The bigintcalc package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>

2016/05/16 v1.4

Abstract

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

Contents

1 Documentation **2**
1.1 Introduction . . . 2
1.2 Conditions . . . 2
1.2.1 Preconditions . . . 2
1.2.2 Postconditions . . . 3
1.3 Error handling . . . 3
1.4 Operations . . . 3
1.4.1 Num . . . 3
1.4.2 Inv, Abs, Sgn . . . 4
1.4.3 Min, Max, Cmp . . . 4
1.4.4 Odd . . . 5
1.4.5 Inc, Dec, Add, Sub . . . 5
1.4.6 Shl, Shr . . . 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 . . . 12
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

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

2.18.1 Help macros . . . 34
2.18.2 Recursive calculation . . . 35
2.19 Div . . . 36
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 . . . 50

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 Catalogue **61**

6 History **61**
[2007/09/27 v1.0] . . . 61
[2007/11/11 v1.1] . . . 61
[2011/01/30 v1.2] . . . 61
[2012/04/08 v1.3] . . . 62
[2016/05/16 v1.4] . . . 62

7 Index **62**

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 catcode 12 (other).
- Call by value is simulated. First the arguments are converted to numbers. Then these numbers are used in the calculations.

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

1.3 Error handling

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:

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

1.4.1 Num

`\bigintcalcNum {`$\langle x \rangle$`}`

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.

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

`\bigintcalcAbs {⟨x⟩}`

Macro `\bigintcalcAbs` returns the absolute value of integer ⟨x⟩.

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

`\bigintcalcSgn {⟨x⟩}`

Macro `\bigintcalcSgn` encodes the sign of ⟨x⟩ as number.

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

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.

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

`\bigintcalcMax {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcCmp {⟨x⟩} {⟨y⟩}`

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

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

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 {`$\langle x \rangle$`}`

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

1.4.5 Inc, Dec, Add, Sub

`\bigintcalcInc {`$\langle x \rangle$`}`

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

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

`\bigintcalcDec {`$\langle x \rangle$`}`

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

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

`\bigintcalcAdd {`$\langle x \rangle$`} {`$\langle y \rangle$`}`

Macro `\bigintcalcAdd` adds the two numbers.

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

`\bigintcalcSub {`$\langle x \rangle$`} {`$\langle y \rangle$`}`

Macro `\bigintcalcSub` calculates the difference.

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

1.4.6 Shl, Shr

`\bigintcalcShl {`$\langle x \rangle$`}`

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

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

`\bigintcalcShr {`$\langle x \rangle$`}`

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

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

1.4.7 Mul, Sqr, Fac, Pow

`\bigintcalcMul {⟨x⟩} {⟨y⟩}`

Macro `\bigintcalcMul` calculates the product of ⟨x⟩ and ⟨y⟩.

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

`\bigintcalcSqr {⟨x⟩}`

Macro `\bigintcalcSqr` returns the square product.

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

`\bigintcalcFac {⟨x⟩}`

Macro `\bigintcalcFac` returns the factorial of ⟨x⟩. Negative numbers are not permitted.

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

$(0! = 1)$

`\bigintcalcPow` Mx My

Macro `\bigintcalcPow` calculates the value of ⟨x⟩ to the power of ⟨y⟩. The error "division by zero" is thrown if ⟨x⟩ is zero and ⟨y⟩ is negative. permitted:

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

$(0^0 = 1)$

1.4.8 Div, Mul

`\bigintcalcDiv {⟨x⟩} {⟨y⟩}`

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

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

`\bigintcalcMod {⟨x⟩} {⟨y⟩}`

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

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

The result ranges:

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

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` ⟨*number*⟩ !

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

`\BigIntCalcInc` ⟨*number*⟩ !

Incrementation.

`\BigIntCalcDec` ⟨*number*⟩ !

Decrementation, positive number without zero.

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

Addition, $A \geq B$.

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

Subtraction, $A \geq B$.

`\BigIntCalcShl` ⟨*number*⟩ !

Left shift (multiplication with two).

`\BigIntCalcShr` ⟨*number*⟩ !

Right shift (integer division by two).

`\BigIntCalcMul` ⟨*number A*⟩ ! ⟨*number B*⟩ !

Multiplication, $A \geq B$.

`\BigIntCalcDiv` ⟨*number A*⟩ ! ⟨*number B*⟩ !

Division operation.

`\BigIntCalcMod` ⟨*number A*⟩ ! ⟨*number B*⟩ !

Modulo operation.

2 Implementation

```
⟨*package⟩
```

2.1 Reload check and package identification

Reload check, especially if the package is not used with LaTeX.

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\let\expandafter\x\csname ver@bigintcalc.sty\endcsname
  \ifx\x\relax % plain-TeX, first loading
  \else
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading,
      % variable is initialized, but \ProvidesPackage not yet seen
    \else
      \expandafter\ifx\csname PackageInfo\endcsname\relax
        \def\x#1#2{%
          \immediate\write-1{Package #1 Info: #2.}%
        }%
      \else
        \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
      \fi
      \x{bigintcalc}{The package is already loaded}%
      \aftergroup\endinput
    \fi
  \fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
```

```
    #2[{#3}]%
    \ifx#1\@undefined
      \xdef#1{#3}%
    \fi
    \ifx#1\relax
      \xdef#1{#3}%
    \fi
  }%
 \fi
\expandafter\x\csname ver@bigintcalc.sty\endcsname
\ProvidesPackage{bigintcalc}%
 [2016/05/16 v1.4 Expandable calculations on big integers (HO)]%
```

2.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
 \catcode13=5 % ^^M
 \endlinechar=13 %
 \catcode123=1 % {
 \catcode125=2 % }
 \catcode64=11 % @
 \def\x{\endgroup
   \expandafter\edef\csname BIC@AtEnd\endcsname{%
     \endlinechar=\the\endlinechar\relax
     \catcode13=\the\catcode13\relax
     \catcode32=\the\catcode32\relax
     \catcode35=\the\catcode35\relax
     \catcode61=\the\catcode61\relax
     \catcode64=\the\catcode64\relax
     \catcode123=\the\catcode123\relax
     \catcode125=\the\catcode125\relax
   }%
 }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
 \edef\BIC@AtEnd{%
   \BIC@AtEnd
   \catcode#1=\the\catcode#1\relax
 }%
 \catcode#1=#2\relax
}
\TMP@EnsureCode{33}{12}% !
\TMP@EnsureCode{36}{14}% $ (comment!)
\TMP@EnsureCode{38}{14}% & (comment!)
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{43}{12}% +
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{58}{11}% : (letter!)
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{63}{14}% ? (comment!)
\TMP@EnsureCode{91}{12}% [
```

```
\TMP@EnsureCode{93}{12}% ]
\edef\BIC@AtEnd{\BIC@AtEnd\noexpand\endinput}
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname BIC@TestMode\endcsname\relax
\else
  \catcode63=9 % ? (ignore)
\fi
? \let\BIC@@TestMode\BIC@TestMode
```

2.3 ε-TEX detection

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname numexpr\endcsname\relax
  \catcode36=9 % $ (ignore)
\else
  \catcode38=9 % & (ignore)
\fi
```

2.4 Help macros

\BIC@Fi

```
\let\BIC@Fi\fi
```

\BIC@AfterFi

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

\BIC@AfterFiFi

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

\BIC@AfterFiFiFi

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

\BIC@Space

```
\begingroup
  \def\x#1{\endgroup
    \let\BIC@Space= #1%
  }%
\x{ }
```

2.5 Expand number

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \def\TMP@RequirePackage#1[#2]{%
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname ver@#1.sty\endcsname\relax
      \input #1.sty\relax
    \fi
  }%
  \TMP@RequirePackage{pdftexcmds}[2007/11/11]%
\else
  \RequirePackage{pdftexcmds}[2007/11/11]%
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@escapehex\endcsname\relax
```

\BIC@Expand

```
  \def\BIC@Expand#1{%
    \romannumeral0%
    \BIC@@Expand#1!\@nil{}%
  }%
```

\BIC@@Expand

```
\def\BIC@@Expand#1#2\@nil#3{%
  \expandafter\ifcat\noexpand#1\relax
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
  {%
    \expandafter\BIC@@Expand#1#2\@nil{#3}%
  }{%
    \ifx#1!%
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    { #3}{%
      \BIC@@Expand#2\@nil{#3#1}%
    }%
  }%
}%
```

\@firstoftwo

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

\@secondoftwo

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

```
\else
```

\BIC@Expand

```
\def\BIC@Expand#1{%
  \romannumeral0\expandafter\expandafter\expandafter\BIC@Space
  \pdf@unescapehex{%
    \expandafter\expandafter\expandafter
    \BIC@StripHexSpace\pdf@escapehex{#1}20\@nil
  }%
}%
```

\BIC@StripHexSpace

```
\def\BIC@StripHexSpace#120#2\@nil{%
  #1%
  \ifx\\#2\\%
  \else
    \BIC@AfterFi{%
      \BIC@StripHexSpace#2\@nil
    }%
  \BIC@Fi
}%
```

```
\fi
```

2.6 Normalize expanded number

\BIC@Normalize

#1: result sign
#2: first token of number

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

```
      \BIC@AfterFiFi{%
        \BIC@Normalize-%
      }%
    \else
      \BIC@AfterFiFi{%
        \BIC@Normalize{}%
      }%
    \fi
  \else
    \ifx#2+%
      \BIC@AfterFiFi{%
        \BIC@Normalize{#1}%
      }%
    \else
      \ifx#20%
        \BIC@AfterFiFiFi{%
          \BIC@NormalizeZero{#1}%
        }%
      \else
        \BIC@AfterFiFiFi{%
          \BIC@NormalizeDigits#1#2%
        }%
      \fi
    \fi
  \BIC@Fi
}
```

\BIC@NormalizeZero

```
\def\BIC@NormalizeZero#1#2{%
  \ifx#2!%
    \BIC@AfterFi{ 0}%
  \else
    \ifx#20%
      \BIC@AfterFiFi{%
        \BIC@NormalizeZero{#1}%
      }%
    \else
      \BIC@AfterFiFi{%
        \BIC@NormalizeDigits#1#2%
      }%
    \fi
  \BIC@Fi
}
```

\BIC@NormalizeDigits

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

2.7 Num

\bigintcalcNum

```
\def\bigintcalcNum#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Normalize
  \expandafter\expandafter\expandafter{%
  \expandafter\expandafter\expandafter}%
  \BIC@Expand{#1}!%
}
```

2.8 Inv, Abs, Sgn

\bigintcalcInv

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

\bigintcalcAbs

```
\def\bigintcalcAbs#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Abs
  \bigintcalcNum{#1}%
}
```

\BIC@Abs

```
\def\BIC@Abs#1{%
  \ifx#1-%
    \expandafter\BIC@Space
  \else
    \expandafter\BIC@Space
    \expandafter#1%
  \fi
}
```

\bigintcalcSgn

```
\def\bigintcalcSgn#1{%
  \number
  \expandafter\expandafter\expandafter\BIC@Sgn
  \bigintcalcNum{#1}! %
}
```

\BIC@Sgn

```
\def\BIC@Sgn#1#2!{%
  \ifx#1-%
    -1%
  \else
    \ifx#10%
      0%
    \else
      1%
    \fi
  \fi
}
```

2.9 Cmp, Min, Max

\bigintcalcCmp

```
\def\bigintcalcCmp#1#2{%
  \number
  \expandafter\expandafter\expandafter\BIC@Cmp
  \bigintcalcNum{#2}!{#1}%
}
```

\BIC@Cmp

```
\def\BIC@Cmp#1!#2{%
  \expandafter\expandafter\expandafter\BIC@@Cmp
  \bigintcalcNum{#2}!#1!%
}
```

\BIC@@Cmp

```
\def\BIC@@Cmp#1#2!#3#4!{%
  \ifx#1-%
    \ifx#3-%
```

```
      \BIC@AfterFiFi{%
        \BIC@@Cmp#4!#2!%
      }%
    \else
      \BIC@AfterFiFi{%
        -1 %
      }%
    \fi
  \else
    \ifx#3-%
      \BIC@AfterFiFi{%
        1 %
      }%
    \else
      \BIC@AfterFiFi{%
        \BIC@CmpLength#1#2!#3#4!#1#2!#3#4!%
      }%
    \fi
  \BIC@Fi
}
```

\BIC@PosCmp

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

\BIC@CmpLength

```
\def\BIC@CmpLength#1#2!#3#4!{%
  \ifx\\#2\\%
    \ifx\\#4\\%
      \BIC@AfterFiFi\BIC@CmpDiff
    \else
      \BIC@AfterFiFi{%
        \BIC@CmpResult{-1}%
      }%
    \fi
  \else
    \ifx\\#4\\%
      \BIC@AfterFiFi{%
        \BIC@CmpResult1%
      }%
    \else
      \BIC@AfterFiFi{%
        \BIC@CmpLength#2!#4!%
      }%
    \fi
  \BIC@Fi
}
```

\BIC@CmpResult

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

\BIC@CmpDiff

```
\def\BIC@CmpDiff#1#2!#3#4!{%
  \ifnum#1<#3 %
    \BIC@AfterFi{%
      -1 %
    }%
  \else
    \ifnum#1>#3 %
      \BIC@AfterFiFi{%
        1 %
```

```
    }%
   \else
    \ifx\\#2\\%
     \BIC@AfterFiFiFi{%
      0 %
     }%
    \else
     \BIC@AfterFiFiFi{%
      \BIC@CmpDiff#2!#4!%
     }%
    \fi
   \fi
  \BIC@Fi
}
```

\bigintcalcMin

```
\def\bigintcalcMin#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@MinMax
  \bigintcalcNum{#1}!-!%
}
```

\bigintcalcMax

```
\def\bigintcalcMax#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@MinMax
  \bigintcalcNum{#1}!!%
}
```

\BIC@MinMax #1: x
#2: sign for comparison
#3: y

```
\def\BIC@MinMax#1!#2!#3{%
  \expandafter\expandafter\expandafter\BIC@@MinMax
  \bigintcalcNum{#3}!#1!#2!%
}
```

\BIC@@MinMax #1: y
#2: x
#3: sign for comparison

```
\def\BIC@@MinMax#1!#2!#3!{%
  \ifnum\BIC@@Cmp#1!#2!=#31 %
    \BIC@AfterFi{ #1}%
  \else
    \BIC@AfterFi{ #2}%
  \fi
}
```

2.10 Odd

\bigintcalcOdd

```
\def\bigintcalcOdd#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Odd
  \bigintcalcAbs{#1}!%
}
```

\BigIntCalcOdd

```
\def\BigIntCalcOdd#1!{%
  \romannumeral0%
  \BIC@Odd#1!%
}
```

\BIC@Odd #1: x

```
\def\BIC@Odd#1#2{%
  \ifx#2!%
    \ifodd#1 %
      \BIC@AfterFiFi{ 1}%
    \else
      \BIC@AfterFiFi{ 0}%
    \fi
  \else
    \expandafter\BIC@Odd\expandafter#2%
  \BIC@Fi
}
```

