ must not be zero.

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

The result ranges:

$$-|y| < \text{Mod}(x, y) \leq 0 \quad \text{for } y < 0$$

$$0 \leq \text{Mod}(x, y) < y \quad \text{for } y \geq 0$$

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 $\langle number \rangle$!`

1/0 is returned if $\langle number \rangle$ is odd/even.

`\BigIntCalcInc $\langle number \rangle$!`

Incrementation.

`\BigIntCalcDec $\langle number \rangle$!`

Decrementation, positive number without zero.

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

Addition, $A \geq B$.

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

Subtraction, $A \geq B$.

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

Left shift (multiplication with two).

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

Right shift (integer division by two).

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

Multiplication, $A \geq B$.

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

Division operation.

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

Modulo operation.

2 Implementation

```
1 ⟨*package⟩
```

2.1 Reload check and package identification

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

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

Package identification:

```
30 \begingroup
31 \catcode35 6 % #
32 \catcode40 12 % (
33 \catcode41 12 % )
34 \catcode44 12 % ,
35 \catcode45 12 % -
36 \catcode46 12 % .
37 \catcode47 12 % /
38 \catcode58 12 % :
39 \catcode64 11 % @
40 \catcode91 12 % [
41 \catcode93 12 % ]
42 \catcode123 1 % {
43 \catcode125 2 % }
44 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45 \def\x#1#2#3[#4]{\endgroup
46 \immediate\write-1{Package: #3 #4}%
47 \xdef#1{#4}%
48 }%
49 \else
50 \def\x#1#2[#3]{\endgroup
51 #2[#3]}%
52 \ifx#1\@undefined
53 \xdef#1{#3}%
54 \fi
55 \ifx#1\relax
56 \xdef#1{#3}%
```



```

57     \fi
58   }%
59 \fi
60 \expandafter\x\csname ver@bigintcalc.sty\endcsname
61 \ProvidesPackage{bigintcalc}%
62 [2007/11/11 v1.1 Expandable big integer calculations (H0)]

```

2.2 Catcodes

```

63 \begingroup
64   \catcode123 1 % {
65   \catcode125 2 % }
66   \def\x{\endgroup
67     \expandafter\edef\csname BIC@AtEnd\endcsname{%
68       \catcode35 \the\catcode35\relax
69       \catcode64 \the\catcode64\relax
70       \catcode123 \the\catcode123\relax
71       \catcode125 \the\catcode125\relax
72     }%
73   }%
74 \x
75 \catcode35 6 % #
76 \catcode64 11 % @
77 \catcode123 1 % {
78 \catcode125 2 % }
79 \def\TMP@EnsureCode#1#2{%
80   \edef\BIC@AtEnd{%
81     \BIC@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{33}{12}% !
87 \TMP@EnsureCode{36}{14}% $ (comment!)
88 \TMP@EnsureCode{38}{14}% & (comment!)
89 \TMP@EnsureCode{40}{12}% (
90 \TMP@EnsureCode{41}{12}% )
91 \TMP@EnsureCode{42}{12}% *
92 \TMP@EnsureCode{43}{12}% +
93 \TMP@EnsureCode{45}{12}% -
94 \TMP@EnsureCode{46}{12}% .
95 \TMP@EnsureCode{47}{12}% /
96 \TMP@EnsureCode{58}{11}% : (letter!)
97 \TMP@EnsureCode{60}{12}% <
98 \TMP@EnsureCode{61}{12}% =
99 \TMP@EnsureCode{62}{12}% >
100 \TMP@EnsureCode{63}{14}% ? (comment!)
101 \begingroup\expandafter\expandafter\expandafter\endgroup
102 \expandafter\ifx\csname BIC@TestMode\endcsname\relax
103 \else
104   \catcode63=9 % ? (ignore)
105 \fi
106 ? \let\BIC@@TestMode\BIC@TestMode

```

2.3 ε -TeX detection

```

107 \begingroup\expandafter\expandafter\expandafter\endgroup
108 \expandafter\ifx\csname numexpr\endcsname\relax
109   \catcode36=9 % $ (ignore)
110 \else
111   \catcode38=9 % & (ignore)
112 \fi

```

2.4 Help macros

```

\BIC@Fi
113 \let\BIC@Fi\fi

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

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

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

\BIC@Space
117 \begingroup
118   \def\x#1{\endgroup
119     \let\BIC@Space= #1%
120   }%
121 \x{ }

```

2.5 Expand number

```

122 \begingroup\expandafter\expandafter\expandafter\endgroup
123 \expandafter\ifx\csname RequirePackage\endcsname\relax
124   \input pdftexcmds.sty\relax
125 \else
126   \RequirePackage{pdftexcmds}[2007/11/11]%
127 \fi

128 \begingroup\expandafter\expandafter\expandafter\endgroup
129 \expandafter\ifx\csname pdf@escapehex\endcsname\relax

\BIC@Expand
130   \def\BIC@Expand#1{%
131     \romannumeral0%
132     \BIC@@Expand#1!\@nil{}}%
133   }%

\BIC@@Expand
134   \def\BIC@@Expand#1#2\@nil#3{%
135     \expandafter\ifcat\noexpand#1\relax
136       \expandafter\@firstoftwo
137     \else
138       \expandafter\@secondoftwo
139     \fi
140     {%
141       \expandafter\BIC@@Expand#1#2\@nil{#3}%
142     }{%
143       \ifx#1!%
144         \expandafter\@firstoftwo
145       \else
146         \expandafter\@secondoftwo
147       \fi
148       { #3}{%
149         \BIC@@Expand#2\@nil{#3#1}%
150       }%
151     }%
152   }%

```

```

\@firstoftwo
153 \expandafter\ifx\csname @firstoftwo\endcsname\relax
154 \long\def\@firstoftwo#1#2{#1}%
155 \fi

\@secondoftwo
156 \expandafter\ifx\csname @secondoftwo\endcsname\relax
157 \long\def\@secondoftwo#1#2{#2}%
158 \fi

159 \else

\BIC@Expand
160 \def\BIC@Expand#1{%
161 \romannumeral0\expandafter\expandafter\expandafter\BIC@Space
162 \pdf@unescapehex{%
163 \expandafter\expandafter\expandafter
164 \BIC@StripHexSpace\pdf@escapehex{#1}20\@nil
165 }%
166 }%

\BIC@StripHexSpace
167 \def\BIC@StripHexSpace#120#2\@nil{%
168 #1%
169 \ifx\#2\%
170 \else
171 \BIC@AfterFi{%
172 \BIC@StripHexSpace#2\@nil
173 }%
174 \BIC@Fi
175 }%

176 \fi

```

2.6 Normalize expanded number

```

\BIC@Normalize #1: result sign
#2: first token of number
177 \def\BIC@Normalize#1#2{%
178 \ifx#2-%
179 \ifx\#1\%
180 \BIC@AfterFiFi{%
181 \BIC@Normalize-%
182 }%
183 \else
184 \BIC@AfterFiFi{%
185 \BIC@Normalize{%
186 }%
187 \fi
188 \else
189 \ifx#2+%
190 \BIC@AfterFiFi{%
191 \BIC@Normalize{#1}%
192 }%
193 \else
194 \ifx#20%
195 \BIC@AfterFiFiFi{%
196 \BIC@NormalizeZero{#1}%
197 }%
198 \else
199 \BIC@AfterFiFiFi{%

```

```

200          \BIC@NormalizeDigits#1#2%
201      }%
202      \fi
203      \fi
204      \BIC@Fi
205  }

```

\BIC@NormalizeZero

```

206 \def\BIC@NormalizeZero#1#2{%
207   \ifx#2!%
208     \BIC@AfterFi{ 0}%
209   \else
210     \ifx#20%
211       \BIC@AfterFiFi{%
212         \BIC@NormalizeZero{#1}%
213       }%
214     \else
215       \BIC@AfterFiFi{%
216         \BIC@NormalizeDigits#1#2%
217       }%
218     \fi
219     \BIC@Fi
220 }

```

\BIC@NormalizeDigits

```

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

```

2.7 Num

\bigintcalcNum

```

222 \def\bigintcalcNum#1{%
223   \romannumeral0%
224   \expandafter\expandafter\expandafter\BIC@Normalize
225   \expandafter\expandafter\expandafter{%
226     \expandafter\expandafter\expandafter}%
227   \BIC@Expand{#1}!%
228 }

```

2.8 Inv, Abs, Sgn

\bigintcalcInv

```

229 \def\bigintcalcInv#1{%
230   \romannumeral0\expandafter\expandafter\expandafter\BIC@Space
231   \bigintcalcNum{-#1}%
232 }

```

\bigintcalcAbs

```

233 \def\bigintcalcAbs#1{%
234   \romannumeral0%
235   \expandafter\expandafter\expandafter\BIC@Abs
236   \bigintcalcNum{#1}%
237 }

```

\BIC@Abs

```

238 \def\BIC@Abs#1{%
239   \ifx#1-%
240     \expandafter\BIC@Space
241   \else
242     \expandafter\BIC@Space
243     \expandafter#1%

```

```

244 \fi
245 }

\bigintcalcSgn

246 \def\bigintcalcSgn#1{%
247 \number
248 \expandafter\expandafter\expandafter\BIC@Sgn
249 \bigintcalcNum{#1}! %
250 }

\BIC@Sgn

251 \def\BIC@Sgn#1#2!{%
252 \ifx#1-%
253 -1%
254 \else
255 \ifx#10%
256 0%
257 \else
258 1%
259 \fi
260 \fi
261 }

```

2.9 Cmp, Min, Max

```

\bigintcalcCmp

262 \def\bigintcalcCmp#1#2{%
263 \number
264 \expandafter\expandafter\expandafter\BIC@Cmp
265 \bigintcalcNum{#2}!{#1}%
266 }

\BIC@Cmp

267 \def\BIC@Cmp#1!#2{%
268 \expandafter\expandafter\expandafter\BIC@@Cmp
269 \bigintcalcNum{#2}!#1!%
270 }

\BIC@@Cmp

271 \def\BIC@@Cmp#1#2!#3#4!{%
272 \ifx#1-%
273 \ifx#3-%
274 \BIC@AfterFiFi{%
275 \BIC@@Cmp#4!#2!%
276 }%
277 \else
278 \BIC@AfterFiFi{%
279 -1 %
280 }%
281 \fi
282 \else
283 \ifx#3-%
284 \BIC@AfterFiFi{%
285 1 %
286 }%
287 \else
288 \BIC@AfterFiFi{%
289 \BIC@CmpLength#1#2!#3#4!#1#2!#3#4!%
290 }%
291 \fi
292 \BIC@Fi
293 }

```

```

\BIC@PosCmp
294 \def\BIC@PosCmp#1!#2!{%
295   \BIC@CmpLength#1!#2!#1!#2!%
296 }

\BIC@CmpLength
297 \def\BIC@CmpLength#1#2!#3#4!{%
298   \ifx\#2\%
299     \ifx\#4\%
300       \BIC@AfterFiFi\BIC@CmpDiff
301     \else
302       \BIC@AfterFiFi{%
303         \BIC@CmpResult{-1}%
304       }%
305     \fi
306   \else
307     \ifx\#4\%
308       \BIC@AfterFiFi{%
309         \BIC@CmpResult1%
310       }%
311     \else
312       \BIC@AfterFiFi{%
313         \BIC@CmpLength#2!#4!%
314       }%
315     \fi
316   \BIC@Fi
317 }

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

\BIC@CmpDiff
319 \def\BIC@CmpDiff#1#2!#3#4!{%
320   \ifnum#1<#3 %
321     \BIC@AfterFi{%
322       -1 %
323     }%
324   \else
325     \ifnum#1>#3 %
326       \BIC@AfterFiFi{%
327         1 %
328       }%
329     \else
330       \ifx\#2\%
331         \BIC@AfterFiFiFi{%
332           0 %
333         }%
334       \else
335         \BIC@AfterFiFiFi{%
336           \BIC@CmpDiff#2!#4!%
337         }%
338       \fi
339     \fi
340   \BIC@Fi
341 }

\bigintcalcMin
342 \def\bigintcalcMin#1{%
343   \romannumeral0%
344   \expandafter\expandafter\expandafter\BIC@MinMax
345   \bigintcalcNum{#1}!-!%
346 }

```

```

\bigintcalcMax
347 \def\bigintcalcMax#1{%
348   \romannumeral0%
349   \expandafter\expandafter\expandafter\BIC@MinMax
350   \bigintcalcNum{#1}!%!%
351 }

\BIC@MinMax #1:  $x$ 
#2: sign for comparison
#3:  $y$ 
352 \def\BIC@MinMax#1!#2!#3{%
353   \expandafter\expandafter\expandafter\BIC@@MinMax
354   \bigintcalcNum{#3}!#1!#2!%
355 }

\BIC@@MinMax #1:  $y$ 
#2:  $x$ 
#3: sign for comparison
356 \def\BIC@@MinMax#1!#2!#3!{%
357   \ifnum\BIC@@Cmp#1!#2!=#31 %
358     \BIC@AfterFi{ #1}%
359   \else
360     \BIC@AfterFi{ #2}%
361   \BIC@Fi
362 }

```

2.10 Odd

```

\bigintcalcOdd
363 \def\bigintcalcOdd#1{%
364   \romannumeral0%
365   \expandafter\expandafter\expandafter\BIC@Odd
366   \bigintcalcAbs{#1}!%!%
367 }

\BigIntCalcOdd
368 \def\BigIntCalcOdd#1!{%
369   \romannumeral0%
370   \BIC@Odd#1!%
371 }

\BIC@Odd #1:  $x$ 
372 \def\BIC@Odd#1#2{%
373   \ifx#2!%
374     \ifodd#1 %
375       \BIC@AfterFiFi{ 1}%
376     \else
377       \BIC@AfterFiFi{ 0}%
378     \fi
379   \else
380     \expandafter\BIC@Odd\expandafter#2%
381   \BIC@Fi
382 }

```

2.11 Inc, Dec

```

\bigintcalcInc
383 \def\bigintcalcInc#1{%
384   \romannumeral0%
385   \expandafter\expandafter\expandafter\BIC@IncSwitch

```

```

386 \bigintcalcNum{#1}!%
387 }

\BIC@IncSwitch

388 \def\BIC@IncSwitch#1#2!{%
389 \ifcase\BIC@@Cmp#1#2!-1!%
390 \BIC@AfterFi{ 0}%
391 \or
392 \BIC@AfterFi{%
393 \BIC@Inc#1#2!{%}%
394 }%
395 \else
396 \BIC@AfterFi{%
397 \expandafter-\romannumeral0%
398 \BIC@Dec#2!{%}%
399 }%
400 \BIC@Fi
401 }

\bigintcalcDec

402 \def\bigintcalcDec#1{%
403 \romannumeral0%
404 \expandafter\expandafter\expandafter\BIC@DecSwitch
405 \bigintcalcNum{#1}!%
406 }

\BIC@DecSwitch

407 \def\BIC@DecSwitch#1#2!{%
408 \ifcase\BIC@Sgn#1#2! %
409 \BIC@AfterFi{ -1}%
410 \or
411 \BIC@AfterFi{%
412 \BIC@Dec#1#2!{%}%
413 }%
414 \else
415 \BIC@AfterFi{%
416 \expandafter-\romannumeral0%
417 \BIC@Inc#2!{%}%
418 }%
419 \BIC@Fi
420 }

\BigIntCalcInc

421 \def\BigIntCalcInc#1!{%
422 \romannumeral0\BIC@Inc#1!{%}%
423 }

\BigIntCalcDec

424 \def\BigIntCalcDec#1!{%
425 \romannumeral0\BIC@Dec#1!{%}%
426 }

\BIC@Inc

427 \def\BIC@Inc#1#2!#3{%
428 \ifx\#2\%
429 \BIC@AfterFi{%
430 \BIC@@Inc#1#3!{%}%
431 }%
432 \else
433 \BIC@AfterFi{%
434 \BIC@Inc#2!{#1#3}%

```



```

435     }%
436 \BIC@Fi
437 }

\BIC@@Inc

438 \def\BIC@@Inc#1#2#3!#4{%
439 \ifcase#1 %
440 \ifx\#3\%
441 \BIC@AfterFiFi{ #2#4}%
442 \else
443 \BIC@AfterFiFi{%
444 \BIC@@Inc0#3!{#2#4}%
445 }%
446 \fi
447 \else
448 \ifnum#2<9 %
449 \BIC@AfterFiFi{%
450 & \expandafter\BIC@@Inc\the\numexpr#2+1\relax
451 $ \expandafter\expandafter\expandafter\BIC@@Inc
452 $ \ifcase#2 \expandafter1%
453 $ \or\expandafter2%
454 $ \or\expandafter3%
455 $ \or\expandafter4%
456 $ \or\expandafter5%
457 $ \or\expandafter6%
458 $ \or\expandafter7%
459 $ \or\expandafter8%
460 $ \or\expandafter9%
461 $? \else\BigIntCalcError:ThisCannotHappen%
462 $ \fi
463 0#3!{#4}%
464 }%
465 \else
466 \BIC@AfterFiFi{%
467 \BIC@@Inc01#3!{#4}%
468 }%
469 \fi
470 \BIC@Fi
471 }

\BIC@@@Inc

472 \def\BIC@@@Inc#1#2#3!#4{%
473 \ifx\#3\%
474 \ifnum#2=1 %
475 \BIC@AfterFiFi{ 1#1#4}%
476 \else
477 \BIC@AfterFiFi{ #1#4}%
478 \fi
479 \else
480 \BIC@AfterFi{%
481 \BIC@@Inc#2#3!{#1#4}%
482 }%
483 \BIC@Fi
484 }

\BIC@Dec

485 \def\BIC@Dec#1#2!#3{%
486 \ifx\#2\%
487 \BIC@AfterFi{%
488 \BIC@@Dec1#1#3!{ }%
489 }%
490 \else

