

The `bigintcalc` package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

2011/01/30 v1.2

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	10
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	13
2.9	Cmp, Min, Max	13
2.10	Odd	15
2.11	Inc, Dec	16
2.12	Add, Sub	19
2.13	Shl, Shr	24
2.14	\BIC@Tim	27
2.15	Mul	29
2.16	Sqr	31
2.17	Fac	31
2.18	Pow	32
2.18.1	Help macros	35
2.18.2	Recursive calculation	35

2.19	Div	37
2.20	Mod	42
3	Test	45
3.1	Catcode checks for loading	45
3.2	Macro tests	46
3.2.1	Preamble with test macro definitions	46
3.2.2	Time	50
3.2.3	Test sets	51
4	Installation	59
4.1	Download	59
4.2	Bundle installation	60
4.3	Package installation	60
4.4	Refresh file name databases	60
4.5	Some details for the interested	60
5	History	61
	[2007/09/27 v1.0]	61
	[2007/11/11 v1.1]	61
	[2011/01/30 v1.2]	61
6	Index	61

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, Mod

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

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^^M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51 \def\x#1#2#3[#4]{\endgroup
52 \immediate\write-1{Package: #3 #4}%
53 \xdef#1{#4}%
54 }%
55 \else
56 \def\x#1#2[#3]{\endgroup
```



```

57     #2[#{#3}]%
58     \ifx#1\@undefined
59         \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62         \xdef#1{#3}%
63     \fi
64 }%
65 \fi
66 \expandafter\x\csname ver@bigintcalc.sty\endcsname
67 \ProvidesPackage{bigintcalc}%
68 [2011/01/30 v1.2 Expandable big integer calculations (H0)]%

```

2.2 Catcodes

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^~M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76     \expandafter\edef\csname BIC@AtEnd\endcsname{%
77         \endlinechar=\the\endlinechar\relax
78         \catcode13=\the\catcode13\relax
79         \catcode32=\the\catcode32\relax
80         \catcode35=\the\catcode35\relax
81         \catcode61=\the\catcode61\relax
82         \catcode64=\the\catcode64\relax
83         \catcode123=\the\catcode123\relax
84         \catcode125=\the\catcode125\relax
85     }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^~M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95     \edef\BIC@AtEnd{%
96         \BIC@AtEnd
97         \catcode#1=\the\catcode#1\relax
98     }%
99     \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{33}{12}% !
102 \TMP@EnsureCode{36}{14}% $ (comment!)
103 \TMP@EnsureCode{38}{14}% & (comment!)
104 \TMP@EnsureCode{40}{12}% (
105 \TMP@EnsureCode{41}{12}% )
106 \TMP@EnsureCode{42}{12}% *
107 \TMP@EnsureCode{43}{12}% +
108 \TMP@EnsureCode{45}{12}% -
109 \TMP@EnsureCode{46}{12}% .
110 \TMP@EnsureCode{47}{12}% /
111 \TMP@EnsureCode{58}{11}% : (letter!)
112 \TMP@EnsureCode{60}{12}% <
113 \TMP@EnsureCode{62}{12}% >
114 \TMP@EnsureCode{63}{14}% ? (comment!)
115 \TMP@EnsureCode{91}{12}% [

```

```

116 \TMP@EnsureCode{93}{12}% ]
117 \edef\BIC@AtEnd{\BIC@AtEnd\noexpand\endinput}
118 \begingroup\expandafter\expandafter\expandafter\endgroup
119 \expandafter\ifx\csname BIC@TestMode\endcsname\relax
120 \else
121 \catcode63=9 % ? (ignore)
122 \fi
123 ? \let\BIC@@TestMode\BIC@TestMode

```

2.3 ϵ -TeX detection

```

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

```

2.4 Help macros

```

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

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

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

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

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

```

2.5 Expand number

```

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

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

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

```

`\BIC@@Expand`

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

`\@firstoftwo`

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

`\@secondoftwo`

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

`\BIC@Expand`

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

`\BIC@StripHexSpace`

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

2.6 Normalize expanded number

`\BIC@Normalize`

`#1`: result sign

`#2`: first token of number

```
199 \def\BIC@Normalize#1#2{%
200   \ifx#2-%
```

```

201 \ifx\#1\%
202 \BIC@AfterFiFi{%
203 \BIC@Normalize-%
204 }%
205 \else
206 \BIC@AfterFiFi{%
207 \BIC@Normalize{%}
208 }%
209 \fi
210 \else
211 \ifx#2+%
212 \BIC@AfterFiFi{%
213 \BIC@Normalize{#1}%
214 }%
215 \else
216 \ifx#20%
217 \BIC@AfterFiFiFi{%
218 \BIC@NormalizeZero{#1}%
219 }%
220 \else
221 \BIC@AfterFiFiFi{%
222 \BIC@NormalizeDigits#1#2%
223 }%
224 \fi
225 \fi
226 \BIC@Fi
227 }

```

\BIC@NormalizeZero

```

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

```

\BIC@NormalizeDigits

```

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

```

2.7 Num

\bigintcalcNum

```

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

```

2.8 Inv, Abs, Sgn

`\bigintcalcInv`

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

`\bigintcalcAbs`

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

`\BIC@Abs`

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

`\bigintcalcSgn`

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

`\BIC@Sgn`

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

2.9 Cmp, Min, Max

`\bigintcalcCmp`

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

`\BIC@Cmp`

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

\BIC@@Cmp

```
293 \def\BIC@@Cmp#1#2!#3#4!{%
294   \ifx#1-%
295     \ifx#3-%
296       \BIC@AfterFiFi{%
297         \BIC@@Cmp#4!#2!%
298       }%
299     \else
300       \BIC@AfterFiFi{%
301         -1 %
302       }%
303     \fi
304   \else
305     \ifx#3-%
306       \BIC@AfterFiFi{%
307         1 %
308       }%
309     \else
310       \BIC@AfterFiFi{%
311         \BIC@CmpLength#1#2!#3#4!#1#2!#3#4!%
312       }%
313     \fi
314   \BIC@Fi
315 }
```

\BIC@PosCmp

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

\BIC@CmpLength

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

\BIC@CmpResult

```
340 \def\BIC@CmpResult#1#2!#3!#1 }
```

\BIC@CmpDiff

```
341 \def\BIC@CmpDiff#1#2!#3#4!{%
342   \ifnum#1<#3 %
343     \BIC@AfterFiFi{%
344       -1 %
```

```

345   }%
346  \else
347    \ifnum#1>#3 %
348      \BIC@AfterFiFi{%
349        1 %
350      }%
351  \else
352    \ifx\#2\%
353      \BIC@AfterFiFiFi{%
354        0 %
355      }%
356  \else
357    \BIC@AfterFiFiFiFi{%
358      \BIC@CmpDiff#2!#4!%
359    }%
360  \fi
361  \fi
362  \BIC@Fi
363 }

```

\bigintcalcMin

```

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

```

\bigintcalcMax

```

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

```

\BIC@MinMax

```

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

```

\BIC@@MinMax

```

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

```

2.10 Odd

\bigintcalcOdd

```

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

```

\BigIntCalcOdd

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

\BIC@Odd #1: x

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

2.11 Inc, Dec

\bigintcalcInc

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

\BIC@IncSwitch

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

\bigintcalcDec

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

\BIC@DecSwitch

```
429 \def\BIC@DecSwitch#1#2!{%
430   \ifcase\BIC@Sgn#1#2! %
431     \BIC@AfterFi{ -1}%
432   \or
433     \BIC@AfterFi{%
434       \BIC@Dec#1#2!}%
435   }%
436   \else
437     \BIC@AfterFi{%
```



```

438     \expandafter-\romannumeral0%
439     \BIC@Inc#2!{%}
440   }%
441   \BIC@Fi
442 }

\BigIntCalcInc

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

\BigIntCalcDec

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

\BIC@Inc

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

\BIC@@Inc

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

```

```

492 \BIC@Fi
493 }

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

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

\BIC@@@Dec
518 \def\BIC@@@Dec#1#2#3!#4{%
519 \ifcase#1 %
520 \ifx\#3\%
521 \BIC@AfterFiFi{ #2#4}%
522 \else
523 \BIC@AfterFiFi{%
524 \BIC@@@Dec#3!{#2#4}%
525 }%
526 \fi
527 \else
528 \ifnum#2>0 %
529 \BIC@AfterFiFi{%
530 & \expandafter\BIC@@@Dec\the\numexpr#2-1\relax
531 $ \expandafter\expandafter\expandafter\BIC@@@Dec
532 $ \ifcase#2
533 $? \BigIntCalcError:ThisCannotHappen%
534 $ \or\expandafter0%
535 $ \or\expandafter1%
536 $ \or\expandafter2%
537 $ \or\expandafter3%
538 $ \or\expandafter4%
539 $ \or\expandafter5%
540 $ \or\expandafter6%
541 $ \or\expandafter7%
542 $ \or\expandafter8%
543 $? \else\BigIntCalcError:ThisCannotHappen%
544 $ \fi
545 0#3!{#4}%
546 }%
547 \else

```

```

548     \BIC@AfterFiFi{%
549     \BIC@@@Dec91#3!{#4}%
550     }%
551     \fi
552     \BIC@Fi
553 }

```

\BIC@@@Dec

```

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

```

2.12 Add, Sub

\bigintcalcAdd

```

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

```

\BIC@Add

```

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

```

\bigintcalcSub

```

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

```

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

$x < 0$	$y < 0$	$-x > -y$	-	Add($-x, -y$)
		else		Add($-y, -x$)
	else	$-x > y$	-	Sub($-x, y$)
		$-x = y$		0
else	$y < 0$	$x > -y$	+	Sub($x, -y$)
		$x = -y$		0
		else	-	Sub($-y, x$)
	else	$x > y$	+	Add(x, y)
		else		Add(y, x)

```

585 \def\BIC@AddSwitch#1#2!#3#4!{%
586 \ifx#1-% x < 0
587 \ifx#3-% y < 0
588 \expandafter-\romannumeral0%
589 \ifnum\BIC@PosCmp#2!#4!=1 % -x > -y
590 \BIC@AfterFiFiFi{%
591 \BIC@AddXY#2!#4!!!%
592 }%
593 \else % -x <= -y
594 \BIC@AfterFiFiFi{%
595 \BIC@AddXY#4!#2!!!%
596 }%
597 \fi
598 \else % y >= 0
599 \ifcase\BIC@PosCmp#2!#3#4!% -x = y
600 \BIC@AfterFiFiFi{ 0}%
601 \or % -x > y
602 \expandafter-\romannumeral0%
603 \BIC@AfterFiFiFi{%
604 \BIC@SubXY#2!#3#4!!!%
605 }%
606 \else % -x <= y
607 \BIC@AfterFiFiFi{%
608 \BIC@SubXY#3#4!#2!!!%
609 }%
610 \fi
611 \fi
612 \else % x >= 0
613 \ifx#3-% y < 0
614 \ifcase\BIC@PosCmp#1#2!#4!% x = -y
615 \BIC@AfterFiFiFi{ 0}%
616 \or % x > -y
617 \BIC@AfterFiFiFi{%
618 \BIC@SubXY#1#2!#4!!!%
619 }%
620 \else % x <= -y
621 \expandafter-\romannumeral0%
622 \BIC@AfterFiFiFi{%
623 \BIC@SubXY#4!#1#2!!!%
624 }%
625 \fi
626 \else % y >= 0
627 \ifnum\BIC@PosCmp#1#2!#3#4!=1 % x > y
628 \BIC@AfterFiFiFi{%
629 \BIC@AddXY#1#2!#3#4!!!%
630 }%
631 \else % x <= y
632 \BIC@AfterFiFiFi{%
633 \BIC@AddXY#3#4!#1#2!!!%
634 }%
635 \fi
636 \fi
637 \BIC@Fi
638 }

```

\BigIntCalcAdd

```

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

```

\BigIntCalcSub

```

642 \def\BigIntCalcSub#1!#2!{%

```

```

643 \romannumeral0\BIC@SubXY#1!#2!!!%
644 }

```

\BIC@AddXY

```

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

```

\BIC@DoAdd #1: carry

#2: reverted result

#3#4: reverted x

#5#6: reverted y

```

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

```

\BIC@AddResult

```

689 $ \def\BIC@AddResult#1{%
690 $ \ifx#10%
691 $ \expandafter\BIC@Space
692 $ \else
693 $ \expandafter\BIC@Space\expandafter#1%
694 $ \fi
695 $ }%

```

```

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

```

\BIC@@AddDigit

```

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

```

\BIC@AddCarry0

```

716 $ \expandafter\def\csname BIC@AddCarry0\endcsname#1{#1}%

```

\BIC@AddCarry10

```

717 $ \expandafter\def\csname BIC@AddCarry10\endcsname#1{1#1}%

```

\BIC@AddCarry[1-9]

```

718 $ \def\BIC@Temp#1#2{%
719 $   \expandafter\def\csname BIC@AddCarry#1\endcsname##1{%
720 $     \ifcase##1 #1\or
721 $       #2%
722 $?   \else\BigIntCalcError:ThisCannotHappen%
723 $     \fi
724 $   }%
725 $ }%
726 $ \BIC@Temp 0{1\or2\or3\or4\or5\or6\or7\or8\or9}%
727 $ \BIC@Temp 1{2\or3\or4\or5\or6\or7\or8\or9\or10}%
728 $ \BIC@Temp 2{3\or4\or5\or6\or7\or8\or9\or10\or11}%
729 $ \BIC@Temp 3{4\or5\or6\or7\or8\or9\or10\or11\or12}%
730 $ \BIC@Temp 4{5\or6\or7\or8\or9\or10\or11\or12\or13}%
731 $ \BIC@Temp 5{6\or7\or8\or9\or10\or11\or12\or13\or14}%
732 $ \BIC@Temp 6{7\or8\or9\or10\or11\or12\or13\or14\or15}%
733 $ \BIC@Temp 7{8\or9\or10\or11\or12\or13\or14\or15\or16}%
734 $ \BIC@Temp 8{9\or10\or11\or12\or13\or14\or15\or16\or17}%
735 $ \BIC@Temp 9{10\or11\or12\or13\or14\or15\or16\or17\or18}%

```

\BIC@SubXY Preconditions:

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

```

736 \def\BIC@SubXY#1#2!#3#4!#5!#6!{%
737   \ifx\#2\%
738     \ifx\#3\%
739       \BIC@AfterFiFi{%

```

```

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

\BIC@DoSub #1: carry
#2: reverted result
#3#4: reverted x
#5#6: reverted y
765 \def\BIC@DoSub#1#2!#3#4!#5#6!{%
766 \ifx\#4\%
767 \BIC@AfterFi{%
768 \expandafter\expandafter\expandafter\BIC@SubResult
769 \BIC@SubDigit#1#3#5#2%
770 }%
771 \else
772 \BIC@AfterFi{%
773 \expandafter\expandafter\expandafter\BIC@DoSub
774 \BIC@SubDigit#1#3#5#2!#4!#6!%
775 }%
776 \BIC@Fi
777 }

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

\BIC@SubDigit #1: carry
#2: digit of  $x$ 
#3: digit of  $y$ 
785 \def\BIC@SubDigit#1#2#3{%
786 \romannumeral0%
787 & \expandafter\BIC@SubDigit\the\numexpr#2-#3-#1!%
788 $ \expandafter\BIC@AddDigit\number
789 $ \csname
790 $ BIC@SubCarry%