2.11 Inc, Dec

\bigintcalcInc

```
\def\bigintcalcInc#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@IncSwitch
  \bigintcalcNum{#1}!%
}
```

\BIC@IncSwitch

```
\def\BIC@IncSwitch#1#2!{%
  \ifcase\BIC@@Cmp#1#2!-1!%
    \BIC@AfterFi{ 0}%
  \or
    \BIC@AfterFi{%
      \BIC@Inc#1#2!{}%
    }%
  \else
    \BIC@AfterFi{%
      \expandafter-\romannumeral0%
      \BIC@Dec#2!{}%
    }%
  \BIC@Fi
}
```

\bigintcalcDec

```
\def\bigintcalcDec#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@DecSwitch
  \bigintcalcNum{#1}!%
}
```

\BIC@DecSwitch

```
\def\BIC@DecSwitch#1#2!{%
  \ifcase\BIC@Sgn#1#2! %
    \BIC@AfterFi{ -1}%
  \or
    \BIC@AfterFi{%
      \BIC@Dec#1#2!{}%
    }%
  \else
    \BIC@AfterFi{%
      \expandafter-\romannumeral0%
      \BIC@Inc#2!{}%
    }%
  \BIC@Fi
}
```

\BigIntCalcInc

```
\def\BigIntCalcInc#1!{%
  \romannumeral0\BIC@Inc#1!{}%
}
```

\BigIntCalcDec

```
\def\BigIntCalcDec#1!{%
  \romannumeral0\BIC@Dec#1!{}%
}
```

\BIC@Inc

```
\def\BIC@Inc#1#2!#3{%
  \ifx\\#2\\%
    \BIC@AfterFi{%
      \BIC@@Inc1#1#3!{}%
    }%
  \else
    \BIC@AfterFi{%
      \BIC@Inc#2!{#1#3}%
    }%
  \BIC@Fi
}
```

\BIC@@Inc

```
\def\BIC@@Inc#1#2#3!#4{%
  \ifcase#1 %
    \ifx\\#3\\%
      \BIC@AfterFiFi{ #2#4}%
    \else
      \BIC@AfterFiFi{%
        \BIC@@Inc0#3!{#2#4}%
      }%
    \fi
  \else
    \ifnum#2<9 %
      \BIC@AfterFiFi{%
&       \expandafter\BIC@@@Inc\the\numexpr#2+1\relax
$       \expandafter\expandafter\expandafter\BIC@@@Inc
$       \ifcase#2 \expandafter1%
$       \or\expandafter2%
$       \or\expandafter3%
$       \or\expandafter4%
$       \or\expandafter5%
$       \or\expandafter6%
$       \or\expandafter7%
$       \or\expandafter8%
$       \or\expandafter9%
$?      \else\BigIntCalcError:ThisCannotHappen%
$       \fi
        0#3!{#4}%
      }%
    \else
      \BIC@AfterFiFi{%
        \BIC@@@Inc01#3!{#4}%
      }%
    \fi
  \BIC@Fi
}
```

\BIC@@@Inc

```
\def\BIC@@@Inc#1#2#3!#4{%
  \ifx\\#3\\%
```

```
    \ifnum#2=1 %
      \BIC@AfterFiFi{ 1#1#4}%
    \else
      \BIC@AfterFiFi{ #1#4}%
    \fi
  \else
    \BIC@AfterFi{%
      \BIC@@Inc#2#3!{#1#4}%
    }%
  \BIC@Fi
}
```

\BIC@Dec

```
\def\BIC@Dec#1#2!#3{%
  \ifx\\#2\\%
    \BIC@AfterFi{%
      \BIC@@Dec1#1#3!{}%
    }%
  \else
    \BIC@AfterFi{%
      \BIC@Dec#2!{#1#3}%
    }%
  \BIC@Fi
}
```

\BIC@@Dec

```
\def\BIC@@Dec#1#2#3!#4{%
  \ifcase#1 %
    \ifx\\#3\\%
      \BIC@AfterFiFi{ #2#4}%
    \else
      \BIC@AfterFiFi{%
        \BIC@@Dec0#3!{#2#4}%
      }%
    \fi
  \else
    \ifnum#2>0 %
      \BIC@AfterFiFi{%
&       \expandafter\BIC@@@Dec\the\numexpr#2-1\relax
$       \expandafter\expandafter\expandafter\BIC@@@Dec
$       \ifcase#2
$?        \BigIntCalcError:ThisCannotHappen%
$       \or\expandafter0%
$       \or\expandafter1%
$       \or\expandafter2%
$       \or\expandafter3%
$       \or\expandafter4%
$       \or\expandafter5%
$       \or\expandafter6%
$       \or\expandafter7%
$       \or\expandafter8%
$?      \else\BigIntCalcError:ThisCannotHappen%
$       \fi
        0#3!{#4}%
      }%
    \else
      \BIC@AfterFiFi{%
        \BIC@@@Dec91#3!{#4}%
      }%
    \fi
  \BIC@Fi
}
```

\BIC@@@Dec

```
\def\BIC@@@Dec#1#2#3!#4{%
  \ifx\\#3\\%
    \ifcase#1 %
      \ifx\\#4\\%
        \BIC@AfterFiFiFi{ 0}%
      \else
        \BIC@AfterFiFiFi{ #4}%
      \fi
    \else
      \BIC@AfterFiFi{ #1#4}%
    \fi
  \else
    \BIC@AfterFi{%
      \BIC@@Dec#2#3!{#1#4}%
    }%
  \BIC@Fi
}
```

2.12 Add, Sub

\bigintcalcAdd

```
\def\bigintcalcAdd#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Add
  \bigintcalcNum{#1}!%
}
```

\BIC@Add

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

\bigintcalcSub

```
\def\bigintcalcSub#1#2{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Add
  \bigintcalcNum{-#2}!{#1}%
}
```

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

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

```
\def\BIC@AddSwitch#1#2!#3#4!{%
  \ifx#1-% x < 0
    \ifx#3-% y < 0
      \expandafter-\romannumeral0%
      \ifnum\BIC@PosCmp#2!#4!=1 % -x > -y
        \BIC@AfterFiFiFi{%
          \BIC@AddXY#2!#4!!!%
```

```
      }%
    \else % -x <= -y
      \BIC@AfterFiFiFi{%
        \BIC@AddXY#4!#2!!!%
      }%
    \fi
  \else % y >= 0
    \ifcase\BIC@PosCmp#2!#3#4!% -x = y
      \BIC@AfterFiFiFi{ 0}%
    \or % -x > y
      \expandafter-\romannumeral0%
      \BIC@AfterFiFiFi{%
        \BIC@SubXY#2!#3#4!!!%
      }%
    \else % -x <= y
      \BIC@AfterFiFiFi{%
        \BIC@SubXY#3#4!#2!!!%
      }%
    \fi
  \fi
\else % x >= 0
  \ifx#3-% y < 0
    \ifcase\BIC@PosCmp#1#2!#4!% x = -y
      \BIC@AfterFiFiFi{ 0}%
    \or % x > -y
      \BIC@AfterFiFiFi{%
        \BIC@SubXY#1#2!#4!!!%
      }%
    \else % x <= -y
      \expandafter-\romannumeral0%
      \BIC@AfterFiFiFi{%
        \BIC@SubXY#4!#1#2!!!%
      }%
    \fi
  \else % y >= 0
    \ifnum\BIC@PosCmp#1#2!#3#4!=1 % x > y
      \BIC@AfterFiFiFi{%
        \BIC@AddXY#1#2!#3#4!!!%
      }%
    \else % x <= y
      \BIC@AfterFiFiFi{%
        \BIC@AddXY#3#4!#1#2!!!%
      }%
    \fi
  \fi
  \BIC@Fi
}
```

\BigIntCalcAdd

```
\def\BigIntCalcAdd#1!#2!{%
  \romannumeral0\BIC@AddXY#1!#2!!!%
}
```

\BigIntCalcSub

```
\def\BigIntCalcSub#1!#2!{%
  \romannumeral0\BIC@SubXY#1!#2!!!%
}
```

\BIC@AddXY

```
\def\BIC@AddXY#1#2!#3#4!#5!#6!{%
  \ifx\\#2\\%
    \ifx\\#3\\%
```

```
      \BIC@AfterFiFi{%
        \BIC@DoAdd0!#1#5!#60!%
      }%
    \else
      \BIC@AfterFiFi{%
        \BIC@DoAdd0!#1#5!#3#6!%
      }%
    \fi
  \else
    \ifx\\#4\\%
      \ifx\\#3\\%
        \BIC@AfterFiFiFi{%
          \BIC@AddXY#2!{}!#1#5!#60!%
        }%
      \else
        \BIC@AfterFiFiFi{%
          \BIC@AddXY#2!{}!#1#5!#3#6!%
        }%
      \fi
    \else
      \BIC@AfterFiFi{%
        \BIC@AddXY#2!#4!#1#5!#3#6!%
      }%
    \fi
  \BIC@Fi
}
```

\BIC@DoAdd

#1: carry
#2: reverted result
#3#4: reverted x
#5#6: reverted y

```
\def\BIC@DoAdd#1#2!#3#4!#5#6!{%
  \ifx\\#4\\%
    \BIC@AfterFi{%
&     \expandafter\BIC@Space
&     \the\numexpr#1+#3+#5\relax#2%
$     \expandafter\expandafter\expandafter\BIC@AddResult
$     \BIC@AddDigit#1#3#5#2%
    }%
  \else
    \BIC@AfterFi{%
      \expandafter\expandafter\expandafter\BIC@DoAdd
      \BIC@AddDigit#1#3#5#2!#4!#6!%
    }%
  \BIC@Fi
}
```

\BIC@AddResult

```
$ \def\BIC@AddResult#1{%
$   \ifx#10%
$     \expandafter\BIC@Space
$   \else
$     \expandafter\BIC@Space\expandafter#1%
$   \fi
$ }%
```

\BIC@AddDigit

#1: carry
#2: digit of x
#3: digit of y

```
\def\BIC@AddDigit#1#2#3{%
  \romannumeral0%
& \expandafter\BIC@@AddDigit\the\numexpr#1+#2+#3!%
```

```
$ \expandafter\BIC@@AddDigit\number%
$ \csname
$   BIC@AddCarry%
$   \ifcase#1 %
$     #2%
$   \else
$     \ifcase#2 1\or2\or3\or4\or5\or6\or7\or8\or9\or10\fi
$   \fi
$ \endcsname#3!%
}
```

\BIC@@AddDigit

```
\def\BIC@@AddDigit#1!{%
  \ifnum#1<10 %
    \BIC@AfterFi{ 0#1}%
  \else
    \BIC@AfterFi{ #1}%
  \BIC@Fi
}
```

\BIC@AddCarry0

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

\BIC@AddCarry10

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

\BIC@AddCarry[1-9]

```
$ \def\BIC@Temp#1#2{%
$   \expandafter\def\csname BIC@AddCarry#1\endcsname##1{%
$     \ifcase##1 #1\or
$     #2%
$?    \else\BigIntCalcError:ThisCannotHappen%
$     \fi
$   }%
$ }%
$ \BIC@Temp 0{1\or2\or3\or4\or5\or6\or7\or8\or9}%
$ \BIC@Temp 1{2\or3\or4\or5\or6\or7\or8\or9\or10}%
$ \BIC@Temp 2{3\or4\or5\or6\or7\or8\or9\or10\or11}%
$ \BIC@Temp 3{4\or5\or6\or7\or8\or9\or10\or11\or12}%
$ \BIC@Temp 4{5\or6\or7\or8\or9\or10\or11\or12\or13}%
$ \BIC@Temp 5{6\or7\or8\or9\or10\or11\or12\or13\or14}%
$ \BIC@Temp 6{7\or8\or9\or10\or11\or12\or13\or14\or15}%
$ \BIC@Temp 7{8\or9\or10\or11\or12\or13\or14\or15\or16}%
$ \BIC@Temp 8{9\or10\or11\or12\or13\or14\or15\or16\or17}%
$ \BIC@Temp 9{10\or11\or12\or13\or14\or15\or16\or17\or18}%
```

\BIC@SubXY Preconditions:

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

```
\def\BIC@SubXY#1#2!#3#4!#5!#6!{%
  \ifx\\#2\\%
    \ifx\\#3\\%
      \BIC@AfterFiFi{%
        \BIC@DoSub0!#1#5!#60!%
      }%
    \else
      \BIC@AfterFiFi{%
        \BIC@DoSub0!#1#5!#3#6!%
      }%
    \fi
```

```
  \else
    \ifx\\#4\\%
      \ifx\\#3\\%
        \BIC@AfterFiFiFi{%
          \BIC@SubXY#2!{}!#1#5!#60!%
        }%
      \else
        \BIC@AfterFiFiFi{%
          \BIC@SubXY#2!{}!#1#5!#3#6!%
        }%
      \fi
    \else
      \BIC@AfterFiFi{%
        \BIC@SubXY#2!#4!#1#5!#3#6!%
      }%
    \fi
  \BIC@Fi
}
```

\BIC@DoSub

#1: carry
#2: reverted result
#3#4: reverted x
#5#6: reverted y

```
\def\BIC@DoSub#1#2!#3#4!#5#6!{%
  \ifx\\#4\\%
    \BIC@AfterFi{%
      \expandafter\expandafter\expandafter\BIC@SubResult
      \BIC@SubDigit#1#3#5#2%
    }%
  \else
    \BIC@AfterFi{%
      \expandafter\expandafter\expandafter\BIC@DoSub
      \BIC@SubDigit#1#3#5#2!#4!#6!%
    }%
  \BIC@Fi
}
```

\BIC@SubResult

```
\def\BIC@SubResult#1{%
  \ifx#10%
    \expandafter\BIC@SubResult
  \else
    \expandafter\BIC@Space\expandafter#1%
  \fi
}
```

\BIC@SubDigit

#1: carry
#2: digit of x
#3: digit of y

```
\def\BIC@SubDigit#1#2#3{%
  \romannumeral0%
& \expandafter\BIC@@SubDigit\the\numexpr#2-#3-#1!%
$ \expandafter\BIC@@AddDigit\number
$   \csname
$     BIC@SubCarry%
$     \ifcase#1 %
$       #3%
$     \else
$       \ifcase#3 1\or2\or3\or4\or5\or6\or7\or8\or9\or10\fi
$     \fi
$   \endcsname#2!%
}
```

\BIC@@SubDigit

```
& \def\BIC@@SubDigit#1!{%
&   \ifnum#1<0 %
&     \BIC@AfterFi{%
&       \expandafter\BIC@Space
&       \expandafter1\the\numexpr#1+10\relax
&     }%
&   \else
&     \BIC@AfterFi{ 0#1}%
&   \BIC@Fi
& }%
```