```

```

491 \BIC@AfterFi{%
492 \BIC@Dec#2!{#1#3}%
493 }%
494 \BIC@Fi
495 }

```

\BIC@@Dec

```

496 \def\BIC@@Dec#1#2#3!#4{%
497 \ifcase#1 %
498 \ifx\#3\%
499 \BIC@AfterFiFi{ #2#4}%
500 \else
501 \BIC@AfterFiFi{%
502 \BIC@@Dec0#3!{#2#4}%
503 }%
504 \fi
505 \else
506 \ifnum#2>0 %
507 \BIC@AfterFiFi{%
508 & \expandafter\BIC@@Dec\the\numexpr#2-1\relax
509 $ \expandafter\expandafter\expandafter\BIC@@Dec
510 $ \ifcase#2
511 $? \BigIntCalcError:ThisCannotHappen%
512 $ \or\expandafter0%
513 $ \or\expandafter1%
514 $ \or\expandafter2%
515 $ \or\expandafter3%
516 $ \or\expandafter4%
517 $ \or\expandafter5%
518 $ \or\expandafter6%
519 $ \or\expandafter7%
520 $ \or\expandafter8%
521 $? \else\BigIntCalcError:ThisCannotHappen%
522 $ \fi
523 0#3!{#4}%
524 }%
525 \else
526 \BIC@AfterFiFi{%
527 \BIC@@Dec91#3!{#4}%
528 }%
529 \fi
530 \BIC@Fi
531 }

```

\BIC@@@Dec

```

532 \def\BIC@@@Dec#1#2#3!#4{%
533 \ifx\#3\%
534 \ifcase#1 %
535 \ifx\#4\%
536 \BIC@AfterFiFiFi{ 0}%
537 \else
538 \BIC@AfterFiFiFi{ #4}%
539 \fi
540 \else
541 \BIC@AfterFiFi{ #1#4}%
542 \fi
543 \else
544 \BIC@AfterFi{%
545 \BIC@@Dec#2#3!{#1#4}%
546 }%
547 \BIC@Fi
548 }

```

2.12 Add, Sub

\bigintcalcAdd

```
549 \def\bigintcalcAdd#1{%
550   \romannumeral0%
551   \expandafter\expandafter\expandafter\BIC@Add
552   \bigintcalcNum{#1}!%
553 }
```

\BIC@Add

```
554 \def\BIC@Add#1!#2{%
555   \expandafter\expandafter\expandafter
556   \BIC@AddSwitch\bigintcalcNum{#2}!#1!%
557 }
```

\bigintcalcSub

```
558 \def\bigintcalcSub#1#2{%
559   \romannumeral0%
560   \expandafter\expandafter\expandafter\BIC@Add
561   \bigintcalcNum{-#2}!{#1}%
562 }
```

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

```
563 \def\BIC@AddSwitch#1#2!#3#4!{%
564   \ifx#1-% x < 0
565     \ifx#3-% y < 0
566       \expandafter-\romannumeral0%
567       \ifnum\BIC@PosCmp#2!#4!=1 % -x > -y
568         \BIC@AfterFiFiFi{%
569           \BIC@AddXY#2!#4!!%
570         }%
571       \else % -x <= -y
572         \BIC@AfterFiFiFi{%
573           \BIC@AddXY#4!#2!!%
574         }%
575       \fi
576     \else % y >= 0
577       \ifcase\BIC@PosCmp#2!#3#4!% -x = y
578         \BIC@AfterFiFiFi{ 0}%
579       \or % -x > y
580         \expandafter-\romannumeral0%
581         \BIC@AfterFiFiFi{%
582           \BIC@SubXY#2!#3#4!!%
583         }%
584       \else % -x <= y
585         \BIC@AfterFiFiFi{%
586           \BIC@SubXY#3#4!#2!!%
587         }%
588       \fi
```

```

589     \fi
590 \else % x >= 0
591     \ifx#3-% y < 0
592         \ifcase\BIC@PosCmp#1#2!#4!% x = -y
593             \BIC@AfterFiFiFi{ 0}%
594         \or % x > -y
595             \BIC@AfterFiFiFi{%
596                 \BIC@SubXY#1#2!#4!!!%
597             }%
598         \else % x <= -y
599             \expandafter-\romannumeral0%
600             \BIC@AfterFiFiFi{%
601                 \BIC@SubXY#4!#1#2!!!%
602             }%
603     \fi
604 \else % y >= 0
605     \ifnum\BIC@PosCmp#1#2!#3#4!=1 % x > y
606         \BIC@AfterFiFiFi{%
607             \BIC@AddXY#1#2!#3#4!!!%
608         }%
609     \else % x <= y
610         \BIC@AfterFiFiFi{%
611             \BIC@AddXY#3#4!#1#2!!!%
612         }%
613     \fi
614 \fi
615 \BIC@Fi
616 }

```

\BigIntCalcAdd

```

617 \def\BigIntCalcAdd#1#2!{%
618     \romannumeral0\BIC@AddXY#1!#2!!!%
619 }

```

\BigIntCalcSub

```

620 \def\BigIntCalcSub#1#2!{%
621     \romannumeral0\BIC@SubXY#1!#2!!!%
622 }

```

\BIC@AddXY

```

623 \def\BIC@AddXY#1#2!#3#4!#5!#6!{%
624     \ifx\\#2\\%
625         \ifx\\#3\\%
626             \BIC@AfterFiFiFi{%
627                 \BIC@DoAdd0!#1#5!#60!%
628             }%
629         \else
630             \BIC@AfterFiFiFi{%
631                 \BIC@DoAdd0!#1#5!#3#6!%
632             }%
633         \fi
634     \else
635         \ifx\\#4\\%
636             \ifx\\#3\\%
637                 \BIC@AfterFiFiFi{%
638                     \BIC@AddXY#2!{}!#1#5!#60!%
639                 }%
640             \else
641                 \BIC@AfterFiFiFi{%
642                     \BIC@AddXY#2!{}!#1#5!#3#6!%
643                 }%
644             \fi

```

```

645     \else
646         \BIC@AfterFiFi{%
647             \BIC@AddXY#2!#4!#1#5!#3#6!%
648         }%
649     \fi
650 \BIC@Fi
651 }

\BIC@DoAdd #1: carry
#2: reverted result
#3#4: reverted  $x$ 
#5#6: reverted  $y$ 
652 \def\BIC@DoAdd#1#2!#3#4!#5#6!{%
653     \ifx\#4\%
654         \BIC@AfterFiFi{%
655             \expandafter\BIC@Space
656             \the\numexpr#1+#3+#5\relax#2%
657             \expandafter\expandafter\expandafter\BIC@AddResult
658             \BIC@AddDigit#1#3#5#2%
659         }%
660     \else
661         \BIC@AfterFiFi{%
662             \expandafter\expandafter\expandafter\BIC@DoAdd
663             \BIC@AddDigit#1#3#5#2!#4!#6!%
664         }%
665     \BIC@Fi
666 }

\BIC@AddResult
667 $ \def\BIC@AddResult#1{%
668 $     \ifx#10%
669 $         \expandafter\BIC@Space
670 $     \else
671 $         \expandafter\BIC@Space\expandafter#1%
672 $     \fi
673 $ }%

\BIC@AddDigit #1: carry
#2: digit of  $x$ 
#3: digit of  $y$ 
674 \def\BIC@AddDigit#1#2#3{%
675     \romannumeral0%
676 & \expandafter\BIC@@AddDigit\the\numexpr#1+#2+#3!%
677 $ \expandafter\BIC@@AddDigit\number%
678 $ \csname
679 $     BIC@AddCarry%
680 $     \ifcase#1 %
681 $         #2%
682 $     \else
683 $         \ifcase#2 1\or2\or3\or4\or5\or6\or7\or8\or9\or10\fi
684 $     \fi
685 $ \endcsname#3!%
686 }

\BIC@@AddDigit
687 \def\BIC@@AddDigit#1!{%
688     \ifnum#1<10 %
689         \BIC@AfterFiFi{ 0#1}%
690     \else
691         \BIC@AfterFiFi{ #1}%
692     \BIC@Fi
693 }

```

\BIC@AddCarry0

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

\BIC@AddCarry10

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

\BIC@AddCarry[1-9]

```

696 $ \def\BIC@Temp#1#2{%
697 $   \expandafter\def\csname BIC@AddCarry#1\endcsname##1{%
698 $     \ifcase##1 #1\or
699 $       #2%
700 $?   \else\BigIntCalcError:ThisCannotHappen%
701 $     \fi
702 $   }%
703 $ }%
704 $ \BIC@Temp 0{1\or2\or3\or4\or5\or6\or7\or8\or9}%
705 $ \BIC@Temp 1{2\or3\or4\or5\or6\or7\or8\or9\or10}%
706 $ \BIC@Temp 2{3\or4\or5\or6\or7\or8\or9\or10\or11}%
707 $ \BIC@Temp 3{4\or5\or6\or7\or8\or9\or10\or11\or12}%
708 $ \BIC@Temp 4{5\or6\or7\or8\or9\or10\or11\or12\or13}%
709 $ \BIC@Temp 5{6\or7\or8\or9\or10\or11\or12\or13\or14}%
710 $ \BIC@Temp 6{7\or8\or9\or10\or11\or12\or13\or14\or15}%
711 $ \BIC@Temp 7{8\or9\or10\or11\or12\or13\or14\or15\or16}%
712 $ \BIC@Temp 8{9\or10\or11\or12\or13\or14\or15\or16\or17}%
713 $ \BIC@Temp 9{10\or11\or12\or13\or14\or15\or16\or17\or18}%

```

\BIC@SubXY Preconditions:

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

```

714 \def\BIC@SubXY#1#2!#3#4!#5!#6!{%
715   \ifx\#2\%
716     \ifx\#3\%
717       \BIC@AfterFiFi{%
718         \BIC@DoSub0!#1#5!#60!%
719       }%
720     \else
721       \BIC@AfterFiFi{%
722         \BIC@DoSub0!#1#5!#3#6!%
723       }%
724     \fi
725   \else
726     \ifx\#4\%
727       \ifx\#3\%
728         \BIC@AfterFiFiFi{%
729           \BIC@SubXY#2!{ }!#1#5!#60!%
730         }%
731       \else
732         \BIC@AfterFiFiFi{%
733           \BIC@SubXY#2!{ }!#1#5!#3#6!%
734         }%
735       \fi
736     \else
737       \BIC@AfterFiFi{%
738         \BIC@SubXY#2!#4!#1#5!#3#6!%
739       }%
740     \fi
741   \BIC@Fi
742 }

```

```

\BIC@DoSub #1: carry
#2: reverted result
#3#4: reverted x
#5#6: reverted y
743 \def\BIC@DoSub#1#2!#3#4!#5#6!{%
744 \ifx\#4\%
745 \BIC@AfterFi{%
746 \expandafter\expandafter\expandafter\BIC@SubResult
747 \BIC@SubDigit#1#3#5#2%
748 }%
749 \else
750 \BIC@AfterFi{%
751 \expandafter\expandafter\expandafter\BIC@DoSub
752 \BIC@SubDigit#1#3#5#2!#4!#6!%
753 }%
754 \BIC@Fi
755 }

\BIC@SubResult
756 \def\BIC@SubResult#1{%
757 \ifx#10%
758 \expandafter\BIC@SubResult
759 \else
760 \expandafter\BIC@Space\expandafter#1%
761 \fi
762 }

\BIC@SubDigit #1: carry
#2: digit of  $x$ 
#3: digit of  $y$ 
763 \def\BIC@SubDigit#1#2#3{%
764 \romannumeral0%
765 & \expandafter\BIC@SubDigit\the\numexpr#2-#3-#1!%
766 $ \expandafter\BIC@@AddDigit\number
767 $ \csname
768 $ BIC@SubCarry%
769 $ \ifcase#1 %
770 $ #3%
771 $ \else
772 $ \ifcase#3 1\or2\or3\or4\or5\or6\or7\or8\or9\or10\fi
773 $ \fi
774 $ \endcsname#2!%
775 }

\BIC@@SubDigit
776 & \def\BIC@@SubDigit#1!{%
777 & \ifnum#1<0 %
778 & \BIC@AfterFi{%
779 & \expandafter\BIC@Space
780 & \expandafter1\the\numexpr#1+10\relax
781 & }%
782 & \else
783 & \BIC@AfterFi{ 0#1}%
784 & \BIC@Fi
785 & }%

\BIC@SubCarry0
786 $ \expandafter\def\csname BIC@SubCarry0\endcsname#1{#1}%

\BIC@SubCarry10
787 $ \expandafter\def\csname BIC@SubCarry10\endcsname#1{#1}%

```

\BIC@SubCarry[1-9]

```

788 $ \def\BIC@Temp#1#2{%
789 $   \expandafter\def\csname BIC@SubCarry#1\endcsname##1{%
790 $     \ifcase##1 #2%
791 $?    \else\BigIntCalcError:ThisCannotHappen%
792 $     \fi
793 $   }%
794 $ }%
795 $ \BIC@Temp 1{19\or0\or1\or2\or3\or4\or5\or6\or7\or8}%
796 $ \BIC@Temp 2{18\or19\or0\or1\or2\or3\or4\or5\or6\or7}%
797 $ \BIC@Temp 3{17\or18\or19\or0\or1\or2\or3\or4\or5\or6}%
798 $ \BIC@Temp 4{16\or17\or18\or19\or0\or1\or2\or3\or4\or5}%
799 $ \BIC@Temp 5{15\or16\or17\or18\or19\or0\or1\or2\or3\or4}%
800 $ \BIC@Temp 6{14\or15\or16\or17\or18\or19\or0\or1\or2\or3}%
801 $ \BIC@Temp 7{13\or14\or15\or16\or17\or18\or19\or0\or1\or2}%
802 $ \BIC@Temp 8{12\or13\or14\or15\or16\or17\or18\or19\or0\or1}%
803 $ \BIC@Temp 9{11\or12\or13\or14\or15\or16\or17\or18\or19\or0}%

```

2.13 Shl, Shr

\bigintcalcShl

```

804 \def\bigintcalcShl#1{%
805   \romannumeral0%
806   \expandafter\expandafter\expandafter\BIC@Shl
807   \bigintcalcNum{#1}!%
808 }

```

\BIC@Shl

```

809 \def\BIC@Shl#1#2!{%
810   \ifx#1-%
811     \BIC@AfterFi{%
812       \expandafter-\romannumeral0%
813 &     \BIC@@Shl#2!!%
814 $     \BIC@AddXY#2!#2!!!%
815   }%
816   \else
817     \BIC@AfterFi{%
818 &     \BIC@@Shl#1#2!!%
819 $     \BIC@AddXY#1#2!#1#2!!!%
820   }%
821   \BIC@Fi
822 }

```

\BigIntCalcShl

```

823 \def\BigIntCalcShl#1!{%
824   \romannumeral0%
825 & \BIC@@Shl#1!!%
826 $ \BIC@AddXY#1!#1!!!%
827 }

```

\BIC@@Shl

```

828 & \def\BIC@@Shl#1#2!{%
829 &   \ifx\#2\%
830 &     \BIC@AfterFi{%
831 &       \BIC@@@Shl0!#1%
832 &     }%
833 &   \else
834 &     \BIC@AfterFi{%
835 &       \BIC@@Shl#2!#1%
836 &     }%
837 &   \BIC@Fi
838 & }%

```



```

\BIC@@@Shl #1: carry
#2: result
#3#4: reverted number
839 & \def\BIC@@@Shl#1#2!#3#4!{%
840 & \ifx\#4\%
841 & \BIC@AfterFi{%
842 & \expandafter\BIC@Space
843 & \the\numexpr#3*2+#1\relax#2%
844 & }%
845 & \else
846 & \BIC@AfterFi{%
847 & \expandafter\BIC@@@Shl\the\numexpr#3*2+#1!#2!#4!%
848 & }%
849 & \BIC@Fi
850 & }%

\BIC@@@Shl
851 & \def\BIC@@@Shl#1!{%
852 & \ifnum#1<10 %
853 & \BIC@AfterFi{%
854 & \BIC@@Shl0#1%
855 & }%
856 & \else
857 & \BIC@AfterFi{%
858 & \BIC@@Shl#1%
859 & }%
860 & \BIC@Fi
861 & }%

\bigintcalcShr
862 \def\bigintcalcShr#1{%
863 \romannumeral0%
864 \expandafter\expandafter\expandafter\BIC@Shr
865 \bigintcalcNum{#1}!%
866 }

\BIC@Shr
867 \def\BIC@Shr#1#2!{%
868 \ifx#1-%
869 \expandafter-\romannumeral0%
870 \BIC@AfterFi{%
871 \BIC@@Shr#2!%
872 }%
873 \else
874 \BIC@AfterFi{%
875 \BIC@@Shr#1#2!%
876 }%
877 \BIC@Fi
878 }

\BigIntCalcShr
879 \def\BigIntCalcShr#1!{%
880 \romannumeral0%
881 \BIC@@Shr#1!%
882 }

\BIC@@Shr
883 \def\BIC@@Shr#1#2!{%
884 \ifcase#1 %
885 \BIC@AfterFi{ 0}%
886 \or