```

```

791 $ \ifcase#1 %
792 $ #3%
793 $ \else
794 $ \ifcase#3 1\or2\or3\or4\or5\or6\or7\or8\or9\or10\fi
795 $ \fi
796 $ \endcsname#2!%
797 }

```

\BIC@SubDigit

```

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

```

\BIC@SubCarry0

```
808 $ \expandafter\def\csname BIC@SubCarry0\endcsname#1{#1}%
```

\BIC@SubCarry10

```
809 $ \expandafter\def\csname BIC@SubCarry10\endcsname#1{1#1}%
```

\BIC@SubCarry[1-9]

```

810 $ \def\BIC@Temp#1#2{%
811 $ \expandafter\def\csname BIC@SubCarry#1\endcsname##1{%
812 $ \ifcase##1 #2%
813 $? \else\BigIntCalcError:ThisCannotHappen%
814 $ \fi
815 $ }%
816 $ }%
817 $ \BIC@Temp 1{19\or0\or1\or2\or3\or4\or5\or6\or7\or8}%
818 $ \BIC@Temp 2{18\or19\or0\or1\or2\or3\or4\or5\or6\or7}%
819 $ \BIC@Temp 3{17\or18\or19\or0\or1\or2\or3\or4\or5\or6}%
820 $ \BIC@Temp 4{16\or17\or18\or19\or0\or1\or2\or3\or4\or5}%
821 $ \BIC@Temp 5{15\or16\or17\or18\or19\or0\or1\or2\or3\or4}%
822 $ \BIC@Temp 6{14\or15\or16\or17\or18\or19\or0\or1\or2\or3}%
823 $ \BIC@Temp 7{13\or14\or15\or16\or17\or18\or19\or0\or1\or2}%
824 $ \BIC@Temp 8{12\or13\or14\or15\or16\or17\or18\or19\or0\or1}%
825 $ \BIC@Temp 9{11\or12\or13\or14\or15\or16\or17\or18\or19\or0}%

```

2.13 Shl, Shr

\bigintcalcShl

```

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

```

\BIC@Shl

```

831 \def\BIC@Shl#1#2!{%
832 \ifx#1-%
833 \BIC@AfterFi{%
834 \expandafter-\romannumeral0%
835 & \BIC@Shl#2!!%
836 $ \BIC@AddXY#2!#2!!!%
837 }%

```



```

838 \else
839 \BIC@AfterFi{%
840 & \BIC@Sh1#1#2!!!%
841 $ \BIC@AddXY#1#2!#1#2!!!%
842 }%
843 \BIC@Fi
844 }

\BigIntCalcSh1

845 \def\BigIntCalcSh1#1!{%
846 \romannumeral0%
847 & \BIC@Sh1#1!!!%
848 $ \BIC@AddXY#1!#1!!!%
849 }

\BIC@Sh1

850 & \def\BIC@Sh1#1#2!{%
851 & \ifx\#2\%
852 & \BIC@AfterFi{%
853 & \BIC@@@Sh10!#1%
854 & }%
855 & \else
856 & \BIC@AfterFi{%
857 & \BIC@Sh1#2!#1%
858 & }%
859 & \BIC@Fi
860 & }%

\BIC@@@Sh1 #1: carry
#2: result
#3#4: reverted number

861 & \def\BIC@@@Sh1#1#2!#3#4!{%
862 & \ifx\#4\%
863 & \BIC@AfterFi{%
864 & \expandafter\BIC@Space
865 & \the\numexpr#3*2+#1\relax#2%
866 & }%
867 & \else
868 & \BIC@AfterFi{%
869 & \expandafter\BIC@@@Sh1\the\numexpr#3*2+#1!#2!#4!%
870 & }%
871 & \BIC@Fi
872 & }%

\BIC@@@Sh1

873 & \def\BIC@@@Sh1#1!{%
874 & \ifnum#1<10 %
875 & \BIC@AfterFi{%
876 & \BIC@@@Sh10!#1%
877 & }%
878 & \else
879 & \BIC@AfterFi{%
880 & \BIC@@@Sh1#1%
881 & }%
882 & \BIC@Fi
883 & }%

\bigintcalcShr

884 \def\bigintcalcShr#1{%
885 \romannumeral0%
886 \expandafter\expandafter\expandafter\BIC@Shr

```

```

887 \bigintcalcNum{#1}!%
888 }

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

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

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

\BIC@@@Shr #1: carry
#2#3: number
#4: result
922 \def\BIC@@@Shr#1#2#3!#4!{%
923 \ifx\\#3\\%
924 \ifodd#1#2 %
925 \BIC@AfterFiFi{%
926 & \expandafter\BIC@ShrResult\the\numexpr(#1#2-1)/2\relax
927 $ \expandafter\expandafter\expandafter\BIC@ShrResult
928 $ \csname BIC@ShrDigit#1#2\endcsname
929 #4!%
930 }%
931 \else
932 \BIC@AfterFiFi{%
933 & \expandafter\BIC@ShrResult\the\numexpr#1#2/2\relax
934 $ \expandafter\expandafter\expandafter\BIC@ShrResult
935 $ \csname BIC@ShrDigit#1#2\endcsname
936 #4!%
937 }%
938 \fi

```

```

939 \else
940   \ifodd#1#2 %
941     \BIC@AfterFiFi{%
942 &     \expandafter\BIC@@@Shr\the\numexpr(#1#2-1)/2\relax1%
943 $     \expandafter\expandafter\expandafter\BIC@@@Shr
944 $     \csname BIC@ShrDigit#1#2\endcsname
945     #3!#4!%
946   }%
947 \else
948   \BIC@AfterFiFi{%
949 &     \expandafter\BIC@@@Shr\the\numexpr#1#2/2\relax0%
950 $     \expandafter\expandafter\expandafter\BIC@@@Shr
951 $     \csname BIC@ShrDigit#1#2\endcsname
952     #3!#4!%
953   }%
954 \fi
955 \BIC@Fi
956 }

```

\BIC@ShrResult

```

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

```

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

```

959 \def\BIC@@@Shr#1#2#3!#4!{%
960   \BIC@@@Shr#2#3!#4#1!%
961 }

```

\BIC@ShrDigit[00-19]

```

962 $ \def\BIC@Temp#1#2#3#4{%
963 $   \expandafter\def\csname BIC@ShrDigit#1#2\endcsname{#3#4}%
964 $ }%
965 $ \BIC@Temp 0000%
966 $ \BIC@Temp 0101%
967 $ \BIC@Temp 0210%
968 $ \BIC@Temp 0311%
969 $ \BIC@Temp 0420%
970 $ \BIC@Temp 0521%
971 $ \BIC@Temp 0630%
972 $ \BIC@Temp 0731%
973 $ \BIC@Temp 0840%
974 $ \BIC@Temp 0941%
975 $ \BIC@Temp 1050%
976 $ \BIC@Temp 1151%
977 $ \BIC@Temp 1260%
978 $ \BIC@Temp 1361%
979 $ \BIC@Temp 1470%
980 $ \BIC@Temp 1571%
981 $ \BIC@Temp 1680%
982 $ \BIC@Temp 1781%
983 $ \BIC@Temp 1890%
984 $ \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
985 \def\BIC@Tim#1#2!{%
986 \ifx\#2\%
987 \BIC@AfterFi{%
988 \BIC@ProcessTim0!#1%
989 }%
990 \else
991 \BIC@AfterFi{%
992 \BIC@Tim#2!#1%
993 }%
994 \BIC@Fi
995 }

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

\BIC@@ProcessTim #1#2: carry?, new digit
#3: new number
#4: old number
#5: digit
1013 \def\BIC@@ProcessTim#1#2!{%
1014 \ifx\#2\%
1015 \BIC@AfterFi{%
1016 \BIC@ProcessTim0#1%
1017 }%
1018 \else
1019 \BIC@AfterFi{%
1020 \BIC@ProcessTim#1#2%
1021 }%
1022 \BIC@Fi
1023 }

\BIC@TimDigit #1: digit 0–9
#2: digit 3–9
#3: carry 0–9
1024 $ \def\BIC@TimDigit#1#2#3{%
1025 $ \ifcase#1 % 0
1026 $ \BIC@AfterFi{ #3}%
1027 $ \or % 1
1028 $ \BIC@AfterFi{%
1029 $ \expandafter\BIC@Space

```

```

1030 $ \number\csname BIC@AddCarry#2\endcsname#3 %
1031 $ }%
1032 $ \else
1033 $ \ifcase#3 %
1034 $ \BIC@AfterFiFi{%
1035 $ \expandafter\BIC@Space
1036 $ \number\csname BIC@MulDigit#2\endcsname#1 %
1037 $ }%
1038 $ \else
1039 $ \BIC@AfterFiFi{%
1040 $ \expandafter\BIC@Space
1041 $ \romannumeral0%
1042 $ \expandafter\BIC@AddXY
1043 $ \number\csname BIC@MulDigit#2\endcsname#1!%
1044 $ #3!!!%
1045 $ }%
1046 $ \fi
1047 $ \BIC@Fi
1048 $ }%

```

\BIC@MulDigit[3-9]

```

1049 $ \def\BIC@Temp#1#2{%
1050 $ \expandafter\def\csname BIC@MulDigit#1\endcsname##1{%
1051 $ \ifcase##1 0%
1052 $ \or ##1%
1053 $ \or ##2%
1054 $? \else\BigIntCalcError:ThisCannotHappen%
1055 $ \fi
1056 $ }%
1057 $ }%
1058 $ \BIC@Temp 3{6\or9\or12\or15\or18\or21\or24\or27}%
1059 $ \BIC@Temp 4{8\or12\or16\or20\or24\or28\or32\or36}%
1060 $ \BIC@Temp 5{10\or15\or20\or25\or30\or35\or40\or45}%
1061 $ \BIC@Temp 6{12\or18\or24\or30\or36\or42\or48\or54}%
1062 $ \BIC@Temp 7{14\or21\or28\or35\or42\or49\or56\or63}%
1063 $ \BIC@Temp 8{16\or24\or32\or40\or48\or56\or64\or72}%
1064 $ \BIC@Temp 9{18\or27\or36\or45\or54\or63\or72\or81}%

```

2.15 Mul

\bigintcalcMul

```

1065 \def\bigintcalcMul#1#2{%
1066 \romannumeral0%
1067 \expandafter\expandafter\expandafter\BIC@Mul
1068 \bigintcalcNum{#1}!{#2}%
1069 }

```

\BIC@Mul

```

1070 \def\BIC@Mul#1!#2{%
1071 \expandafter\expandafter\expandafter\BIC@MulSwitch
1072 \bigintcalcNum{#2}!#1!%
1073 }

```

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

```

1074 \def\BIC@MulSwitch#1#2!#3#4!{%
1075 \ifcase\BIC@Sgn#1#2! % x = 0
1076 \BIC@AfterFi{ 0}%
1077 \or % x > 0
1078 \ifcase\BIC@Sgn#3#4! % y = 0
1079 \BIC@AfterFiFi{ 0}%
1080 \or % y > 0
1081 \ifnum\BIC@PosCmp#1#2!#3#4!=1 % x > y
1082 \BIC@AfterFiFiFi{%
1083 \BIC@ProcessMul0!#1#2!#3#4!%
1084 }%
1085 \else % x <= y
1086 \BIC@AfterFiFiFi{%
1087 \BIC@ProcessMul0!#3#4!#1#2!%
1088 }%
1089 \fi
1090 \else % y < 0
1091 \expandafter-\romannumeral0%
1092 \ifnum\BIC@PosCmp#1#2!#4!=1 % x > -y
1093 \BIC@AfterFiFiFiFi{%
1094 \BIC@ProcessMul0!#1#2!#4!%
1095 }%
1096 \else % x <= -y
1097 \BIC@AfterFiFiFiFi{%
1098 \BIC@ProcessMul0!#4!#1#2!%
1099 }%
1100 \fi
1101 \fi
1102 \else % x < 0
1103 \ifcase\BIC@Sgn#3#4! % y = 0
1104 \BIC@AfterFiFiFi{ 0}%
1105 \or % y > 0
1106 \expandafter-\romannumeral0%
1107 \ifnum\BIC@PosCmp#2!#3#4!=1 % -x > y
1108 \BIC@AfterFiFiFiFi{%
1109 \BIC@ProcessMul0!#2!#3#4!%
1110 }%
1111 \else % -x <= y
1112 \BIC@AfterFiFiFiFi{%
1113 \BIC@ProcessMul0!#3#4!#2!%
1114 }%
1115 \fi
1116 \else % y < 0
1117 \ifnum\BIC@PosCmp#2!#4!=1 % -x > -y
1118 \BIC@AfterFiFiFiFi{%
1119 \BIC@ProcessMul0!#2!#4!%
1120 }%
1121 \else % -x <= -y
1122 \BIC@AfterFiFiFiFi{%

```

```

1123         \BIC@ProcessMul0!#4!#2!%
1124     }%
1125     \fi
1126     \fi
1127     \BIC@Fi
1128 }

```

\BigIntCalcMul

```

1129 \def\BigIntCalcMul#1!#2!{%
1130     \romannumeral0%
1131     \BIC@ProcessMul0!#1!#2!%
1132 }

```

\BIC@ProcessMul

```

#1: result
#2: number  $x$ 
#3#4: number  $y$ 
1133 \def\BIC@ProcessMul#1!#2!#3#4!{%
1134     \ifx\#4\%
1135         \BIC@AfterFi{%
1136             \expandafter\expandafter\expandafter\BIC@Space
1137             \bigintcalcAdd{\BIC@Tim#2!#3}{#10}%
1138         }%
1139     \else
1140         \BIC@AfterFi{%
1141             \expandafter\expandafter\expandafter\BIC@ProcessMul
1142             \bigintcalcAdd{\BIC@Tim#2!#3}{#10}!#2!#4!%
1143         }%
1144     \BIC@Fi
1145 }

```

2.16 Sqr

\bigintcalcSqr

```

1146 \def\bigintcalcSqr#1{%
1147     \romannumeral0%
1148     \expandafter\expandafter\expandafter\BIC@Sqr
1149     \bigintcalcNum{#1}!%
1150 }

```

\BIC@Sqr

```

1151 \def\BIC@Sqr#1{%
1152     \ifx#1-%
1153         \expandafter\BIC@@Sqr
1154     \else
1155         \expandafter\BIC@@Sqr\expandafter#1%
1156     \fi
1157 }

```

\BIC@@Sqr

```

1158 \def\BIC@@Sqr#1!{%
1159     \BIC@ProcessMul0!#1!#1!%
1160 }

```

2.17 Fac

\bigintcalcFac

```

1161 \def\bigintcalcFac#1{%
1162     \romannumeral0%
1163     \expandafter\expandafter\expandafter\BIC@Fac
1164     \bigintcalcNum{#1}!%
1165 }