\BIC@SubCarry0

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

\BIC@SubCarry10

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

\BIC@SubCarry[1-9]

```
$ \def\BIC@Temp#1#2{%
$   \expandafter\def\csname BIC@SubCarry#1\endcsname##1{%
$     \ifcase##1 #2%
$?    \else\BigIntCalcError:ThisCannotHappen%
$     \fi
$   }%
$ }%
$ \BIC@Temp 1{19\or0\or1\or2\or3\or4\or5\or6\or7\or8}%
$ \BIC@Temp 2{18\or19\or0\or1\or2\or3\or4\or5\or6\or7}%
$ \BIC@Temp 3{17\or18\or19\or0\or1\or2\or3\or4\or5\or6}%
$ \BIC@Temp 4{16\or17\or18\or19\or0\or1\or2\or3\or4\or5}%
$ \BIC@Temp 5{15\or16\or17\or18\or19\or0\or1\or2\or3\or4}%
$ \BIC@Temp 6{14\or15\or16\or17\or18\or19\or0\or1\or2\or3}%
$ \BIC@Temp 7{13\or14\or15\or16\or17\or18\or19\or0\or1\or2}%
$ \BIC@Temp 8{12\or13\or14\or15\or16\or17\or18\or19\or0\or1}%
$ \BIC@Temp 9{11\or12\or13\or14\or15\or16\or17\or18\or19\or0}%
```

2.13 Shl, Shr

\bigintcalcShl

```
\def\bigintcalcShl#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Shl
  \bigintcalcNum{#1}!%
}
```

\BIC@Shl

```
\def\BIC@Shl#1#2!{%
  \ifx#1-%
    \BIC@AfterFi{%
      \expandafter-\romannumeral0%
&     \BIC@@Shl#2!!%
$     \BIC@AddXY#2!#2!!!%
    }%
  \else
    \BIC@AfterFi{%
&     \BIC@@Shl#1#2!!%
$     \BIC@AddXY#1#2!#1#2!!!%
    }%
  \BIC@Fi
}
```

\BigIntCalcShl

```
\def\BigIntCalcShl#1!{%
  \romannumeral0%
& \BIC@@Shl#1!!%
$ \BIC@AddXY#1!#1!!!%
}
```

\BIC@@Shl

```
& \def\BIC@@Shl#1#2!{%
&   \ifx\\#2\\%
&     \BIC@AfterFi{%
&       \BIC@@@Shl0!#1%
&     }%
&   \else
&     \BIC@AfterFi{%
&       \BIC@@Shl#2!#1%
&     }%
&   \BIC@Fi
& }%
```

\BIC@@@Shl

#1: carry
#2: result
#3#4: reverted number

```
& \def\BIC@@@Shl#1#2!#3#4!{%
&   \ifx\\#4\\%
&     \BIC@AfterFi{%
&       \expandafter\BIC@Space
&       \the\numexpr#3*2+#1\relax#2%
&     }%
&   \else
&     \BIC@AfterFi{%
&       \expandafter\BIC@@@@Shl\the\numexpr#3*2+#1!#2!#4!%
&     }%
&   \BIC@Fi
& }%
```

\BIC@@@@Shl

```
& \def\BIC@@@@Shl#1!{%
&   \ifnum#1<10 %
&     \BIC@AfterFi{%
&       \BIC@@@Shl0#1%
&     }%
&   \else
&     \BIC@AfterFi{%
&       \BIC@@@Shl#1%
&     }%
&   \BIC@Fi
& }%
```

\bigintcalcShr

```
\def\bigintcalcShr#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Shr
  \bigintcalcNum{#1}!%
}
```

\BIC@Shr

```
\def\BIC@Shr#1#2!{%
  \ifx#1-%
    \expandafter-\romannumeral0%
    \BIC@AfterFi{%
      \BIC@@Shr#2!%
```

```
    }%
  \else
    \BIC@AfterFi{%
      \BIC@@Shr#1#2!%
    }%
  \BIC@Fi
}
```

\BigIntCalcShr

```
\def\BigIntCalcShr#1!{%
  \romannumeral0%
  \BIC@@Shr#1!%
}
```

\BIC@@Shr

```
\def\BIC@@Shr#1#2!{%
  \ifcase#1 %
    \BIC@AfterFi{ 0}%
  \or
    \ifx\\#2\\%
      \BIC@AfterFiFi{ 0}%
    \else
      \BIC@AfterFiFi{%
        \BIC@@@Shr#1#2!!%
      }%
    \fi
  \else
    \BIC@AfterFi{%
      \BIC@@@Shr0#1#2!!%
    }%
  \BIC@Fi
}
```

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

```
\def\BIC@@@Shr#1#2#3!#4!{%
  \ifx\\#3\\%
    \ifodd#1#2 %
      \BIC@AfterFiFi{%
&       \expandafter\BIC@ShrResult\the\numexpr(#1#2-1)/2\relax
$       \expandafter\expandafter\expandafter\BIC@ShrResult
$       \csname BIC@ShrDigit#1#2\endcsname
        #4!%
      }%
    \else
      \BIC@AfterFiFi{%
&       \expandafter\BIC@ShrResult\the\numexpr#1#2/2\relax
$       \expandafter\expandafter\expandafter\BIC@ShrResult
$       \csname BIC@ShrDigit#1#2\endcsname
        #4!%
      }%
    \fi
  \else
    \ifodd#1#2 %
      \BIC@AfterFiFi{%
&       \expandafter\BIC@@@@Shr\the\numexpr(#1#2-1)/2\relax1%
$       \expandafter\expandafter\expandafter\BIC@@@@Shr
$       \csname BIC@ShrDigit#1#2\endcsname
        #3!#4!%
      }%
    \else
```

```
      \BIC@AfterFiFi{%
& \expandafter\BIC@@@@Shr\the\numexpr#1#2/2\relax0%
$ \expandafter\expandafter\expandafter\BIC@@@@Shr
$ \csname BIC@ShrDigit#1#2\endcsname
    #3!#4!%
  }%
 \fi
\BIC@Fi
}
```

\BIC@ShrResult

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

\BIC@@@@Shr

#1: new digit
#2: carry
#3: remaining number
#4: result

```
\def\BIC@@@@Shr#1#2#3!#4!{%
 \BIC@@@Shr#2#3!#4#1!%
}
```

\BIC@ShrDigit[00-19]

```
$ \def\BIC@Temp#1#2#3#4{%
$  \expandafter\def\csname BIC@ShrDigit#1#2\endcsname{#3#4}%
$ }%
$ \BIC@Temp 0000%
$ \BIC@Temp 0101%
$ \BIC@Temp 0210%
$ \BIC@Temp 0311%
$ \BIC@Temp 0420%
$ \BIC@Temp 0521%
$ \BIC@Temp 0630%
$ \BIC@Temp 0731%
$ \BIC@Temp 0840%
$ \BIC@Temp 0941%
$ \BIC@Temp 1050%
$ \BIC@Temp 1151%
$ \BIC@Temp 1260%
$ \BIC@Temp 1361%
$ \BIC@Temp 1470%
$ \BIC@Temp 1571%
$ \BIC@Temp 1680%
$ \BIC@Temp 1781%
$ \BIC@Temp 1890%
$ \BIC@Temp 1991%
```

2.14 \BIC@Tim

\BIC@Tim

Macro \BIC@Tim implements "Number *time*s digit".
#1: plain number without sign
#2: digit

\BIC@@Tim

#1#2: number
#3: reverted number

```
\def\BIC@@Tim#1#2!{%
 \ifx\\#2\\%
  \BIC@AfterFi{%
   \BIC@ProcessTim0!#1%
  }%
 \else
```

```
    \BIC@AfterFi{%
      \BIC@@Tim#2!#1%
    }%
  \BIC@Fi
}
```

\BIC@ProcessTim
#1: carry
#2: result
#3#4: reverted number
#5: digit

```
\def\BIC@ProcessTim#1#2!#3#4!#5{%
  \ifx\\#4\\%
    \BIC@AfterFi{%
      \expandafter\BIC@Space
&     \the\numexpr#3*#5+#1\relax
$     \romannumeral0\BIC@TimDigit#3#5#1%
      #2%
    }%
  \else
    \BIC@AfterFi{%
      \expandafter\BIC@@ProcessTim
&     \the\numexpr#3*#5+#1%
$     \romannumeral0\BIC@TimDigit#3#5#1%
      !#2!#4!#5%
    }%
  \BIC@Fi
}
```

\BIC@@ProcessTim
#1#2: carry?, new digit
#3: new number
#4: old number
#5: digit

```
\def\BIC@@ProcessTim#1#2!{%
  \ifx\\#2\\%
    \BIC@AfterFi{%
      \BIC@ProcessTim0#1%
    }%
  \else
    \BIC@AfterFi{%
      \BIC@ProcessTim#1#2%
    }%
  \BIC@Fi
}
```

\BIC@TimDigit
#1: digit 0–9
#2: digit 3–9
#3: carry 0–9

```
$ \def\BIC@TimDigit#1#2#3{%
$   \ifcase#1 % 0
$     \BIC@AfterFi{ #3}%
$   \or % 1
$     \BIC@AfterFi{%
$       \expandafter\BIC@Space
$       \number\csname BIC@AddCarry#2\endcsname#3 %
$     }%
$   \else
$     \ifcase#3 %
$       \BIC@AfterFiFi{%
$         \expandafter\BIC@Space
$         \number\csname BIC@MulDigit#2\endcsname#1 %
$       }%
$     \else
```

```
$       \BIC@AfterFiFi{%
$         \expandafter\BIC@Space
$         \romannumeral0%
$         \expandafter\BIC@AddXY
$         \number\csname BIC@MulDigit#2\endcsname#1!%
$         #3!!!%
$       }%
$     \fi
$   \BIC@Fi
$ }%
```

\BIC@MulDigit[3-9]

```
$ \def\BIC@Temp#1#2{%
$   \expandafter\def\csname BIC@MulDigit#1\endcsname##1{%
$     \ifcase##1 0%
$     \or ##1%
$     \or #2%
$?    \else\BigIntCalcError:ThisCannotHappen%
$     \fi
$   }%
$ }%
$ \BIC@Temp 3{6\or9\or12\or15\or18\or21\or24\or27}%
$ \BIC@Temp 4{8\or12\or16\or20\or24\or28\or32\or36}%
$ \BIC@Temp 5{10\or15\or20\or25\or30\or35\or40\or45}%
$ \BIC@Temp 6{12\or18\or24\or30\or36\or42\or48\or54}%
$ \BIC@Temp 7{14\or21\or28\or35\or42\or49\or56\or63}%
$ \BIC@Temp 8{16\or24\or32\or40\or48\or56\or64\or72}%
$ \BIC@Temp 9{18\or27\or36\or45\or54\or63\or72\or81}%
```

2.15 Mul

\bigintcalcMul

```
\def\bigintcalcMul#1#2{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Mul
  \bigintcalcNum{#1}!{#2}%
}
```

\BIC@Mul

```
\def\BIC@Mul#1!#2{%
  \expandafter\expandafter\expandafter\BIC@MulSwitch
  \bigintcalcNum{#2}!#1!%
}
```

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

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

```
\def\BIC@MulSwitch#1#2!#3#4!{%
  \ifcase\BIC@Sgn#1#2! % x = 0
    \BIC@AfterFi{ 0}%
```

```
  \or % x > 0
    \ifcase\BIC@Sgn#3#4! % y = 0
      \BIC@AfterFiFi{ 0}%
    \or % y > 0
      \ifnum\BIC@PosCmp#1#2!#3#4!=1 % x > y
        \BIC@AfterFiFiFi{%
          \BIC@ProcessMul0!#1#2!#3#4!%
        }%
      \else % x <= y
        \BIC@AfterFiFiFi{%
          \BIC@ProcessMul0!#3#4!#1#2!%
        }%
      \fi
    \else % y < 0
      \expandafter-\romannumeral0%
      \ifnum\BIC@PosCmp#1#2!#4!=1 % x > -y
        \BIC@AfterFiFiFi{%
          \BIC@ProcessMul0!#1#2!#4!%
        }%
      \else % x <= -y
        \BIC@AfterFiFiFi{%
          \BIC@ProcessMul0!#4!#1#2!%
        }%
      \fi
    \fi
  \else % x < 0
    \ifcase\BIC@Sgn#3#4! % y = 0
      \BIC@AfterFiFi{ 0}%
    \or % y > 0
      \expandafter-\romannumeral0%
      \ifnum\BIC@PosCmp#2!#3#4!=1 % -x > y
        \BIC@AfterFiFiFi{%
          \BIC@ProcessMul0!#2!#3#4!%
        }%
      \else % -x <= y
        \BIC@AfterFiFiFi{%
          \BIC@ProcessMul0!#3#4!#2!%
        }%
      \fi
    \else % y < 0
      \ifnum\BIC@PosCmp#2!#4!=1 % -x > -y
        \BIC@AfterFiFiFi{%
          \BIC@ProcessMul0!#2!#4!%
        }%
      \else % -x <= -y
        \BIC@AfterFiFiFi{%
          \BIC@ProcessMul0!#4!#2!%
        }%
      \fi
    \fi
  \BIC@Fi
}
```

\BigIntCalcMul

```
\def\BigIntCalcMul#1!#2!{%
  \romannumeral0%
  \BIC@ProcessMul0!#1!#2!%
}
```

\BIC@ProcessMul

#1: result
#2: number x
#3#4: number y

```
\def\BIC@ProcessMul#1!#2!#3#4!{%
  \ifx\\#4\\%
    \BIC@AfterFi{%
      \expandafter\expandafter\expandafter\BIC@Space
      \bigintcalcAdd{\BIC@Tim#2!#3}{#10}%
    }%
  \else
    \BIC@AfterFi{%
      \expandafter\expandafter\expandafter\BIC@ProcessMul
      \bigintcalcAdd{\BIC@Tim#2!#3}{#10}!#2!#4!%
    }%
  \BIC@Fi
}
```

2.16 Sqr

\bigintcalcSqr

```
\def\bigintcalcSqr#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Sqr
  \bigintcalcNum{#1}!%
}
```

\BIC@Sqr

```
\def\BIC@Sqr#1{%
   \ifx#1-%
     \expandafter\BIC@@Sqr
   \else
     \expandafter\BIC@@Sqr\expandafter#1%
   \fi
}
```

\BIC@@Sqr

```
\def\BIC@@Sqr#1!{%
  \BIC@ProcessMul0!#1!#1!%
}
```

2.17 Fac

\bigintcalcFac

```
\def\bigintcalcFac#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Fac
  \bigintcalcNum{#1}!%
}
```

\BIC@Fac

```
\def\BIC@Fac#1#2!{%
  \ifx#1-%
    \BIC@AfterFi{ 0\BigIntCalcError:FacNegative}%
  \else
    \ifnum\BIC@PosCmp#1#2!13!<0 %
      \ifcase#1#2 %
        \BIC@AfterFiFiFi{ 1}% 0!
      \or\BIC@AfterFiFiFi{ 1}% 1!
      \or\BIC@AfterFiFiFi{ 2}% 2!
      \or\BIC@AfterFiFiFi{ 6}% 3!
      \or\BIC@AfterFiFiFi{ 24}% 4!
      \or\BIC@AfterFiFiFi{ 120}% 5!
      \or\BIC@AfterFiFiFi{ 720}% 6!
      \or\BIC@AfterFiFiFi{ 5040}% 7!
```

```
      \or\BIC@AfterFiFiFi{ 40320}% 8!
      \or\BIC@AfterFiFiFi{ 362880}% 9!
      \or\BIC@AfterFiFiFi{ 3628800}% 10!
      \or\BIC@AfterFiFiFi{ 39916800}% 11!
      \or\BIC@AfterFiFiFi{ 479001600}% 12!
?     \else\BigIntCalcError:ThisCannotHappen%
      \fi
    \else
      \BIC@AfterFiFi{%
        \BIC@ProcessFac#1#2!479001600!%
      }%
    \fi
  \BIC@Fi
}
```