```

```

887 \ifx\#2\%
888 \BIC@AfterFiFi{ 0}%
889 \else
890 \BIC@AfterFiFi{%
891 \BIC@@@Shr#1#2!!%
892 }%
893 \fi
894 \else
895 \BIC@AfterFiFi{%
896 \BIC@@@Shr0#1#2!!%
897 }%
898 \BIC@Fi
899 }

\BIC@@@Shr #1: carry
#2#3: number
#4: result
900 \def\BIC@@@Shr#1#2#3!#4!{%
901 \ifx\#3\%
902 \ifodd#1#2 %
903 \BIC@AfterFiFi{%
904 & \expandafter\BIC@ShrResult\the\numexpr(#1#2-1)/2\relax
905 $ \expandafter\expandafter\expandafter\BIC@ShrResult
906 $ \csname BIC@ShrDigit#1#2\endcsname
907 #4!%
908 }%
909 \else
910 \BIC@AfterFiFi{%
911 & \expandafter\BIC@ShrResult\the\numexpr#1#2/2\relax
912 $ \expandafter\expandafter\expandafter\BIC@ShrResult
913 $ \csname BIC@ShrDigit#1#2\endcsname
914 #4!%
915 }%
916 \fi
917 \else
918 \ifodd#1#2 %
919 \BIC@AfterFiFi{%
920 & \expandafter\BIC@@@Shr\the\numexpr(#1#2-1)/2\relax1%
921 $ \expandafter\expandafter\expandafter\BIC@@@Shr
922 $ \csname BIC@ShrDigit#1#2\endcsname
923 #3!#4!%
924 }%
925 \else
926 \BIC@AfterFiFi{%
927 & \expandafter\BIC@@@Shr\the\numexpr#1#2/2\relax0%
928 $ \expandafter\expandafter\expandafter\BIC@@@Shr
929 $ \csname BIC@ShrDigit#1#2\endcsname
930 #3!#4!%
931 }%
932 \fi
933 \BIC@Fi
934 }

\BIC@ShrResult
935 & \def\BIC@ShrResult#1#2!{ #2#1}%
936 $ \def\BIC@ShrResult#1#2#3!{ #3#1}%

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

```

```

938 \BIC@@@Shr#2#3!#4#1!%
939 }

```

\BIC@ShrDigit[00-19]

```

940 $ \def\BIC@Temp#1#2#3#4{%
941 $ \expandafter\def\csname BIC@ShrDigit#1#2\endcsname{#3#4}%
942 $ }%
943 $ \BIC@Temp 0000%
944 $ \BIC@Temp 0101%
945 $ \BIC@Temp 0210%
946 $ \BIC@Temp 0311%
947 $ \BIC@Temp 0420%
948 $ \BIC@Temp 0521%
949 $ \BIC@Temp 0630%
950 $ \BIC@Temp 0731%
951 $ \BIC@Temp 0840%
952 $ \BIC@Temp 0941%
953 $ \BIC@Temp 1050%
954 $ \BIC@Temp 1151%
955 $ \BIC@Temp 1260%
956 $ \BIC@Temp 1361%
957 $ \BIC@Temp 1470%
958 $ \BIC@Temp 1571%
959 $ \BIC@Temp 1680%
960 $ \BIC@Temp 1781%
961 $ \BIC@Temp 1890%
962 $ \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

```

963 \def\BIC@@Tim#1#2!{%
964 \ifx\#2\%
965 \BIC@AfterFi{%
966 \BIC@ProcessTim0!#1%
967 }%
968 \else
969 \BIC@AfterFi{%
970 \BIC@@Tim#2!#1%
971 }%
972 \BIC@Fi
973 }

```

\BIC@ProcessTim #1: carry

#2: result

#3#4: reverted number

#5: digit

```

974 \def\BIC@ProcessTim#1#2!#3#4!#5{%
975 \ifx\#4\%
976 \BIC@AfterFi{%
977 \expandafter\BIC@Space
978 & \the\numexpr#3*#5+#1\relax
979 $ \romannumeral0\BIC@TimDigit#3#5#1%
980 #2%
981 }%
982 \else

```

```

983     \BIC@AfterFi{%
984     \expandafter\BIC@@ProcessTim
985 &     \the\numexpr#3*#5+#1%
986 $     \romannumeral0\BIC@TimDigit#3#5#1%
987     !#2!#4!#5%
988     }%
989     \BIC@Fi
990 }

\BIC@@ProcessTim #1#2: carry?, new digit
#3: new number
#4: old number
#5: digit
991 \def\BIC@@ProcessTim#1#2!{%
992     \ifx\#2\%
993     \BIC@AfterFi{%
994     \BIC@ProcessTim0#1%
995     }%
996     \else
997     \BIC@AfterFi{%
998     \BIC@ProcessTim#1#2%
999     }%
1000     \BIC@Fi
1001 }

\BIC@TimDigit #1: digit 0-9
#2: digit 3-9
#3: carry 0-9
1002 $ \def\BIC@TimDigit#1#2#3{%
1003 $     \ifcase#1 % 0
1004 $         \BIC@AfterFi{ #3}%
1005 $         \or % 1
1006 $         \BIC@AfterFi{%
1007 $             \expandafter\BIC@Space
1008 $             \number\csname BIC@AddCarry#2\endcsname#3 %
1009 $         }%
1010 $     \else
1011 $         \ifcase#3 %
1012 $             \BIC@AfterFiFi{%
1013 $                 \expandafter\BIC@Space
1014 $                 \number\csname BIC@MulDigit#2\endcsname#1 %
1015 $             }%
1016 $         \else
1017 $             \BIC@AfterFiFi{%
1018 $                 \expandafter\BIC@Space
1019 $                 \romannumeral0%
1020 $                 \expandafter\BIC@AddXY
1021 $                 \number\csname BIC@MulDigit#2\endcsname#1!%
1022 $                 #3!!!%
1023 $             }%
1024 $         \fi
1025 $     \BIC@Fi
1026 $ }%

\BIC@MulDigit[3-9]
1027 $ \def\BIC@Temp#1#2{%
1028 $     \expandafter\def\csname BIC@MulDigit#1\endcsname##1{%
1029 $         \ifcase##1 0%
1030 $         \or ##1%
1031 $         \or ##2%
1032 $     }? \else\BigIntCalcError:ThisCannotHappen%
1033 $     \fi

```

```

1034 $ }%
1035 $ }%
1036 $ \BIC@Temp 3{6\or9\or12\or15\or18\or21\or24\or27}%
1037 $ \BIC@Temp 4{8\or12\or16\or20\or24\or28\or32\or36}%
1038 $ \BIC@Temp 5{10\or15\or20\or25\or30\or35\or40\or45}%
1039 $ \BIC@Temp 6{12\or18\or24\or30\or36\or42\or48\or54}%
1040 $ \BIC@Temp 7{14\or21\or28\or35\or42\or49\or56\or63}%
1041 $ \BIC@Temp 8{16\or24\or32\or40\or48\or56\or64\or72}%
1042 $ \BIC@Temp 9{18\or27\or36\or45\or54\or63\or72\or81}%

```

2.15 Mul

\bigintcalcMul

```

1043 \def\bigintcalcMul#1#2{%
1044   \romannumeral0%
1045   \expandafter\expandafter\expandafter\BIC@Mul
1046   \bigintcalcNum{#1}!{#2}%
1047 }

```

\BIC@Mul

```

1048 \def\BIC@Mul#1!#2{%
1049   \expandafter\expandafter\expandafter\BIC@MulSwitch
1050   \bigintcalcNum{#2}!#1!%
1051 }

```

\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		$\text{Mul}(-y, -x)$

```

1052 \def\BIC@MulSwitch#1#2!#3#4!{%
1053   \ifcase\BIC@Sgn#1#2! % x = 0
1054     \BIC@AfterFi{ 0}%
1055   \or % x > 0
1056     \ifcase\BIC@Sgn#3#4! % y = 0
1057       \BIC@AfterFiFi{ 0}%
1058     \or % y > 0
1059       \ifnum\BIC@PosCmp#1#2!#3#4!=1 % x > y
1060         \BIC@AfterFiFiFi{%
1061           \BIC@ProcessMul0!#1#2!#3#4!%
1062         }%
1063       \else % x <= y
1064         \BIC@AfterFiFiFi{%
1065           \BIC@ProcessMul0!#3#4!#1#2!%
1066         }%
1067     \fi
1068   \else % y < 0
1069     \expandafter-\romannumeral0%
1070     \ifnum\BIC@PosCmp#1#2!#4!=1 % x > -y
1071       \BIC@AfterFiFiFi{%
1072         \BIC@ProcessMul0!#1#2!#4!%
1073       }%

```

```

1074     \else % x <= -y
1075         \BIC@AfterFiFiFi{%
1076             \BIC@ProcessMul0!#4!#1#2!%
1077         }%
1078     \fi
1079 \fi
1080 \else % x < 0
1081     \ifcase\BIC@Sgn#3#4! % y = 0
1082         \BIC@AfterFiFiFi{ 0}%
1083     \or % y > 0
1084         \expandafter-\romannumeral0%
1085         \ifnum\BIC@PosCmp#2!#3#4!=1 % -x > y
1086             \BIC@AfterFiFiFi{%
1087                 \BIC@ProcessMul0!#2!#3#4!%
1088             }%
1089         \else % -x <= y
1090             \BIC@AfterFiFiFi{%
1091                 \BIC@ProcessMul0!#3#4!#2!%
1092             }%
1093         \fi
1094     \else % y < 0
1095         \ifnum\BIC@PosCmp#2!#4!=1 % -x > -y
1096             \BIC@AfterFiFiFi{%
1097                 \BIC@ProcessMul0!#2!#4!%
1098             }%
1099         \else % -x <= -y
1100             \BIC@AfterFiFiFi{%
1101                 \BIC@ProcessMul0!#4!#2!%
1102             }%
1103         \fi
1104     \fi
1105 \BIC@Fi
1106 }

```

\BigIntCalcMul

```

1107 \def\BigIntCalcMul#1!#2!{%
1108     \romannumeral0%
1109     \BIC@ProcessMul0!#1!#2!%
1110 }

```

\BIC@ProcessMul

```

#1: result
#2: number x
#3#4: number y
1111 \def\BIC@ProcessMul#1!#2!#3#4!{%
1112     \ifx\#4\%
1113         \BIC@AfterFiFi{%
1114             \expandafter\expandafter\expandafter\BIC@Space
1115             \bigintcalcAdd{\BIC@Tim#2!#3}{#10}%
1116         }%
1117     \else
1118         \BIC@AfterFiFi{%
1119             \expandafter\expandafter\expandafter\BIC@ProcessMul
1120             \bigintcalcAdd{\BIC@Tim#2!#3}{#10}!#2!#4!%
1121         }%
1122     \BIC@Fi
1123 }

```

2.16 Sqr

\bigintcalcSqr

```

1124 \def\bigintcalcSqr#1{%
1125     \romannumeral0%

```

```

1126 \expandafter\expandafter\expandafter\BIC@Sqr
1127 \bigintcalcNum{#1}!%
1128 }

```

\BIC@Sqr

```

1129 \def\BIC@Sqr#1{%
1130   \ifx#1-%
1131     \expandafter\BIC@@Sqr
1132   \else
1133     \expandafter\BIC@@Sqr\expandafter#1%
1134   \fi
1135 }

```

\BIC@@Sqr

```

1136 \def\BIC@@Sqr#1!{%
1137   \BIC@ProcessMul0!#1!#1!%
1138 }

```

2.17 Fac

\bigintcalcFac

```

1139 \def\bigintcalcFac#1{%
1140   \romannumeral0%
1141   \expandafter\expandafter\expandafter\BIC@Fac
1142   \bigintcalcNum{#1}!%
1143 }

```

\BIC@Fac

```

1144 \def\BIC@Fac#1#2!{%
1145   \ifx#1-%
1146     \BIC@AfterFi{ 0\BigIntCalcError:FacNegative}%
1147   \else
1148     \ifnum\BIC@PosCmp#1#2!13!<0 %
1149       \ifcase#1#2 %
1150         \BIC@AfterFiFiFi{ 1}% 0!
1151       \or\BIC@AfterFiFiFi{ 1}% 1!
1152       \or\BIC@AfterFiFiFi{ 2}% 2!
1153       \or\BIC@AfterFiFiFi{ 6}% 3!
1154       \or\BIC@AfterFiFiFi{ 24}% 4!
1155       \or\BIC@AfterFiFiFi{ 120}% 5!
1156       \or\BIC@AfterFiFiFi{ 720}% 6!
1157       \or\BIC@AfterFiFiFi{ 5040}% 7!
1158       \or\BIC@AfterFiFiFi{ 40320}% 8!
1159       \or\BIC@AfterFiFiFi{ 362880}% 9!
1160       \or\BIC@AfterFiFiFi{ 3628800}% 10!
1161       \or\BIC@AfterFiFiFi{ 39916800}% 11!
1162       \or\BIC@AfterFiFiFi{ 479001600}% 12!
1163 ?   \else\BigIntCalcError:ThisCannotHappen%
1164     \fi
1165   \else
1166     \BIC@AfterFiFi{%
1167       \BIC@ProcessFac#1#2!479001600!%
1168     }%
1169   \fi
1170   \BIC@Fi
1171 }

```

\BIC@ProcessFac #1: *n*

#2: result

```

1172 \def\BIC@ProcessFac#1!#2!{%
1173   \ifnum\BIC@PosCmp#1!12!=0 %

```

```

1174 \BIC@AfterFi{ #2}%
1175 \else
1176 \BIC@AfterFi{%
1177 \expandafter\BIC@ProcessFac
1178 \romannumeral0\BIC@ProcessMul0!#2!#1!%
1179 !#1!%
1180 }%
1181 \BIC@Fi
1182 }

```

```

\BIC@ProcessFac #1: result
#2: n
1183 \def\BIC@ProcessFac#1!#2!{%
1184 \expandafter\BIC@ProcessFac
1185 \romannumeral0\BIC@Dec#2!{%
1186 !#1!%
1187 }

```

2.18 Pow

```

\bigintcalcPow #1: basis
#2: power
1188 \def\bigintcalcPow#1{%
1189 \romannumeral0%
1190 \expandafter\expandafter\expandafter\BIC@Pow
1191 \bigintcalcNum{#1}!%
1192 }

\BIC@Pow #1: basis
#2: power
1193 \def\BIC@Pow#1!#2{%
1194 \expandafter\expandafter\expandafter\BIC@PowSwitch
1195 \bigintcalcNum{#2}!#1!%
1196 }

\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 ($x > 1$)	$\text{Pow}(x, y)$	

```

1197 \def\BIC@PowSwitch#1#2!#3#4!{%
1198 \ifcase\ifx\\#2\\%
1199 \ifx#100 %  $y = 0$ 
1200 \else\ifx#111 %  $y = 1$ 

```



```

1201         \else\ifx#122 % y = 2
1202         \else4 % y > 2
1203         \fi\fi\fi
1204     \else
1205         \ifx#1-3 % y < 0
1206         \else4 % y > 2
1207         \fi
1208     \fi
1209     \BIC@AfterFi{ 1}% y = 0
1210 \or % y = 1
1211     \BIC@AfterFi{ #3#4}%
1212 \or % y = 2
1213     \ifx#3-% x < 0
1214         \BIC@AfterFiFi{%
1215             \BIC@ProcessMul0!#4!#4!%
1216         }%
1217     \else % x >= 0
1218         \BIC@AfterFiFi{%
1219             \BIC@ProcessMul0!#3#4!#3#4!%
1220         }%
1221     \fi
1222 \or % y < 0
1223     \ifcase\ifx\#4\%
1224         \ifx#300 % x = 0
1225         \else\ifx#311 % x = 1
1226         \else3 % x > 1
1227         \fi\fi
1228     \else
1229         \ifcase\BIC@MinusOne#3#4! %
1230             3 % |x| > 1
1231         \or
1232             2 % x = -1
1233 ?         \else\BigIntCalcError:ThisCannotHappen%
1234         \fi
1235     \fi
1236     \BIC@AfterFiFi{ 0\BigIntCalcError:DivisionByZero}% x = 0
1237 \or % x = 1
1238     \BIC@AfterFiFi{ 1}% x = 1
1239 \or % x = -1
1240     \ifcase\BIC@ModTwo#2! % even(y)
1241         \BIC@AfterFiFiFi{ 1}%
1242     \or % odd(y)
1243         \BIC@AfterFiFiFi{ -1}%
1244 ?     \else\BigIntCalcError:ThisCannotHappen%
1245     \fi
1246 \or % |x| > 1
1247     \BIC@AfterFiFi{ 0}%
1248 ?     \else\BigIntCalcError:ThisCannotHappen%
1249     \fi
1250 \or % y > 2
1251     \ifcase\ifx\#4\%
1252         \ifx#300 % x = 0
1253         \else\ifx#311 % x = 1
1254         \else4 % x > 1
1255         \fi\fi
1256     \else
1257         \ifx#3-%
1258             \ifcase\BIC@MinusOne#3#4! %
1259                 3 % x < -1
1260             \else
1261                 2 % x = -1
1262             \fi