```

\BIC@Fac

```
1166 \def\BIC@Fac#1#2!{%
1167   \ifx#1-%
1168     \BIC@AfterFi{ 0\BigIntCalcError:FacNegative}%
1169   \else
1170     \ifnum\BIC@PosCmp#1#2!13!<0 %
1171       \ifcase#1#2 %
1172         \BIC@AfterFiFiFi{ 1}% 0!
1173         \or\BIC@AfterFiFiFi{ 1}% 1!
1174         \or\BIC@AfterFiFiFi{ 2}% 2!
1175         \or\BIC@AfterFiFiFi{ 6}% 3!
1176         \or\BIC@AfterFiFiFi{ 24}% 4!
1177         \or\BIC@AfterFiFiFi{ 120}% 5!
1178         \or\BIC@AfterFiFiFi{ 720}% 6!
1179         \or\BIC@AfterFiFiFi{ 5040}% 7!
1180         \or\BIC@AfterFiFiFi{ 40320}% 8!
1181         \or\BIC@AfterFiFiFi{ 362880}% 9!
1182         \or\BIC@AfterFiFiFi{ 3628800}% 10!
1183         \or\BIC@AfterFiFiFi{ 39916800}% 11!
1184         \or\BIC@AfterFiFiFi{ 479001600}% 12!
1185       \else\BigIntCalcError:ThisCannotHappen%
1186       \fi
1187     \else
1188       \BIC@AfterFiFiFi{%
1189         \BIC@ProcessFac#1#2!479001600!%
1190       }%
1191     \fi
1192   \BIC@Fi
1193 }
```

\BIC@ProcessFac #1: n

#2: result

```
1194 \def\BIC@ProcessFac#1!#2!{%
1195   \ifnum\BIC@PosCmp#1!12!=0 %
1196     \BIC@AfterFi{ #2}%
1197   \else
1198     \BIC@AfterFi{%
1199       \expandafter\BIC@ProcessFac
1200       \romannumeral0\BIC@ProcessMul0!#2!#1!%
1201       !#1!%
1202     }%
1203   \BIC@Fi
1204 }
```

\BIC@@ProcessFac #1: result

#2: n

```
1205 \def\BIC@@ProcessFac#1!#2!{%
1206   \expandafter\BIC@ProcessFac
1207   \romannumeral0\BIC@Dec#2!{%
1208     !#1!%
1209   }
```

2.18 Pow

\bigintcalcPow #1: basis

#2: power

```
1210 \def\bigintcalcPow#1{%
1211   \romannumeral0%
1212   \expandafter\expandafter\expandafter\BIC@Pow
1213   \bigintcalcNum{#1}!%
1214 }
```



```

\BIC@Pow #1: basis
#2: power
1215 \def\BIC@Pow#1!#2{%
1216 \expandafter\expandafter\expandafter\BIC@PowSwitch
1217 \bigintcalcNum{#2}!#1!%
1218 }

```

```

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

```

1219 \def\BIC@PowSwitch#1#2!#3#4!{%
1220 \ifcase\ifx\#2\%
1221 \ifx#100 %  $y = 0$ 
1222 \else\ifx#111 %  $y = 1$ 
1223 \else\ifx#122 %  $y = 2$ 
1224 \else4 %  $y > 2$ 
1225 \fi\fi\fi
1226 \else
1227 \ifx#1-3 %  $y < 0$ 
1228 \else4 %  $y > 2$ 
1229 \fi
1230 \fi
1231 \BIC@AfterFi{ 1}%  $y = 0$ 
1232 \or %  $y = 1$ 
1233 \BIC@AfterFi{ #3#4}%
1234 \or %  $y = 2$ 
1235 \ifx#3-%  $x < 0$ 
1236 \BIC@AfterFiFi{%
1237 \BIC@ProcessMulo!#4!#4!%
1238 }%
1239 \else %  $x \geq 0$ 
1240 \BIC@AfterFiFi{%
1241 \BIC@ProcessMulo!#3#4!#3#4!%
1242 }%
1243 \fi
1244 \or %  $y < 0$ 
1245 \ifcase\ifx\#4\%
1246 \ifx#300 %  $x = 0$ 
1247 \else\ifx#311 %  $x = 1$ 
1248 \else3 %  $x > 1$ 
1249 \fi\fi
1250 \else
1251 \ifcase\BIC@MinusOne#3#4! %

```

```

1252         3 % |x| > 1
1253         \or
1254         2 % x = -1
1255 ?         \else\BigIntCalcError:ThisCannotHappen%
1256         \fi
1257         \fi
1258         \BIC@AfterFiFi{ 0}\BigIntCalcError:DivisionByZero}% x = 0
1259     \or % x = 1
1260         \BIC@AfterFiFi{ 1}% x = 1
1261     \or % x = -1
1262         \ifcase\BIC@ModTwo#2! % even(y)
1263             \BIC@AfterFiFiFi{ 1}%
1264         \or % odd(y)
1265             \BIC@AfterFiFiFi{ -1}%
1266 ?         \else\BigIntCalcError:ThisCannotHappen%
1267         \fi
1268     \or % |x| > 1
1269         \BIC@AfterFiFi{ 0}%
1270 ?         \else\BigIntCalcError:ThisCannotHappen%
1271         \fi
1272     \or % y > 2
1273         \ifcase\ifx\#4\%
1274             \ifx#300 % x = 0
1275             \else\ifx#311 % x = 1
1276             \else4 % x > 1
1277             \fi\fi
1278         \else
1279             \ifx#3-%
1280                 \ifcase\BIC@MinusOne#3#4! %
1281                     3 % x < -1
1282                 \else
1283                     2 % x = -1
1284                 \fi
1285             \else
1286                 4 % x > 1
1287             \fi
1288         \fi
1289         \BIC@AfterFiFi{ 0}% x = 0
1290     \or % x = 1
1291         \BIC@AfterFiFi{ 1}% x = 1
1292     \or % x = -1
1293         \ifcase\BIC@ModTwo#1#2! % even(y)
1294             \BIC@AfterFiFiFi{ 1}%
1295         \or % odd(y)
1296             \BIC@AfterFiFiFi{ -1}%
1297 ?         \else\BigIntCalcError:ThisCannotHappen%
1298         \fi
1299     \or % x < -1
1300         \ifcase\BIC@ModTwo#1#2! % even(y)
1301             \BIC@AfterFiFiFi{%
1302                 \BIC@PowRec#4!#1#2!1!%
1303             }%
1304         \or % odd(y)
1305             \expandafter-\romannumeral0%
1306             \BIC@AfterFiFiFi{%
1307                 \BIC@PowRec#4!#1#2!1!%
1308             }%
1309 ?         \else\BigIntCalcError:ThisCannotHappen%
1310         \fi
1311     \or % x > 1
1312         \BIC@AfterFiFi{%
1313             \BIC@PowRec#3#4!#1#2!1!%

```

```

1314     }%
1315 ?   \else\BigIntCalcError:ThisCannotHappen%
1316     \fi
1317 ?   \else\BigIntCalcError:ThisCannotHappen%
1318     \BIC@Fi
1319 }

```

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.

```

1320 \def\BIC@ModTwo#1#2!{%
1321   \ifx\#2\%
1322     \ifodd#1 %
1323       \BIC@AfterFiFi1%
1324     \else
1325       \BIC@AfterFiFi0%
1326     \fi
1327   \else
1328     \BIC@AfterFi{%
1329     \BIC@ModTwo#2!%
1330   }%
1331   \BIC@Fi
1332 }

```

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

```

1333 \def\BIC@MinusOne#1#2!{%
1334   \ifx#1-%
1335     \BIC@@MinusOne#2!%
1336   \else
1337     0%
1338   \fi
1339 }

```

`\BIC@@MinusOne`

```

1340 \def\BIC@@MinusOne#1#2!{%
1341   \ifx#11%
1342     \ifx\#2\%
1343       1%
1344     \else
1345       0%
1346     \fi
1347   \else
1348     0%
1349   \fi
1350 }

```

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)
1351 \def\BIC@PowRec#1!#2#3!#4!{%
1352 \ifcase\ifx#21\ifx#\#3\0 \else1 \fi\else1 \fi %  $y = 1$ 
1353 \ifnum\BIC@PosCmp#1!#4!=1 %  $x > r$ 
1354 \BIC@AfterFiFi{%
1355 \BIC@ProcessMul0!#1!#4!%
1356 }%
1357 \else
1358 \BIC@AfterFiFi{%
1359 \BIC@ProcessMul0!#4!#1!%
1360 }%
1361 \fi
1362 \or
1363 \ifcase\BIC@ModTwo#2#3! % even( $y$ )
1364 \BIC@AfterFiFi{%
1365 \expandafter\BIC@@PowRec\romannumeral0%
1366 \BIC@Shr#2#3!%
1367 !#1!#4!%
1368 }%
1369 \or % odd( $y$ )
1370 \ifnum\BIC@PosCmp#1!#4!=1 %  $x > r$ 
1371 \BIC@AfterFiFiFiFi{%
1372 \expandafter\BIC@@@PowRec\romannumeral0%
1373 \BIC@ProcessMul0!#1!#4!%
1374 !#1!#2#3!%
1375 }%
1376 \else
1377 \BIC@AfterFiFiFiFi{%
1378 \expandafter\BIC@@@PowRec\romannumeral0%
1379 \BIC@ProcessMul0!#1!#4!%
1380 !#1!#2#3!%
1381 }%
1382 \fi
1383 ? \else\BigIntCalcError:ThisCannotHappen%
1384 \fi
1385 ? \else\BigIntCalcError:ThisCannotHappen%
1386 \BIC@Fi
1387 }

\BIC@@PowRec #1:  $y/2$ 
#2:  $x$ 
#3: new  $r$  ( $r$  or  $r * x$ )
1388 \def\BIC@@PowRec#1!#2!#3!{%
1389 \expandafter\BIC@PowRec\romannumeral0%
1390 \BIC@ProcessMul0!#2!#2!%
1391 !#1!#3!%
1392 }

\BIC@@@PowRec #1:  $r * x$  #2:  $x$  #3:  $y$ 
1393 \def\BIC@@@PowRec#1!#2!#3!{%
1394 \expandafter\BIC@@PowRec\romannumeral0%
1395 \BIC@Shr#3!%
1396 !#2!#1!%
1397 }

```

2.19 Div

```

\bigintcalcDiv #1: x
                #2: y (divisor)
1398 \def\bigintcalcDiv#1{%
1399   \romannumeral0%
1400   \expandafter\expandafter\expandafter\BIC@Div
1401   \bigintcalcNum{#1}!%
1402 }

\BIC@Div #1: x
          #2: y
1403 \def\BIC@Div#1!#2{%
1404   \expandafter\expandafter\expandafter\BIC@DivSwitchSign
1405   \bigintcalcNum{#2}!#1!%
1406 }

\BigIntCalcDiv
1407 \def\BigIntCalcDiv#1!#2!{%
1408   \romannumeral0%
1409   \BIC@DivSwitchSign#2!#1!%
1410 }

```

\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
1411 \def\BIC@DivSwitchSign#1#2!#3#4!{%
1412   \ifcase\BIC@Sgn#1#2! % y = 0
1413     \BIC@AfterFi{ 0\BigIntCalcError:DivisionByZero}%
1414   \or % y > 0
1415     \ifcase\BIC@Sgn#3#4! % x = 0
1416       \BIC@AfterFiFi{ 0}%
1417     \or % x > 0
1418       \BIC@AfterFiFi{%
1419         \BIC@DivSwitch{ }#3#4!#1#2!%
1420       }%
1421     \else % x < 0
1422       \BIC@AfterFiFi{%
1423         \BIC@DivSwitch-#4!#1#2!%
1424       }%
1425     \fi
1426   \else % y < 0
1427     \ifcase\BIC@Sgn#3#4! % x = 0
1428       \BIC@AfterFiFi{ 0}%
1429     \or % x > 0
1430       \BIC@AfterFiFi{%
1431         \BIC@DivSwitch-#3#4!#2!%
1432       }%
1433     \else % x < 0
1434       \BIC@AfterFiFi{%
1435         \BIC@DivSwitch{ }#4!#2!%
1436       }%
1437     \fi

```

```
1438 \BIC@Fi
1439 }
```

\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 ≠ 0)
1440 \def\BIC@DivSwitch#1#2!#3#4!{%
1441 \ifcase\BIC@PosCmp#3#4!#2!% y = x
1442 \BIC@AfterFi{ #1}%
1443 \or % y > x
1444 \BIC@AfterFi{ 0}%
1445 \else % y < x
1446 \ifx\#1\%
1447 \else
1448 \expandafter-\romannumeral0%
1449 \fi
1450 \ifcase\ifx\#4\%
1451 \ifx#310 % y = 1
1452 \else\ifx#321 % y = 2
1453 \else\ifx#342 % y = 4
1454 \else3 % y > 2
1455 \fi\fi\fi
1456 \else
1457 3 % y > 2
1458 \fi
1459 \BIC@AfterFiFi{ #2}% y = 1
1460 \or % y = 2
1461 \BIC@AfterFiFi{%
1462 \BIC@@Shr#2!%
1463 }%
1464 \or % y = 4
1465 \BIC@AfterFiFi{%
1466 \expandafter\BIC@@Shr\romannumeral0%
1467 \BIC@@Shr#2!!%
1468 }%
1469 \or % y > 2
1470 \BIC@AfterFiFi{%
1471 \BIC@DivStartX#2!#3#4!!!%
1472 }%
1473 ? \else\BigIntCalcError:ThisCannotHappen%
1474 \fi
1475 \BIC@Fi
1476 }
```

```
\BIC@ProcessDiv #1#2: x
#3#4: y
#5: collect first digits of x
#6: corresponding digits of y
1477 \def\BIC@DivStartX#1#2!#3#4!#5!#6!{%
1478 \ifx\#4\%
1479 \BIC@AfterFi{%
1480 \BIC@DivStartYii#6#3#4!{#5#1}#2=!%
1481 }%
```

```

1482 \else
1483   \BIC@AfterFi{%
1484     \BIC@DivStartX#2!#4!#5#1!#6#3!%
1485   }%
1486 \BIC@Fi
1487 }

\BIC@DivStartYii #1:  $y$ 
                 #2:  $x, =$ 
1488 \def\BIC@DivStartYii#1!{%
1489   \expandafter\BIC@DivStartYiv\romannumeral0%
1490   \BIC@Shl#1!%
1491   !#1!%
1492 }

\BIC@DivStartYiv #1:  $2y$ 
                 #2:  $y$ 
                 #3:  $x, =$ 
1493 \def\BIC@DivStartYiv#1!{%
1494   \expandafter\BIC@DivStartYvi\romannumeral0%
1495   \BIC@Shl#1!%
1496   !#1!%
1497 }

\BIC@DivStartYvi #1:  $4y$ 
                 #2:  $2y$ 
                 #3:  $y$ 
                 #4:  $x, =$ 
1498 \def\BIC@DivStartYvi#1!#2!{%
1499   \expandafter\BIC@DivStartYviii\romannumeral0%
1500   \BIC@AddXY#1!#2!!!%
1501   !#1!#2!%
1502 }

\BIC@DivStartYviii #1:  $6y$ 
                  #2:  $4y$ 
                  #3:  $2y$ 
                  #4:  $y$ 
                  #5:  $x, =$ 
1503 \def\BIC@DivStartYviii#1!#2!{%
1504   \expandafter\BIC@DivStart\romannumeral0%
1505   \BIC@Shl#2!%
1506   !#1!#2!%
1507 }

\BIC@DivStart #1:  $8y$ 
              #2:  $6y$ 
              #3:  $4y$ 
              #4:  $2y$ 
              #5:  $y$ 
              #6:  $x, =$ 
1508 \def\BIC@DivStart#1!#2!#3!#4!#5!#6!{%
1509   \BIC@ProcessDiv#6!!#5!#4!#3!#2!#1!=%
1510 }

\BIC@ProcessDiv #1#2#3:  $x, =$ 
                #4: result
                #5:  $y$ 
                #6:  $2y$ 
                #7:  $4y$ 