\BIC@ProcessFac
#1: n
#2: result

```
\def\BIC@ProcessFac#1!#2!{%
  \ifnum\BIC@PosCmp#1!12!=0 %
    \BIC@AfterFi{ #2}%
  \else
    \BIC@AfterFi{%
      \expandafter\BIC@@ProcessFac
      \romannumeral0\BIC@ProcessMul0!#2!#1!%
      !#1!%
    }%
  \BIC@Fi
}
```

\BIC@@ProcessFac
#1: result
#2: n

```
\def\BIC@@ProcessFac#1!#2!{%
  \expandafter\BIC@ProcessFac
  \romannumeral0\BIC@Dec#2!{}%
  !#1!%
}
```

2.18 Pow

\bigintcalcPow
#1: basis
#2: power

```
\def\bigintcalcPow#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Pow
  \bigintcalcNum{#1}!%
}
```

\BIC@Pow
#1: basis
#2: power

```
\def\BIC@Pow#1!#2{%
  \expandafter\expandafter\expandafter\BIC@PowSwitch
  \bigintcalcNum{#2}!#1!%
}
```

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

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

```
\def\BIC@PowSwitch#1#2!#3#4!{%
  \ifcase\ifx\\#2\\%
        \ifx#100 % y = 0
        \else\ifx#111 % y = 1
        \else\ifx#122 % y = 2
        \else4 % y > 2
        \fi\fi\fi
       \else
        \ifx#1-3 % y < 0
        \else4 % y > 2
        \fi
       \fi
    \BIC@AfterFi{ 1}% y = 0
  \or % y = 1
    \BIC@AfterFi{ #3#4}%
  \or % y = 2
    \ifx#3-% x < 0
      \BIC@AfterFiFi{%
        \BIC@ProcessMul0!#4!#4!%
      }%
    \else % x >= 0
      \BIC@AfterFiFi{%
        \BIC@ProcessMul0!#3#4!#3#4!%
      }%
    \fi
  \or % y < 0
    \ifcase\ifx\\#4\\%
          \ifx#300 % x = 0
          \else\ifx#311 % x = 1
          \else3 % x > 1
          \fi\fi
         \else
          \ifcase\BIC@MinusOne#3#4! %
            3 % |x| > 1
          \or
            2 % x = -1
? \else\BigIntCalcError:ThisCannotHappen%
          \fi
         \fi
      \BIC@AfterFiFi{ 0\BigIntCalcError:DivisionByZero}% x = 0
    \or % x = 1
      \BIC@AfterFiFi{ 1}% x = 1
    \or % x = -1
      \ifcase\BIC@ModTwo#2! % even(y)
```

```
      \BIC@AfterFiFiFi{ 1}%
    \or % odd(y)
      \BIC@AfterFiFiFi{ -1}%
  ?   \else\BigIntCalcError:ThisCannotHappen%
    \fi
  \or % |x| > 1
    \BIC@AfterFiFi{ 0}%
  ?  \else\BigIntCalcError:ThisCannotHappen%
  \fi
 \or % y > 2
  \ifcase\ifx\\#4\\%
        \ifx#300 % x = 0
        \else\ifx#311 % x = 1
        \else4 % x > 1
        \fi\fi
      \else
        \ifx#3-%
          \ifcase\BIC@MinusOne#3#4! %
            3 % x < -1
          \else
            2 % x = -1
          \fi
        \else
          4 % x > 1
        \fi
      \fi
    \BIC@AfterFiFi{ 0}% x = 0
  \or % x = 1
    \BIC@AfterFiFi{ 1}% x = 1
  \or % x = -1
    \ifcase\BIC@ModTwo#1#2! % even(y)
      \BIC@AfterFiFiFi{ 1}%
    \or % odd(y)
      \BIC@AfterFiFiFi{ -1}%
  ?   \else\BigIntCalcError:ThisCannotHappen%
    \fi
  \or % x < -1
    \ifcase\BIC@ModTwo#1#2! % even(y)
      \BIC@AfterFiFiFi{%
        \BIC@PowRec#4!#1#2!1!%
      }%
    \or % odd(y)
      \expandafter-\romannumeral0%
      \BIC@AfterFiFiFi{%
        \BIC@PowRec#4!#1#2!1!%
      }%
  ?   \else\BigIntCalcError:ThisCannotHappen%
    \fi
  \or % x > 1
    \BIC@AfterFiFi{%
      \BIC@PowRec#3#4!#1#2!1!%
    }%
  ?  \else\BigIntCalcError:ThisCannotHappen%
  \fi
? \else\BigIntCalcError:ThisCannotHappen%
 \BIC@Fi
}
```

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.

```
\def\BIC@ModTwo#1#2!{%
  \ifx\\#2\\%
    \ifodd#1 %
      \BIC@AfterFiFi1%
    \else
      \BIC@AfterFiFi0%
    \fi
  \else
    \BIC@AfterFi{%
      \BIC@ModTwo#2!%
    }%
  \BIC@Fi
}
```

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

```
\def\BIC@MinusOne#1#2!{%
  \ifx#1-%
    \BIC@@MinusOne#2!%
  \else
    0%
  \fi
}
```

\BIC@@MinusOne

```
\def\BIC@@MinusOne#1#2!{%
  \ifx#11%
    \ifx\\#2\\%
      1%
    \else
      0%
    \fi
  \else
    0%
  \fi
}
```

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)

```
\def\BIC@PowRec#1!#2#3!#4!{%
  \ifcase\ifx#21\ifx\\#3\\0 \else1 \fi\else1 \fi % y = 1
    \ifnum\BIC@PosCmp#1!#4!=1 % x > r
      \BIC@AfterFiFi{%
        \BIC@ProcessMul0!#1!#4!%
      }%
```

```
    \else
      \BIC@AfterFiFi{%
        \BIC@ProcessMul0!#4!#1!%
      }%
    \fi
  \or
    \ifcase\BIC@ModTwo#2#3! % even(y)
      \BIC@AfterFiFi{%
        \expandafter\BIC@@PowRec\romannumeral0%
        \BIC@@Shr#2#3!%
        !#1!#4!%
      }%
    \or % odd(y)
      \ifnum\BIC@PosCmp#1!#4!=1 % x > r
        \BIC@AfterFiFiFi{%
          \expandafter\BIC@@@PowRec\romannumeral0%
          \BIC@ProcessMul0!#1!#4!%
          !#1!#2#3!%
        }%
      \else
        \BIC@AfterFiFiFi{%
          \expandafter\BIC@@@PowRec\romannumeral0%
          \BIC@ProcessMul0!#1!#4!%
          !#1!#2#3!%
        }%
      \fi
? \else\BigIntCalcError:ThisCannotHappen%
    \fi
? \else\BigIntCalcError:ThisCannotHappen%
  \BIC@Fi
}
```

\BIC@@PowRec

#1: $y/2$
#2: x
#3: new r (r or $r * x$)

```
\def\BIC@@PowRec#1!#2!#3!{%
  \expandafter\BIC@PowRec\romannumeral0%
  \BIC@ProcessMul0!#2!#2!%
  !#1!#3!%
}
```

\BIC@@@PowRec

#1: $r * x$ #2: x #3: y

```
\def\BIC@@@PowRec#1!#2!#3!{%
  \expandafter\BIC@@PowRec\romannumeral0%
  \BIC@@Shr#3!%
  !#2!#1!%
}
```

2.19 Div

\bigintcalcDiv

#1: x
#2: y (divisor)

```
\def\bigintcalcDiv#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Div
  \bigintcalcNum{#1}!%
}
```

\BIC@Div

#1: x
#2: y

```
\def\BIC@Div#1!#2{%
```

```
  \expandafter\expandafter\expandafter\BIC@DivSwitchSign
  \bigintcalcNum{#2}!#1!%
}
```

\BigIntCalcDiv

```
\def\BigIntCalcDiv#1!#2!{%
  \romannumeral0%
  \BIC@DivSwitchSign#2!#1!%
}
```

\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

```
\def\BIC@DivSwitchSign#1#2!#3#4!{%
  \ifcase\BIC@Sgn#1#2! % y = 0
    \BIC@AfterFi{ 0\BigIntCalcError:DivisionByZero}%
  \or % y > 0
    \ifcase\BIC@Sgn#3#4! % x = 0
      \BIC@AfterFiFi{ 0}%
    \or % x > 0
      \BIC@AfterFiFi{%
        \BIC@DivSwitch{}#3#4!#1#2!%
      }%
    \else % x < 0
      \BIC@AfterFiFi{%
        \BIC@DivSwitch-#4!#1#2!%
      }%
    \fi
  \else % y < 0
    \ifcase\BIC@Sgn#3#4! % x = 0
      \BIC@AfterFiFi{ 0}%
    \or % x > 0
      \BIC@AfterFiFi{%
        \BIC@DivSwitch-#3#4!#2!%
      }%
    \else % x < 0
      \BIC@AfterFiFi{%
        \BIC@DivSwitch{}#4!#2!%
      }%
    \fi
  \BIC@Fi
}
```

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

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

#1: sign
#2: x
#3#4: y ($y \neq 0$)

```
\def\BIC@DivSwitch#1#2!#3#4!{%
  \ifcase\BIC@PosCmp#3#4!#2!% y = x
    \BIC@AfterFi{ #11}%
  \or % y > x
    \BIC@AfterFi{ 0}%
  \else % y < x
    \ifx\\#1\\%
    \else
      \expandafter-\romannumeral0%
    \fi
    \ifcase\ifx\\#4\\%
           \ifx#310 % y = 1
           \else\ifx#321 % y = 2
           \else\ifx#342 % y = 4
           \else3 % y > 2
           \fi\fi\fi
         \else
           3 % y > 2
         \fi
      \BIC@AfterFiFi{ #2}% y = 1
    \or % y = 2
      \BIC@AfterFiFi{%
        \BIC@@Shr#2!%
      }%
    \or % y = 4
      \BIC@AfterFiFi{%
        \expandafter\BIC@@Shr\romannumeral0%
          \BIC@@Shr#2!!%
      }%
    \or % y > 2
      \BIC@AfterFiFi{%
        \BIC@DivStartX#2!#3#4!!!%
      }%
? \else\BigIntCalcError:ThisCannotHappen%
    \fi
  \BIC@Fi
}
```

\BIC@ProcessDiv

#1#2: x
#3#4: y
#5: collect first digits of x
#6: corresponding digits of y

```
\def\BIC@DivStartX#1#2!#3#4!#5!#6!{%
  \ifx\\#4\\%
    \BIC@AfterFi{%
      \BIC@DivStartYii#6#3#4!{#5#1}#2=!%
    }%
  \else
    \BIC@AfterFi{%
      \BIC@DivStartX#2!#4!#5#1!#6#3!%
    }%
  \BIC@Fi
}
```

\BIC@DivStartYii

#1: y
#2: x, =

```
\def\BIC@DivStartYii#1!{%
  \expandafter\BIC@DivStartYiv\romannumeral0%
  \BIC@Shl#1!%
```

```
  !#1!%
}
```

\BIC@DivStartYiv #1: $2y$
#2: y
#3: x, =

```
\def\BIC@DivStartYiv#1!{%
  \expandafter\BIC@DivStartYvi\romannumeral0%
  \BIC@Shl#1!%
  !#1!%
}
```

\BIC@DivStartYvi #1: $4y$
#2: $2y$
#3: y
#4: x, =

```
\def\BIC@DivStartYvi#1!#2!{%
  \expandafter\BIC@DivStartYviii\romannumeral0%
  \BIC@AddXY#1!#2!!!%
  !#1!#2!%
}
```

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

```
\def\BIC@DivStartYviii#1!#2!{%
  \expandafter\BIC@DivStart\romannumeral0%
  \BIC@Shl#2!%
  !#1!#2!%
}
```

\BIC@DivStart #1: $8y$
#2: $6y$
#3: $4y$
#4: $2y$
#5: y
#6: x, =

```
\def\BIC@DivStart#1!#2!#3!#4!#5!#6!{%
  \BIC@ProcessDiv#6!!#5!#4!#3!#2!#1!=%
}
```

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

```
\def\BIC@ProcessDiv#1#2#3!#4!#5!{%
  \ifcase\BIC@PosCmp#5!#1!% y = #1
    \ifx#2=%
      \BIC@AfterFiFi{\BIC@DivCleanup{#41}}%
    \else
      \BIC@AfterFiFi{%
        \BIC@ProcessDiv#2#3!#41!#5!%
      }%
    \fi
  \or % y > #1
```

```
    \ifx#2=%
      \BIC@AfterFiFi{\BIC@DivCleanup{#40}}%
    \else
      \ifx\\#4\\%
        \BIC@AfterFiFiFi{%
          \BIC@ProcessDiv{#1#2}#3!!#5!%
        }%
      \else
        \BIC@AfterFiFiFi{%
          \BIC@ProcessDiv{#1#2}#3!#40!#5!%
        }%
      \fi
    \fi
  \else % y < #1
    \BIC@AfterFi{%
      \BIC@@ProcessDiv{#1}#2#3!#4!#5!%
    }%
  \BIC@Fi
}
```

\BIC@DivCleanup

#1: result
#2: garbage

```
\def\BIC@DivCleanup#1#2={ #1}%
```

\BIC@@ProcessDiv

```
\def\BIC@@ProcessDiv#1#2#3!#4!#5!#6!#7!{%
  \ifcase\BIC@PosCmp#7!#1!% 4y = #1
    \ifx#2=%
      \BIC@AfterFiFi{\BIC@DivCleanup{#44}}%
    \else
     \BIC@AfterFiFi{%
       \BIC@ProcessDiv#2#3!#44!#5!#6!#7!%
     }%
    \fi
  \or % 4y > #1
    \ifcase\BIC@PosCmp#6!#1!% 2y = #1
      \ifx#2=%
        \BIC@AfterFiFiFi{\BIC@DivCleanup{#42}}%
      \else
        \BIC@AfterFiFiFi{%
          \BIC@ProcessDiv#2#3!#42!#5!#6!#7!%
        }%
      \fi
    \or % 2y > #1
      \ifx#2=%
        \BIC@AfterFiFiFi{\BIC@DivCleanup{#41}}%
      \else
        \BIC@AfterFiFiFi{%
          \BIC@DivSub#1!#5!#2#3!#41!#5!#6!#7!%
        }%
      \fi
    \else % 2y < #1
      \BIC@AfterFiFi{%
        \expandafter\BIC@ProcessDivII\romannumeral0%
        \BIC@SubXY#1!#6!!!%
        !#2#3!#4!#5!23%
        #6!#7!%
      }%
    \fi
  \else % 4y < #1
    \BIC@AfterFi{%
      \BIC@@@ProcessDiv{#1}#2#3!#4!#5!#6!#7!%
```

```
    }%
  \BIC@Fi
}
```