```

```

1263         \else
1264         4 % x > 1
1265         \fi
1266     \fi
1267     \BIC@AfterFiFi{ 0}% x = 0
1268 \or % x = 1
1269     \BIC@AfterFiFi{ 1}% x = 1
1270 \or % x = -1
1271     \ifcase\BIC@ModTwo#1#2! % even(y)
1272         \BIC@AfterFiFiFi{ 1}%
1273     \or % odd(y)
1274         \BIC@AfterFiFiFi{ -1}%
1275 ?     \else\BigIntCalcError:ThisCannotHappen%
1276     \fi
1277 \or % x < -1
1278     \ifcase\BIC@ModTwo#1#2! % even(y)
1279         \BIC@AfterFiFiFi{%
1280             \BIC@PowRec#4!#1#2!1!%
1281         }%
1282     \or % odd(y)
1283         \expandafter-\romannumeral0%
1284         \BIC@AfterFiFiFi{%
1285             \BIC@PowRec#4!#1#2!1!%
1286         }%
1287 ?     \else\BigIntCalcError:ThisCannotHappen%
1288     \fi
1289 \or % x > 1
1290     \BIC@AfterFiFi{%
1291         \BIC@PowRec#3#4!#1#2!1!%
1292     }%
1293 ?     \else\BigIntCalcError:ThisCannotHappen%
1294     \fi
1295 ? \else\BigIntCalcError:ThisCannotHappen%
1296 \BIC@Fi
1297 }

```

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.

```

1298 \def\BIC@ModTwo#1#2!{%
1299     \ifx\#2\%
1300         \ifodd#1 %
1301             \BIC@AfterFiFi1%
1302         \else
1303             \BIC@AfterFiFi0%
1304         \fi
1305     \else
1306         \BIC@AfterFi{%
1307             \BIC@ModTwo#2!%
1308         }%
1309     \BIC@Fi
1310 }

```

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

```

1311 \def\BIC@MinusOne#1#2!{%
1312     \ifx#1-%
1313         \BIC@@MinusOne#2!%
1314     \else
1315         0%
1316     \fi

```

```

1317 }

\BIC@@MinusOne

1318 \def\BIC@@MinusOne#1#2!{%
1319   \ifx#11%
1320     \ifx\|#2\|%
1321       1%
1322     \else
1323       0%
1324     \fi
1325   \else
1326     0%
1327   \fi
1328 }

```

2.18.2 Recursive calculation

```

\BIC@PowRec      Pow(x, y) {
                  PowRec(x, y, 1)
                  }
                  PowRec(x, y, r) {
                    if y == 1 then
                      return r
                    else
                      ifodd y then
                        return PowRec(x*x, y div 2, r*x) % y div 2 = (y-1)/2
                      else
                        return PowRec(x*x, y div 2, r)
                      fi
                    fi
                  }
                  #1: x (basis)
                  #2#3: y (power)
                  #4: r (result)

1329 \def\BIC@PowRec#1!#2#3!#4!{%
1330   \ifcase\ifx#21\ifx\|#3\|0 \else1 \fi\else1 \fi % y = 1
1331     \ifnum\BIC@PosCmp#1!#4!=1 % x > r
1332       \BIC@AfterFiFi{%
1333         \BIC@ProcessMul0!#1!#4!%
1334       }%
1335     \else
1336       \BIC@AfterFiFi{%
1337         \BIC@ProcessMul0!#4!#1!%
1338       }%
1339     \fi
1340   \or
1341     \ifcase\BIC@ModTwo#2#3! % even(y)
1342       \BIC@AfterFiFi{%
1343         \expandafter\BIC@@PowRec\romannumeral0%
1344         \BIC@@Shr#2#3!%
1345         !#1!#4!%
1346       }%
1347     \or % odd(y)
1348       \ifnum\BIC@PosCmp#1!#4!=1 % x > r
1349         \BIC@AfterFiFiFi{%
1350           \expandafter\BIC@@@PowRec\romannumeral0%
1351           \BIC@ProcessMul0!#1!#4!%
1352           !#1!#2#3!%
1353         }%
1354       \else
1355         \BIC@AfterFiFiFi{%
1356           \expandafter\BIC@@@PowRec\romannumeral0%

```

```

1357         \BIC@ProcessMul0!#1!#4!%
1358         !#1!#2#3!%
1359     }%
1360     \fi
1361 ?     \else\BigIntCalcError:ThisCannotHappen%
1362     \fi
1363 ? \else\BigIntCalcError:ThisCannotHappen%
1364     \BIC@Fi
1365 }

```

```

\BIC@@PowRec #1:  $y/2$ 
#2:  $x$ 
#3: new  $r$  ( $r$  or  $r * x$ )
1366 \def\BIC@@PowRec#1!#2!#3!{%
1367     \expandafter\BIC@PowRec\romannumeral0%
1368     \BIC@ProcessMul0!#2!#2!%
1369     !#1!#3!%
1370 }

```

```

\BIC@@@PowRec #1:  $r * x$  #2:  $x$  #3:  $y$ 
1371 \def\BIC@@@PowRec#1!#2!#3!{%
1372     \expandafter\BIC@@PowRec\romannumeral0%
1373     \BIC@@Shr#3!%
1374     !#2!#1!%
1375 }

```

2.19 Div

```

\bigintcalcDiv #1:  $x$ 
#2:  $y$  (divisor)
1376 \def\bigintcalcDiv#1{%
1377     \romannumeral0%
1378     \expandafter\expandafter\expandafter\BIC@Div
1379     \bigintcalcNum{#1}!%
1380 }

```

```

\BIC@Div #1:  $x$ 
#2:  $y$ 
1381 \def\BIC@Div#1!#2{%
1382     \expandafter\expandafter\expandafter\BIC@DivSwitchSign
1383     \bigintcalcNum{#2}!#1!%
1384 }

```

```

\BigIntCalcDiv
1385 \def\BigIntCalcDiv#1!#2!{%
1386     \romannumeral0%
1387     \BIC@DivSwitchSign#2!#1!%
1388 }

```

\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:  $x$ 
1389 \def\BIC@DivSwitchSign#1#2!#3#4!{%
1390   \ifcase\BIC@Sgn#1#2! %  $y = 0$ 
1391     \BIC@AfterFi{ 0\BigIntCalcError:DivisionByZero}%
1392   \or %  $y > 0$ 
1393     \ifcase\BIC@Sgn#3#4! %  $x = 0$ 
1394       \BIC@AfterFiFi{ 0}%
1395     \or %  $x > 0$ 
1396       \BIC@AfterFiFi{%
1397         \BIC@DivSwitch{ }#3#4!#1#2!%
1398       }%
1399     \else %  $x < 0$ 
1400       \BIC@AfterFiFi{%
1401         \BIC@DivSwitch-#4!#1#2!%
1402       }%
1403     \fi
1404   \else %  $y < 0$ 
1405     \ifcase\BIC@Sgn#3#4! %  $x = 0$ 
1406       \BIC@AfterFiFi{ 0}%
1407     \or %  $x > 0$ 
1408       \BIC@AfterFiFi{%
1409         \BIC@DivSwitch-#3#4!#2!%
1410       }%
1411     \else %  $x < 0$ 
1412       \BIC@AfterFiFi{%
1413         \BIC@DivSwitch{ }#4!#2!%
1414       }%
1415     \fi
1416   \BIC@Fi
1417 }

```

\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:  $y$  ( $y \neq 0$ )
1418 \def\BIC@DivSwitch#1#2!#3#4!{%
1419   \ifcase\BIC@PosCmp#3#4!#2!%  $y = x$ 
1420     \BIC@AfterFi{ #1}%
1421   \or %  $y > x$ 
1422     \BIC@AfterFi{ 0}%
1423   \else %  $y < x$ 
1424     \ifx\#1\%
1425       \else
1426         \expandafter-\romannumeral0%
1427       \fi
1428     \ifcase\ifx\#4\%
1429       \ifx#310 %  $y = 1$ 
1430       \else\ifx#321 %  $y = 2$ 
1431       \else\ifx#342 %  $y = 4$ 
1432       \else3 %  $y > 2$ 
1433       \fi\fi\fi
1434     \else
1435       3 %  $y > 2$ 

```

```

1436         \fi
1437         \BIC@AfterFiFi{ #2}% y = 1
1438     \or % y = 2
1439         \BIC@AfterFiFi{%
1440             \BIC@Shr#2!%
1441         }%
1442     \or % y = 4
1443         \BIC@AfterFiFi{%
1444             \expandafter\BIC@Shr\romannumeral0%
1445             \BIC@Shr#2!%
1446         }%
1447     \or % y > 2
1448         \BIC@AfterFiFi{%
1449             \BIC@DivStartX#2!#3#4!!!%
1450         }%
1451 ? \else\BigIntCalcError:ThisCannotHappen%
1452     \fi
1453 \BIC@Fi
1454 }

\BIC@ProcessDiv #1#2:  $x$ 
#3#4:  $y$ 
#5: collect first digits of  $x$ 
#6: corresponding digits of  $y$ 
1455 \def\BIC@DivStartX#1#2!#3#4!#5!#6!{%
1456     \ifx\#4\%
1457         \BIC@AfterFi{%
1458             \BIC@DivStartYii#6#3#4!{#5#1}#2=!%
1459         }%
1460     \else
1461         \BIC@AfterFi{%
1462             \BIC@DivStartX#2!#4!#5#1!#6#3!%
1463         }%
1464     \BIC@Fi
1465 }

\BIC@DivStartYii #1:  $y$ 
#2:  $x$ , =
1466 \def\BIC@DivStartYii#1!{%
1467     \expandafter\BIC@DivStartYiv\romannumeral0%
1468     \BIC@Shl#1!%
1469     !#1!%
1470 }

\BIC@DivStartYiv #1:  $2y$ 
#2:  $y$ 
#3:  $x$ , =
1471 \def\BIC@DivStartYiv#1!{%
1472     \expandafter\BIC@DivStartYvi\romannumeral0%
1473     \BIC@Shl#1!%
1474     !#1!%
1475 }

\BIC@DivStartYvi #1:  $4y$ 
#2:  $2y$ 
#3:  $y$ 
#4:  $x$ , =
1476 \def\BIC@DivStartYvi#1!#2!{%
1477     \expandafter\BIC@DivStartYviii\romannumeral0%
1478     \BIC@AddXY#1!#2!!!%
1479     !#1!#2!%
1480 }

```

```

\BIC@DivStartYviii #1: 6y
#2: 4y
#3: 2y
#4: y
#5: x, =
1481 \def\BIC@DivStartYviii#1!#2!{%
1482 \expandafter\BIC@DivStart\romannumeral0%
1483 \BIC@Shl#2!%
1484 !#1!#2!%
1485 }

\BIC@DivStart #1: 8y
#2: 6y
#3: 4y
#4: 2y
#5: y
#6: x, =
1486 \def\BIC@DivStart#1!#2!#3!#4!#5!#6!{%
1487 \BIC@ProcessDiv#6!!#5!#4!#3!#2!#1!=%
1488 }

\BIC@ProcessDiv #1#2#3: x, =
#4: result
#5: y
#6: 2y
#7: 4y
#8: 6y
#9: 8y
1489 \def\BIC@ProcessDiv#1#2#3!#4!#5!{%
1490 \ifcase\BIC@PosCmp#5!#1!% y = #1
1491 \ifx#2=%
1492 \BIC@AfterFiFi{\BIC@DivCleanup{#41}}%
1493 \else
1494 \BIC@AfterFiFi{%
1495 \BIC@ProcessDiv#2#3!#41!#5!%
1496 }%
1497 \fi
1498 \or % y > #1
1499 \ifx#2=%
1500 \BIC@AfterFiFi{\BIC@DivCleanup{#40}}%
1501 \else
1502 \ifx\#4\%
1503 \BIC@AfterFiFiFi{%
1504 \BIC@ProcessDiv{#1#2}#3!#5!%
1505 }%
1506 \else
1507 \BIC@AfterFiFiFi{%
1508 \BIC@ProcessDiv{#1#2}#3!#40!#5!%
1509 }%
1510 \fi
1511 \fi
1512 \else % y < #1
1513 \BIC@AfterFi{%
1514 \BIC@@ProcessDiv{#1}#2#3!#4!#5!%
1515 }%
1516 \BIC@Fi
1517 }

\BIC@DivCleanup #1: result
#2: garbage
1518 \def\BIC@DivCleanup#1#2={ #1}%

```

\BIC@@ProcessDiv

```

1519 \def\BIC@@ProcessDiv#1#2#3!#4!#5!#6!#7!{%
1520   \ifcase\BIC@PosCmp#7!#1!% 4y = #1
1521     \ifx#2=%
1522       \BIC@AfterFiFi{\BIC@DivCleanup{#44}}}%
1523     \else
1524       \BIC@AfterFiFi{%
1525         \BIC@ProcessDiv#2#3!#44!#5!#6!#7!%
1526       }%
1527     \fi
1528   \or % 4y > #1
1529     \ifcase\BIC@PosCmp#6!#1!% 2y = #1
1530       \ifx#2=%
1531         \BIC@AfterFiFiFi{\BIC@DivCleanup{#42}}}%
1532       \else
1533         \BIC@AfterFiFiFi{%
1534           \BIC@ProcessDiv#2#3!#42!#5!#6!#7!%
1535         }%
1536       \fi
1537     \or % 2y > #1
1538       \ifx#2=%
1539         \BIC@AfterFiFiFi{\BIC@DivCleanup{#41}}}%
1540       \else
1541         \BIC@AfterFiFiFi{%
1542           \BIC@DivSub#1!#5!#2#3!#41!#5!#6!#7!%
1543         }%
1544       \fi
1545     \else % 2y < #1
1546       \BIC@AfterFiFi{%
1547         \expandafter\BIC@ProcessDivII\romannumeral0%
1548         \BIC@SubXY#1!#6!!!%
1549         !#2#3!#4!#5!23%
1550         #6!#7!%
1551       }%
1552     \fi
1553   \else % 4y < #1
1554     \BIC@AfterFiFi{%
1555       \BIC@@@ProcessDiv{#1}#2#3!#4!#5!#6!#7!%
1556     }%
1557   \BIC@Fi
1558 }

```

\BIC@DivSub Next token group: #1-#2 and next digit #3.

```

1559 \def\BIC@DivSub#1!#2!#3{%
1560   \expandafter\BIC@ProcessDiv\expandafter{%
1561     \romannumeral0%
1562     \BIC@SubXY#1!#2!!!%
1563     #3%
1564   }%
1565 }

```

\BIC@ProcessDivII #1: $x' - 2y$
 #2#3: remaining x , =
 #4: result
 #5: y
 #6: first possible result digit
 #7: second possible result digit

```

1566 \def\BIC@ProcessDivII#1!#2#3!#4!#5!#6#7{%
1567   \ifcase\BIC@PosCmp#5!#1!% y = #1
1568     \ifx#2=%
1569       \BIC@AfterFiFi{\BIC@DivCleanup{#4#7}}}%
1570     \else

```



```

1571      \BIC@AfterFiFi{%
1572      \BIC@ProcessDiv#2#3!#4#7!#5!%
1573      }%
1574      \fi
1575 \or % y > #1
1576 \ifx#2=%
1577 \BIC@AfterFiFi{\BIC@DivCleanup{#4#6}}%
1578 \else
1579 \BIC@AfterFiFi{%
1580 \BIC@ProcessDiv{#1#2}#3!#4#6!#5!%
1581 }%
1582 \fi
1583 \else % y < #1
1584 \ifx#2=%
1585 \BIC@AfterFiFi{\BIC@DivCleanup{#4#7}}%
1586 \else
1587 \BIC@AfterFiFi{%
1588 \BIC@DivSub#1!#5!#2#3!#4#7!#5!%
1589 }%
1590 \fi
1591 \BIC@Fi
1592 }

```

```

\BIC@ProcessDivIV #1#2#3:  $x, =, x > 4y$ 
#4: result
#5:  $y$ 
#6:  $2y$ 
#7:  $4y$ 
#8:  $6y$ 
#9:  $8y$ 

1593 \def\BIC@@@ProcessDiv#1#2#3!#4!#5!#6!#7!#8!#9!{%
1594 \ifcase\BIC@PosCmp#8!#1!% 6y = #1
1595 \ifx#2=%
1596 \BIC@AfterFiFi{\BIC@DivCleanup{#46}}%
1597 \else
1598 \BIC@AfterFiFi{%
1599 \BIC@ProcessDiv#2#3!#46!#5!#6!#7!#8!#9!%
1600 }%
1601 \fi
1602 \or % 6y > #1
1603 \BIC@AfterFi{%
1604 \expandafter\BIC@ProcessDivII\romannumeral0%
1605 \BIC@SubXY#1!#7!!!%
1606 !#2#3!#4!#5!45%
1607 #6!#7!#8!#9!%
1608 }%
1609 \else % 6y < #1
1610 \ifcase\BIC@PosCmp#9!#1!% 8y = #1
1611 \ifx#2=%
1612 \BIC@AfterFiFiFi{\BIC@DivCleanup{#48}}%
1613 \else
1614 \BIC@AfterFiFiFi{%
1615 \BIC@ProcessDiv#2#3!#48!#5!#6!#7!#8!#9!%
1616 }%
1617 \fi
1618 \or % 8y > #1
1619 \BIC@AfterFiFi{%
1620 \expandafter\BIC@ProcessDivII\romannumeral0%
1621 \BIC@SubXY#1!#8!!!%
1622 !#2#3!#4!#5!67%
1623 #6!#7!#8!#9!%
1624 }%