```

```

#8: 6y
#9: 8y
1511 \def\BIC@ProcessDiv#1#2#3!#4!#5!{%
1512 \ifcase\BIC@PosCmp#5!#1!% y = #1
1513 \ifx#2=%
1514 \BIC@AfterFiFi{\BIC@DivCleanup{#41}}%
1515 \else
1516 \BIC@AfterFiFi{%
1517 \BIC@ProcessDiv#2#3!#41!#5!%
1518 }%
1519 \fi
1520 \or % y > #1
1521 \ifx#2=%
1522 \BIC@AfterFiFi{\BIC@DivCleanup{#40}}%
1523 \else
1524 \ifx\#4\%
1525 \BIC@AfterFiFiFi{%
1526 \BIC@ProcessDiv{#1#2}#3!#5!%
1527 }%
1528 \else
1529 \BIC@AfterFiFiFi{%
1530 \BIC@ProcessDiv{#1#2}#3!#40!#5!%
1531 }%
1532 \fi
1533 \fi
1534 \else % y < #1
1535 \BIC@AfterFi{%
1536 \BIC@@ProcessDiv{#1}#2#3!#4!#5!%
1537 }%
1538 \BIC@Fi
1539 }

```

```

\BIC@DivCleanup #1: result
#2: garbage
1540 \def\BIC@DivCleanup#1#2={ #1}%

```

\BIC@@ProcessDiv

```

1541 \def\BIC@@ProcessDiv#1#2#3!#4!#5!#6!#7!{%
1542 \ifcase\BIC@PosCmp#7!#1!% 4y = #1
1543 \ifx#2=%
1544 \BIC@AfterFiFi{\BIC@DivCleanup{#44}}%
1545 \else
1546 \BIC@AfterFiFi{%
1547 \BIC@ProcessDiv#2#3!#44!#5!#6!#7!%
1548 }%
1549 \fi
1550 \or % 4y > #1
1551 \ifcase\BIC@PosCmp#6!#1!% 2y = #1
1552 \ifx#2=%
1553 \BIC@AfterFiFiFi{\BIC@DivCleanup{#42}}%
1554 \else
1555 \BIC@AfterFiFiFi{%
1556 \BIC@ProcessDiv#2#3!#42!#5!#6!#7!%
1557 }%
1558 \fi
1559 \or % 2y > #1
1560 \ifx#2=%
1561 \BIC@AfterFiFiFi{\BIC@DivCleanup{#41}}%
1562 \else
1563 \BIC@AfterFiFiFi{%
1564 \BIC@DivSub#1!#5!#2#3!#41!#5!#6!#7!%
1565 }%

```



```

1566     \fi
1567     \else % 2y < #1
1568     \BIC@AfterFiFi{%
1569     \expandafter\BIC@ProcessDivII\romannumeral0%
1570     \BIC@SubXY#1!#6!!!%
1571     !#2#3!#4!#5!23%
1572     #6!#7!%
1573     }%
1574     \fi
1575     \else % 4y < #1
1576     \BIC@AfterFiFi{%
1577     \BIC@@@ProcessDiv{#1}#2#3!#4!#5!#6!#7!%
1578     }%
1579     \BIC@Fi
1580 }

```

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

```

1581 \def\BIC@DivSub#1!#2!#3{%
1582 \expandafter\BIC@ProcessDiv\expandafter{%
1583 \romannumeral0%
1584 \BIC@SubXY#1!#2!!!%
1585 #3%
1586 }%
1587 }

```

```

\BIC@ProcessDivII #1:  $x' - 2y$ 
#2#3: remaining  $x$ , =
#4: result
#5:  $y$ 
#6: first possible result digit
#7: second possible result digit
1588 \def\BIC@ProcessDivII#1!#2#3!#4!#5!#6#7{%
1589 \ifcase\BIC@PosCmp#5!#1!%  $y = #1$ 
1590 \ifx#2=%
1591 \BIC@AfterFiFi{\BIC@DivCleanup{#4#7}}%
1592 \else
1593 \BIC@AfterFiFi{%
1594 \BIC@ProcessDiv#2#3!#4#7!#5!%
1595 }%
1596 \fi
1597 \or %  $y > #1$ 
1598 \ifx#2=%
1599 \BIC@AfterFiFi{\BIC@DivCleanup{#4#6}}%
1600 \else
1601 \BIC@AfterFiFi{%
1602 \BIC@ProcessDiv{#1#2}#3!#4#6!#5!%
1603 }%
1604 \fi
1605 \else %  $y < #1$ 
1606 \ifx#2=%
1607 \BIC@AfterFiFi{\BIC@DivCleanup{#4#7}}%
1608 \else
1609 \BIC@AfterFiFi{%
1610 \BIC@DivSub#1!#5!#2#3!#4#7!#5!%
1611 }%
1612 \fi
1613 \BIC@Fi
1614 }

```

```

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

```

```

#6: 2y
#7: 4y
#8: 6y
#9: 8y
1615 \def\BIC@@@ProcessDiv#1#2#3!#4!#5!#6!#7!#8!#9!{%
1616 \ifcase\BIC@PosCmp#8!#1!% 6y = #1
1617 \ifx#2=%
1618 \BIC@AfterFiFi{\BIC@DivCleanup{#46}}}%
1619 \else
1620 \BIC@AfterFiFi{%
1621 \BIC@ProcessDiv#2#3!#46!#5!#6!#7!#8!#9!%
1622 }%
1623 \fi
1624 \or % 6y > #1
1625 \BIC@AfterFiFi{%
1626 \expandafter\BIC@ProcessDivII\romannumeral0%
1627 \BIC@SubXY#1!#7!!!%
1628 !#2#3!#4!#5!45%
1629 #6!#7!#8!#9!%
1630 }%
1631 \else % 6y < #1
1632 \ifcase\BIC@PosCmp#9!#1!% 8y = #1
1633 \ifx#2=%
1634 \BIC@AfterFiFiFi{\BIC@DivCleanup{#48}}}%
1635 \else
1636 \BIC@AfterFiFiFi{%
1637 \BIC@ProcessDiv#2#3!#48!#5!#6!#7!#8!#9!%
1638 }%
1639 \fi
1640 \or % 8y > #1
1641 \BIC@AfterFiFiFi{%
1642 \expandafter\BIC@ProcessDivII\romannumeral0%
1643 \BIC@SubXY#1!#8!!!%
1644 !#2#3!#4!#5!67%
1645 #6!#7!#8!#9!%
1646 }%
1647 \else % 8y < #1
1648 \BIC@AfterFiFiFi{%
1649 \expandafter\BIC@ProcessDivII\romannumeral0%
1650 \BIC@SubXY#1!#9!!!%
1651 !#2#3!#4!#5!89%
1652 #6!#7!#8!#9!%
1653 }%
1654 \fi
1655 \BIC@Fi
1656 }

```

2.20 Mod

```

\bigintcalcMod #1: x
#2: y
1657 \def\bigintcalcMod#1{%
1658 \romannumeral0%
1659 \expandafter\expandafter\expandafter\BIC@Mod
1660 \bigintcalcNum{#1}!%
1661 }

\BIC@Mod #1: x
#2: y
1662 \def\BIC@Mod#1!#2{%
1663 \expandafter\expandafter\expandafter\BIC@ModSwitchSign

```

```

1664 \bigintcalcNum{#2}!#1!%
1665 }

```

\BigIntCalcMod

```

1666 \def\BigIntCalcMod#1!#2!{%
1667 \romannumeral0%
1668 \BIC@ModSwitchSign#2!#1!%
1669 }

```

\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
1670 \def\BIC@ModSwitchSign#1#2!#3#4!{%
1671 \ifcase\ifx\#2\%
1672     \ifx#100 % y = 0
1673     \else1 % y > 0
1674     \fi
1675     \else
1676     \ifx#1-2 % y < 0
1677     \else1 % y > 0
1678     \fi
1679     \fi
1680     \BIC@AfterFi{ 0\BigIntCalcError:DivisionByZero}%
1681 \or % y > 0
1682 \ifcase\ifx\#4\%
1683     \BIC@AfterFiFi{ 0}%
1684     \else
1685     \BIC@AfterFiFi{%
1686     \BIC@ModSwitch{ }#3#4!#1#2!%
1687     }%
1688     \fi
1689 \else % y < 0
1690     \ifcase\ifx\#4\%
1691     \ifx#300 % x = 0
1692     \else1 % x > 0
1693     \fi
1694     \else
1695     \ifx#3-2 % x < 0
1696     \else1 % x > 0
1697     \fi
1698     \fi
1699     \BIC@AfterFiFi{ 0}%
1700 \or % x > 0
1701     \BIC@AfterFiFi{%
1702     \BIC@ModSwitch--#3#4!#2!%
1703     }%
1704 \else % x < 0
1705     \BIC@AfterFiFi{%
1706     \BIC@ModSwitch-#4!#2!%
1707     }%
1708     \fi
1709 \BIC@Fi
1710 }

```

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

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

```

#1: sign
#2#3: x
#4#5: y
1711 \def\BIC@ModSwitch#1#2#3!#4#5!{%
1712   \ifcase\ifx\#5\%
1713       \ifx#410 % y = 1
1714       \else\ifx#421 % y = 2
1715       \else2 % y > 2
1716       \fi\fi
1717       \else2 % y > 2
1718       \fi
1719   \BIC@AfterFi{ 0}% y = 1
1720 \or % y = 2
1721   \ifcase\BIC@ModTwo#2#3! % even(x)
1722   \BIC@AfterFiFi{ 0}%
1723   \or % odd(x)
1724   \BIC@AfterFiFi{ #11}%
1725 ? \else\BigIntCalcError:ThisCannotHappen%
1726   \fi
1727 \or % y > 2
1728   \ifx\#1\%
1729   \else
1730     \expandafter\BIC@Space\romannumeral0%
1731     \expandafter\BIC@ModMinus\romannumeral0%
1732     \fi
1733   \ifx#2-% x < 0
1734     \BIC@AfterFiFi{%
1735       \expandafter\expandafter\expandafter\BIC@ModX
1736       \bigintcalcSub{#2#3}{%
1737         \bigintcalcMul{#4#5}{\bigintcalcDiv{#2#3}{#4#5}}%
1738       }!#4#5!%
1739     }%
1740   \else % x > 0
1741     \BIC@AfterFiFi{%
1742       \expandafter\expandafter\expandafter\BIC@Space
1743       \bigintcalcSub{#2#3}{%
1744         \bigintcalcMul{#4#5}{\bigintcalcDiv{#2#3}{#4#5}}%
1745       }%
1746     }%
1747   \fi
1748 ? \else\BigIntCalcError:ThisCannotHappen%
1749   \BIC@Fi
1750 }

```

\BIC@ModMinus

```

1751 \def\BIC@ModMinus#1{%
1752   \ifx#10%
1753     \BIC@AfterFi{ 0}%
1754   \else
1755     \BIC@AfterFi{ -#1}%
1756   \BIC@Fi
1757 }

```

\BIC@ModX

```

#1#2: z
#3: x
1758 \def\BIC@ModX#1#2!#3!{%

```

```

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

1768 \BIC@AtEnd%
1769 </package>

```

3 Test

3.1 Catcode checks for loading

```

1770 <*test1>
1771 \catcode'\{=1 %
1772 \catcode'\}=2 %
1773 \catcode'\#=6 %
1774 \catcode'\@=11 %
1775 \expandafter\ifx\csname count@\endcsname\relax
1776 \countdef\count@=255 %
1777 \fi
1778 \expandafter\ifx\csname @gobble\endcsname\relax
1779 \long\def\@gobble#1{}%
1780 \fi
1781 \expandafter\ifx\csname @firstofone\endcsname\relax
1782 \long\def\@firstofone#1{#1}%
1783 \fi
1784 \expandafter\ifx\csname loop\endcsname\relax
1785 \expandafter\@firstofone
1786 \else
1787 \expandafter\@gobble
1788 \fi
1789 {%
1790 \def\loop#1\repeat{%
1791   \def\body{#1}%
1792   \iterate
1793 }%
1794 \def\iterate{%
1795   \body
1796   \let\next\iterate
1797   \else
1798   \let\next\relax
1799   \fi
1800   \next
1801 }%
1802 \let\repeat=\fi
1803 }%
1804 \def\RestoreCatcodes{}
1805 \count@=0 %
1806 \loop
1807 \edef\RestoreCatcodes{%
1808   \RestoreCatcodes
1809   \catcode\the\count@=\the\catcode\count@\relax
1810 }%
1811 \ifnum\count@<255 %
1812 \advance\count@ 1 %
1813 \repeat

```

```

1814
1815 \def\RangeCatcodeInvalid#1#2{%
1816   \count@=#1\relax
1817   \loop
1818     \catcode\count@=15 %
1819   \ifnum\count@<#2\relax
1820     \advance\count@ 1 %
1821   \repeat
1822 }
1823 \def\RangeCatcodeCheck#1#2#3{%
1824   \count@=#1\relax
1825   \loop
1826     \ifnum#3=\catcode\count@
1827     \else
1828       \errmessage{%
1829         Character \the\count@\space
1830         with wrong catcode \the\catcode\count@\space
1831         instead of \number#3%
1832       }%
1833     \fi
1834   \ifnum\count@<#2\relax
1835     \advance\count@ 1 %
1836   \repeat
1837 }
1838 \def\space{ }
1839 \expandafter\ifx\csname LoadCommand\endcsname\relax
1840 \def\LoadCommand{\input bigintcalc.sty\relax}%
1841 \fi
1842 \def\Test{%
1843   \RangeCatcodeInvalid{0}{47}%
1844   \RangeCatcodeInvalid{58}{64}%
1845   \RangeCatcodeInvalid{91}{96}%
1846   \RangeCatcodeInvalid{123}{255}%
1847   \catcode'\@=12 %
1848   \catcode'\|=0 %
1849   \catcode'\%=14 %
1850   \LoadCommand
1851   \RangeCatcodeCheck{0}{36}{15}%
1852   \RangeCatcodeCheck{37}{37}{14}%
1853   \RangeCatcodeCheck{38}{47}{15}%
1854   \RangeCatcodeCheck{48}{57}{12}%
1855   \RangeCatcodeCheck{58}{63}{15}%
1856   \RangeCatcodeCheck{64}{64}{12}%
1857   \RangeCatcodeCheck{65}{90}{11}%
1858   \RangeCatcodeCheck{91}{91}{15}%
1859   \RangeCatcodeCheck{92}{92}{0}%
1860   \RangeCatcodeCheck{93}{96}{15}%
1861   \RangeCatcodeCheck{97}{122}{11}%
1862   \RangeCatcodeCheck{123}{255}{15}%
1863   \RestoreCatcodes
1864 }
1865 \Test
1866 \csname @@end\endcsname
1867 \end
1868 </test1>

```

3.2 Macro tests

3.2.1 Preamble with test macro definitions