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

```
\def\BIC@DivSub#1!#2!#3{%
  \expandafter\BIC@ProcessDiv\expandafter{%
    \romannumeral0%
    \BIC@SubXY#1!#2!!!%
    #3%
  }%
}
```

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

```
\def\BIC@ProcessDivII#1!#2#3!#4!#5!#6#7{%
  \ifcase\BIC@PosCmp#5!#1!% y = #1
    \ifx#2=%
      \BIC@AfterFiFi{\BIC@DivCleanup{#4#7}}%
    \else
      \BIC@AfterFiFi{%
        \BIC@ProcessDiv#2#3!#4#7!#5!%
      }%
    \fi
  \or % y > #1
    \ifx#2=%
      \BIC@AfterFiFi{\BIC@DivCleanup{#4#6}}%
    \else
      \BIC@AfterFiFi{%
        \BIC@ProcessDiv{#1#2}#3!#4#6!#5!%
      }%
    \fi
  \else % y < #1
    \ifx#2=%
      \BIC@AfterFiFi{\BIC@DivCleanup{#4#7}}%
    \else
      \BIC@AfterFiFi{%
        \BIC@DivSub#1!#5!#2#3!#4#7!#5!%
      }%
    \fi
  \BIC@Fi
}
```

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

```
\def\BIC@@@ProcessDiv#1#2#3!#4!#5!#6!#7!#8!#9!{%
  \ifcase\BIC@PosCmp#8!#1!% 6y = #1
    \ifx#2=%
      \BIC@AfterFiFi{\BIC@DivCleanup{#46}}%
    \else
      \BIC@AfterFiFi{%
        \BIC@ProcessDiv#2#3!#46!#5!#6!#7!#8!#9!%
```

```
    }%
   \fi
  \or % 6y > #1
   \BIC@AfterFi{%
    \expandafter\BIC@ProcessDivII\romannumeral0%
    \BIC@SubXY#1!#7!!!%
    !#2#3!#4!#5!45%
    #6!#7!#8!#9!%
   }%
  \else % 6y < #1
   \ifcase\BIC@PosCmp#9!#1!% 8y = #1
    \ifx#2=%
     \BIC@AfterFiFiFi{\BIC@DivCleanup{#48}}%
    \else
     \BIC@AfterFiFiFi{%
      \BIC@ProcessDiv#2#3!#48!#5!#6!#7!#8!#9!%
     }%
    \fi
   \or % 8y > #1
    \BIC@AfterFiFi{%
     \expandafter\BIC@ProcessDivII\romannumeral0%
     \BIC@SubXY#1!#8!!!%
     !#2#3!#4!#5!67%
     #6!#7!#8!#9!%
    }%
   \else % 8y < #1
    \BIC@AfterFiFi{%
     \expandafter\BIC@ProcessDivII\romannumeral0%
     \BIC@SubXY#1!#9!!!%
     !#2#3!#4!#5!89%
     #6!#7!#8!#9!%
    }%
   \fi
  \BIC@Fi
}
```

2.20 Mod

\bigintcalcMod

#1: x
#2: y

```
\def\bigintcalcMod#1{%
  \romannumeral0%
  \expandafter\expandafter\expandafter\BIC@Mod
  \bigintcalcNum{#1}!%
}
```

\BIC@Mod

#1: x
#2: y

```
\def\BIC@Mod#1!#2{%
  \expandafter\expandafter\expandafter\BIC@ModSwitchSign
  \bigintcalcNum{#2}!#1!%
}
```

\BigIntCalcMod

```
\def\BigIntCalcMod#1!#2!{%
  \romannumeral0%
  \BIC@ModSwitchSign#2!#1!%
}
```

\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

```
\def\BIC@ModSwitchSign#1#2!#3#4!{%
  \ifcase\ifx\\#2\\%
         \ifx#100 % y = 0
         \else1 % y > 0
         \fi
        \else
         \ifx#1-2 % y < 0
         \else1 % y > 0
         \fi
        \fi
    \BIC@AfterFi{ 0\BigIntCalcError:DivisionByZero}%
  \or % y > 0
    \ifcase\ifx\\#4\\\ifx#300 \else1 \fi\else1 \fi % x = 0
      \BIC@AfterFiFi{ 0}%
    \else
      \BIC@AfterFiFi{%
        \BIC@ModSwitch{}#3#4!#1#2!%
      }%
    \fi
  \else % y < 0
    \ifcase\ifx\\#4\\%
           \ifx#300 % x = 0
           \else1 % x > 0
           \fi
          \else
           \ifx#3-2 % x < 0
           \else1 % x > 0
           \fi
          \fi
      \BIC@AfterFiFi{ 0}%
    \or % x > 0
      \BIC@AfterFiFi{%
        \BIC@ModSwitch--#3#4!#2!%
      }%
    \else % x < 0
      \BIC@AfterFiFi{%
        \BIC@ModSwitch-#4!#2!%
      }%
    \fi
  \BIC@Fi
}
```

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

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

#1: sign
#2#3: x
#4#5: y

```
\def\BIC@ModSwitch#1#2#3!#4#5!{%
  \ifcase\ifx\\#5\\%
```

```
          \ifx#410 % y = 1
          \else\ifx#421 % y = 2
          \else2 % y > 2
          \fi\fi
        \else2 % y > 2
        \fi
    \BIC@AfterFi{ 0}% y = 1
  \or % y = 2
    \ifcase\BIC@ModTwo#2#3! % even(x)
      \BIC@AfterFiFi{ 0}%
    \or % odd(x)
      \BIC@AfterFiFi{ #11}%
? \else\BigIntCalcError:ThisCannotHappen%
    \fi
  \or % y > 2
    \ifx\\#1\\%
    \else
      \expandafter\BIC@Space\romannumeral0%
      \expandafter\BIC@ModMinus\romannumeral0%
    \fi
    \ifx#2-% x < 0
      \BIC@AfterFiFi{%
        \expandafter\expandafter\expandafter\BIC@ModX
        \bigintcalcSub{#2#3}{%
          \bigintcalcMul{#4#5}{\bigintcalcDiv{#2#3}{#4#5}}%
        }!#4#5!%
      }%
    \else % x > 0
      \BIC@AfterFiFi{%
        \expandafter\expandafter\expandafter\BIC@Space
        \bigintcalcSub{#2#3}{%
          \bigintcalcMul{#4#5}{\bigintcalcDiv{#2#3}{#4#5}}%
        }%
      }%
    \fi
? \else\BigIntCalcError:ThisCannotHappen%
  \BIC@Fi
}
```

\BIC@ModMinus

```
\def\BIC@ModMinus#1{%
  \ifx#10%
    \BIC@AfterFi{ 0}%
  \else
    \BIC@AfterFi{ -#1}%
  \BIC@Fi
}
```

\BIC@ModX #1#2: z
#3: x

```
\def\BIC@ModX#1#2!#3!{%
  \ifx#1-% z < 0
    \BIC@AfterFi{%
      \expandafter\BIC@Space\romannumeral0%
      \BIC@SubXY#3!#2!!!%
    }%
  \else % z >= 0
    \BIC@AfterFi{ #1#2}%
  \BIC@Fi
}
```

```
\BIC@AtEnd%
```

⟨/package⟩

3 Test

3.1 Catcode checks for loading

⟨*test1⟩

```
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\RangeCatcodeCheck#1#2#3{%
  \count@=#1\relax
  \loop
    \ifnum#3=\catcode\count@
    \else
```

```
      \errmessage{%
        Character \the\count@\space
        with wrong catcode \the\catcode\count@\space
        instead of \number#3%
      }%
    \fi
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input bigintcalc.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\%=14 %
  \LoadCommand
  \RangeCatcodeCheck{0}{36}{15}%
  \RangeCatcodeCheck{37}{37}{14}%
  \RangeCatcodeCheck{38}{47}{15}%
  \RangeCatcodeCheck{48}{57}{12}%
  \RangeCatcodeCheck{58}{63}{15}%
  \RangeCatcodeCheck{64}{64}{12}%
  \RangeCatcodeCheck{65}{90}{11}%
  \RangeCatcodeCheck{91}{91}{15}%
  \RangeCatcodeCheck{92}{92}{0}%
  \RangeCatcodeCheck{93}{96}{15}%
  \RangeCatcodeCheck{97}{122}{11}%
  \RangeCatcodeCheck{123}{255}{15}%
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 Macro tests

3.2.1 Preamble with test macro definitions

```
⟨*test2⟩
\NeedsTeXFormat{LaTeX2e}
\nofiles
\documentclass{article}
⟨noetex⟩\let\SavedNumexpr\numexpr
⟨noetex⟩\let\numexpr\UNDEFINED
\makeatletter
\chardef\BIC@TestMode=1 %
\makeatother
\usepackage{bigintcalc}[2016/05/16]
⟨noetex⟩\let\numexpr\SavedNumexpr
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\newcommand*{\TestSpaceAtEnd}[1]{%
⟨noetex⟩  \let\SavedNumexpr\numexpr
⟨noetex⟩  \let\numexpr\UNDEFINED
```

```
  \edef\resultA{#1}%
  \edef\resultB{#1 }%
⟨noetex⟩  \let\numexpr\SavedNumexpr
  \Expect*{\resultA\space}*{\resultB}%
}
\newcommand*{\TestResult}[2]{%
⟨noetex⟩  \let\SavedNumexpr\numexpr
⟨noetex⟩  \let\numexpr\UNDEFINED
  \edef\result{#1}%
⟨noetex⟩  \let\numexpr\SavedNumexpr
  \Expect*{\result}{#2}%
}
\newcommand*{\TestResultTwoExpansions}[2]{%
⟨*noetex⟩
  \begingroup
    \let\numexpr\UNDEFINED
    \expandafter\expandafter\expandafter
  \endgroup
⟨/noetex⟩
  \expandafter\expandafter\expandafter\Expect
  \expandafter\expandafter\expandafter{#1}{#2}%
}
\newcount\TestCount
⟨etex⟩\newcommand*{\TestArg}[1]{\numexpr#1\relax}
⟨noetex⟩\newcommand*{\TestArg}[1]{#1}
\newcommand*{\TestTeXDivide}[2]{%
  \TestCount=\TestArg{#1}\relax
  \divide\TestCount by \TestArg{#2}\relax
  \Expect*{\bigintcalcDiv{#1}{#2}}*{\the\TestCount}%
}
\newcommand*{\Test}[2]{%
  \TestResult{#1}{#2}%
  \TestResultTwoExpansions{#1}{#2}%
  \TestSpaceAtEnd{#1}%
}
\newcommand*{\TestExch}[2]{\Test{#2}{#1}}
\newcommand*{\TestInv}[2]{%
  \Test{\bigintcalcInv{#1}}{#2}%
}
\newcommand*{\TestAbs}[2]{%
  \Test{\bigintcalcAbs{#1}}{#2}%
}
\newcommand*{\TestSgn}[2]{%
  \Test{\bigintcalcSgn{#1}}{#2}%
}
\newcommand*{\TestMin}[3]{%
  \Test{\bigintcalcMin{#1}{#2}}{#3}%
}
\newcommand*{\TestMax}[3]{%
  \Test{\bigintcalcMax{#1}{#2}}{#3}%
}
\newcommand*{\TestCmp}[3]{%
  \Test{\bigintcalcCmp{#1}{#2}}{#3}%
}
\newcommand*{\TestOdd}[2]{%
  \Test{\bigintcalcOdd{#1}}{#2}%
  \edef\x{%
    \noexpand\Test{%
      \noexpand\BigIntCalcOdd
      \bigintcalcAbs{#1}!%
    }{#2}%
  }%
```

```
  \x
}
\newcommand*{\TestInc}[2]{%
  \Test{\bigintcalcInc{#1}}{#2}%
  \ifnum\bigintcalcSgn{#1}>-1 %
    \edef\x{%
      \noexpand\Test{%
        \noexpand\BigIntCalcInc\bigintcalcNum{#1}!%
      }{#2}%
    }%
    \x
  \fi
}
\newcommand*{\TestDec}[2]{%
  \Test{\bigintcalcDec{#1}}{#2}%
  \ifnum\bigintcalcSgn{#1}>0 %
    \edef\x{%
      \noexpand\Test{%
        \noexpand\BigIntCalcDec\bigintcalcNum{#1}!%
      }{#2}%
    }%
    \x
  \fi
}
\newcommand*{\TestAdd}[3]{%
  \Test{\bigintcalcAdd{#1}{#2}}{#3}%
  \ifnum\bigintcalcSgn{#1}>0 %
    \ifnum\bigintcalcSgn{#2}> 0 %
      \ifnum\bigintcalcCmp{#1}{#2}>0 %
        \edef\x{%
          \noexpand\Test{%
            \noexpand\BigIntCalcAdd
            \bigintcalcNum{#1}!\bigintcalcNum{#2}!%
          }{#3}%
        }%
        \x
      \else
        \edef\x{%
          \noexpand\Test{%
            \noexpand\BigIntCalcAdd
            \bigintcalcNum{#2}!\bigintcalcNum{#1}!%
          }{#3}%
        }%
        \x
      \fi
    \fi
  \fi
}
\newcommand*{\TestSub}[3]{%
  \Test{\bigintcalcSub{#1}{#2}}{#3}%
  \ifnum\bigintcalcSgn{#1}>0 %
    \ifnum\bigintcalcSgn{#2}> 0 %
      \ifnum\bigintcalcCmp{#1}{#2}>0 %
        \edef\x{%
          \noexpand\Test{%
            \noexpand\BigIntCalcSub
            \bigintcalcNum{#1}!\bigintcalcNum{#2}!%
          }{#3}%
        }%
        \x
      \fi
    \fi
```

```
  \fi
}
\newcommand*{\TestShl}[2]{%
  \Test{\bigintcalcShl{#1}}{#2}%
  \edef\x{%
    \noexpand\Test{%
      \noexpand\BigIntCalcShl\bigintcalcAbs{#1}!%
    }{\bigintcalcAbs{#2}}%
  }%
  \x
}
\newcommand*{\TestShr}[2]{%
  \Test{\bigintcalcShr{#1}}{#2}%
  \edef\x{%
    \noexpand\Test{%
      \noexpand\BigIntCalcShr\bigintcalcAbs{#1}!%
    }{\bigintcalcAbs{#2}}%
  }%
  \x
}
\newcommand*{\TestMul}[3]{%
  \Test{\bigintcalcMul{#1}{#2}}{#3}%
  \edef\x{%
    \noexpand\Test{%
      \noexpand\BigIntCalcMul
      \bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
    }{\bigintcalcAbs{#3}}%
  }%
  \x
}
\newcommand*{\TestSqr}[2]{%
  \Test{\bigintcalcSqr{#1}}{#2}%
}
\newcommand*{\TestFac}[2]{%
  \expandafter\TestExch\expandafter{%
    \the\numexpr#2%
  }{\bigintcalcFac{#1}}%
}
\newcommand*{\TestFacBig}[2]{%
  \Test{\bigintcalcFac{#1}}{#2}%
}
\newcommand*{\TestPow}[3]{%
  \Test{\bigintcalcPow{#1}{#2}}{#3}%
}
\newcommand*{\TestDiv}[3]{%
  \Test{\bigintcalcDiv{#1}{#2}}{#3}%
  \TestTeXDivide{#1}{#2}%
}
\newcommand*{\TestDivBig}[3]{%
  \Test{\bigintcalcDiv{#1}{#2}}{#3}%
  \edef\x{%
    \noexpand\Test{%
      \noexpand\BigIntCalcDiv\bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
    }{\bigintcalcAbs{#3}}%
  }%
}
\newcommand*{\TestMod}[3]{%
  \Test{\bigintcalcMod{#1}{#2}}{#3}%
  \ifcase\ifcase\bigintcalcSgn{#1} 0%
        \or
          \ifcase\bigintcalcSgn{#2} 1%
          \or 0%
```