```

```

1625 \else % 8y < #1
1626 \BIC@AfterFiFi{%
1627 \expandafter\BIC@ProcessDivII\romannumeral0%
1628 \BIC@SubXY#1!#9!!!%
1629 !#2#3!#4!#5!89%
1630 #6!#7!#8!#9!%
1631 }%
1632 \fi
1633 \BIC@Fi
1634 }

```

2.20 Mod

```

\bigintcalcMod #1: x
#2: y
1635 \def\bigintcalcMod#1{%
1636 \romannumeral0%
1637 \expandafter\expandafter\expandafter\BIC@Mod
1638 \bigintcalcNum{#1}!%
1639 }

\BIC@Mod #1: x
#2: y
1640 \def\BIC@Mod#1!#2{%
1641 \expandafter\expandafter\expandafter\BIC@ModSwitchSign
1642 \bigintcalcNum{#2}!#1!%
1643 }

\BigIntCalcMod
1644 \def\BigIntCalcMod#1!#2!{%
1645 \romannumeral0%
1646 \BIC@ModSwitchSign#2!#1!%
1647 }

```

\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
1648 \def\BIC@ModSwitchSign#1#2!#3#4!{%
1649 \ifcase\ifx\#2\%
1650 \ifx#100 % y = 0
1651 \else1 % y > 0
1652 \fi
1653 \else
1654 \ifx#1-2 % y < 0
1655 \else1 % y > 0
1656 \fi
1657 \fi
1658 \BIC@AfterFi{ 0\BigIntCalcError:DivisionByZero}%
1659 \or % y > 0
1660 \ifcase\ifx\#4\%
1661 \BIC@AfterFiFi{ 0}%
1662 \else
1663 \BIC@AfterFiFi{%
1664 \BIC@ModSwitch{ }#3#4!#1#2!%
1665 }%

```

```

1666 \fi
1667 \else % y < 0
1668 \ifcase\ifx\#4\%
1669 \ifx#300 % x = 0
1670 \else1 % x > 0
1671 \fi
1672 \else
1673 \ifx#3-2 % x < 0
1674 \else1 % x > 0
1675 \fi
1676 \fi
1677 \BIC@AfterFiFi{ 0}%
1678 \or % x > 0
1679 \BIC@AfterFiFi{%
1680 \BIC@ModSwitch--#3#4!#2!%
1681 }%
1682 \else % x < 0
1683 \BIC@AfterFiFi{%
1684 \BIC@ModSwitch-#4!#2!%
1685 }%
1686 \fi
1687 \BIC@Fi
1688 }

```

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

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

#1: sign

#2#3: x

#4#5: y

```

1689 \def\BIC@ModSwitch#1#2#3!#4#5!{%
1690 \ifcase\ifx\#5\%
1691 \ifx#410 % y = 1
1692 \else\ifx#421 % y = 2
1693 \else2 % y > 2
1694 \fi\fi
1695 \else2 % y > 2
1696 \fi
1697 \BIC@AfterFi{ 0}% y = 1
1698 \or % y = 2
1699 \ifcase\BIC@ModTwo#2#3! % even(x)
1700 \BIC@AfterFiFi{ 0}%
1701 \or % odd(x)
1702 \BIC@AfterFiFi{ #11}%
1703 ? \else\BigIntCalcError:ThisCannotHappen%
1704 \fi
1705 \or % y > 2
1706 \ifx\#1\%
1707 \else
1708 \expandafter\BIC@Space\romannumeral0%
1709 \expandafter\BIC@ModMinus\romannumeral0%
1710 \fi
1711 \ifx#2-% x < 0
1712 \BIC@AfterFiFi{%
1713 \expandafter\expandafter\expandafter\BIC@ModX
1714 \bigintcalcSub{#2#3}{%
1715 \bigintcalcMul{#4#5}{\bigintcalcDiv{#2#3}{#4#5}}}%

```

```

1716         }!#4#5!%
1717     }%
1718     \else % x > 0
1719         \BIC@AfterFiFi{%
1720             \expandafter\expandafter\expandafter\BIC@Space
1721             \bigintcalcSub{#2#3}{%
1722                 \bigintcalcMul{#4#5}{\bigintcalcDiv{#2#3}{#4#5}}%
1723             }%
1724         }%
1725     \fi
1726 ? \else\BigIntCalcError:ThisCannotHappen%
1727     \BIC@Fi
1728 }

\BIC@ModMinus

1729 \def\BIC@ModMinus#1{%
1730     \ifx#10%
1731         \BIC@AfterFi{ 0}%
1732     \else
1733         \BIC@AfterFi{ -#1}%
1734     \BIC@Fi
1735 }

\BIC@ModX #1#2: z
#3: x
1736 \def\BIC@ModX#1#2!#3!{%
1737     \ifx#1-% z < 0
1738         \BIC@AfterFi{%
1739             \expandafter\BIC@Space\romannumeral0%
1740             \BIC@SubXY#3!#2!!!%
1741         }%
1742     \else % z >= 0
1743         \BIC@AfterFi{ #1#2}%
1744     \BIC@Fi
1745 }

1746 \BIC@AtEnd
1747 </package>

```

3 Test

3.1 Catcode checks for loading

```

1748 <*test1>

1749 \catcode'\{=1 %
1750 \catcode'\}=2 %
1751 \catcode'\#=6 %
1752 \catcode'\@=11 %
1753 \expandafter\ifx\csname count@\endcsname\relax
1754     \countdef\count@=255 %
1755 \fi
1756 \expandafter\ifx\csname @gobble\endcsname\relax
1757     \long\def\@gobble#1{}%
1758 \fi
1759 \expandafter\ifx\csname @firstofone\endcsname\relax
1760     \long\def\@firstofone#1{#1}%
1761 \fi
1762 \expandafter\ifx\csname loop\endcsname\relax
1763     \expandafter\@firstofone
1764 \else
1765     \expandafter\@gobble

```

```

1766 \fi
1767 {%
1768   \def\loop#1\repeat{%
1769     \def\body{#1}%
1770     \iterate
1771   }%
1772   \def\iterate{%
1773     \body
1774     \let\next\iterate
1775   \else
1776     \let\next\relax
1777   \fi
1778   \next
1779 }%
1780 \let\repeat=\fi
1781 }%
1782 \def\RestoreCatcodes{}
1783 \count@=0 %
1784 \loop
1785   \edef\RestoreCatcodes{%
1786     \RestoreCatcodes
1787     \catcode\the\count@=\the\catcode\count@\relax
1788   }%
1789   \ifnum\count@<255 %
1790     \advance\count@ 1 %
1791   \repeat
1792
1793 \def\RangeCatcodeInvalid#1#2{%
1794   \count@=#1\relax
1795   \loop
1796     \catcode\count@=15 %
1797     \ifnum\count@<#2\relax
1798       \advance\count@ 1 %
1799     \repeat
1800 }
1801 \expandafter\ifx\csname LoadCommand\endcsname\relax
1802   \def\LoadCommand{\input bigintcalc.sty\relax}%
1803 \fi
1804 \def\Test{%
1805   \RangeCatcodeInvalid{0}{47}%
1806   \RangeCatcodeInvalid{58}{64}%
1807   \RangeCatcodeInvalid{91}{96}%
1808   \RangeCatcodeInvalid{123}{255}%
1809   \catcode'\@=12 %
1810   \catcode'\@=0 %
1811   \catcode'\{=1 %
1812   \catcode'\}=2 %
1813   \catcode'\#=6 %
1814   \catcode'\[=12 %
1815   \catcode'\]=12 %
1816   \catcode'\%=14 %
1817   \catcode'\ =10 %
1818   \catcode13=5 %
1819   \LoadCommand
1820   \RestoreCatcodes
1821 }
1822 \Test
1823 \csname @@end\endcsname
1824 \end
1825 </test1>

```

3.2 Macro tests

3.2.1 Preamble with test macro definitions

```
1826 <*test2>
1827 \NeedsTeXFormat{LaTeX2e}
1828 \nofiles
1829 \documentclass{article}
1830 <noetex> \let \SavedNumexpr \numexpr
1831 <noetex> \let \numexpr \UNDEFINED
1832 \makeatletter
1833 \chardef \BIC@TestMode=1 %
1834 \makeatother
1835 \usepackage{bigintcalc}[2007/11/11]
1836 <noetex> \let \numexpr \SavedNumexpr
1837 \usepackage{qstest}
1838 \IncludeTests{*}
1839 \LogTests{log}{*}{*}
1840 \newcommand*{\TestSpaceAtEnd}[1]{%
1841 <noetex> \let \SavedNumexpr \numexpr
1842 <noetex> \let \numexpr \UNDEFINED
1843 \edef\resultA{#1}%
1844 \edef\resultB{#1 }%
1845 <noetex> \let \numexpr \SavedNumexpr
1846 \Expect*{\resultA\space}*{\resultB}%
1847 }
1848 \newcommand*{\TestResult}[2]{%
1849 <noetex> \let \SavedNumexpr \numexpr
1850 <noetex> \let \numexpr \UNDEFINED
1851 \edef\result{#1}%
1852 <noetex> \let \numexpr \SavedNumexpr
1853 \Expect*{\result}{#2}%
1854 }
1855 \newcommand*{\TestResultTwoExpansions}[2]{%
1856 <*noetex>
1857 \begingroup
1858 \let \numexpr \UNDEFINED
1859 \expandafter\expandafter\expandafter
1860 \endgroup
1861 </noetex>
1862 \expandafter\expandafter\expandafter\Expect
1863 \expandafter\expandafter\expandafter{#1}{#2}%
1864 }
1865 \newcount\TestCount
1866 <etex> \newcommand*{\TestArg}[1]{\numexpr#1\relax}
1867 <noetex> \newcommand*{\TestArg}[1]{#1}
1868 \newcommand*{\TestTeXDivide}[2]{%
1869 \TestCount=\TestArg{#1}\relax
1870 \divide\TestCount by \TestArg{#2}\relax
1871 \Expect*{\bigintcalcDiv{#1}{#2}}*{\the\TestCount}%
1872 }
1873 \newcommand*{\Test}[2]{%
1874 \TestResult{#1}{#2}%
1875 \TestResultTwoExpansions{#1}{#2}%
1876 \TestSpaceAtEnd{#1}%
1877 }
1878 \newcommand*{\TestExch}[2]{\Test{#2}{#1}}
1879 \newcommand*{\TestInv}[2]{%
1880 \Test{\bigintcalcInv{#1}}{#2}%
1881 }
1882 \newcommand*{\TestAbs}[2]{%
1883 \Test{\bigintcalcAbs{#1}}{#2}%
1884 }
```

```

1885 \newcommand*{\TestSgn}[2]{%
1886   \Test{\bigintcalcSgn{#1}}{#2}%
1887 }
1888 \newcommand*{\TestMin}[3]{%
1889   \Test{\bigintcalcMin{#1}{#2}}{#3}%
1890 }
1891 \newcommand*{\TestMax}[3]{%
1892   \Test{\bigintcalcMax{#1}{#2}}{#3}%
1893 }
1894 \newcommand*{\TestCmp}[3]{%
1895   \Test{\bigintcalcCmp{#1}{#2}}{#3}%
1896 }
1897 \newcommand*{\TestOdd}[2]{%
1898   \Test{\bigintcalcOdd{#1}}{#2}%
1899   \edef\x{%
1900     \noexpand\Test{%
1901       \noexpand\BigIntCalcOdd
1902       \bigintcalcAbs{#1}!%
1903     }{#2}%
1904   }%
1905   \x
1906 }
1907 \newcommand*{\TestInc}[2]{%
1908   \Test{\bigintcalcInc{#1}}{#2}%
1909   \ifnum\bigintcalcSgn{#1}>-1 %
1910     \edef\x{%
1911       \noexpand\Test{%
1912         \noexpand\BigIntCalcInc\bigintcalcNum{#1}!%
1913       }{#2}%
1914     }%
1915     \x
1916   \fi
1917 }
1918 \newcommand*{\TestDec}[2]{%
1919   \Test{\bigintcalcDec{#1}}{#2}%
1920   \ifnum\bigintcalcSgn{#1}>0 %
1921     \edef\x{%
1922       \noexpand\Test{%
1923         \noexpand\BigIntCalcDec\bigintcalcNum{#1}!%
1924       }{#2}%
1925     }%
1926     \x
1927   \fi
1928 }
1929 \newcommand*{\TestAdd}[3]{%
1930   \Test{\bigintcalcAdd{#1}{#2}}{#3}%
1931   \ifnum\bigintcalcSgn{#1}>0 %
1932     \ifnum\bigintcalcSgn{#2}> 0 %
1933       \ifnum\bigintcalcCmp{#1}{#2}>0 %
1934         \edef\x{%
1935           \noexpand\Test{%
1936             \noexpand\BigIntCalcAdd
1937             \bigintcalcNum{#1}!\bigintcalcNum{#2}!%
1938           }{#3}%
1939         }%
1940         \x
1941       \else
1942         \edef\x{%
1943           \noexpand\Test{%
1944             \noexpand\BigIntCalcAdd
1945             \bigintcalcNum{#2}!\bigintcalcNum{#1}!%
1946           }{#3}%

```

```

1947         }%
1948         \x
1949     \fi
1950 \fi
1951 \fi
1952 }
1953 \newcommand*{\TestSub}[3]{%
1954   \Test{\bigintcalcSub{#1}{#2}}{#3}%
1955   \ifnum\bigintcalcSgn{#1}>0 %
1956     \ifnum\bigintcalcSgn{#2}> 0 %
1957       \ifnum\bigintcalcCmp{#1}{#2}>0 %
1958         \edef\x{%
1959           \noexpand\Test{%
1960             \noexpand\BigIntCalcSub
1961             \bigintcalcNum{#1}!\bigintcalcNum{#2}!%
1962           }{#3}%
1963         }%
1964         \x
1965       \fi
1966     \fi
1967   \fi
1968 }
1969 \newcommand*{\TestShl}[2]{%
1970   \Test{\bigintcalcShl{#1}}{#2}%
1971   \edef\x{%
1972     \noexpand\Test{%
1973       \noexpand\BigIntCalcShl\bigintcalcAbs{#1}!%
1974     }{\bigintcalcAbs{#2}}%
1975   }%
1976   \x
1977 }
1978 \newcommand*{\TestShr}[2]{%
1979   \Test{\bigintcalcShr{#1}}{#2}%
1980   \edef\x{%
1981     \noexpand\Test{%
1982       \noexpand\BigIntCalcShr\bigintcalcAbs{#1}!%
1983     }{\bigintcalcAbs{#2}}%
1984   }%
1985   \x
1986 }
1987 \newcommand*{\TestMul}[3]{%
1988   \Test{\bigintcalcMul{#1}{#2}}{#3}%
1989   \edef\x{%
1990     \noexpand\Test{%
1991       \noexpand\BigIntCalcMul
1992       \bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
1993     }{\bigintcalcAbs{#3}}%
1994   }%
1995   \x
1996 }
1997 \newcommand*{\TestSqr}[2]{%
1998   \Test{\bigintcalcSqr{#1}}{#2}%
1999 }
2000 \newcommand*{\TestFac}[2]{%
2001   \expandafter\TestExch\expandafter{%
2002     \the\numexpr#2%
2003   }{\bigintcalcFac{#1}}%
2004 }
2005 \newcommand*{\TestFacBig}[2]{%
2006   \Test{\bigintcalcFac{#1}}{#2}%
2007 }
2008 \newcommand*{\TestPow}[3]{%

```



```

2009 \Test{\bigintcalcPow{#1}{#2}}{#3}%
2010 }
2011 \newcommand*{\TestDiv}[3]{%
2012 \Test{\bigintcalcDiv{#1}{#2}}{#3}%
2013 \TestTeXDivide{#1}{#2}%
2014 }
2015 \newcommand*{\TestDivBig}[3]{%
2016 \Test{\bigintcalcDiv{#1}{#2}}{#3}%
2017 \edef\x{%
2018 \noexpand\Test{%
2019 \noexpand\BigIntCalcDiv\bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
2020 }\bigintcalcAbs{#3}}%
2021 }%
2022 }
2023 \newcommand*{\TestMod}[3]{%
2024 \Test{\bigintcalcMod{#1}{#2}}{#3}%
2025 \ifcase\ifcase\bigintcalcSgn{#1} 0%
2026 \or
2027 \ifcase\bigintcalcSgn{#2} 1%
2028 \or 0%
2029 \else 1%
2030 \fi
2031 \else
2032 \ifcase\bigintcalcSgn{#2} 1%
2033 \or 1%
2034 \else 0%
2035 \fi
2036 \fi\relax
2037 \edef\x{%
2038 \noexpand\Test{%
2039 \noexpand\BigIntCalcMod
2040 \bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
2041 }\bigintcalcAbs{#3}}%
2042 }%
2043 \x
2044 \fi
2045 }

```

3.2.2 Time