```

1869 <*test2>
1870 \NeedsTeXFormat{LaTeX2e}
1871 \nofiles

```

```

1872 \documentclass{article}
1873 <noetex> \let\SavedNumexpr\numexpr
1874 <noetex> \let\numexpr\UNDEFINED
1875 \makeatletter
1876 \chardef\BIC@TestMode=1 %
1877 \makeatother
1878 \usepackage{bigintcalc}[2011/01/30]
1879 <noetex> \let\numexpr\SavedNumexpr
1880 \usepackage{qstest}
1881 \IncludeTests{*}
1882 \LogTests{log}{*}{*}
1883 \newcommand*{\TestSpaceAtEnd}[1]{%
1884 <noetex> \let\SavedNumexpr\numexpr
1885 <noetex> \let\numexpr\UNDEFINED
1886 \edef\resultA{#1}%
1887 \edef\resultB{#1 }%
1888 <noetex> \let\numexpr\SavedNumexpr
1889 \Expect*{\resultA\space}*{\resultB}%
1890 }
1891 \newcommand*{\TestResult}[2]{%
1892 <noetex> \let\SavedNumexpr\numexpr
1893 <noetex> \let\numexpr\UNDEFINED
1894 \edef\result{#1}%
1895 <noetex> \let\numexpr\SavedNumexpr
1896 \Expect*{\result}{#2}%
1897 }
1898 \newcommand*{\TestResultTwoExpansions}[2]{%
1899 <*noetex>
1900 \begingroup
1901 \let\numexpr\UNDEFINED
1902 \expandafter\expandafter\expandafter
1903 \endgroup
1904 </noetex>
1905 \expandafter\expandafter\expandafter\Expect
1906 \expandafter\expandafter\expandafter{#1}{#2}%
1907 }
1908 \newcount\TestCount
1909 <etex> \newcommand*{\TestArg}[1]{\numexpr#1\relax}
1910 <noetex> \newcommand*{\TestArg}[1]{#1}
1911 \newcommand*{\TestTeXDivide}[2]{%
1912 \TestCount=\TestArg{#1}\relax
1913 \divide\TestCount by \TestArg{#2}\relax
1914 \Expect*{\bigintcalcDiv{#1}{#2}}*{\the\TestCount}%
1915 }
1916 \newcommand*{\Test}[2]{%
1917 \TestResult{#1}{#2}%
1918 \TestResultTwoExpansions{#1}{#2}%
1919 \TestSpaceAtEnd{#1}%
1920 }
1921 \newcommand*{\TestExch}[2]{\Test{#2}{#1}}
1922 \newcommand*{\TestInv}[2]{%
1923 \Test{\bigintcalcInv{#1}}{#2}%
1924 }
1925 \newcommand*{\TestAbs}[2]{%
1926 \Test{\bigintcalcAbs{#1}}{#2}%
1927 }
1928 \newcommand*{\TestSgn}[2]{%
1929 \Test{\bigintcalcSgn{#1}}{#2}%
1930 }
1931 \newcommand*{\TestMin}[3]{%
1932 \Test{\bigintcalcMin{#1}{#2}}{#3}%
1933 }

```

```

1934 \newcommand*\TestMax}[3]{%
1935   \Test{\bigintcalcMax{#1}{#2}}{#3}%
1936 }
1937 \newcommand*\TestCmp}[3]{%
1938   \Test{\bigintcalcCmp{#1}{#2}}{#3}%
1939 }
1940 \newcommand*\TestOdd}[2]{%
1941   \Test{\bigintcalcOdd{#1}}{#2}%
1942   \edef\x{%
1943     \noexpand\Test{%
1944       \noexpand\BigIntCalcOdd
1945       \bigintcalcAbs{#1}!%
1946     }{#2}%
1947   }%
1948   \x
1949 }
1950 \newcommand*\TestInc}[2]{%
1951   \Test{\bigintcalcInc{#1}}{#2}%
1952   \ifnum\bigintcalcSgn{#1}>-1 %
1953     \edef\x{%
1954       \noexpand\Test{%
1955         \noexpand\BigIntCalcInc\bigintcalcNum{#1}!%
1956       }{#2}%
1957     }%
1958     \x
1959   \fi
1960 }
1961 \newcommand*\TestDec}[2]{%
1962   \Test{\bigintcalcDec{#1}}{#2}%
1963   \ifnum\bigintcalcSgn{#1}>0 %
1964     \edef\x{%
1965       \noexpand\Test{%
1966         \noexpand\BigIntCalcDec\bigintcalcNum{#1}!%
1967       }{#2}%
1968     }%
1969     \x
1970   \fi
1971 }
1972 \newcommand*\TestAdd}[3]{%
1973   \Test{\bigintcalcAdd{#1}{#2}}{#3}%
1974   \ifnum\bigintcalcSgn{#1}>0 %
1975     \ifnum\bigintcalcSgn{#2}> 0 %
1976       \ifnum\bigintcalcCmp{#1}{#2}>0 %
1977         \edef\x{%
1978           \noexpand\Test{%
1979             \noexpand\BigIntCalcAdd
1980             \bigintcalcNum{#1}!\bigintcalcNum{#2}!%
1981           }{#3}%
1982         }%
1983         \x
1984       \else
1985         \edef\x{%
1986           \noexpand\Test{%
1987             \noexpand\BigIntCalcAdd
1988             \bigintcalcNum{#2}!\bigintcalcNum{#1}!%
1989           }{#3}%
1990         }%
1991         \x
1992       \fi
1993     \fi
1994   \fi
1995 }

```



```

1996 \newcommand*\TestSub}[3]{%
1997   \Test{\bigintcalcSub{#1}{#2}}{#3}%
1998   \ifnum\bigintcalcSgn{#1}>0 %
1999     \ifnum\bigintcalcSgn{#2}> 0 %
2000       \ifnum\bigintcalcCmp{#1}{#2}>0 %
2001         \edef\x{%
2002           \noexpand\Test{%
2003             \noexpand\BigIntCalcSub
2004               \bigintcalcNum{#1}!\bigintcalcNum{#2}!%
2005             }{#3}%
2006         }%
2007         \x
2008       \fi
2009     \fi
2010   \fi
2011 }
2012 \newcommand*\TestShl}[2]{%
2013   \Test{\bigintcalcShl{#1}}{#2}%
2014   \edef\x{%
2015     \noexpand\Test{%
2016       \noexpand\BigIntCalcShl\bigintcalcAbs{#1}!%
2017     }{\bigintcalcAbs{#2}}%
2018   }%
2019   \x
2020 }
2021 \newcommand*\TestShr}[2]{%
2022   \Test{\bigintcalcShr{#1}}{#2}%
2023   \edef\x{%
2024     \noexpand\Test{%
2025       \noexpand\BigIntCalcShr\bigintcalcAbs{#1}!%
2026     }{\bigintcalcAbs{#2}}%
2027   }%
2028   \x
2029 }
2030 \newcommand*\TestMul}[3]{%
2031   \Test{\bigintcalcMul{#1}{#2}}{#3}%
2032   \edef\x{%
2033     \noexpand\Test{%
2034       \noexpand\BigIntCalcMul
2035         \bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
2036     }{\bigintcalcAbs{#3}}%
2037   }%
2038   \x
2039 }
2040 \newcommand*\TestSqr}[2]{%
2041   \Test{\bigintcalcSqr{#1}}{#2}%
2042 }
2043 \newcommand*\TestFac}[2]{%
2044   \expandafter\TestExch\expandafter{%
2045     \the\numexpr#2%
2046   }{\bigintcalcFac{#1}}%
2047 }
2048 \newcommand*\TestFacBig}[2]{%
2049   \Test{\bigintcalcFac{#1}}{#2}%
2050 }
2051 \newcommand*\TestPow}[3]{%
2052   \Test{\bigintcalcPow{#1}{#2}}{#3}%
2053 }
2054 \newcommand*\TestDiv}[3]{%
2055   \Test{\bigintcalcDiv{#1}{#2}}{#3}%
2056   \TestTeXDivide{#1}{#2}%
2057 }

```

```

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

```

3.2.2 Time

```

2089 \begingroup\expandafter\expandafter\expandafter\endgroup
2090 \expandafter\ifx\csname pdfresettimer\endcsname\relax
2091 \else
2092 \makeatletter
2093 \newcount\SummaryTime
2094 \newcount\TestTime
2095 \SummaryTime=\z@
2096 \newcommand*\PrintTime}[2]{%
2097 \typeout{%
2098 [Time #1: \strip@pt\dimexpr\number#2sp\relax\space s]%
2099 }%
2100 }%
2101 \newcommand*\StartTime}[1]{%
2102 \renewcommand*\TimeDescription}{#1}%
2103 \pdfresettimer
2104 }%
2105 \newcommand*\TimeDescription}{%
2106 \newcommand*\StopTime}{%
2107 \TestTime=\pdfelapsedtime
2108 \global\advance\SummaryTime\TestTime
2109 \PrintTime\TimeDescription\TestTime
2110 }%
2111 \let\saved@qstest\qstest
2112 \let\saved@endqstest\endqstest
2113 \def\qstest#1#2{%
2114 \saved@qstest{#1}{#2}%
2115 \StartTime{#1}%
2116 }%
2117 \def\endqstest{%
2118 \StopTime

```

```

2119   \saved@endqstest
2120 }%
2121 \AtEndDocument{%
2122   \PrintTime{summary}\SummaryTime
2123 }%
2124 \makeatother
2125 \fi

```

3.2.3 Test sets

```

2126 \makeatletter
2127
2128 \begin{qstest}{inv}{inv}%
2129   \TestInv{0}{0}%
2130   \TestInv{1}{-1}%
2131   \TestInv{-1}{1}%
2132   \TestInv{10}{-10}%
2133   \TestInv{-10}{10}%
2134   \TestInv{2147483647}{-2147483647}%
2135   \TestInv{-2147483647}{2147483647}%
2136   \TestInv{12345678901234567890}{-12345678901234567890}%
2137   \TestInv{-12345678901234567890}{12345678901234567890}%
2138   \TestInv{ 0 }{0}%
2139   \TestInv{ 1 }{-1}%
2140   \TestInv{--1}{-1}%
2141   \TestInv{\number\z@}{0}%
2142   \TestInv{\ifx\relax\relax1\fi}{-1}%
2143   \TestInv{\ifx\relax\relax-\fi\ifx234\else1\fi}{1}%
2144 \end{qstest}
2145
2146 \begin{qstest}{abs}{abs}%
2147   \TestAbs{0}{0}%
2148   \TestAbs{1}{1}%
2149   \TestAbs{-1}{1}%
2150   \TestAbs{10}{10}%
2151   \TestAbs{-10}{10}%
2152   \TestAbs{2147483647}{2147483647}%
2153   \TestAbs{-2147483647}{2147483647}%
2154   \TestAbs{12345678901234567890}{12345678901234567890}%
2155   \TestAbs{-12345678901234567890}{12345678901234567890}%
2156   \TestAbs{ 0 }{0}%
2157   \TestAbs{ 1 }{1}%
2158   \TestAbs{--1}{1}%
2159   \TestAbs{---+1}{1}%
2160   \TestAbs{000000000000}{0}%
2161   \TestAbs{00000001000}{1000}%
2162   \TestAbs{\ifx\relax\relax 0\else 1\fi}{0}%
2163 \end{qstest}
2164
2165 \begin{qstest}{sign}{sign}%
2166   \TestSgn{0}{0}%
2167   \TestSgn{1}{1}%
2168   \TestSgn{-1}{-1}%
2169   \TestSgn{10}{1}%
2170   \TestSgn{-10}{-1}%
2171   \TestSgn{2147483647}{1}%
2172   \TestSgn{-2147483647}{-1}%
2173   \TestSgn{12345678901234567890}{1}%
2174   \TestSgn{-12345678901234567890}{-1}%
2175   \TestSgn{ 0 }{0}%
2176   \TestSgn{ 2 }{1}%
2177   \TestSgn{-2 }{-1}%
2178   \TestSgn{--2}{1}%
2179   \TestSgn{\number\z@}{0}%

```