```
          \else 1%
          \fi
        \else
          \ifcase\bigintcalcSgn{#2} 1%
          \or 1%
          \else 0%
          \fi
        \fi\relax
    \edef\x{%
      \noexpand\Test{%
        \noexpand\BigIntCalcMod
        \bigintcalcAbs{#1}!\bigintcalcAbs{#2}!%
      }{\bigintcalcAbs{#3}}%
    }%
    \x
  \fi
}
```

3.2.2 Time

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdfresettimer\endcsname\relax
\else
  \makeatletter
  \newcount\SummaryTime
  \newcount\TestTime
  \SummaryTime=\z@
  \newcommand*{\PrintTime}[2]{%
    \typeout{%
      [Time #1: \strip@pt\dimexpr\number#2sp\relax\space s]%
    }%
  }%
  \newcommand*{\StartTime}[1]{%
    \renewcommand*{\TimeDescription}{#1}%
    \pdfresettimer
  }%
  \newcommand*{\TimeDescription}{}%
  \newcommand*{\StopTime}{%
    \TestTime=\pdfelapsedtime
    \global\advance\SummaryTime\TestTime
    \PrintTime\TimeDescription\TestTime
  }%
  \let\saved@qstest\qstest
  \let\saved@endqstest\endqstest
  \def\qstest#1#2{%
    \saved@qstest{#1}{#2}%
    \StartTime{#1}%
  }%
  \def\endqstest{%
    \StopTime
    \saved@endqstest
  }%
  \AtEndDocument{%
    \PrintTime{summary}\SummaryTime
  }%
  \makeatother
\fi
```

3.2.3 Test sets

```
\makeatletter

\begin{qstest}{inv}{inv}%
  \TestInv{0}{0}%
  \TestInv{1}{-1}%
```

```
  \TestInv{-1}{1}%
  \TestInv{10}{-10}%
  \TestInv{-10}{10}%
  \TestInv{2147483647}{-2147483647}%
  \TestInv{-2147483647}{2147483647}%
  \TestInv{12345678901234567890}{-12345678901234567890}%
  \TestInv{-12345678901234567890}{12345678901234567890}%
  \TestInv{ 0 }{0}%
  \TestInv{ 1 }{-1}%
  \TestInv{--1}{-1}%
  \TestInv{\number\z@}{0}%
  \TestInv{\ifx\relax\relax1\fi}{-1}%
  \TestInv{\ifx\relax\relax-\fi\ifx234\else1\fi}{1}%
\end{qstest}

\begin{qstest}{abs}{abs}%
  \TestAbs{0}{0}%
  \TestAbs{1}{1}%
  \TestAbs{-1}{1}%
  \TestAbs{10}{10}%
  \TestAbs{-10}{10}%
  \TestAbs{2147483647}{2147483647}%
  \TestAbs{-2147483647}{2147483647}%
  \TestAbs{12345678901234567890}{12345678901234567890}%
  \TestAbs{-12345678901234567890}{12345678901234567890}%
  \TestAbs{ 0 }{0}%
  \TestAbs{ 1 }{1}%
  \TestAbs{--1}{1}%
  \TestAbs{-+-+1}{1}%
  \TestAbs{00000000000}{0}%
  \TestAbs{00000001000}{1000}%
  \TestAbs{\ifx\relax\relax 0\else 1\fi}{0}%
\end{qstest}

\begin{qstest}{sign}{sign}%
  \TestSgn{0}{0}%
  \TestSgn{1}{1}%
  \TestSgn{-1}{-1}%
  \TestSgn{10}{1}%
  \TestSgn{-10}{-1}%
  \TestSgn{2147483647}{1}%
  \TestSgn{-2147483647}{-1}%
  \TestSgn{12345678901234567890}{1}%
  \TestSgn{-12345678901234567890}{-1}%
  \TestSgn{ 0 }{0}%
  \TestSgn{ 2 }{1}%
  \TestSgn{ -2 }{-1}%
  \TestSgn{--2}{1}%
  \TestSgn{\number\z@}{0}%
  \TestSgn{\number\@ne}{1}%
  \TestSgn{\number\m@ne}{-1}%
  \TestSgn{%
    -+-+\number\z@\number\z@
    \iftrue1\fi\iftrue2\fi\iftrue3\fi
  }{1}%
\end{qstest}

\begin{qstest}{min}{min}%
  \TestMin{0}{1}{0}%
  \TestMin{1}{0}{0}%
  \TestMin{-10}{-20}{-20}%
  \TestMin{ 1 }{ 2 }{1}%
```

```
  \TestMin{ 2 }{ 1 }{1}%
  \TestMin{1}{1}{1}%
  \TestMin{\number\z@}{\number\@ne}{0}%
  \TestMin{\number\@ne}{\number\m@ne}{-1}%
\end{qstest}

\begin{qstest}{max}{max}%
  \TestMax{0}{1}{1}%
  \TestMax{1}{0}{1}%
  \TestMax{-10}{-20}{-10}%
  \TestMax{ 1 }{ 2 }{2}%
  \TestMax{ 2 }{ 1 }{2}%
  \TestMax{1}{1}{1}%
  \TestMax{\number\z@}{\number\@ne}{1}%
  \TestMax{\number\@ne}{\number\m@ne}{1}%
\end{qstest}

\begin{qstest}{cmp}{cmp}%
  \TestCmp{0}{0}{0}%
  \TestCmp{-21}{17}{-1}%
  \TestCmp{3}{4}{-1}%
  \TestCmp{-10}{-10}{0}%
  \TestCmp{-10}{-11}{1}%
  \TestCmp{100}{5}{1}%
  \TestCmp{9}{10}{-1}%
  \TestCmp{10}{9}{1}%
  \TestCmp{ 3 }{ 3 }{0}%
  \TestCmp{-9}{-10}{1}%
  \TestCmp{-10}{-9}{-1}%
  \TestCmp{-3}{-3}{0}%
  \TestCmp{0}{-2}{1}%
  \TestCmp{0}{2}{-1}%
  \TestCmp{2}{0}{1}%
  \TestCmp{-2}{0}{-1}%
  \TestCmp{12}{11}{1}%
  \TestCmp{11}{12}{-1}%
  \TestCmp{2147483647}{-2147483647}{1}%
  \TestCmp{-2147483647}{2147483647}{-1}%
  \TestCmp{2147483647}{2147483647}{0}%
  \TestCmp{\number\z@}{\number\@ne}{-1}%
  \TestCmp{\number\@ne}{\number\m@ne}{1}%
  \TestCmp{ 4 }{ 5 }{-1}%
  \TestCmp{ -3 }{ -7 }{1}%
\end{qstest}

\begin{qstest}{odd}{odd}
\tracingmacros=1
  \TestOdd{0}{0}%
  \TestOdd{1}{1}%
  \TestOdd{2}{0}%
  \TestOdd{3}{1}%
  \TestOdd{14}{0}%
  \TestOdd{15}{1}%
  \TestOdd{1234567890123456789 6}{0}%
  \TestOdd{1234567890123456789 7}{1}%
\end{qstest}

\begin{qstest}{inc}{inc}%
  \TestInc{0}{1}%
  \TestInc{1}{2}%
  \TestInc{-1}{0}%
  \TestInc{10}{11}%
```

```
  \TestInc{-10}{-9}%
  \TestInc{ 3 }{4}%
  \TestInc{999}{1000}%
  \TestInc{-1000}{-999}%
  \TestInc{129}{130}%
  \TestInc{2147483646}{2147483647}%
  \TestInc{-2147483647}{-2147483646}%
  \TestInc{12345678901234567890}{12345678901234567891}%
  \TestInc{9999999999999999999}{10000000000000000000}%
  \TestInc{-12345678901234567891}{-12345678901234567890}%
  \TestInc{-10000000000000000000}{-9999999999999999999}%
\end{qstest}

\begin{qstest}{dec}{dec}%
  \TestDec{0}{-1}%
  \TestDec{1}{0}%
  \TestDec{-1}{-2}%
  \TestDec{10}{9}%
  \TestDec{-10}{-11}%
  \TestDec{1000}{999}%
  \TestDec{-999}{-1000}%
  \TestDec{130}{129}%
  \TestDec{2147483647}{2147483646}%
  \TestDec{-2147483646}{-2147483647}%
  \TestDec{12345678901234567891}{12345678901234567890}%
  \TestDec{10000000000000000000}{99999999999999999999}%
  \TestDec{-12345678901234567890}{-12345678901234567891}%
  \TestDec{-99999999999999999999}{-10000000000000000000}%
\end{qstest}

\begin{qstest}{add}{add}%
  \TestAdd{0}{0}{0}%
  \TestAdd{1}{0}{1}%
  \TestAdd{0}{1}{1}%
  \TestAdd{1}{2}{3}%
  \TestAdd{-1}{-1}{-2}%
  \TestAdd{2147483646}{1}{2147483647}%
  \TestAdd{-2147483647}{2147483647}{0}%
  \TestAdd{20}{-5}{15}%
  \TestAdd{-4}{-1}{-5}%
  \TestAdd{-1}{-4}{-5}%
  \TestAdd{-4}{1}{-3}%
  \TestAdd{-1}{4}{3}%
  \TestAdd{4}{-1}{3}%
  \TestAdd{1}{-4}{-3}%
  \TestAdd{-4}{-1}{-5}%
  \TestAdd{-1}{-4}{-5}%
  \TestAdd{ -4 }{ -1 }{-5}%
  \TestAdd{ -1 }{ -4 }{-5}%
  \TestAdd{ -4 }{ 1 }{-3}%
  \TestAdd{ -1 }{ 4 }{3}%
  \TestAdd{ 4 }{ -1 }{3}%
  \TestAdd{ 1 }{ -4 }{-3}%
  \TestAdd{ -4 }{ -1 }{-5}%
  \TestAdd{ -1 }{ -4 }{-5}%
  \TestAdd{876543210}{111111111}{987654321}%
  \TestAdd{999999999}{2}{1000000001}%
\end{qstest}

\begin{qstest}{sub}{sub}
  \TestSub{0}{0}{0}%
  \TestSub{1}{0}{1}%
```

```
  \TestSub{1}{2}{-1}%
  \TestSub{-1}{-1}{0}%
  \TestSub{2147483646}{-1}{2147483647}%
  \TestSub{-2147483647}{-2147483647}{0}%
  \TestSub{-4}{-1}{-3}%
  \TestSub{-1}{-4}{3}%
  \TestSub{-4}{1}{-5}%
  \TestSub{-1}{4}{-5}%
  \TestSub{4}{-1}{5}%
  \TestSub{1}{-4}{5}%
  \TestSub{-4}{-1}{-3}%
  \TestSub{-1}{-4}{3}%
  \TestSub{ -4 }{ -1 }{-3}%
  \TestSub{ -1 }{ -4 }{3}%
  \TestSub{ -4 }{ 1 }{-5}%
  \TestSub{ -1 }{ 4 }{-5}%
  \TestSub{ 4 }{ -1 }{5}%
  \TestSub{ 1 }{ -4 }{5}%
  \TestSub{ -4 }{ -1 }{-3}%
  \TestSub{ -1 }{ -4 }{3}%
  \TestSub{1000000000}{2}{999999998}%
  \TestSub{987654321}{111111111}{876543210}%
\end{qstest}

\begin{qstest}{shl}{shl}
  \TestShl{0}{0}%
  \TestShl{1}{2}%
  \TestShl{2}{4}%
  \TestShl{5621}{11242}%
  \TestShl{1073741823}{2147483646}%
\end{qstest}

\begin{qstest}{shr}{shr}
  \TestShr{0}{0}%
  \TestShr{1}{0}%
  \TestShr{2}{1}%
  \TestShr{3}{1}%
  \TestShr{4}{2}%
  \TestShr{5}{2}%
  \TestShr{6}{3}%
  \TestShr{7}{3}%
  \TestShr{8}{4}%
  \TestShr{9}{4}%
  \TestShr{10}{5}%
  \TestShr{11}{5}%
  \TestShr{12}{6}%
  \TestShr{13}{6}%
  \TestShr{14}{7}%
  \TestShr{15}{7}%
  \TestShr{16}{8}%
  \TestShr{17}{8}%
  \TestShr{18}{9}%
  \TestShr{19}{9}%
  \TestShr{20}{10}%
  \TestShr{21}{10}%
  \TestShr{22}{11}%
  \TestShr{11241}{5620}%
  \TestShr{73054202}{36527101}%
  \TestShr{2147483646}{1073741823}%
\end{qstest}

\begin{qstest}{mul}{mul}
```

```
  \TestMul{0}{0}{0}%
  \TestMul{1}{0}{0}%
  \TestMul{0}{1}{0}%
  \TestMul{1}{1}{1}%
  \TestMul{3}{1}{3}%
  \TestMul{1}{-3}{-3}%
  \TestMul{-4}{-5}{20}%
  \TestMul{3}{7}{21}%
  \TestMul{7}{3}{21}%
  \TestMul{3}{-7}{-21}%
  \TestMul{7}{-3}{-21}%
  \TestMul{-3}{7}{-21}%
  \TestMul{-7}{3}{-21}%
  \TestMul{-3}{-7}{21}%
  \TestMul{-7}{-3}{21}%
  \TestMul{12}{11}{132}%
  \TestMul{999}{333}{332667}%
  \TestMul{1000}{4321}{4321000}%
  \TestMul{12345}{173955}{2147474475}%
  \TestMul{1073741823}{2}{2147483646}%
  \TestMul{2}{1073741823}{2147483646}%
  \TestMul{-1073741823}{2}{-2147483646}%
  \TestMul{2}{-1073741823}{-2147483646}%
  \TestMul{6706022400}{13}{87178291200}%
\end{qstest}

\begin{qstest}{sqr}{sqr}
  \TestSqr{0}{0}%
  \TestSqr{1}{1}%
  \TestSqr{2}{4}%
  \TestSqr{3}{9}%
  \TestSqr{4}{16}%
  \TestSqr{9}{81}%
  \TestSqr{10}{100}%
  \TestSqr{46340}{2147395600}%
  \TestSqr{-1}{1}%
  \TestSqr{-2}{4}%
  \TestSqr{-46340}{2147395600}%
\end{qstest}

\begin{qstest}{fac}{fac}
  \TestFac{0}{1}%
  \TestFac{1}{1}%
  \TestFac{2}{2}%
  \TestFac{3}{2*3}%
  \TestFac{4}{2*3*4}%
  \TestFac{5}{2*3*4*5}%
  \TestFac{6}{2*3*4*5*6}%
  \TestFac{7}{2*3*4*5*6*7}%
  \TestFac{8}{2*3*4*5*6*7*8}%
  \TestFac{9}{2*3*4*5*6*7*8*9}%
  \TestFac{10}{2*3*4*5*6*7*8*9*10}%
  \TestFac{11}{2*3*4*5*6*7*8*9*10*11}%
  \TestFac{12}{2*3*4*5*6*7*8*9*10*11*12}%
  \TestFacBig{13}{6227020800}%
  \TestFacBig{14}{87178291200}%
  \TestFacBig{15}{1307674368000}%
  \TestFacBig{16}{20922789888000}%
  \TestFacBig{17}{355687428096000}%
  \TestFacBig{18}{6402373705728000}%
  \TestFacBig{19}{121645100408832000}%
  \TestFacBig{20}{2432902008176640000}%
```

```
  \TestFacBig{21}{5109094217170944000}%
  \TestFacBig{22}{112400072777760768000}%
\end{qstest}

\begin{qstest}{pow}{pow}
  \TestPow{-2}{0}{1}%
  \TestPow{-1}{0}{1}%
  \TestPow{0}{0}{1}%
  \TestPow{1}{0}{1}%
  \TestPow{2}{0}{1}%
  \TestPow{3}{0}{1}%
  \TestPow{-2}{1}{-2}%
  \TestPow{-1}{1}{-1}%
  \TestPow{1}{1}{1}%
  \TestPow{2}{1}{2}%
  \TestPow{3}{1}{3}%
  \TestPow{-2}{2}{4}%
  \TestPow{-1}{2}{1}%
  \TestPow{0}{2}{0}%
  \TestPow{1}{2}{1}%
  \TestPow{2}{2}{4}%
  \TestPow{3}{2}{9}%
  \TestPow{0}{1}{0}%
  \TestPow{1}{-2}{1}%
  \TestPow{1}{-1}{1}%
  \TestPow{-1}{-2}{1}%
  \TestPow{-1}{-1}{-1}%
  \TestPow{-1}{3}{-1}%
  \TestPow{-1}{4}{1}%
  \TestPow{-2}{-1}{0}%
  \TestPow{-2}{-2}{0}%
  \TestPow{2}{3}{8}%
  \TestPow{2}{4}{16}%
  \TestPow{2}{5}{32}%
  \TestPow{2}{6}{64}%
  \TestPow{2}{7}{128}%
  \TestPow{2}{8}{256}%
  \TestPow{2}{9}{512}%
  \TestPow{2}{10}{1024}%
  \TestPow{-2}{3}{-8}%
  \TestPow{-2}{4}{16}%
  \TestPow{-2}{5}{-32}%
  \TestPow{-2}{6}{64}%
  \TestPow{-2}{7}{-128}%
  \TestPow{-2}{8}{256}%
  \TestPow{-2}{9}{-512}%
  \TestPow{-2}{10}{1024}%
  \TestPow{3}{3}{27}%
  \TestPow{3}{4}{81}%
  \TestPow{3}{5}{243}%
  \TestPow{-3}{3}{-27}%
  \TestPow{-3}{4}{81}%
  \TestPow{-3}{5}{-243}%
  \TestPow{2}{30}{1073741824}%
  \TestPow{-3}{19}{-1162261467}%
  \TestPow{5}{13}{1220703125}%
  \TestPow{-7}{11}{-1977326743}%
\end{qstest}

\begin{qstest}{div}{div}
  \TestDiv{1}{1}{1}%
  \TestDiv{2}{1}{2}%
```

```
\TestDiv{-2}{1}{-2}%
\TestDiv{2}{-1}{-2}%
\TestDiv{-2}{-1}{2}%
\TestDiv{15}{2}{7}%
\TestDiv{-16}{2}{-8}%
\TestDiv{1}{2}{0}%
\TestDiv{1}{3}{0}%
\TestDiv{2}{3}{0}%
\TestDiv{-2}{3}{0}%
\TestDiv{2}{-3}{0}%
\TestDiv{-2}{-3}{0}%
\TestDiv{13}{3}{4}%
\TestDiv{-13}{-3}{4}%
\TestDiv{-13}{3}{-4}%
\TestDiv{-6}{5}{-1}%
\TestDiv{-5}{5}{-1}%
\TestDiv{-4}{5}{0}%
\TestDiv{-3}{5}{0}%
\TestDiv{-2}{5}{0}%
\TestDiv{-1}{5}{0}%
\TestDiv{0}{5}{0}%
\TestDiv{1}{5}{0}%
\TestDiv{2}{5}{0}%
\TestDiv{3}{5}{0}%
\TestDiv{4}{5}{0}%
\TestDiv{5}{5}{1}%
\TestDiv{6}{5}{1}%
\TestDiv{-5}{4}{-1}%
\TestDiv{-4}{4}{-1}%
\TestDiv{-3}{4}{0}%
\TestDiv{-2}{4}{0}%
\TestDiv{-1}{4}{0}%
\TestDiv{0}{4}{0}%
\TestDiv{1}{4}{0}%
\TestDiv{2}{4}{0}%
\TestDiv{3}{4}{0}%
\TestDiv{4}{4}{1}%
\TestDiv{5}{4}{1}%
\TestDiv{12345}{678}{18}%
\TestDiv{32372}{5952}{5}%
\TestDiv{284271294}{18162}{15651}%
\TestDiv{217652429}{12561}{17327}%
\TestDiv{462028434}{5439}{84947}%
\TestDiv{2147483647}{1000}{2147483}%
\TestDiv{2147483647}{-1000}{-2147483}%
\TestDiv{-2147483647}{1000}{-2147483}%
\TestDiv{-2147483647}{-1000}{2147483}%
\TestDiv{0}{3}{0}%
\TestDiv{1}{3}{0}%
\TestDiv{2}{3}{0}%
\TestDiv{3}{3}{1}%
\TestDiv{4}{3}{1}%
\TestDiv{5}{3}{1}%
\TestDiv{6}{3}{2}%
\TestDiv{7}{3}{2}%
\TestDiv{8}{3}{2}%
\TestDiv{9}{3}{3}%
\TestDiv{10}{3}{3}%
\TestDiv{11}{3}{3}%
\TestDiv{12}{3}{4}%
\TestDiv{13}{3}{4}%
\TestDiv{14}{3}{4}%
```

```
  \TestDiv{15}{3}{5}%
  \TestDiv{16}{3}{5}%
  \TestDiv{17}{3}{5}%
  \TestDiv{18}{3}{6}%
  \TestDiv{19}{3}{6}%
  \TestDiv{20}{3}{6}%
  \TestDiv{21}{3}{7}%
  \TestDiv{22}{3}{7}%
  \TestDiv{23}{3}{7}%
  \TestDiv{24}{3}{8}%
  \TestDiv{25}{3}{8}%
  \TestDiv{26}{3}{8}%
  \TestDiv{27}{3}{9}%
  \TestDiv{28}{3}{9}%
  \TestDiv{29}{3}{9}%
  \TestDiv{30}{3}{10}%
  \TestDiv{31}{3}{10}%
  \TestDivBig{17363436332507}{24702}{702916214}%
\end{qstest}

\begin{qstest}{mod}{mod}
  \TestMod{-6}{5}{4}%
  \TestMod{-5}{5}{0}%
  \TestMod{-4}{5}{1}%
  \TestMod{-3}{5}{2}%
  \TestMod{-2}{5}{3}%
  \TestMod{-1}{5}{4}%
  \TestMod{0}{5}{0}%
  \TestMod{1}{5}{1}%
  \TestMod{2}{5}{2}%
  \TestMod{3}{5}{3}%
  \TestMod{4}{5}{4}%
  \TestMod{5}{5}{0}%
  \TestMod{6}{5}{1}%
  \TestMod{-5}{4}{3}%
  \TestMod{-4}{4}{0}%
  \TestMod{-3}{4}{1}%
  \TestMod{-2}{4}{2}%
  \TestMod{-1}{4}{3}%
  \TestMod{0}{4}{0}%
  \TestMod{1}{4}{1}%
  \TestMod{2}{4}{2}%
  \TestMod{3}{4}{3}%
  \TestMod{4}{4}{0}%
  \TestMod{5}{4}{1}%
  \TestMod{-6}{-5}{-1}%
  \TestMod{-5}{-5}{0}%
  \TestMod{-4}{-5}{-4}%
  \TestMod{-3}{-5}{-3}%
  \TestMod{-2}{-5}{-2}%
  \TestMod{-1}{-5}{-1}%
  \TestMod{0}{-5}{0}%
  \TestMod{1}{-5}{-4}%
  \TestMod{2}{-5}{-3}%
  \TestMod{3}{-5}{-2}%
  \TestMod{4}{-5}{-1}%
  \TestMod{5}{-5}{0}%
  \TestMod{6}{-5}{-4}%
  \TestMod{-5}{-4}{-1}%
  \TestMod{-4}{-4}{0}%
  \TestMod{-3}{-4}{-3}%
  \TestMod{-2}{-4}{-2}%
```

```
  \TestMod{-1}{-4}{-1}%
  \TestMod{0}{-4}{0}%
  \TestMod{1}{-4}{-3}%
  \TestMod{2}{-4}{-2}%
  \TestMod{3}{-4}{-1}%
  \TestMod{4}{-4}{0}%
  \TestMod{5}{-4}{-3}%
  \TestMod{2147483647}{1000}{647}%
  \TestMod{2147483647}{-1000}{-353}%
  \TestMod{-2147483647}{1000}{353}%
  \TestMod{-2147483647}{-1000}{-647}%
  \TestMod{ 0 }{ 4 }{0}%
  \TestMod{ 1 }{ 4 }{1}%
  \TestMod{ -1 }{ 4 }{3}%
  \TestMod{ 0 }{ -4 }{0}%
  \TestMod{ 1 }{ -4 }{-3}%
  \TestMod{ -1 }{ -4 }{-1}%
  \TestMod{18362}{25}{12}%
\end{qstest}

\newcommand*{\TestError}[2]{%
  \begingroup
    \expandafter\def\csname BigIntCalcError:#1\endcsname{}%
    \Expect*{#2}{0}%
    \expandafter\def\csname BigIntCalcError:#1\endcsname{ERROR}%
    \Expect*{#2}{0ERROR}%
  \endgroup
}
\begin{qstest}{error}{error}
  \TestError{FacNegative}{\bigintcalcFac{-1}}%
  \TestError{FacNegative}{\bigintcalcFac{-2147483647}}%
  \TestError{DivisionByZero}{\bigintcalcPow{0}{-1}}%
  \TestError{DivisionByZero}{\bigintcalcDiv{1}{0}}%
  \TestError{DivisionByZero}{\bigintcalcMod{1}{0}}%
\end{qstest}

\begin{document}
\end{document}
⟨/test2⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/oberdiek/bigintcalc.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/bigintcalc.pdf Documentation.

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

CTAN:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:tds/tds.pdf). Directories with `texmf` in their name are usually organized this way.

[1] http://ctan.org/pkg/bigintcalc

4.2 Bundle installation

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

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

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

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

4.3 Package installation

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