```

2046 \beginingroup\expandafter\expandafter\expandafter\endgroup
2047 \expandafter\ifx\csname pdfresettimer\endcsname\relax
2048 \else
2049 \makeatletter
2050 \newcount\SummaryTime
2051 \newcount\TestTime
2052 \SummaryTime=\z@
2053 \newcommand*{\PrintTime}[2]{%
2054 \typeout{%
2055 [Time #1: \strip@pt\dimexpr\number#2sp\relax\space s]%
2056 }%
2057 }%
2058 \newcommand*{\StartTime}[1]{%
2059 \renewcommand*{\TimeDescription}{#1}%
2060 \pdfresettimer
2061 }%
2062 \newcommand*{\TimeDescription}{}%
2063 \newcommand*{\StopTime}{%
2064 \TestTime=\pdfelapsedtime
2065 \global\advance\SummaryTime\TestTime
2066 \PrintTime\TimeDescription\TestTime
2067 }%
2068 \let\saved@qstest\qstest
2069 \let\saved@endqstest\endqstest

```

```

2070 \def\qstest#1#2{%
2071   \saved@qstest{#1}{#2}%
2072   \StartTime{#1}%
2073 }%
2074 \def\endqstest{%
2075   \StopTime
2076   \saved@endqstest
2077 }%
2078 \AtEndDocument{%
2079   \PrintTime{summary}\SummaryTime
2080 }%
2081 \makeatother
2082 \fi

```

3.2.3 Test sets

```

2083 \makeatletter
2084
2085 \begin{qstest}{inv}{inv}%
2086   \TestInv{0}{0}%
2087   \TestInv{1}{-1}%
2088   \TestInv{-1}{1}%
2089   \TestInv{10}{-10}%
2090   \TestInv{-10}{10}%
2091   \TestInv{2147483647}{-2147483647}%
2092   \TestInv{-2147483647}{2147483647}%
2093   \TestInv{12345678901234567890}{-12345678901234567890}%
2094   \TestInv{-12345678901234567890}{12345678901234567890}%
2095   \TestInv{ 0 }{0}%
2096   \TestInv{ 1 }{-1}%
2097   \TestInv{--1}{-1}%
2098   \TestInv{\number\z@}{0}%
2099   \TestInv{\ifx\relax\relax1\fi}{-1}%
2100   \TestInv{\ifx\relax\relax-\fi\ifx234\else1\fi}{1}%
2101 \end{qstest}
2102
2103 \begin{qstest}{abs}{abs}%
2104   \TestAbs{0}{0}%
2105   \TestAbs{1}{1}%
2106   \TestAbs{-1}{1}%
2107   \TestAbs{10}{10}%
2108   \TestAbs{-10}{10}%
2109   \TestAbs{2147483647}{2147483647}%
2110   \TestAbs{-2147483647}{2147483647}%
2111   \TestAbs{12345678901234567890}{12345678901234567890}%
2112   \TestAbs{-12345678901234567890}{12345678901234567890}%
2113   \TestAbs{ 0 }{0}%
2114   \TestAbs{ 1 }{1}%
2115   \TestAbs{--1}{1}%
2116   \TestAbs{-++1}{1}%
2117   \TestAbs{00000000000}{0}%
2118   \TestAbs{00000001000}{1000}%
2119   \TestAbs{\ifx\relax\relax 0\else 1\fi}{0}%
2120 \end{qstest}
2121
2122 \begin{qstest}{sign}{sign}%
2123   \TestSgn{0}{0}%
2124   \TestSgn{1}{1}%
2125   \TestSgn{-1}{-1}%
2126   \TestSgn{10}{1}%
2127   \TestSgn{-10}{-1}%
2128   \TestSgn{2147483647}{1}%
2129   \TestSgn{-2147483647}{-1}%
2130   \TestSgn{12345678901234567890}{1}%

```

```

2131 \TestSgn{-12345678901234567890}{-1}%
2132 \TestSgn{ 0 }{0}%
2133 \TestSgn{ 2 }{1}%
2134 \TestSgn{ -2 }{-1}%
2135 \TestSgn{--2}{1}%
2136 \TestSgn{\number\z@}{0}%
2137 \TestSgn{\number\@ne}{1}%
2138 \TestSgn{\number\m@ne}{-1}%
2139 \TestSgn{%
2140   -++\number\z@\number\z@
2141   \iftrue1\fi\iftrue2\fi\iftrue3\fi
2142 }{1}%
2143 \end{qstest}
2144
2145 \begin{qstest}{min}{min}%
2146 \TestMin{0}{1}{0}%
2147 \TestMin{1}{0}{0}%
2148 \TestMin{-10}{-20}{-20}%
2149 \TestMin{ 1 }{ 2 }{1}%
2150 \TestMin{ 2 }{ 1 }{1}%
2151 \TestMin{1}{1}{1}%
2152 \TestMin{\number\z@}{\number\@ne}{0}%
2153 \TestMin{\number\@ne}{\number\m@ne}{-1}%
2154 \end{qstest}
2155
2156 \begin{qstest}{max}{max}%
2157 \TestMax{0}{1}{1}%
2158 \TestMax{1}{0}{1}%
2159 \TestMax{-10}{-20}{-10}%
2160 \TestMax{ 1 }{ 2 }{2}%
2161 \TestMax{ 2 }{ 1 }{2}%
2162 \TestMax{1}{1}{1}%
2163 \TestMax{\number\z@}{\number\@ne}{1}%
2164 \TestMax{\number\@ne}{\number\m@ne}{1}%
2165 \end{qstest}
2166
2167 \begin{qstest}{cmp}{cmp}%
2168 \TestCmp{0}{0}{0}%
2169 \TestCmp{-21}{17}{-1}%
2170 \TestCmp{3}{4}{-1}%
2171 \TestCmp{-10}{-10}{0}%
2172 \TestCmp{-10}{-11}{1}%
2173 \TestCmp{100}{5}{1}%
2174 \TestCmp{9}{10}{-1}%
2175 \TestCmp{10}{9}{1}%
2176 \TestCmp{ 3 }{ 3 }{0}%
2177 \TestCmp{-9}{-10}{1}%
2178 \TestCmp{-10}{-9}{-1}%
2179 \TestCmp{-3}{-3}{0}%
2180 \TestCmp{0}{-2}{1}%
2181 \TestCmp{0}{2}{-1}%
2182 \TestCmp{2}{0}{1}%
2183 \TestCmp{-2}{0}{-1}%
2184 \TestCmp{12}{11}{1}%
2185 \TestCmp{11}{12}{-1}%
2186 \TestCmp{2147483647}{-2147483647}{1}%
2187 \TestCmp{-2147483647}{2147483647}{-1}%
2188 \TestCmp{2147483647}{2147483647}{0}%
2189 \TestCmp{\number\z@}{\number\@ne}{-1}%
2190 \TestCmp{\number\@ne}{\number\m@ne}{1}%
2191 \TestCmp{ 4 }{ 5 }{-1}%
2192 \TestCmp{ -3 }{ -7 }{1}%

```

```

2193 \end{qstest}
2194
2195 \begin{qstest}{odd}{odd}
2196 \tracingmacros=1
2197 \TestOdd{0}{0}%
2198 \TestOdd{1}{1}%
2199 \TestOdd{2}{0}%
2200 \TestOdd{3}{1}%
2201 \TestOdd{14}{0}%
2202 \TestOdd{15}{1}%
2203 \TestOdd{12345678901234567896}{0}%
2204 \TestOdd{12345678901234567897}{1}%
2205 \end{qstest}
2206
2207 \begin{qstest}{inc}{inc}%
2208 \TestInc{0}{1}%
2209 \TestInc{1}{2}%
2210 \TestInc{-1}{0}%
2211 \TestInc{10}{11}%
2212 \TestInc{-10}{-9}%
2213 \TestInc{ 3 }{4}%
2214 \TestInc{999}{1000}%
2215 \TestInc{-1000}{-999}%
2216 \TestInc{129}{130}%
2217 \TestInc{2147483646}{2147483647}%
2218 \TestInc{-2147483647}{-2147483646}%
2219 \TestInc{12345678901234567890}{12345678901234567891}%
2220 \TestInc{9999999999999999999}{10000000000000000000}%
2221 \TestInc{-12345678901234567891}{-12345678901234567890}%
2222 \TestInc{-10000000000000000000}{-9999999999999999999}%
2223 \end{qstest}
2224
2225 \begin{qstest}{dec}{dec}%
2226 \TestDec{0}{-1}%
2227 \TestDec{1}{0}%
2228 \TestDec{-1}{-2}%
2229 \TestDec{10}{9}%
2230 \TestDec{-10}{-11}%
2231 \TestDec{1000}{999}%
2232 \TestDec{-999}{-1000}%
2233 \TestDec{130}{129}%
2234 \TestDec{2147483647}{2147483646}%
2235 \TestDec{-2147483646}{-2147483647}%
2236 \TestDec{12345678901234567891}{12345678901234567890}%
2237 \TestDec{10000000000000000000}{9999999999999999999}%
2238 \TestDec{-12345678901234567890}{-12345678901234567891}%
2239 \TestDec{-9999999999999999999}{-10000000000000000000}%
2240 \end{qstest}
2241
2242 \begin{qstest}{add}{add}%
2243 \TestAdd{0}{0}{0}%
2244 \TestAdd{1}{0}{1}%
2245 \TestAdd{0}{1}{1}%
2246 \TestAdd{1}{2}{3}%
2247 \TestAdd{-1}{-1}{-2}%
2248 \TestAdd{2147483646}{1}{2147483647}%
2249 \TestAdd{-2147483647}{2147483647}{0}%
2250 \TestAdd{20}{-5}{15}%
2251 \TestAdd{-4}{-1}{-5}%
2252 \TestAdd{-1}{-4}{-5}%
2253 \TestAdd{-4}{1}{-3}%
2254 \TestAdd{-1}{4}{3}%

```

```

2255 \TestAdd{4}{-1}{3}%
2256 \TestAdd{1}{-4}{-3}%
2257 \TestAdd{-4}{-1}{-5}%
2258 \TestAdd{-1}{-4}{-5}%
2259 \TestAdd{ -4 }{ -1 }{-5}%
2260 \TestAdd{ -1 }{ -4 }{-5}%
2261 \TestAdd{ -4 }{ 1 }{-3}%
2262 \TestAdd{ -1 }{ 4 }{3}%
2263 \TestAdd{ 4 }{ -1 }{3}%
2264 \TestAdd{ 1 }{ -4 }{-3}%
2265 \TestAdd{ -4 }{ -1 }{-5}%
2266 \TestAdd{ -1 }{ -4 }{-5}%
2267 \TestAdd{876543210}{111111111}{987654321}%
2268 \TestAdd{999999999}{2}{1000000001}%
2269 \end{qstest}
2270
2271 \begin{qstest}{sub}{sub}
2272 \TestSub{0}{0}{0}%
2273 \TestSub{1}{0}{1}%
2274 \TestSub{1}{2}{-1}%
2275 \TestSub{-1}{-1}{0}%
2276 \TestSub{2147483646}{-1}{2147483647}%
2277 \TestSub{-2147483647}{-2147483647}{0}%
2278 \TestSub{-4}{-1}{-3}%
2279 \TestSub{-1}{-4}{3}%
2280 \TestSub{-4}{1}{-5}%
2281 \TestSub{-1}{4}{-5}%
2282 \TestSub{4}{-1}{5}%
2283 \TestSub{1}{-4}{5}%
2284 \TestSub{-4}{-1}{-3}%
2285 \TestSub{-1}{-4}{3}%
2286 \TestSub{ -4 }{ -1 }{-3}%
2287 \TestSub{ -1 }{ -4 }{3}%
2288 \TestSub{ -4 }{ 1 }{-5}%
2289 \TestSub{ -1 }{ 4 }{-5}%
2290 \TestSub{ 4 }{ -1 }{5}%
2291 \TestSub{ 1 }{ -4 }{5}%
2292 \TestSub{ -4 }{ -1 }{-3}%
2293 \TestSub{ -1 }{ -4 }{3}%
2294 \TestSub{1000000000}{2}{999999998}%
2295 \TestSub{987654321}{111111111}{876543210}%
2296 \end{qstest}
2297
2298 \begin{qstest}{shl}{shl}
2299 \TestShl{0}{0}%
2300 \TestShl{1}{2}%
2301 \TestShl{2}{4}%
2302 \TestShl{5621}{11242}%
2303 \TestShl{1073741823}{2147483646}%
2304 \end{qstest}
2305
2306 \begin{qstest}{shr}{shr}
2307 \TestShr{0}{0}%
2308 \TestShr{1}{0}%
2309 \TestShr{2}{1}%
2310 \TestShr{3}{1}%
2311 \TestShr{4}{2}%
2312 \TestShr{5}{2}%
2313 \TestShr{6}{3}%
2314 \TestShr{7}{3}%
2315 \TestShr{8}{4}%
2316 \TestShr{9}{4}%

```

```

2317 \TestShr{10}{5}%
2318 \TestShr{11}{5}%
2319 \TestShr{12}{6}%
2320 \TestShr{13}{6}%
2321 \TestShr{14}{7}%
2322 \TestShr{15}{7}%
2323 \TestShr{16}{8}%
2324 \TestShr{17}{8}%
2325 \TestShr{18}{9}%
2326 \TestShr{19}{9}%
2327 \TestShr{20}{10}%
2328 \TestShr{21}{10}%
2329 \TestShr{22}{11}%
2330 \TestShr{11241}{5620}%
2331 \TestShr{73054202}{36527101}%
2332 \TestShr{2147483646}{1073741823}%
2333 \end{qstest}
2334
2335 \begin{qstest}{mul}{mul}
2336 \TestMul{0}{0}{0}%
2337 \TestMul{1}{0}{0}%
2338 \TestMul{0}{1}{0}%
2339 \TestMul{1}{1}{1}%
2340 \TestMul{3}{1}{3}%
2341 \TestMul{1}{-3}{-3}%
2342 \TestMul{-4}{-5}{20}%
2343 \TestMul{3}{7}{21}%
2344 \TestMul{7}{3}{21}%
2345 \TestMul{3}{-7}{-21}%
2346 \TestMul{7}{-3}{-21}%
2347 \TestMul{-3}{7}{-21}%
2348 \TestMul{-7}{3}{-21}%
2349 \TestMul{-3}{-7}{21}%
2350 \TestMul{-7}{-3}{21}%
2351 \TestMul{12}{11}{132}%
2352 \TestMul{999}{333}{332667}%
2353 \TestMul{1000}{4321}{4321000}%
2354 \TestMul{12345}{173955}{2147474475}%
2355 \TestMul{1073741823}{2}{2147483646}%
2356 \TestMul{2}{1073741823}{2147483646}%
2357 \TestMul{-1073741823}{2}{-2147483646}%
2358 \TestMul{2}{-1073741823}{-2147483646}%
2359 \TestMul{6706022400}{13}{87178291200}%
2360 \end{qstest}
2361
2362 \begin{qstest}{sqr}{sqr}
2363 \TestSqr{0}{0}%
2364 \TestSqr{1}{1}%
2365 \TestSqr{2}{4}%
2366 \TestSqr{3}{9}%
2367 \TestSqr{4}{16}%
2368 \TestSqr{9}{81}%
2369 \TestSqr{10}{100}%
2370 \TestSqr{46340}{2147395600}%
2371 \TestSqr{-1}{1}%
2372 \TestSqr{-2}{4}%
2373 \TestSqr{-46340}{2147395600}%
2374 \end{qstest}
2375
2376 \begin{qstest}{fac}{fac}
2377 \TestFac{0}{1}%
2378 \TestFac{1}{1}%

```