```

2180 \TestSgn{\number\@ne}{1}%
2181 \TestSgn{\number\m@ne}{-1}%
2182 \TestSgn{%
2183   -+-+\number\z@\number\z@
2184   \iftrue1\fi\iftrue2\fi\iftrue3\fi
2185 }{1}%
2186 \end{qstest}
2187
2188 \begin{qstest}{min}{min}%
2189 \TestMin{0}{1}{0}%
2190 \TestMin{1}{0}{0}%
2191 \TestMin{-10}{-20}{-20}%
2192 \TestMin{ 1 }{ 2 }{1}%
2193 \TestMin{ 2 }{ 1 }{1}%
2194 \TestMin{1}{1}{1}%
2195 \TestMin{\number\z@}{\number\@ne}{0}%
2196 \TestMin{\number\@ne}{\number\m@ne}{-1}%
2197 \end{qstest}
2198
2199 \begin{qstest}{max}{max}%
2200 \TestMax{0}{1}{1}%
2201 \TestMax{1}{0}{1}%
2202 \TestMax{-10}{-20}{-10}%
2203 \TestMax{ 1 }{ 2 }{2}%
2204 \TestMax{ 2 }{ 1 }{2}%
2205 \TestMax{1}{1}{1}%
2206 \TestMax{\number\z@}{\number\@ne}{1}%
2207 \TestMax{\number\@ne}{\number\m@ne}{1}%
2208 \end{qstest}
2209
2210 \begin{qstest}{cmp}{cmp}%
2211 \TestCmp{0}{0}{0}%
2212 \TestCmp{-21}{17}{-1}%
2213 \TestCmp{3}{4}{-1}%
2214 \TestCmp{-10}{-10}{0}%
2215 \TestCmp{-10}{-11}{1}%
2216 \TestCmp{100}{5}{1}%
2217 \TestCmp{9}{10}{-1}%
2218 \TestCmp{10}{9}{1}%
2219 \TestCmp{ 3 }{ 3 }{0}%
2220 \TestCmp{-9}{-10}{1}%
2221 \TestCmp{-10}{-9}{-1}%
2222 \TestCmp{-3}{-3}{0}%
2223 \TestCmp{0}{-2}{1}%
2224 \TestCmp{0}{2}{-1}%
2225 \TestCmp{2}{0}{1}%
2226 \TestCmp{-2}{0}{-1}%
2227 \TestCmp{12}{11}{1}%
2228 \TestCmp{11}{12}{-1}%
2229 \TestCmp{2147483647}{-2147483647}{1}%
2230 \TestCmp{-2147483647}{2147483647}{-1}%
2231 \TestCmp{2147483647}{2147483647}{0}%
2232 \TestCmp{\number\z@}{\number\@ne}{-1}%
2233 \TestCmp{\number\@ne}{\number\m@ne}{1}%
2234 \TestCmp{ 4 }{ 5 }{-1}%
2235 \TestCmp{ -3 }{ -7 }{1}%
2236 \end{qstest}
2237
2238 \begin{qstest}{odd}{odd}
2239 \tracingmacros=1
2240 \TestOdd{0}{0}%
2241 \TestOdd{1}{1}%

```


2304 \TestAdd{ -4 }{ 1 }{-3}%
2305 \TestAdd{ -1 }{ 4 }{3}%
2306 \TestAdd{ 4 }{ -1 }{3}%
2307 \TestAdd{ 1 }{ -4 }{-3}%
2308 \TestAdd{ -4 }{ -1 }{-5}%
2309 \TestAdd{ -1 }{ -4 }{-5}%
2310 \TestAdd{876543210}{111111111}{987654321}%
2311 \TestAdd{999999999}{2}{100000001}%
2312 \end{qstest}
2313
2314 \begin{qstest}{sub}{sub}
2315 \TestSub{0}{0}{0}%
2316 \TestSub{1}{0}{1}%
2317 \TestSub{1}{2}{-1}%
2318 \TestSub{-1}{-1}{0}%
2319 \TestSub{2147483646}{-1}{2147483647}%
2320 \TestSub{-2147483647}{-2147483647}{0}%
2321 \TestSub{-4}{-1}{-3}%
2322 \TestSub{-1}{-4}{3}%
2323 \TestSub{-4}{1}{-5}%
2324 \TestSub{-1}{4}{-5}%
2325 \TestSub{4}{-1}{5}%
2326 \TestSub{1}{-4}{5}%
2327 \TestSub{-4}{-1}{-3}%
2328 \TestSub{-1}{-4}{3}%
2329 \TestSub{ -4 }{ -1 }{-3}%
2330 \TestSub{ -1 }{ -4 }{3}%
2331 \TestSub{ -4 }{ 1 }{-5}%
2332 \TestSub{ -1 }{ 4 }{-5}%
2333 \TestSub{ 4 }{ -1 }{5}%
2334 \TestSub{ 1 }{ -4 }{5}%
2335 \TestSub{ -4 }{ -1 }{-3}%
2336 \TestSub{ -1 }{ -4 }{3}%
2337 \TestSub{100000000}{2}{999999998}%
2338 \TestSub{987654321}{111111111}{876543210}%
2339 \end{qstest}
2340
2341 \begin{qstest}{shl}{shl}
2342 \TestShl{0}{0}%
2343 \TestShl{1}{2}%
2344 \TestShl{2}{4}%
2345 \TestShl{5621}{11242}%
2346 \TestShl{1073741823}{2147483646}%
2347 \end{qstest}
2348
2349 \begin{qstest}{shr}{shr}
2350 \TestShr{0}{0}%
2351 \TestShr{1}{0}%
2352 \TestShr{2}{1}%
2353 \TestShr{3}{1}%
2354 \TestShr{4}{2}%
2355 \TestShr{5}{2}%
2356 \TestShr{6}{3}%
2357 \TestShr{7}{3}%
2358 \TestShr{8}{4}%
2359 \TestShr{9}{4}%
2360 \TestShr{10}{5}%
2361 \TestShr{11}{5}%
2362 \TestShr{12}{6}%
2363 \TestShr{13}{6}%
2364 \TestShr{14}{7}%
2365 \TestShr{15}{7}%

2366 \TestShr{16}{8}%
2367 \TestShr{17}{8}%
2368 \TestShr{18}{9}%
2369 \TestShr{19}{9}%
2370 \TestShr{20}{10}%
2371 \TestShr{21}{10}%
2372 \TestShr{22}{11}%
2373 \TestShr{11241}{5620}%
2374 \TestShr{73054202}{36527101}%
2375 \TestShr{2147483646}{1073741823}%
2376 \end{qstest}
2377
2378 \begin{qstest}{mul}{mul}
2379 \TestMul{0}{0}{0}%
2380 \TestMul{1}{0}{0}%
2381 \TestMul{0}{1}{0}%
2382 \TestMul{1}{1}{1}%
2383 \TestMul{3}{1}{3}%
2384 \TestMul{1}{-3}{-3}%
2385 \TestMul{-4}{-5}{20}%
2386 \TestMul{3}{7}{21}%
2387 \TestMul{7}{3}{21}%
2388 \TestMul{3}{-7}{-21}%
2389 \TestMul{7}{-3}{-21}%
2390 \TestMul{-3}{7}{-21}%
2391 \TestMul{-7}{3}{-21}%
2392 \TestMul{-3}{-7}{21}%
2393 \TestMul{-7}{-3}{21}%
2394 \TestMul{12}{11}{132}%
2395 \TestMul{999}{333}{332667}%
2396 \TestMul{1000}{4321}{4321000}%
2397 \TestMul{12345}{173955}{2147474475}%
2398 \TestMul{1073741823}{2}{2147483646}%
2399 \TestMul{2}{1073741823}{2147483646}%
2400 \TestMul{-1073741823}{2}{-2147483646}%
2401 \TestMul{2}{-1073741823}{-2147483646}%
2402 \TestMul{6706022400}{13}{87178291200}%
2403 \end{qstest}
2404
2405 \begin{qstest}{sqr}{sqr}
2406 \TestSqr{0}{0}%
2407 \TestSqr{1}{1}%
2408 \TestSqr{2}{4}%
2409 \TestSqr{3}{9}%
2410 \TestSqr{4}{16}%
2411 \TestSqr{9}{81}%
2412 \TestSqr{10}{100}%
2413 \TestSqr{46340}{2147395600}%
2414 \TestSqr{-1}{1}%
2415 \TestSqr{-2}{4}%
2416 \TestSqr{-46340}{2147395600}%
2417 \end{qstest}
2418
2419 \begin{qstest}{fac}{fac}
2420 \TestFac{0}{1}%
2421 \TestFac{1}{1}%
2422 \TestFac{2}{2}%
2423 \TestFac{3}{2*3}%
2424 \TestFac{4}{2*3*4}%
2425 \TestFac{5}{2*3*4*5}%
2426 \TestFac{6}{2*3*4*5*6}%
2427 \TestFac{7}{2*3*4*5*6*7}%

2428 $\backslash\text{TestFac}\{8\}\{2*3*4*5*6*7*8\}\%$
2429 $\backslash\text{TestFac}\{9\}\{2*3*4*5*6*7*8*9\}\%$
2430 $\backslash\text{TestFac}\{10\}\{2*3*4*5*6*7*8*9*10\}\%$
2431 $\backslash\text{TestFac}\{11\}\{2*3*4*5*6*7*8*9*10*11\}\%$
2432 $\backslash\text{TestFac}\{12\}\{2*3*4*5*6*7*8*9*10*11*12\}\%$
2433 $\backslash\text{TestFacBig}\{13\}\{6227020800\}\%$
2434 $\backslash\text{TestFacBig}\{14\}\{87178291200\}\%$
2435 $\backslash\text{TestFacBig}\{15\}\{1307674368000\}\%$
2436 $\backslash\text{TestFacBig}\{16\}\{20922789888000\}\%$
2437 $\backslash\text{TestFacBig}\{17\}\{355687428096000\}\%$
2438 $\backslash\text{TestFacBig}\{18\}\{6402373705728000\}\%$
2439 $\backslash\text{TestFacBig}\{19\}\{121645100408832000\}\%$
2440 $\backslash\text{TestFacBig}\{20\}\{2432902008176640000\}\%$
2441 $\backslash\text{TestFacBig}\{21\}\{51090942171709440000\}\%$
2442 $\backslash\text{TestFacBig}\{22\}\{112400072777607680000\}\%$
2443 $\backslash\text{end}\{qstest\}$
2444
2445 $\backslash\text{begin}\{qstest\}\{pow\}\{pow\}$
2446 $\backslash\text{TestPow}\{-2\}\{0\}\{1\}\%$
2447 $\backslash\text{TestPow}\{-1\}\{0\}\{1\}\%$
2448 $\backslash\text{TestPow}\{0\}\{0\}\{1\}\%$
2449 $\backslash\text{TestPow}\{1\}\{0\}\{1\}\%$
2450 $\backslash\text{TestPow}\{2\}\{0\}\{1\}\%$
2451 $\backslash\text{TestPow}\{3\}\{0\}\{1\}\%$
2452 $\backslash\text{TestPow}\{-2\}\{1\}\{-2\}\%$
2453 $\backslash\text{TestPow}\{-1\}\{1\}\{-1\}\%$
2454 $\backslash\text{TestPow}\{1\}\{1\}\{1\}\%$
2455 $\backslash\text{TestPow}\{2\}\{1\}\{2\}\%$
2456 $\backslash\text{TestPow}\{3\}\{1\}\{3\}\%$
2457 $\backslash\text{TestPow}\{-2\}\{2\}\{4\}\%$
2458 $\backslash\text{TestPow}\{-1\}\{2\}\{1\}\%$
2459 $\backslash\text{TestPow}\{0\}\{2\}\{0\}\%$
2460 $\backslash\text{TestPow}\{1\}\{2\}\{1\}\%$
2461 $\backslash\text{TestPow}\{2\}\{2\}\{4\}\%$
2462 $\backslash\text{TestPow}\{3\}\{2\}\{9\}\%$
2463 $\backslash\text{TestPow}\{0\}\{1\}\{0\}\%$
2464 $\backslash\text{TestPow}\{1\}\{-2\}\{1\}\%$
2465 $\backslash\text{TestPow}\{1\}\{-1\}\{1\}\%$
2466 $\backslash\text{TestPow}\{-1\}\{-2\}\{1\}\%$
2467 $\backslash\text{TestPow}\{-1\}\{-1\}\{-1\}\%$
2468 $\backslash\text{TestPow}\{-1\}\{3\}\{-1\}\%$
2469 $\backslash\text{TestPow}\{-1\}\{4\}\{1\}\%$
2470 $\backslash\text{TestPow}\{-2\}\{-1\}\{0\}\%$
2471 $\backslash\text{TestPow}\{-2\}\{-2\}\{0\}\%$
2472 $\backslash\text{TestPow}\{2\}\{3\}\{8\}\%$
2473 $\backslash\text{TestPow}\{2\}\{4\}\{16\}\%$
2474 $\backslash\text{TestPow}\{2\}\{5\}\{32\}\%$
2475 $\backslash\text{TestPow}\{2\}\{6\}\{64\}\%$
2476 $\backslash\text{TestPow}\{2\}\{7\}\{128\}\%$
2477 $\backslash\text{TestPow}\{2\}\{8\}\{256\}\%$
2478 $\backslash\text{TestPow}\{2\}\{9\}\{512\}\%$
2479 $\backslash\text{TestPow}\{2\}\{10\}\{1024\}\%$
2480 $\backslash\text{TestPow}\{-2\}\{3\}\{-8\}\%$
2481 $\backslash\text{TestPow}\{-2\}\{4\}\{-16\}\%$
2482 $\backslash\text{TestPow}\{-2\}\{5\}\{-32\}\%$
2483 $\backslash\text{TestPow}\{-2\}\{6\}\{-64\}\%$
2484 $\backslash\text{TestPow}\{-2\}\{7\}\{-128\}\%$
2485 $\backslash\text{TestPow}\{-2\}\{8\}\{-256\}\%$
2486 $\backslash\text{TestPow}\{-2\}\{9\}\{-512\}\%$
2487 $\backslash\text{TestPow}\{-2\}\{10\}\{-1024\}\%$
2488 $\backslash\text{TestPow}\{3\}\{3\}\{27\}\%$
2489 $\backslash\text{TestPow}\{3\}\{4\}\{81\}\%$


```

2490 \TestPow{3}{5}{243}%
2491 \TestPow{-3}{3}{-27}%
2492 \TestPow{-3}{4}{81}%
2493 \TestPow{-3}{5}{-243}%
2494 \TestPow{2}{30}{1073741824}%
2495 \TestPow{-3}{19}{-1162261467}%
2496 \TestPow{5}{13}{1220703125}%
2497 \TestPow{-7}{11}{-1977326743}%
2498 \end{qstest}
2499
2500 \begin{qstest}{div}{div}
2501 \TestDiv{1}{1}{1}%
2502 \TestDiv{2}{1}{2}%
2503 \TestDiv{-2}{1}{-2}%
2504 \TestDiv{2}{-1}{-2}%
2505 \TestDiv{-2}{-1}{2}%
2506 \TestDiv{15}{2}{7}%
2507 \TestDiv{-16}{2}{-8}%
2508 \TestDiv{1}{2}{0}%
2509 \TestDiv{1}{3}{0}%
2510 \TestDiv{2}{3}{0}%
2511 \TestDiv{-2}{3}{0}%
2512 \TestDiv{2}{-3}{0}%
2513 \TestDiv{-2}{-3}{0}%
2514 \TestDiv{13}{3}{4}%
2515 \TestDiv{-13}{-3}{4}%
2516 \TestDiv{-13}{3}{-4}%
2517 \TestDiv{-6}{5}{-1}%
2518 \TestDiv{-5}{5}{-1}%
2519 \TestDiv{-4}{5}{0}%
2520 \TestDiv{-3}{5}{0}%
2521 \TestDiv{-2}{5}{0}%
2522 \TestDiv{-1}{5}{0}%
2523 \TestDiv{0}{5}{0}%
2524 \TestDiv{1}{5}{0}%
2525 \TestDiv{2}{5}{0}%
2526 \TestDiv{3}{5}{0}%
2527 \TestDiv{4}{5}{0}%
2528 \TestDiv{5}{5}{1}%
2529 \TestDiv{6}{5}{1}%
2530 \TestDiv{-5}{4}{-1}%
2531 \TestDiv{-4}{4}{-1}%
2532 \TestDiv{-3}{4}{0}%
2533 \TestDiv{-2}{4}{0}%
2534 \TestDiv{-1}{4}{0}%
2535 \TestDiv{0}{4}{0}%
2536 \TestDiv{1}{4}{0}%
2537 \TestDiv{2}{4}{0}%
2538 \TestDiv{3}{4}{0}%
2539 \TestDiv{4}{4}{1}%
2540 \TestDiv{5}{4}{1}%
2541 \TestDiv{12345}{678}{18}%
2542 \TestDiv{32372}{5952}{5}%
2543 \TestDiv{284271294}{18162}{15651}%
2544 \TestDiv{217652429}{12561}{17327}%
2545 \TestDiv{462028434}{5439}{84947}%
2546 \TestDiv{2147483647}{1000}{2147483}%
2547 \TestDiv{2147483647}{-1000}{-2147483}%
2548 \TestDiv{-2147483647}{1000}{-2147483}%
2549 \TestDiv{-2147483647}{-1000}{2147483}%
2550 \TestDiv{0}{3}{0}%
2551 \TestDiv{1}{3}{0}%

```

```

2552 \TestDiv{2}{3}{0}%
2553 \TestDiv{3}{3}{1}%
2554 \TestDiv{4}{3}{1}%
2555 \TestDiv{5}{3}{1}%
2556 \TestDiv{6}{3}{2}%
2557 \TestDiv{7}{3}{2}%
2558 \TestDiv{8}{3}{2}%
2559 \TestDiv{9}{3}{3}%
2560 \TestDiv{10}{3}{3}%
2561 \TestDiv{11}{3}{3}%
2562 \TestDiv{12}{3}{4}%
2563 \TestDiv{13}{3}{4}%
2564 \TestDiv{14}{3}{4}%
2565 \TestDiv{15}{3}{5}%
2566 \TestDiv{16}{3}{5}%
2567 \TestDiv{17}{3}{5}%
2568 \TestDiv{18}{3}{6}%
2569 \TestDiv{19}{3}{6}%
2570 \TestDiv{20}{3}{6}%
2571 \TestDiv{21}{3}{7}%
2572 \TestDiv{22}{3}{7}%
2573 \TestDiv{23}{3}{7}%
2574 \TestDiv{24}{3}{8}%
2575 \TestDiv{25}{3}{8}%
2576 \TestDiv{26}{3}{8}%
2577 \TestDiv{27}{3}{9}%
2578 \TestDiv{28}{3}{9}%
2579 \TestDiv{29}{3}{9}%
2580 \TestDiv{30}{3}{10}%
2581 \TestDiv{31}{3}{10}%
2582 \TestDivBig{17363436332507}{24702}{702916214}%
2583 \end{qstest}
2584
2585 \begin{qstest}{mod}{mod}
2586 \TestMod{-6}{5}{4}%
2587 \TestMod{-5}{5}{0}%
2588 \TestMod{-4}{5}{1}%
2589 \TestMod{-3}{5}{2}%
2590 \TestMod{-2}{5}{3}%
2591 \TestMod{-1}{5}{4}%
2592 \TestMod{0}{5}{0}%
2593 \TestMod{1}{5}{1}%
2594 \TestMod{2}{5}{2}%
2595 \TestMod{3}{5}{3}%
2596 \TestMod{4}{5}{4}%
2597 \TestMod{5}{5}{0}%
2598 \TestMod{6}{5}{1}%
2599 \TestMod{-5}{4}{3}%
2600 \TestMod{-4}{4}{0}%
2601 \TestMod{-3}{4}{1}%
2602 \TestMod{-2}{4}{2}%
2603 \TestMod{-1}{4}{3}%
2604 \TestMod{0}{4}{0}%
2605 \TestMod{1}{4}{1}%
2606 \TestMod{2}{4}{2}%
2607 \TestMod{3}{4}{3}%
2608 \TestMod{4}{4}{0}%
2609 \TestMod{5}{4}{1}%
2610 \TestMod{-6}{-5}{-1}%
2611 \TestMod{-5}{-5}{0}%
2612 \TestMod{-4}{-5}{-4}%
2613 \TestMod{-3}{-5}{-3}%

```