```
tex bigintcalc.dtx
```

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

```
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 (teTEX, mikTEX, ...) relies on file name databases, you must refresh these. For example, teTEX users run texhash or mktexlsr.

4.5 Some details for the interested

Unpacking with LATEX. The .dtx chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

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

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

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

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

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

An example follows how to generate the documentation with pdfLaTeX:

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

5 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements caption and description are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is bigintcalc.xml.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='bigintcalc'>
  <name>bigintcalc</name>
  <caption>Integer calculations on very large numbers.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2007,2011,2012'/>
  <license type='lppl1.3'/>
  <version number='1.4'/>
  <description>
    This package provides expandable arithmetic operations
    with big integers that can exceed TeX's number limits.
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref> bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/bigintcalc.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/bigintcalc.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

6 History

[2007/09/27 v1.0]

- First version.

[2007/11/11 v1.1]

- Use of package pdftexcmds for LuaTeX support.

[2011/01/30 v1.2]

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

[2012/04/08 v1.3]

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

[2016/05/16 v1.4]

- Documentation updates.

7 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

\# 1774
\% 1850
\@ 1775, 1848
\@firstofone 1783, 1786
\@firstoftwo 159, 167, 176
\@gobble 1780, 1788
\@ne 2181, 2196, 2197, 2207, 2208, 2233, 2234
\@nil . . 155, 157, 164, 172, 187, 190, 195
\@secondoftwo 161, 169, 179
\@undefined 58
\\ 192, 202, 321, 322, 330, 353, 451, 463, 496, 509, 521, 556, 558, 647, 648, 658, 659, 676, 738, 739, 749, 750, 767, 852, 863, 910, 924, 987, 998, 1015, 1135, 1221, 1246, 1274, 1322, 1343, 1353, 1447, 1451, 1479, 1525, 1672, 1683, 1691, 1713, 1729, 1849
\{ 1772
\} 1773

A

\advance 1813, 1821, 1836, 2109
\aftergroup 29
\AtEndDocument 2122

B

\begin 2129, 2147, 2166, 2189, 2200, 2211, 2239, 2251, 2269, 2286, 2315, 2342, 2350, 2379, 2406, 2420, 2446, 2501, 2586, 2656, 2664
\BIC@@@@Shl 870, 874
\BIC@@@@Shr . . 943, 944, 950, 951, 960
\BIC@@@Dec 531, 532, 550, 555
\BIC@@@Inc 473, 474, 490, 495
\BIC@@@PowRec . . . 1373, 1379, 1394
\BIC@@@ProcessDiv 1578, 1616
\BIC@@@Shl 854, 862, 877, 881
\BIC@@@Shr 914, 919, 923, 961
\BIC@@AddDigit . . . 699, 700, 710, 789
\BIC@@Cmp 291, 294, 380, 412
\BIC@@Dec 511, 519, 568
\BIC@@Expand 155, 157
\BIC@@Inc 453, 461, 504
\BIC@@MinMax 376, 379
\BIC@@MinusOne 1336, 1341
\BIC@@PowRec 1366, 1389, 1395
\BIC@@ProcessDiv 1537, 1542
\BIC@@ProcessFac 1200, 1206
\BIC@@ProcessTim 1007, 1014
\BIC@@Shl 836, 841, 848, 851
\BIC@@Shr 894, 898, 904, 906, 1367, 1396, 1463, 1467, 1468
\BIC@@Sqr 1154, 1156, 1159
\BIC@@SubDigit 788, 799
\BIC@@TestMode 123
\BIC@@Tim 986
\BIC@Abs 258, 261
\BIC@Add 574, 577, 583
\BIC@AddCarry0 717
\BIC@AddCarry10 718
\BIC@AddCarry[1-9] 719
\BIC@AddDigit 681, 686, 697
\BIC@AddResult 680, 690
\BIC@AddSwitch 579, 586
\BIC@AddXY . . 592, 596, 630, 634, 641, 646, 837, 842, 849, 1043, 1501
\BIC@AfterFi 131, 194, 231, 344, 381, 383, 413, 415, 419, 432, 434, 438, 452, 456, 503, 510, 514, 567, 677, 684, 712, 714, 768, 773, 801, 806, 834, 840, 853, 857, 864, 869, 876, 880, 893, 897, 908, 918, 988, 992, 999, 1006, 1016, 1020, 1027, 1029, 1077, 1136, 1141, 1169, 1197, 1199, 1232, 1234, 1329, 1414, 1443, 1445, 1480, 1484, 1536, 1577, 1626, 1681, 1720, 1754, 1756, 1761, 1766
\BIC@AfterFiFi 132, 203, 207, 213, 234, 238, 297, 301, 307, 311, 323, 325, 331, 335, 349, 398, 400, 464, 466, 472, 489, 498, 500, 522, 524, 530, 549, 564, 649, 653, 669, 740, 744, 760, 911, 913, 926, 933, 942, 949, 1035, 1040, 1080, 1105, 1189, 1237, 1241, 1259, 1261, 1270, 1290, 1292, 1313, 1324, 1326, 1355, 1359, 1365, 1417, 1419, 1423, 1429, 1431, 1435, 1460, 1462, 1466,