```

2379 \TestFac{2}{2}%
2380 \TestFac{3}{2*3}%
2381 \TestFac{4}{2*3*4}%
2382 \TestFac{5}{2*3*4*5}%
2383 \TestFac{6}{2*3*4*5*6}%
2384 \TestFac{7}{2*3*4*5*6*7}%
2385 \TestFac{8}{2*3*4*5*6*7*8}%
2386 \TestFac{9}{2*3*4*5*6*7*8*9}%
2387 \TestFac{10}{2*3*4*5*6*7*8*9*10}%
2388 \TestFac{11}{2*3*4*5*6*7*8*9*10*11}%
2389 \TestFac{12}{2*3*4*5*6*7*8*9*10*11*12}%
2390 \TestFacBig{13}{6227020800}%
2391 \TestFacBig{14}{87178291200}%
2392 \TestFacBig{15}{1307674368000}%
2393 \TestFacBig{16}{20922789888000}%
2394 \TestFacBig{17}{355687428096000}%
2395 \TestFacBig{18}{6402373705728000}%
2396 \TestFacBig{19}{121645100408832000}%
2397 \TestFacBig{20}{2432902008176640000}%
2398 \TestFacBig{21}{51090942171709440000}%
2399 \TestFacBig{22}{112400072777607680000}%
2400 \end{qstest}
2401
2402 \begin{qstest}{pow}{pow}
2403 \TestPow{-2}{0}{1}%
2404 \TestPow{-1}{0}{1}%
2405 \TestPow{0}{0}{1}%
2406 \TestPow{1}{0}{1}%
2407 \TestPow{2}{0}{1}%
2408 \TestPow{3}{0}{1}%
2409 \TestPow{-2}{1}{-2}%
2410 \TestPow{-1}{1}{-1}%
2411 \TestPow{1}{1}{1}%
2412 \TestPow{2}{1}{2}%
2413 \TestPow{3}{1}{3}%
2414 \TestPow{-2}{2}{4}%
2415 \TestPow{-1}{2}{1}%
2416 \TestPow{0}{2}{0}%
2417 \TestPow{1}{2}{1}%
2418 \TestPow{2}{2}{4}%
2419 \TestPow{3}{2}{9}%
2420 \TestPow{0}{1}{0}%
2421 \TestPow{1}{-2}{1}%
2422 \TestPow{1}{-1}{1}%
2423 \TestPow{-1}{-2}{1}%
2424 \TestPow{-1}{-1}{-1}%
2425 \TestPow{-1}{3}{-1}%
2426 \TestPow{-1}{4}{1}%
2427 \TestPow{-2}{-1}{0}%
2428 \TestPow{-2}{-2}{0}%
2429 \TestPow{2}{3}{8}%
2430 \TestPow{2}{4}{16}%
2431 \TestPow{2}{5}{32}%
2432 \TestPow{2}{6}{64}%
2433 \TestPow{2}{7}{128}%
2434 \TestPow{2}{8}{256}%
2435 \TestPow{2}{9}{512}%
2436 \TestPow{2}{10}{1024}%
2437 \TestPow{-2}{3}{-8}%
2438 \TestPow{-2}{4}{16}%
2439 \TestPow{-2}{5}{-32}%
2440 \TestPow{-2}{6}{64}%

```

```

2441 \TestPow{-2}{7}{-128}%
2442 \TestPow{-2}{8}{256}%
2443 \TestPow{-2}{9}{-512}%
2444 \TestPow{-2}{10}{1024}%
2445 \TestPow{3}{3}{27}%
2446 \TestPow{3}{4}{81}%
2447 \TestPow{3}{5}{243}%
2448 \TestPow{-3}{3}{-27}%
2449 \TestPow{-3}{4}{81}%
2450 \TestPow{-3}{5}{-243}%
2451 \TestPow{2}{30}{1073741824}%
2452 \TestPow{-3}{19}{-1162261467}%
2453 \TestPow{5}{13}{1220703125}%
2454 \TestPow{-7}{11}{-1977326743}%
2455 \end{qstest}
2456
2457 \begin{qstest}{div}{div}
2458 \TestDiv{1}{1}{1}%
2459 \TestDiv{2}{1}{2}%
2460 \TestDiv{-2}{1}{-2}%
2461 \TestDiv{2}{-1}{-2}%
2462 \TestDiv{-2}{-1}{2}%
2463 \TestDiv{15}{2}{7}%
2464 \TestDiv{-16}{2}{-8}%
2465 \TestDiv{1}{2}{0}%
2466 \TestDiv{1}{3}{0}%
2467 \TestDiv{2}{3}{0}%
2468 \TestDiv{-2}{3}{0}%
2469 \TestDiv{2}{-3}{0}%
2470 \TestDiv{-2}{-3}{0}%
2471 \TestDiv{13}{3}{4}%
2472 \TestDiv{-13}{-3}{4}%
2473 \TestDiv{-13}{3}{-4}%
2474 \TestDiv{-6}{5}{-1}%
2475 \TestDiv{-5}{5}{-1}%
2476 \TestDiv{-4}{5}{0}%
2477 \TestDiv{-3}{5}{0}%
2478 \TestDiv{-2}{5}{0}%
2479 \TestDiv{-1}{5}{0}%
2480 \TestDiv{0}{5}{0}%
2481 \TestDiv{1}{5}{0}%
2482 \TestDiv{2}{5}{0}%
2483 \TestDiv{3}{5}{0}%
2484 \TestDiv{4}{5}{0}%
2485 \TestDiv{5}{5}{1}%
2486 \TestDiv{6}{5}{1}%
2487 \TestDiv{-5}{4}{-1}%
2488 \TestDiv{-4}{4}{-1}%
2489 \TestDiv{-3}{4}{0}%
2490 \TestDiv{-2}{4}{0}%
2491 \TestDiv{-1}{4}{0}%
2492 \TestDiv{0}{4}{0}%
2493 \TestDiv{1}{4}{0}%
2494 \TestDiv{2}{4}{0}%
2495 \TestDiv{3}{4}{0}%
2496 \TestDiv{4}{4}{1}%
2497 \TestDiv{5}{4}{1}%
2498 \TestDiv{12345}{678}{18}%
2499 \TestDiv{32372}{5952}{5}%
2500 \TestDiv{284271294}{18162}{15651}%
2501 \TestDiv{217652429}{12561}{17327}%
2502 \TestDiv{462028434}{5439}{84947}%

```



```

2503 \TestDiv{2147483647}{1000}{2147483}%
2504 \TestDiv{2147483647}{-1000}{-2147483}%
2505 \TestDiv{-2147483647}{1000}{-2147483}%
2506 \TestDiv{-2147483647}{-1000}{2147483}%
2507 \TestDiv{0}{3}{0}%
2508 \TestDiv{1}{3}{0}%
2509 \TestDiv{2}{3}{0}%
2510 \TestDiv{3}{3}{1}%
2511 \TestDiv{4}{3}{1}%
2512 \TestDiv{5}{3}{1}%
2513 \TestDiv{6}{3}{2}%
2514 \TestDiv{7}{3}{2}%
2515 \TestDiv{8}{3}{2}%
2516 \TestDiv{9}{3}{3}%
2517 \TestDiv{10}{3}{3}%
2518 \TestDiv{11}{3}{3}%
2519 \TestDiv{12}{3}{4}%
2520 \TestDiv{13}{3}{4}%
2521 \TestDiv{14}{3}{4}%
2522 \TestDiv{15}{3}{5}%
2523 \TestDiv{16}{3}{5}%
2524 \TestDiv{17}{3}{5}%
2525 \TestDiv{18}{3}{6}%
2526 \TestDiv{19}{3}{6}%
2527 \TestDiv{20}{3}{6}%
2528 \TestDiv{21}{3}{7}%
2529 \TestDiv{22}{3}{7}%
2530 \TestDiv{23}{3}{7}%
2531 \TestDiv{24}{3}{8}%
2532 \TestDiv{25}{3}{8}%
2533 \TestDiv{26}{3}{8}%
2534 \TestDiv{27}{3}{9}%
2535 \TestDiv{28}{3}{9}%
2536 \TestDiv{29}{3}{9}%
2537 \TestDiv{30}{3}{10}%
2538 \TestDiv{31}{3}{10}%
2539 \TestDivBig{17363436332507}{24702}{702916214}%
2540 \end{qstest}
2541
2542 \begin{qstest}{mod}{mod}
2543 \TestMod{-6}{5}{4}%
2544 \TestMod{-5}{5}{0}%
2545 \TestMod{-4}{5}{1}%
2546 \TestMod{-3}{5}{2}%
2547 \TestMod{-2}{5}{3}%
2548 \TestMod{-1}{5}{4}%
2549 \TestMod{0}{5}{0}%
2550 \TestMod{1}{5}{1}%
2551 \TestMod{2}{5}{2}%
2552 \TestMod{3}{5}{3}%
2553 \TestMod{4}{5}{4}%
2554 \TestMod{5}{5}{0}%
2555 \TestMod{6}{5}{1}%
2556 \TestMod{-5}{4}{3}%
2557 \TestMod{-4}{4}{0}%
2558 \TestMod{-3}{4}{1}%
2559 \TestMod{-2}{4}{2}%
2560 \TestMod{-1}{4}{3}%
2561 \TestMod{0}{4}{0}%
2562 \TestMod{1}{4}{1}%
2563 \TestMod{2}{4}{2}%
2564 \TestMod{3}{4}{3}%

```

```

2565 \TestMod{4}{4}{0}%
2566 \TestMod{5}{4}{1}%
2567 \TestMod{-6}{-5}{-1}%
2568 \TestMod{-5}{-5}{0}%
2569 \TestMod{-4}{-5}{-4}%
2570 \TestMod{-3}{-5}{-3}%
2571 \TestMod{-2}{-5}{-2}%
2572 \TestMod{-1}{-5}{-1}%
2573 \TestMod{0}{-5}{0}%
2574 \TestMod{1}{-5}{-4}%
2575 \TestMod{2}{-5}{-3}%
2576 \TestMod{3}{-5}{-2}%
2577 \TestMod{4}{-5}{-1}%
2578 \TestMod{5}{-5}{0}%
2579 \TestMod{6}{-5}{-4}%
2580 \TestMod{-5}{-4}{-1}%
2581 \TestMod{-4}{-4}{0}%
2582 \TestMod{-3}{-4}{-3}%
2583 \TestMod{-2}{-4}{-2}%
2584 \TestMod{-1}{-4}{-1}%
2585 \TestMod{0}{-4}{0}%
2586 \TestMod{1}{-4}{-3}%
2587 \TestMod{2}{-4}{-2}%
2588 \TestMod{3}{-4}{-1}%
2589 \TestMod{4}{-4}{0}%
2590 \TestMod{5}{-4}{-3}%
2591 \TestMod{2147483647}{1000}{647}%
2592 \TestMod{2147483647}{-1000}{-353}%
2593 \TestMod{-2147483647}{1000}{353}%
2594 \TestMod{-2147483647}{-1000}{-647}%
2595 \TestMod{ 0 }{ 4 }{0}%
2596 \TestMod{ 1 }{ 4 }{1}%
2597 \TestMod{ -1 }{ 4 }{3}%
2598 \TestMod{ 0 }{ -4 }{0}%
2599 \TestMod{ 1 }{ -4 }{-3}%
2600 \TestMod{ -1 }{ -4 }{-1}%
2601 \TestMod{18362}{25}{12}%
2602 \end{qstest}
2603
2604 \newcommand*{\TestError}[2]{%
2605   \begingroup
2606     \expandafter\def\csname BigIntCalcError:#1\endcsname{%
2607       \Expect*{#2}{0}%
2608     \expandafter\def\csname BigIntCalcError:#1\endcsname{ERROR}%
2609     \Expect*{#2}{OERROR}%
2610   \endgroup
2611 }
2612 \begin{qstest}{error}{error}
2613   \TestError{FacNegative}{\bigintcalcFac{-1}}%
2614   \TestError{FacNegative}{\bigintcalcFac{-2147483647}}%
2615   \TestError{DivisionByZero}{\bigintcalcPow{0}{-1}}%
2616   \TestError{DivisionByZero}{\bigintcalcDiv{1}{0}}%
2617   \TestError{DivisionByZero}{\bigintcalcMod{1}{0}}%
2618 \end{qstest}
2619
2620 \begin{document}
2621 \end{document}
2622 </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.

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-T_EX:

```
tex bigintcalc.dtx
```

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

<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 T_EX distribution (teT_EX, miK_TE_X, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktextlsr`.

¹<http://ftp.ctan.org/tex-archive/>

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk bigintcalc.pdf unpack_files output .
```

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

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

L^AT_EX: Generate the documentation.

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

```
latex \let\install=y\input{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 L^AT_EX support.

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; numbers in roman refer to the code lines where the entry is used.

Symbols	
<code>\#</code>	1751 , 1813
<code>\%</code>	1816
<code>\@</code>	1752 , 1809
<code>\@firstofone</code>	1760 , 1763
<code>\@firstoftwo</code>	136 , 144 , 153
<code>\@gobble</code>	1757 , 1765
<code>\@ne</code>	2137 , 2152 , 2153 , 2163 , 2164 , 2189 , 2190
<code>\@nil</code> .	132 , 134 , 141 , 149 , 164 , 167 , 172
<code>\@secondoftwo</code>	138 , 146 , 156
<code>\@undefined</code>	52
<code>\[</code>	1814
<code>\]</code>	169 , 179 , 298 , 299 , 307 , 330 , 428 , 440 , 473 , 486 , 498 , 533 , 535 , 624 , 625 , 635 , 636 , 653 , 715 , 716 , 726 , 727 , 744 , 829 , 840 , 887 , 901 , 964 , 975 , 992 , 1112 ,

	1198, 1223, 1251, 1299, 1320, 1330, 1424, 1428, 1456, 1502, 1649, 1660, 1668, 1690, 1706, 1810	895, 965, 969, 976, 983, 993, 997, 1004, 1006, 1054, 1113, 1118, 1146, 1174, 1176, 1209, 1211, 1306, 1391, 1420, 1422, 1457, 1461, 1513, 1554, 1603, 1658, 1697, 1731, 1733, 1738, 1743
\{	1749, 1811	
\}	1750, 1812	
\]	1815	
_	1817	
A		
\advance	1790, 1798, 2065	
\aftergroup	26	
\AtEndDocument	2078	
B		
\begin	2085, 2103, 2122, 2145, 2156, 2167, 2195, 2207, 2225, 2242, 2271, 2298, 2306, 2335, 2362, 2376, 2402, 2457, 2542, 2612, 2620	
\BIC@@@Shl	847, 851	
\BIC@@@Shr	920, 921, 927, 928, 937	
\BIC@@@Dec	508, 509, 527, 532	
\BIC@@@Inc	450, 451, 467, 472	
\BIC@@@PowRec	1350, 1356, 1371	
\BIC@@@ProcessDiv	1555, 1593	
\BIC@@@Shl	831, 839, 854, 858	
\BIC@@@Shr	891, 896, 900, 938	
\BIC@@AddDigit	676, 677, 687, 766	
\BIC@@Cmp	268, 271, 357, 389	
\BIC@@Dec	488, 496, 545	
\BIC@@Expand	132, 134	
\BIC@@Inc	430, 438, 481	
\BIC@@MinMax	353, 356	
\BIC@@MinusOne	1313, 1318	
\BIC@@PowRec	1343, 1366, 1372	
\BIC@@ProcessDiv	1514, 1519	
\BIC@@ProcessFac	1177, 1183	
\BIC@@ProcessTim	984, 991	
\BIC@@Shl	813, 818, 825, 828	
\BIC@@Shr	871, 875, 881, 883, 1344, 1373, 1440, 1444, 1445	
\BIC@@Sqr	1131, 1133, 1136	
\BIC@@SubDigit	765, 776	
\BIC@@TestMode	106	
\BIC@@Tim	963	
\BIC@@Abs	235, 238	
\BIC@@Add	551, 554, 560	
\BIC@@AddCarry0	694	
\BIC@@AddCarry10	695	
\BIC@@AddCarry[1-9]	696	
\BIC@@AddDigit	658, 663, 674	
\BIC@@AddResult	657, 667	
\BIC@@AddSwitch	556, 563	
\BIC@@AddXY	569, 573, 607, 611, 618, 623, 814, 819, 826, 1020, 1478	
\BIC@AfterFi	114, 171, 208, 321, 358, 360, 390, 392, 396, 409, 411, 415, 429, 433, 480, 487, 491, 544, 654, 661, 689, 691, 745, 750, 778, 783, 811, 817, 830, 834, 841, 846, 853, 857, 870, 874, 885,	
\BIC@AfterFiFi	115, 180, 184, 190, 211, 215, 274, 278, 284, 288, 300, 302, 308, 312, 326, 375, 377, 441, 443, 449, 466, 475, 477, 499, 501, 507, 526, 541, 626, 630, 646, 717, 721, 737, 888, 890, 903, 910, 919, 926, 1012, 1017, 1057, 1082, 1166, 1214, 1218, 1236, 1238, 1247, 1267, 1269, 1290, 1301, 1303, 1332, 1336, 1342, 1394, 1396, 1400, 1406, 1408, 1412, 1437, 1439, 1443, 1448, 1492, 1494, 1500, 1522, 1524, 1546, 1569, 1571, 1577, 1579, 1585, 1587, 1596, 1598, 1619, 1626, 1661, 1663, 1677, 1679, 1683, 1700, 1702, 1712, 1719	
\BIC@AfterFiFiFi	116, 195, 199, 331, 335, 536, 538, 568, 572, 578, 581, 585, 593, 595, 600, 606, 610, 637, 641, 728, 732, 1060, 1064, 1071, 1075, 1086, 1090, 1096, 1100, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1241, 1243, 1272, 1274, 1279, 1284, 1349, 1355, 1503, 1507, 1531, 1533, 1539, 1541, 1612, 1614	
\BIC@AtEnd	80, 81, 1746	
\BIC@Cmp	264, 267	
\BIC@CmpDiff	300, 319	
\BIC@CmpLength	289, 295, 297	
\BIC@CmpResult	303, 309, 318	
\BIC@Dec	398, 412, 425, 485, 1185	
\BIC@DecSwitch	404, 407	
\BIC@Div	1378, 1381	
\BIC@DivCleanup	1492, 1500, 1518, 1522, 1531, 1539, 1569, 1577, 1585, 1596, 1612	
\BIC@DivStart	1482, 1486	
\BIC@DivStartX	1449, 1455, 1462	
\BIC@DivStartYii	1458, 1466	
\BIC@DivStartYiv	1467, 1471	
\BIC@DivStartYvi	1472, 1476	
\BIC@DivStartYviii	1477, 1481	
\BIC@DivSub	1542, 1559, 1588	
\BIC@DivSwitch	1397, 1401, 1409, 1413, 1418	
\BIC@DivSwitchSign	1382, 1387, 1389	
\BIC@DoAdd	627, 631, 652	
\BIC@DoSub	718, 722, 743	
\BIC@Expand	130, 160, 227	
\BIC@Fac	1141, 1144	
\BIC@Fi	113, 114, 115, 116, 174, 204, 219, 292, 316, 340, 361,	