```

2614 \TestMod{-2}{-5}{-2}%
2615 \TestMod{-1}{-5}{-1}%
2616 \TestMod{0}{-5}{0}%
2617 \TestMod{1}{-5}{-4}%
2618 \TestMod{2}{-5}{-3}%
2619 \TestMod{3}{-5}{-2}%
2620 \TestMod{4}{-5}{-1}%
2621 \TestMod{5}{-5}{0}%
2622 \TestMod{6}{-5}{-4}%
2623 \TestMod{-5}{-4}{-1}%
2624 \TestMod{-4}{-4}{0}%
2625 \TestMod{-3}{-4}{-3}%
2626 \TestMod{-2}{-4}{-2}%
2627 \TestMod{-1}{-4}{-1}%
2628 \TestMod{0}{-4}{0}%
2629 \TestMod{1}{-4}{-3}%
2630 \TestMod{2}{-4}{-2}%
2631 \TestMod{3}{-4}{-1}%
2632 \TestMod{4}{-4}{0}%
2633 \TestMod{5}{-4}{-3}%
2634 \TestMod{2147483647}{1000}{647}%
2635 \TestMod{2147483647}{-1000}{-353}%
2636 \TestMod{-2147483647}{1000}{353}%
2637 \TestMod{-2147483647}{-1000}{-647}%
2638 \TestMod{ 0 }{ 4 }{0}%
2639 \TestMod{ 1 }{ 4 }{1}%
2640 \TestMod{ -1 }{ 4 }{3}%
2641 \TestMod{ 0 }{ -4 }{0}%
2642 \TestMod{ 1 }{ -4 }{-3}%
2643 \TestMod{ -1 }{ -4 }{-1}%
2644 \TestMod{18362}{25}{12}%
2645 \end{qstest}
2646
2647 \newcommand*{\TestError}[2]{%
2648 \begingroup
2649 \expandafter\def\csname BigIntCalcError:#1\endcsname{%
2650 \Expect*{#2}{0}%
2651 \expandafter\def\csname BigIntCalcError:#1\endcsname{ERROR}%
2652 \Expect*{#2}{0ERROR}%
2653 \endgroup
2654 }
2655 \begin{qstest}{error}{error}
2656 \TestError{FacNegative}{\bigintcalcFac{-1}}%
2657 \TestError{FacNegative}{\bigintcalcFac{-2147483647}}%
2658 \TestError{DivisionByZero}{\bigintcalcPow{0}{-1}}%
2659 \TestError{DivisionByZero}{\bigintcalcDiv{1}{0}}%
2660 \TestError{DivisionByZero}{\bigintcalcMod{1}{0}}%
2661 \end{qstest}
2662
2663 \begin{document}
2664 \end{document}
2665 </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.

¹<http://ftp.ctan.org/tex-archive/>

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 \TeX 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 \TeX :

```
tex bigintcalc.dtx
```

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

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

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

4.4 Refresh file name databases

If your \TeX distribution (`te \TeX` , `mik \TeX` , ...) relies on file name databases, you must refresh these. For example, `te \TeX` users run `texhash` or `mktextlsr`.

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 pdftexcmds for LuaT_EX support.

[2011/01/30 v1.2]

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

6 Index

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

Symbols		
\# 1773	462, 495, 508, 520, 555, 557, 646, 647, 657, 658, 675, 737,
\% 1849	738, 748, 749, 766, 851, 862,
\@ 1774, 1847	909, 923, 986, 997, 1014, 1134,
\@firstofone 1782, 1785	1220, 1245, 1273, 1321, 1342,
\@firstoftwo 158, 166, <u>175</u>	1352, 1446, 1450, 1478, 1524,
\@gobble 1779, 1787	1671, 1682, 1690, 1712, 1728, 1848
\@ne 2180,	\{ 1771
	2195, 2196, 2206, 2207, 2232, 2233	\} 1772
\@nil	. 154, 156, 163, 171, 186, 189, 194	
\@secondoftwo 160, 168, <u>178</u>	
\@undefined 58	
\\	191, 201, 320, 321, 329, 352, 450,	

A

\advance 1812, 1820, 1835, 2108
\aftergroup 29

<code>\AtEndDocument</code>	2121	925, 932, 941, 948, 1034, 1039, 1079, 1104, 1188, 1236, 1240, 1258, 1260, 1269, 1289, 1291, 1312, 1323, 1325, 1354, 1358, 1364, 1416, 1418, 1422, 1428, 1430, 1434, 1459, 1461, 1465, 1470, 1514, 1516, 1522, 1544, 1546, 1568, 1591, 1593, 1599, 1601, 1607, 1609, 1618, 1620, 1641, 1648, 1683, 1685, 1699, 1701, 1705, 1722, 1724, 1734, 1741
B		
<code>\begin</code>	2128, 2146, 2165, 2188, 2199, 2210, 2238, 2250, 2268, 2285, 2314, 2341, 2349, 2378, 2405, 2419, 2445, 2500, 2585, 2655, 2663	
<code>\BIC@@@Shl</code>	869, <u>873</u>	
<code>\BIC@@@Shr</code> ...	942, 943, 949, 950, <u>959</u>	
<code>\BIC@@@Dec</code>	530, 531, 549, <u>554</u>	
<code>\BIC@@@Inc</code>	472, 473, 489, <u>494</u>	
<code>\BIC@@@PowRec</code>	1372, 1378, <u>1393</u>	<code>\BIC@AfterFiFiFi</code> <u>133</u> , 217, 221, 353, 357, 558, 560, 590, 594, 600, 603, 607, 615, 617, 622, 628, 632, 659, 663, 750, 754, 1082, 1086, 1093, 1097, 1108, 1112, 1118, 1122, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1263, 1265, 1294, 1296, 1301, 1306, 1371, 1377, 1525, 1529, 1553, 1555, 1561, 1563, 1634, 1636
<code>\BIC@@@ProcessDiv</code>	1577, 1615	<code>\BIC@AtEnd</code>
<code>\BIC@@@Shl</code>	853, <u>861</u> , 876, 880	95, 96, 117, 1768
<code>\BIC@@@Shr</code>	913, 918, <u>922</u> , 960	<code>\BIC@Cmp</code>
<code>\BIC@@AddDigit</code>	698, 699, <u>709</u> , 788	286, <u>289</u>
<code>\BIC@@Cmp</code>	290, <u>293</u> , 379, 411	<code>\BIC@CmpDiff</code>
<code>\BIC@@Dec</code>	510, <u>518</u> , 567	322, <u>341</u>
<code>\BIC@@Expand</code>	154, <u>156</u>	<code>\BIC@CmpLength</code>
<code>\BIC@@Inc</code>	452, <u>460</u> , 503	311, 317, <u>319</u>
<code>\BIC@MinMax</code>	375, <u>378</u>	<code>\BIC@CmpResult</code>
<code>\BIC@MinusOne</code>	1335, <u>1340</u>	325, 331, <u>340</u>
<code>\BIC@PowRec</code>	1365, <u>1388</u> , 1394	<code>\BIC@Dec</code>
<code>\BIC@ProcessDiv</code>	1536, <u>1541</u>	420, 434, 447, <u>507</u> , 1207
<code>\BIC@ProcessFac</code>	1199, <u>1205</u>	<code>\BIC@DecSwitch</code>
<code>\BIC@ProcessTim</code>	1006, <u>1013</u>	426, <u>429</u>
<code>\BIC@Shl</code>	835, 840, 847, <u>850</u>	<code>\BIC@Div</code>
<code>\BIC@Shr</code>	893, 897, 903, 905, 1366, 1395, 1462, 1466, 1467	1400, <u>1403</u>
<code>\BIC@Sqr</code>	1153, 1155, <u>1158</u>	<code>\BIC@DivCleanup</code>
<code>\BIC@SubDigit</code>	787, <u>798</u>	. 1514, 1522, <u>1540</u> , 1544, 1553, 1561, 1591, 1599, 1607, 1618, 1634
<code>\BIC@TestMode</code>	123	<code>\BIC@DivStart</code>
<code>\BIC@Tim</code>	985	1504, <u>1508</u>
<code>\BIC@Abs</code>	257, <u>260</u>	<code>\BIC@DivStartX</code>
<code>\BIC@Add</code>	573, <u>576</u> , 582	1471, 1477, 1484
<code>\BIC@AddCarry0</code>	716	<code>\BIC@DivStartYii</code>
<code>\BIC@AddCarry10</code>	717	1480, <u>1488</u>
<code>\BIC@AddCarry[1-9]</code>	718	<code>\BIC@DivStartYiv</code>
<code>\BIC@AddDigit</code>	680, 685, <u>696</u>	1489, <u>1493</u>
<code>\BIC@AddResult</code>	679, <u>689</u>	<code>\BIC@DivStartYvi</code>
<code>\BIC@AddSwitch</code>	578, <u>585</u>	1494, <u>1498</u>
<code>\BIC@AddXY</code>	591, 595, 629, 633, 640, <u>645</u> , 836, 841, 848, 1042, 1500	<code>\BIC@DivStartYviii</code>
<code>\BIC@AfterFi</code>	131, 193, 230, 343, 380, 382, 412, 414, 418, 431, 433, 437, 451, 455, 502, 509, 513, 566, 676, 683, 711, 713, 767, 772, 800, 805, 833, 839, 852, 856, 863, 868, 875, 879, 892, 896, 907, 917, 987, 991, 998, 1005, 1015, 1019, 1026, 1028, 1076, 1135, 1140, 1168, 1196, 1198, 1231, 1233, 1328, 1413, 1442, 1444, 1479, 1483, 1535, 1576, 1625, 1680, 1719, 1753, 1755, 1760, 1765	1499, <u>1503</u>
<code>\BIC@AfterFiFi</code> <u>132</u> , 202, 206, 212, 233, 237, 296, 300, 306, 310, 322, 324, 330, 334, 348, 397, 399, 463, 465, 471, 488, 497, 499, 521, 523, 529, 548, 563, 648, 652, 668, 739, 743, 759, 910, 912,	<code>\BIC@DivSub</code>
		1564, <u>1581</u> , 1610
		<code>\BIC@DivSwitch</code>
	 1419, 1423, 1431, 1435, <u>1440</u>
		<code>\BIC@DivSwitchSign</code> .. 1404, 1409, <u>1411</u>
		<code>\BIC@DoAdd</code>
		649, 653, <u>674</u>
		<code>\BIC@DoSub</code>
		740, 744, <u>765</u>
		<code>\BIC@Expand</code>
		<u>152</u> , <u>182</u> , 249
		<code>\BIC@Fac</code>
		1163, <u>1166</u>
		<code>\BIC@Fi</code> ... <u>130</u> , 131, 132, 133, 196, 226, 241, 314, 338, 362, 383, 403, 422, 441, 458, 492, 505, 516, 552, 569, 637, 672, 687, 714, 763, 776, 806, 843, 859, 871, 882, 899, 920, 955, 994, 1011, 1022, 1047, 1127, 1144, 1192, 1203, 1318, 1331, 1386, 1438, 1475, 1486, 1538, 1579, 1613, 1655, 1709, 1749, 1756, 1766
		<code>\BIC@Inc</code>
		415, 439, 444, <u>449</u>
		<code>\BIC@IncSwitch</code>
		407, <u>410</u>
		<code>\BIC@MinMax</code>
		366, 371, <u>374</u>
		<code>\BIC@MinusOne</code>
		1251, 1280, <u>1333</u>
		<code>\BIC@Mod</code>
		1659, <u>1662</u>