381, 400, 419, 436, 470, 483,
 494, 530, 547, 615, 650, 665,
 692, 741, 754, 784, 821, 837,
 849, 860, 877, 898, 933, 972,
 989, 1000, 1025, 1105, 1122,
 1170, 1181, 1296, 1309, 1364,
 1416, 1453, 1464, 1516, 1557,
 1591, 1633, 1687, 1727, 1734, 1744
 \BIC@Inc 393, 417, 422, 427
 \BIC@IncSwitch 385, 388
 \BIC@MinMax 344, 349, 352
 \BIC@MinusOne 1229, 1258, 1311
 \BIC@Mod 1637, 1640
 \BIC@ModMinus 1709, 1729
 \BIC@ModSwitch . 1664, 1680, 1684, 1689
 \BIC@ModSwitchSign .. 1641, 1646, 1648
 \BIC@ModTwo
 1240, 1271, 1278, 1298, 1341, 1699
 \BIC@ModX 1713, 1736
 \BIC@Mul 1045, 1048
 \BIC@MulDigit[3-9] 1027
 \BIC@MulSwitch 1049, 1052
 \BIC@Normalize 177, 224
 \BIC@NormalizeDigits .. 200, 216, 221
 \BIC@NormalizeZero 196, 206
 \BIC@Odd 365, 370, 372
 \BIC@PosCmp 294, 567, 577, 592,
 605, 1059, 1070, 1085, 1095,
 1148, 1173, 1331, 1348, 1419,
 1490, 1520, 1529, 1567, 1594, 1610
 \BIC@Pow 1190, 1193
 \BIC@PowRec 1280, 1285, 1291, 1329, 1367
 \BIC@PowSwitch 1194, 1197
 \BIC@ProcessDiv
 1455, 1487, 1489, 1525,
 1534, 1560, 1572, 1580, 1599, 1615
 \BIC@ProcessDivII
 1547, 1566, 1604, 1620, 1627
 \BIC@ProcessDivIV 1593
 \BIC@ProcessFac 1167, 1172, 1184
 \BIC@ProcessMul . 1061, 1065, 1072,
 1076, 1087, 1091, 1097, 1101,
 1109, 1111, 1137, 1178, 1215,
 1219, 1333, 1337, 1351, 1357, 1368
 \BIC@ProcessTim ... 966, 974, 994, 998
 \BIC@Sgn 248, 251, 408,
 1053, 1056, 1081, 1390, 1393, 1405
 \BIC@Shl ... 806, 809, 1468, 1473, 1483
 \BIC@Shr 864, 867
 \BIC@ShrDigit[00-19] 940
 \BIC@ShrResult 904, 905, 911, 912, 935
 \BIC@Space 117,
 161, 230, 240, 242, 655, 669,
 671, 760, 779, 842, 977, 1007,
 1013, 1018, 1114, 1708, 1720, 1739
 \BIC@Sqr 1126, 1129
 \BIC@StripHexSpace 164, 167
 \BIC@SubCarry0 786
 \BIC@SubCarry10 787
 \BIC@SubCarry[1-9] 788
 \BIC@SubDigit 747, 752, 763
 \BIC@SubResult 746, 756
 \BIC@SubXY
 . 582, 586, 596, 601, 621, 714,
 1548, 1562, 1605, 1621, 1628, 1740
 \BIC@Temp 696, 704,
 705, 706, 707, 708, 709, 710,
 711, 712, 713, 788, 795, 796,
 797, 798, 799, 800, 801, 802,
 803, 940, 943, 944, 945, 946,
 947, 948, 949, 950, 951, 952,
 953, 954, 955, 956, 957, 958,
 959, 960, 961, 962, 1027, 1036,
 1037, 1038, 1039, 1040, 1041, 1042
 \BIC@TestMode 106, 1833
 \BIC@Tim 963, 1115, 1120
 \BIC@TimDigit 979, 986, 1002
 \bigintcalcAbs .. 4, 233, 366, 1883,
 1902, 1973, 1974, 1982, 1983,
 1992, 1993, 2019, 2020, 2040, 2041
 \BigIntCalcAdd 7, 617, 1936, 1944
 \bigintcalcAdd 5, 549, 1115, 1120, 1930
 \bigintcalcCmp 4, 262, 1895, 1933, 1957
 \BigIntCalcDec 7, 424, 1923
 \bigintcalcDec 5, 402, 1919
 \BigIntCalcDiv 7, 1385, 2019
 \bigintcalcDiv 6, 1376,
 1715, 1722, 1871, 2012, 2016, 2616
 \BigIntCalcError 461,
 511, 521, 700, 791, 1032, 1146,
 1163, 1233, 1236, 1244, 1248,
 1275, 1287, 1293, 1295, 1361,
 1363, 1391, 1451, 1658, 1703, 1726
 \bigintcalcFac
 .. 6, 1139, 2003, 2006, 2613, 2614
 \BigIntCalcInc 7, 421, 1912
 \bigintcalcInc 5, 383, 1908
 \bigintcalcInv 3, 229, 1880
 \bigintcalcMax 4, 347, 1892
 \bigintcalcMin 4, 342, 1889
 \BigIntCalcMod 7, 1644, 2039
 \bigintcalcMod ... 6, 1635, 2024, 2617
 \BigIntCalcMul 7, 1107, 1991
 \bigintcalcMul 6, 1043, 1715, 1722, 1988
 \bigintcalcNum .. 3, 222, 231, 236,
 249, 265, 269, 345, 350, 354,
 386, 405, 552, 556, 561, 807,
 865, 1046, 1050, 1127, 1142,
 1191, 1195, 1379, 1383, 1638,
 1642, 1912, 1923, 1937, 1945, 1961
 \BigIntCalcOdd 7, 368, 1901
 \bigintcalcOdd 5, 363, 1898
 \bigintcalcPow ... 6, 1188, 2009, 2615
 \bigintcalcSgn
 4, 246, 1886, 1909, 1920, 1931,
 1932, 1955, 1956, 2025, 2027, 2032
 \BigIntCalcShl 7, 823, 1973
 \bigintcalcShl 5, 804, 1970
 \BigIntCalcShr 7, 879, 1982
 \bigintcalcShr 5, 862, 1979
 \bigintcalcSqr 6, 1124, 1998
 \BigIntCalcSub 7, 620, 1960
 \bigintcalcSub 5, 558, 1714, 1721, 1954
 \body 1769, 1773

C		
<code>\catcode</code>	3, 4, 5, 6, 7, 8, 9, 17, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 64, 65, 68, 69, 70, 71, 75, 76, 77, 78, 82, 84, 104, 109, 111, 1749, 1750, 1751, 1752, 1787, 1796, 1809, 1810, 1811, 1812, 1813, 1814, 1815, 1816, 1817, 1818	
<code>\chardef</code>	1833	
<code>\count@</code>	1754, 1783, 1787, 1789, 1790, 1794, 1796, 1797, 1798	
<code>\countdef</code>	1754	
<code>\csname</code>	10, 18, 44, 60, 67, 102, 108, 123, 129, 153, 156, 678, 694, 695, 697, 767, 786, 787, 789, 906, 913, 922, 929, 941, 1008, 1014, 1021, 1028, 1753, 1756, 1759, 1762, 1801, 1823, 2047, 2606, 2608	
D		
<code>\dimexpr</code>	2055	
<code>\divide</code>	1870	
<code>\documentclass</code>	1829	
E		
<code>\empty</code>	13, 14	
<code>\end</code>	1824, 2101, 2120, 2143, 2154, 2165, 2193, 2205, 2223, 2240, 2269, 2296, 2304, 2333, 2360, 2374, 2400, 2455, 2540, 2602, 2618, 2621	
<code>\endcsname</code>	10, 18, 44, 60, 67, 102, 108, 123, 129, 153, 156, 685, 694, 695, 697, 774, 786, 787, 789, 906, 913, 922, 929, 941, 1008, 1014, 1021, 1028, 1753, 1756, 1759, 1762, 1801, 1823, 2047, 2606, 2608	
<code>\endinput</code>	26	
<code>\endqstest</code>	2069, 2074	
<code>\Expect</code>	1846, 1853, 1862, 1871, 2607, 2609	
I		
<code>\ifcase</code>	389, 408, 439, 452, 497, 510, 534, 577, 592, 680, 683, 698, 769, 772, 790, 884, 1003, 1011, 1029, 1053, 1056, 1081, 1149, 1198, 1223, 1229, 1240, 1251, 1258, 1271, 1278, 1330, 1341, 1390, 1393, 1405, 1419, 1428, 1490, 1520, 1529, 1567, 1594, 1610, 1649, 1660, 1668, 1690, 1699, 2025, 2027, 2032	
<code>\ifcat</code>	135	
<code>\ifnum</code>	320, 325, 357, 448, 474, 506, 567, 605, 688, 777, 852, 1059, 1070, 1085, 1095, 1148, 1173, 1331, 1348, 1789, 1797, 1909, 1920, 1931, 1932, 1933, 1955, 1956, 1957	
<code>\ifodd</code>	374, 902, 918, 1300	
<code>\iftrue</code>	2141	
<code>\ifx</code>	11, 14, 18, 44, 52, 55, 102, 108, 123, 129, 143, 153, 156, 169, 178, 179, 189, 194, 207, 210, 239, 252, 255, 272, 273, 283, 298, 299, 307, 330, 373, 428, 440, 473, 486, 498, 533, 535, 564, 565, 591, 624, 625, 635, 636, 653, 668, 715, 716, 726, 727, 744, 757, 810, 829, 840, 868, 887, 901, 964, 975, 992, 1112, 1130, 1145, 1198, 1199, 1200, 1201, 1205, 1213, 1223, 1224, 1225, 1251, 1252, 1253, 1257, 1299, 1312, 1319, 1320, 1330, 1424, 1428, 1429, 1430, 1431, 1456, 1491, 1499, 1502, 1521, 1530, 1538, 1568, 1576, 1584, 1595, 1611, 1649, 1650, 1654, 1660, 1668, 1669, 1673, 1690, 1691, 1692, 1706, 1711, 1730, 1737, 1753, 1756, 1759, 1762, 1801, 2047, 2099, 2100, 2119	
<code>\immediate</code>	20, 46	
<code>\IncludeTests</code>	1838	
<code>\input</code>	124, 1802	
<code>\iterate</code>	1770, 1772, 1774	
L		
<code>\LoadCommand</code>	1802, 1819	
<code>\LogTests</code>	1839	
<code>\loop</code>	1768, 1784, 1795	
M		
<code>\m@ne</code>	2138, 2153, 2164, 2190	
<code>\makeatletter</code>	1832, 2049, 2083	
<code>\makeatother</code>	1834, 2081	
N		
<code>\NeedsTeXFormat</code>	1827	
<code>\newcommand</code>	1840, 1848, 1855, 1866, 1867, 1868, 1873, 1878, 1879, 1882, 1885, 1888, 1891, 1894, 1897, 1907, 1918, 1929, 1953, 1969, 1978, 1987, 1997, 2000, 2005, 2008, 2011, 2015, 2023, 2053, 2058, 2062, 2063, 2604	
<code>\newcount</code>	1865, 2050, 2051	
<code>\next</code>	1774, 1776, 1778	
<code>\nofiles</code>	1828	
<code>\number</code>	247, 263, 677, 766, 1008, 1014, 1021, 2055, 2098, 2136, 2137, 2138, 2140, 2152, 2153, 2163, 2164, 2189, 2190	
<code>\numexpr</code>	450, 508, 656, 676, 765, 780, 843, 847, 904, 911, 920, 927, 978, 985, 1830, 1831, 1836, 1841, 1842, 1845, 1849, 1850, 1852, 1858, 1866, 2002	
P		
<code>\PackageInfo</code>	23	
<code>\pdf@escapehex</code>	164	
<code>\pdf@unescapehex</code>	162	

\pdfelapsedtime	2064	2177, 2178, 2179, 2180, 2181,
\pdfresettimer	2060	2182, 2183, 2184, 2185, 2186,
\PrintTime	2053, 2066, 2079	2187, 2188, 2189, 2190, 2191, 2192
\ProvidesPackage	15, 61	\TestCount
		1865, 1869, 1870, 1871
		\TestDec ..
		1918, 2226, 2227, 2228,
		2229, 2230, 2231, 2232, 2233,
		2234, 2235, 2236, 2237, 2238, 2239
		\TestDiv
		2011,
		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, 2499, 2500, 2501, 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
		\TestDivBig
		2015, 2539
		\TestError
		2604, 2613, 2614, 2615, 2616, 2617
		\TestExch
		1878, 2001
		\TestFac
		2000, 2377, 2378,
		2379, 2380, 2381, 2382, 2383,
		2384, 2385, 2386, 2387, 2388, 2389
		\TestFacBig
		2005, 2390, 2391, 2392, 2393,
		2394, 2395, 2396, 2397, 2398, 2399
		\TestInc
		1907, 2208, 2209, 2210, 2211,
		2212, 2213, 2214, 2215, 2216,
		2217, 2218, 2219, 2220, 2221, 2222
		\TestInv
		1879, 2086, 2087, 2088, 2089,
		2090, 2091, 2092, 2093, 2094,
		2095, 2096, 2097, 2098, 2099, 2100
		\TestMax
		1891, 2157, 2158,
		2159, 2160, 2161, 2162, 2163, 2164
		\TestMin
		1888, 2146, 2147,
		2148, 2149, 2150, 2151, 2152, 2153
		\TestMod ..
		2023, 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, 2583, 2584, 2585,
		2586, 2587, 2588, 2589, 2590,
		2591, 2592, 2593, 2594, 2595,
		2596, 2597, 2598, 2599, 2600, 2601
		\TestMul ..
		1987, 2336, 2337, 2338,
		2339, 2340, 2341, 2342, 2343,
		2344, 2345, 2346, 2347, 2348,
		2349, 2350, 2351, 2352, 2353,
		2354, 2355, 2356, 2357, 2358, 2359
		\TestOdd
		1897, 2197, 2198,

Q	
\qstest	2068, 2070

R	
\RangeCatcodeInvalid	1793, 1805, 1806, 1807, 1808
\renewcommand	2059
\repeat	1768, 1780, 1791, 1799
\RequirePackage	126
\RestoreCatcodes	1782, 1785, 1786, 1820
\result	1851, 1853
\resultA	1843, 1846
\resultB	1844, 1846
\romannumeral	131, 161, 223, 230, 234,
	343, 348, 364, 369, 384, 397,
	403, 416, 422, 425, 550, 559,
	566, 580, 599, 618, 621, 675,
	764, 805, 812, 824, 863, 869,
	880, 979, 986, 1019, 1044, 1069,
	1084, 1108, 1125, 1140, 1178,
	1185, 1189, 1283, 1343, 1350,
	1356, 1367, 1372, 1377, 1386,
	1426, 1444, 1467, 1472, 1477,
	1482, 1547, 1561, 1604, 1620,
	1627, 1636, 1645, 1708, 1709, 1739

S	
\saved@endqstest	2069, 2076
\saved@qstest	2068, 2071
\SavedNumexpr	1830, 1836, 1841, 1845, 1849, 1852
\space	1846, 2055
\StartTime	2058, 2072
\StopTime	2063, 2075
\strip@pt	2055
\SummaryTime ...	2050, 2052, 2065, 2079

T	
\Test	1804, 1822, 1873, 1878,
	1880, 1883, 1886, 1889, 1892,
	1895, 1898, 1900, 1908, 1911,
	1919, 1922, 1930, 1935, 1943,
	1954, 1959, 1970, 1972, 1979,
	1981, 1988, 1990, 1998, 2006,
	2009, 2012, 2016, 2018, 2024, 2038
\TestAbs	1882,
	2104, 2105, 2106, 2107, 2108,
	2109, 2110, 2111, 2112, 2113,
	2114, 2115, 2116, 2117, 2118, 2119
\TestAdd	1929,
	2243, 2244, 2245, 2246, 2247,
	2248, 2249, 2250, 2251, 2252,
	2253, 2254, 2255, 2256, 2257,
	2258, 2259, 2260, 2261, 2262,
	2263, 2264, 2265, 2266, 2267, 2268
\TestArg	1866, 1867, 1869, 1870
\TestCmp	1894, 2168, 2169, 2170, 2171,
	2172, 2173, 2174, 2175, 2176,