<code>\BIC@ModMinus</code>	1731, 1751	<code>\BIC@Tim</code>	985, 1137, 1142
<code>\BIC@ModSwitch</code>	1686, 1702, 1706, 1711	<code>\BIC@TimDigit</code>	1001, 1008, 1024
<code>\BIC@ModSwitchSign</code>	1663, 1668, 1670	<code>\bigintcalcAbs</code>	4, 255 , 388, 1926, 1945, 2016, 2017, 2025, 2026, 2035, 2036, 2062, 2063, 2083, 2084
<code>\BIC@ModTwo</code>	1262, 1293, 1300, 1320 , 1363, 1721	<code>\BigIntCalcAdd</code>	7, 639 , 1979, 1987
<code>\BIC@ModX</code>	1735, 1758	<code>\bigintcalcAdd</code>	5, 571 , 1137, 1142, 1973
<code>\BIC@Mul</code>	1067, 1070	<code>\bigintcalcCmp</code>	4, 284 , 1938, 1976, 2000
<code>\BIC@MulDigit[3-9]</code>	1049	<code>\BigIntCalcDec</code>	7, 446 , 1966
<code>\BIC@MulSwitch</code>	1071, 1074	<code>\bigintcalcDec</code>	5, 424 , 1962
<code>\BIC@Normalize</code>	199, 246	<code>\BigIntCalcDiv</code>	7, 1407 , 2062
<code>\BIC@NormalizeDigits</code>	222, 238, 243	<code>\bigintcalcDiv</code>	6, 1398 , 1737, 1744, 1914, 2055, 2059, 2659
<code>\BIC@NormalizeZero</code>	218, 228	<code>\BigIntCalcError</code>	483, 533, 543, 722, 813, 1054, 1168, 1185, 1255, 1258, 1266, 1270, 1297, 1309, 1315, 1317, 1383, 1385, 1413, 1473, 1680, 1725, 1748
<code>\BIC@Odd</code>	387, 392, 394	<code>\bigintcalcFac</code>	6, 1161 , 2046, 2049, 2656, 2657
<code>\BIC@PosCmp</code>	316, 589, 599, 614, 627, 1081, 1092, 1107, 1117, 1170, 1195, 1353, 1370, 1441, 1512, 1542, 1551, 1589, 1616, 1632	<code>\BigIntCalcInc</code>	7, 443 , 1955
<code>\BIC@Pow</code>	1212, 1215	<code>\bigintcalcInc</code>	5, 405 , 1951
<code>\BIC@PowRec</code>	1302, 1307, 1313, 1351 , 1389	<code>\bigintcalcInv</code>	3, 251 , 1923
<code>\BIC@PowSwitch</code>	1216, 1219	<code>\bigintcalcMax</code>	4, 369 , 1935
<code>\BIC@ProcessDiv</code>	1477, 1509, 1511 , 1547, 1556, 1582, 1594, 1602, 1621, 1637	<code>\bigintcalcMin</code>	4, 364 , 1932
<code>\BIC@ProcessDivII</code>	1569, 1588 , 1626, 1642, 1649	<code>\BigIntCalcMod</code>	7, 1666 , 2082
<code>\BIC@ProcessDivIV</code>	1615	<code>\bigintcalcMod</code>	6, 1657 , 2067, 2660
<code>\BIC@ProcessFac</code>	1189, 1194 , 1206	<code>\BigIntCalcMul</code>	7, 1129 , 2034
<code>\BIC@ProcessMul</code>	1083, 1087, 1094, 1098, 1109, 1113, 1119, 1123, 1131, 1133 , 1159, 1200, 1237, 1241, 1355, 1359, 1373, 1379, 1390	<code>\bigintcalcMul</code>	6, 1065 , 1737, 1744, 2031
<code>\BIC@ProcessTim</code>	988, 996 , 1016, 1020	<code>\bigintcalcNum</code>	3, 244 , 253, 258, 271, 287, 291, 367, 372, 376, 408, 427, 574, 578, 583, 829, 887, 1068, 1072, 1149, 1164, 1213, 1217, 1401, 1405, 1660, 1664, 1955, 1966, 1980, 1988, 2004
<code>\BIC@Sgn</code>	270, 273 , 430, 1075, 1078, 1103, 1412, 1415, 1427	<code>\BigIntCalcOdd</code>	7, 390 , 1944
<code>\BIC@Shl</code>	828, 831 , 1490, 1495, 1505	<code>\bigintcalcOdd</code>	5, 385 , 1941
<code>\BIC@Shr</code>	886, 889	<code>\bigintcalcPow</code>	6, 1210 , 2052, 2658
<code>\BIC@ShrDigit[00-19]</code>	962	<code>\bigintcalcSgn</code>	4, 268 , 1929, 1952, 1963, 1974, 1975, 1998, 1999, 2068, 2070, 2075
<code>\BIC@ShrResult</code>	926, 927, 933, 934, 957	<code>\BigIntCalcShl</code>	7, 845 , 2016
<code>\BIC@Space</code>	134, 183, 252, 262, 264, 677, 691, 693, 782, 801, 864, 999, 1029, 1035, 1040, 1136, 1730, 1742, 1761	<code>\bigintcalcShl</code>	5, 826 , 2013
<code>\BIC@Sqr</code>	1148, 1151	<code>\BigIntCalcShr</code>	7, 901 , 2025
<code>\BIC@StripHexSpace</code>	186, 189	<code>\bigintcalcShr</code>	5, 884 , 2022
<code>\BIC@SubCarry0</code>	808	<code>\bigintcalcSqr</code>	6, 1146 , 2041
<code>\BIC@SubCarry10</code>	809	<code>\BigIntCalcSub</code>	7, 642 , 2003
<code>\BIC@SubCarry[1-9]</code>	810	<code>\bigintcalcSub</code>	5, 580 , 1736, 1743, 1997
<code>\BIC@SubDigit</code>	769, 774, 785	<code>\body</code>	1791, 1795
<code>\BIC@SubResult</code>	768, 778		
<code>\BIC@SubXY</code>	604, 608, 618, 623, 643, 736 , 1570, 1584, 1627, 1643, 1650, 1762		
<code>\BIC@Temp</code>	718, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 810, 817, 818, 819, 820, 821, 822, 823, 824, 825, 962, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 1049, 1058, 1059, 1060, 1061, 1062, 1063, 1064		
<code>\BIC@TestMode</code>	123, 1876		

C

<code>\catcode</code>	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 121, 126, 128, 1771, 1772, 1773, 1774, 1809, 1818, 1826, 1830, 1847, 1848, 1849
<code>\chardef</code>	1876
<code>\count@</code>	1776, 1805, 1809, 1811, 1812, 1816, 1818, 1819, 1820, 1824, 1826, 1829, 1830, 1834, 1835

<code>\countdef</code>	1776	749, 766, 779, 832, 851, 862,	
<code>\csname</code>	14, 21,	890, 909, 923, 986, 997, 1014,	
	50, 66, 76, 119, 125, 140, 143,	1134, 1152, 1167, 1220, 1221,	
	151, 175, 178, 700, 716, 717,	1222, 1223, 1227, 1235, 1245,	
	719, 789, 808, 809, 811, 928,	1246, 1247, 1273, 1274, 1275,	
	935, 944, 951, 963, 1030, 1036,	1279, 1321, 1334, 1341, 1342,	
	1043, 1050, 1775, 1778, 1781,	1352, 1446, 1450, 1451, 1452,	
	1784, 1839, 1866, 2090, 2649, 2651	1453, 1478, 1513, 1521, 1524,	
		1543, 1552, 1560, 1590, 1598,	
		1606, 1617, 1633, 1671, 1672,	
		1676, 1682, 1690, 1691, 1695,	
		1712, 1713, 1714, 1728, 1733,	
		1752, 1759, 1775, 1778, 1781,	
		1784, 1839, 2090, 2142, 2143, 2162	
D			
<code>\dimexpr</code>	2098		
<code>\divide</code>	1913		
<code>\documentclass</code>	1872		
E			
<code>\empty</code>	17, 18	<code>\immediate</code>	
<code>\end</code>	1867,	<code>\IncludeTests</code>	
	2144, 2163, 2186, 2197, 2208,	<code>\input</code>	
	2236, 2248, 2266, 2283, 2312,	<code>\iterate</code>	
	2339, 2347, 2376, 2403, 2417,	1792, 1794, 1796	
	2443, 2498, 2583, 2645, 2661, 2664		
<code>\endcsname</code>	14, 21,	L	
	50, 66, 76, 119, 125, 140, 143,	<code>\LoadCommand</code>	
	151, 175, 178, 707, 716, 717,	1840, 1850	
	719, 796, 808, 809, 811, 928,	<code>\LogTests</code>	
	935, 944, 951, 963, 1030, 1036,	1882	
	1043, 1050, 1775, 1778, 1781,	<code>\loop</code>	
	1784, 1839, 1866, 2090, 2649, 2651	1790, 1806, 1817, 1825	
<code>\endinput</code>	29, 117	M	
<code>\endlinechar</code>	4, 35, 71, 77, 89	<code>\m@ne</code>	
<code>\endqstest</code>	2112, 2117	2181, 2196, 2207, 2233	
<code>\errmessage</code>	1828	<code>\makeatletter</code>	
<code>\Expect</code>	1889, 1896, 1905, 1914, 2650, 2652	1875, 2092, 2126	
		<code>\makeatother</code>	
		1877, 2124	
I			
<code>\ifcase</code>	411, 430, 461, 474,	N	
	519, 532, 556, 599, 614, 702,	<code>\NeedsTeXFormat</code>	
	705, 720, 791, 794, 812, 906,	1870	
	1025, 1033, 1051, 1075, 1078,	<code>\newcommand</code>	
	1103, 1171, 1220, 1245, 1251,	1883, 1891, 1898,	
	1262, 1273, 1280, 1293, 1300,	1909, 1910, 1911, 1916, 1921,	
	1352, 1363, 1412, 1415, 1427,	1922, 1925, 1928, 1931, 1934,	
	1441, 1450, 1512, 1542, 1551,	1937, 1940, 1950, 1961, 1972,	
	1589, 1616, 1632, 1671, 1682,	1996, 2012, 2021, 2030, 2040,	
	1690, 1712, 1721, 2068, 2070, 2075	2043, 2048, 2051, 2054, 2058,	
<code>\ifcat</code>	157	2066, 2096, 2101, 2105, 2106, 2647	
<code>\ifnum</code>	342, 347, 379, 470,	<code>\newcount</code>	
	496, 528, 589, 627, 710, 799,	1908, 2093, 2094	
	874, 1081, 1092, 1107, 1117,	<code>\next</code>	
	1170, 1195, 1353, 1370, 1811,	1796, 1798, 1800	
	1819, 1826, 1834, 1952, 1963,	<code>\nofiles</code>	
	1974, 1975, 1976, 1998, 1999, 2000	1871	
<code>\ifodd</code>	396, 924, 940, 1322	<code>\number</code>	
<code>\iftrue</code>	2184	269, 285, 699, 788,	
<code>\ifx</code>	15,	1030, 1036, 1043, 1831, 2098,	
	18, 21, 50, 58, 61, 119, 125, 140,	2141, 2179, 2180, 2181, 2183,	
	143, 151, 165, 175, 178, 191,	2195, 2196, 2206, 2207, 2232, 2233	
	200, 201, 211, 216, 229, 232,	<code>\numexpr</code>	
	261, 274, 277, 294, 295, 305,	472, 530, 678,	
	320, 321, 329, 352, 395, 450,	698, 787, 802, 865, 869, 926,	
	462, 495, 508, 520, 555, 557,	933, 942, 949, 1000, 1007, 1873,	
	586, 587, 613, 646, 647, 657,	1874, 1879, 1884, 1885, 1888,	
	658, 675, 690, 737, 738, 748,	1892, 1893, 1895, 1901, 1909, 2045	
		P	
		<code>\PackageInfo</code>	
		26	
		<code>\pdf@escapehex</code>	
		186	
		<code>\pdf@unescapehex</code>	
		184	
		<code>\pdfelapsedtime</code>	
		2107	
		<code>\pdfresettimer</code>	
		2103	
		<code>\PrintTime</code>	
		2096, 2109, 2122	
		<code>\ProvidesPackage</code>	
		19, 67	
Q			
<code>\qstest</code>	2111, 2113		

R

`\RangeCatcodeCheck` . . . 1823, 1851,
 1852, 1853, 1854, 1855, 1856,
 1857, 1858, 1859, 1860, 1861, 1862
`\RangeCatcodeInvalid`
 1815, 1843, 1844, 1845, 1846
`\renewcommand` 2102
`\repeat` . . . 1790, 1802, 1813, 1821, 1836
`\RestoreCatcodes` 1804, 1807, 1808, 1863
`\result` 1894, 1896
`\resultA` 1886, 1889
`\resultB` 1887, 1889
`\romannumeral`
 . 153, 183, 245, 252, 256, 365,
 370, 386, 391, 406, 419, 425,
 438, 444, 447, 572, 581, 588,
 602, 621, 640, 643, 697, 786,
 827, 834, 846, 885, 891, 902,
 1001, 1008, 1041, 1066, 1091,
 1106, 1130, 1147, 1162, 1200,
 1207, 1211, 1305, 1365, 1372,
 1378, 1389, 1394, 1399, 1408,
 1448, 1466, 1489, 1494, 1499,
 1504, 1569, 1583, 1626, 1642,
 1649, 1658, 1667, 1730, 1731, 1761

S

`\saved@endqstest` 2112, 2119
`\saved@qstest` 2111, 2114
`\SavedNumexpr`
 1873, 1879, 1884, 1888, 1892, 1895
`\space` . . . 1829, 1830, 1838, 1889, 2098
`\StartTime` 2101, 2115
`\StopTime` 2106, 2118
`\strip@pt` 2098
`\SummaryTime` . . . 2093, 2095, 2108, 2122

T

`\Test` 1842, 1865, 1916, 1921,
 1923, 1926, 1929, 1932, 1935,
 1938, 1941, 1943, 1951, 1954,
 1962, 1965, 1973, 1978, 1986,
 1997, 2002, 2013, 2015, 2022,
 2024, 2031, 2033, 2041, 2049,
 2052, 2055, 2059, 2061, 2067, 2081
`\TestAbs` 1925,
 2147, 2148, 2149, 2150, 2151,
 2152, 2153, 2154, 2155, 2156,
 2157, 2158, 2159, 2160, 2161, 2162
`\TestAdd` 1972,
 2286, 2287, 2288, 2289, 2290,
 2291, 2292, 2293, 2294, 2295,
 2296, 2297, 2298, 2299, 2300,
 2301, 2302, 2303, 2304, 2305,
 2306, 2307, 2308, 2309, 2310, 2311
`\TestArg` 1909, 1910, 1912, 1913
`\TestCmp` 1937, 2211, 2212, 2213, 2214,
 2215, 2216, 2217, 2218, 2219,
 2220, 2221, 2222, 2223, 2224,
 2225, 2226, 2227, 2228, 2229,
 2230, 2231, 2232, 2233, 2234, 2235
`\TestCount` 1908, 1912, 1913, 1914

`\TestDec` . . . 1961, 2269, 2270, 2271,
 2272, 2273, 2274, 2275, 2276,
 2277, 2278, 2279, 2280, 2281, 2282
`\TestDiv` 2054,
 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, 2539, 2540,
 2541, 2542, 2543, 2544, 2545,
 2546, 2547, 2548, 2549, 2550,
 2551, 2552, 2553, 2554, 2555,
 2556, 2557, 2558, 2559, 2560,
 2561, 2562, 2563, 2564, 2565,
 2566, 2567, 2568, 2569, 2570,
 2571, 2572, 2573, 2574, 2575,
 2576, 2577, 2578, 2579, 2580, 2581
`\TestDivBig` 2058, 2582
`\TestError`
 2647, 2656, 2657, 2658, 2659, 2660
`\TestExch` 1921, 2044
`\TestFac` 2043, 2420, 2421,
 2422, 2423, 2424, 2425, 2426,
 2427, 2428, 2429, 2430, 2431, 2432
`\TestFacBig`
 . 2048, 2433, 2434, 2435, 2436,
 2437, 2438, 2439, 2440, 2441, 2442
`\TestInc` 1950, 2251, 2252, 2253, 2254,
 2255, 2256, 2257, 2258, 2259,
 2260, 2261, 2262, 2263, 2264, 2265
`\TestInv` 1922, 2129, 2130, 2131, 2132,
 2133, 2134, 2135, 2136, 2137,
 2138, 2139, 2140, 2141, 2142, 2143
`\TestMax` 1934, 2200, 2201,
 2202, 2203, 2204, 2205, 2206, 2207
`\TestMin` 1931, 2189, 2190,
 2191, 2192, 2193, 2194, 2195, 2196
`\TestMod` . . . 2066, 2586, 2587, 2588,
 2589, 2590, 2591, 2592, 2593,
 2594, 2595, 2596, 2597, 2598,
 2599, 2600, 2601, 2602, 2603,
 2604, 2605, 2606, 2607, 2608,
 2609, 2610, 2611, 2612, 2613,
 2614, 2615, 2616, 2617, 2618,
 2619, 2620, 2621, 2622, 2623,
 2624, 2625, 2626, 2627, 2628,
 2629, 2630, 2631, 2632, 2633,
 2634, 2635, 2636, 2637, 2638,
 2639, 2640, 2641, 2642, 2643, 2644
`\TestMul` . . . 2030, 2379, 2380, 2381,
 2382, 2383, 2384, 2385, 2386,
 2387, 2388, 2389, 2390, 2391,
 2392, 2393, 2394, 2395, 2396,
 2397, 2398, 2399, 2400, 2401, 2402
`\TestOdd` 1940, 2240, 2241,
 2242, 2243, 2244, 2245, 2246, 2247
`\TestPow` 2051, 2446,
 2447, 2448, 2449, 2450, 2451,
 2452, 2453, 2454, 2455, 2456