1471, 1515, 1517, 1523, 1545, 1547, 1569, 1592, 1594, 1600, 1602, 1608, 1610, 1619, 1621, 1642, 1649, 1684, 1686, 1700, 1702, 1706, 1723, 1725, 1735, 1742
\BIC@AfterFiFiFi 133, 218, 222, 354, 358, 559, 561, 591, 595, 601, 604, 608, 616, 618, 623, 629, 633, 660, 664, 751, 755, 1083, 1087, 1094, 1098, 1109, 1113, 1119, 1123, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1264, 1266, 1295, 1297, 1302, 1307, 1372, 1378, 1526, 1530, 1554, 1556, 1562, 1564, 1635, 1637
\BIC@AtEnd 95, 96, 117, 1769
\BIC@Cmp 287, 290
\BIC@CmpDiff 323, 342
\BIC@CmpLength 312, 318, 320
\BIC@CmpResult 326, 332, 341
\BIC@Dec 421, 435, 448, 508, 1208
\BIC@DecSwitch 427, 430
\BIC@Div 1401, 1404
\BIC@DivCleanup 1515, 1523, 1541, 1545, 1554, 1562, 1592, 1600, 1608, 1619, 1635
\BIC@DivStart 1505, 1509
\BIC@DivStartX 1472, 1478, 1485
\BIC@DivStartYii 1481, 1489
\BIC@DivStartYiv 1490, 1494
\BIC@DivStartYvi 1495, 1499
\BIC@DivStartYviii 1500, 1504
\BIC@DivSub 1565, 1582, 1611
\BIC@DivSwitch 1420, 1424, 1432, 1436, 1441
\BIC@DivSwitchSign . 1405, 1410, 1412
\BIC@DoAdd 650, 654, 675
\BIC@DoSub 741, 745, 766
\BIC@Expand 153, 183, 250
\BIC@Fac 1164, 1167
\BIC@Fi . . . 130, 131, 132, 133, 197, 227, 242, 315, 339, 363, 384, 404, 423, 442, 459, 493, 506, 517, 553, 570, 638, 673, 688, 715, 764, 777, 807, 844, 860, 872, 883, 900, 921, 956, 995, 1012, 1023, 1048, 1128, 1145, 1193, 1204, 1319, 1332, 1387, 1439, 1476, 1487, 1539, 1580, 1614, 1656, 1710, 1750, 1757, 1767
\BIC@Inc 416, 440, 445, 450
\BIC@IncSwitch 408, 411
\BIC@MinMax 367, 372, 375
\BIC@MinusOne 1252, 1281, 1334
\BIC@Mod 1660, 1663
\BIC@ModMinus 1732, 1752
\BIC@ModSwitch 1687, 1703, 1707, 1712
\BIC@ModSwitchSign 1664, 1669, 1671
\BIC@ModTwo 1263, 1294, 1301, 1321, 1364, 1722
\BIC@ModX 1736, 1759
\BIC@Mul 1068, 1071
\BIC@MulDigit[3-9] 1050
\BIC@MulSwitch 1072, 1075
\BIC@Normalize 200, 247
\BIC@NormalizeDigits . . . 223, 239, 244
\BIC@NormalizeZero 219, 229
\BIC@Odd 388, 393, 395
\BIC@PosCmp . 317, 590, 600, 615, 628, 1082, 1093, 1108, 1118, 1171, 1196, 1354, 1371, 1442, 1513, 1543, 1552, 1590, 1617, 1633
\BIC@Pow 1213, 1216
\BIC@PowRec 1303, 1308, 1314, 1352, 1390
\BIC@PowSwitch 1217, 1220
\BIC@ProcessDiv 1478, 1510, 1512, 1548, 1557, 1583, 1595, 1603, 1622, 1638
\BIC@ProcessDivII 1570, 1589, 1627, 1643, 1650
\BIC@ProcessDivIV 1616
\BIC@ProcessFac 1190, 1195, 1207
\BIC@ProcessMul 1084, 1088, 1095, 1099, 1110, 1114, 1120, 1124, 1132, 1134, 1160, 1201, 1238, 1242, 1356, 1360, 1374, 1380, 1391
\BIC@ProcessTim . 989, 997, 1017, 1021
\BIC@Sgn 271, 274, 431, 1076, 1079, 1104, 1413, 1416, 1428
\BIC@Shl . . 829, 832, 1491, 1496, 1506
\BIC@Shr 887, 890
\BIC@ShrDigit[00-19] 963
\BIC@ShrResult 927, 928, 934, 935, 958
\BIC@Space 134, 184, 253, 263, 265, 678, 692, 694, 783, 802, 865, 1000, 1030, 1036, 1041, 1137, 1731, 1743, 1762
\BIC@Sqr 1149, 1152
\BIC@StripHexSpace 187, 190
\BIC@SubCarry0 809
\BIC@SubCarry10 810
\BIC@SubCarry[1-9] 811
\BIC@SubDigit 770, 775, 786
\BIC@SubResult 769, 779
\BIC@SubXY 605, 609, 619, 624, 644, 737, 1571, 1585, 1628, 1644, 1651, 1763
\BIC@Temp 719, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 811, 818, 819, 820, 821, 822, 823, 824, 825, 826, 963, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 1050, 1059, 1060, 1061, 1062, 1063, 1064, 1065
\BIC@TestMode 123, 1877
\BIC@Tim 986, 1138, 1143
\BIC@TimDigit 1002, 1009, 1025
\bigintcalcAbs . . . 4, 256, 389, 1927, 1946, 2017, 2018, 2026, 2027, 2036, 2037, 2063, 2064, 2084, 2085

\BigIntCalcAdd 7, 640, 1980, 1988
\bigintcalcAdd . 5, 572, 1138, 1143, 1974
\bigintcalcCmp 4, 285, 1939, 1977, 2001
\BigIntCalcDec 7, 447, 1967
\bigintcalcDec 5, 425, 1963
\BigIntCalcDiv 7, 1408, 2063
\bigintcalcDiv 6, 1399, 1738, 1745, 1915, 2056, 2060, 2660
\BigIntCalcError 484, 534, 544, 723, 814, 1055, 1169, 1186, 1256, 1259, 1267, 1271, 1298, 1310, 1316, 1318, 1384, 1386, 1414, 1474, 1681, 1726, 1749
\bigintcalcFac 6, 1162, 2047, 2050, 2657, 2658
\BigIntCalcInc 7, 444, 1956
\bigintcalcInc 5, 406, 1952
\bigintcalcInv 4, 252, 1924
\bigintcalcMax 4, 370, 1936
\bigintcalcMin 4, 365, 1933
\BigIntCalcMod 7, 1667, 2083
\bigintcalcMod 6, 1658, 2068, 2661
\BigIntCalcMul 7, 1130, 2035
\bigintcalcMul 6, 1066, 1738, 1745, 2032
\bigintcalcNum . . . 3, 245, 254, 259, 272, 288, 292, 368, 373, 377, 409, 428, 575, 579, 584, 830, 888, 1069, 1073, 1150, 1165, 1214, 1218, 1402, 1406, 1661, 1665, 1956, 1967, 1981, 1989, 2005
\BigIntCalcOdd 7, 391, 1945
\bigintcalcOdd 5, 386, 1942
\bigintcalcPow 6, 1211, 2053, 2659
\bigintcalcSgn 4, 269, 1930, 1953, 1964, 1975, 1976, 1999, 2000, 2069, 2071, 2076
\BigIntCalcShl 7, 846, 2017
\bigintcalcShl 5, 827, 2014
\BigIntCalcShr 7, 902, 2026
\bigintcalcShr 5, 885, 2023
\bigintcalcSqr 6, 1147, 2042
\BigIntCalcSub 7, 643, 2004
\bigintcalcSub . 5, 581, 1737, 1744, 1998
\body 1792, 1796

C

\catcode 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, 1772, 1773, 1774, 1775, 1810, 1819, 1827, 1831, 1848, 1849, 1850
\chardef 1877
\count@ 1777, 1806, 1810, 1812, 1813, 1817, 1819, 1820, 1821, 1825, 1827, 1830, 1831, 1835, 1836
\countdef 1777
\csname 14, 21, 50, 66, 76, 119, 125, 140, 143, 152, 176, 179, 701, 717, 718, 720, 790, 809, 810, 812, 929, 936, 945, 952, 964, 1031, 1037, 1044, 1051, 1776, 1779, 1782, 1785, 1840, 1867, 2091, 2650, 2652

D

\dimexpr 2099
\divide 1914
\documentclass 1873

E

\empty 17, 18
\end 1868, 2145, 2164, 2187, 2198, 2209, 2237, 2249, 2267, 2284, 2313, 2340, 2348, 2377, 2404, 2418, 2444, 2499, 2584, 2646, 2662, 2665
\endcsname 14, 21, 50, 66, 76, 119, 125, 140, 143, 152, 176, 179, 708, 717, 718, 720, 797, 809, 810, 812, 929, 936, 945, 952, 964, 1031, 1037, 1044, 1051, 1776, 1779, 1782, 1785, 1840, 1867, 2091, 2650, 2652
\endinput 29, 117
\endlinechar 4, 35, 71, 77, 89
\endqstest 2113, 2118
\errmessage 1829
\Expect 1890, 1897, 1906, 1915, 2651, 2653

I

\ifcase 412, 431, 462, 475, 520, 533, 557, 600, 615, 703, 706, 721, 792, 795, 813, 907, 1026, 1034, 1052, 1076, 1079, 1104, 1172, 1221, 1246, 1252, 1263, 1274, 1281, 1294, 1301, 1353, 1364, 1413, 1416, 1428, 1442, 1451, 1513, 1543, 1552, 1590, 1617, 1633, 1672, 1683, 1691, 1713, 1722, 2069, 2071, 2076
\ifcat 158
\ifnum 343, 348, 380, 471, 497, 529, 590, 628, 711, 800, 875, 1082, 1093, 1108, 1118, 1171, 1196, 1354, 1371, 1812, 1820, 1827, 1835, 1953, 1964, 1975, 1976, 1977, 1999, 2000, 2001
\ifodd 397, 925, 941, 1323
\iftrue 2185
\ifx 15, 18, 21, 50, 58, 61, 119, 125, 140, 143, 152, 166, 176, 179, 192, 201, 202, 212, 217, 230, 233, 262, 275, 278, 295, 296, 306, 321, 322, 330, 353, 396, 451, 463, 496, 509, 521, 556, 558, 587, 588, 614, 647, 648, 658, 659, 676, 691, 738, 739, 749, 750, 767, 780, 833, 852, 863, 891, 910, 924, 987, 998, 1015, 1135, 1153, 1168, 1221, 1222, 1223, 1224, 1228, 1236, 1246,

1247, 1248, 1274, 1275, 1276, 1280, 1322, 1335, 1342, 1343, 1353, 1447, 1451, 1452, 1453, 1454, 1479, 1514, 1522, 1525, 1544, 1553, 1561, 1591, 1599, 1607, 1618, 1634, 1672, 1673, 1677, 1683, 1691, 1692, 1696, 1713, 1714, 1715, 1729, 1734, 1753, 1760, 1776, 1779, 1782, 1785, 1840, 2091, 2143, 2144, 2163
\immediate 23, 52
\IncludeTests 1882
\input 144, 1841
\iterate 1793, 1795, 1797

L

\LoadCommand 1841, 1851
\LogTests 1883
\loop 1791, 1807, 1818, 1826

M

\m@ne 2182, 2197, 2208, 2234
\makeatletter 1876, 2093, 2127
\makeatother 1878, 2125

N

\NeedsTeXFormat 1871
\newcommand . . . 1884, 1892, 1899, 1910, 1911, 1912, 1917, 1922, 1923, 1926, 1929, 1932, 1935, 1938, 1941, 1951, 1962, 1973, 1997, 2013, 2022, 2031, 2041, 2044, 2049, 2052, 2055, 2059, 2067, 2097, 2102, 2106, 2107, 2648
\newcount 1909, 2094, 2095
\next 1797, 1799, 1801
\nofiles 1872
\number 270, 286, 700, 789, 1031, 1037, 1044, 1832, 2099, 2142, 2180, 2181, 2182, 2184, 2196, 2197, 2207, 2208, 2233, 2234
\numexpr 473, 531, 679, 699, 788, 803, 866, 870, 927, 934, 943, 950, 1001, 1008, 1874, 1875, 1880, 1885, 1886, 1889, 1893, 1894, 1896, 1902, 1910, 2046

P

\PackageInfo 26
\pdf@escapehex 187
\pdf@unescapehex 185
\pdfelapsedtime 2108
\pdfresettimer 2104
\PrintTime 2097, 2110, 2123
\ProvidesPackage 19, 67

Q

\qstest 2112, 2114

R

\RangeCatcodeCheck . . 1824, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863
\RangeCatcodeInvalid 1816, 1844, 1845, 1846, 1847
\renewcommand 2103
\repeat . . . 1791, 1803, 1814, 1822, 1837
\RequirePackage 149
\RestoreCatcodes 1805, 1808, 1809, 1864
\result 1895, 1897
\resultA 1887, 1890
\resultB 1888, 1890
\romannumeral 154, 184, 246, 253, 257, 366, 371, 387, 392, 407, 420, 426, 439, 445, 448, 573, 582, 589, 603, 622, 641, 644, 698, 787, 828, 835, 847, 886, 892, 903, 1002, 1009, 1042, 1067, 1092, 1107, 1131, 1148, 1163, 1201, 1208, 1212, 1306, 1366, 1373, 1379, 1390, 1395, 1400, 1409, 1449, 1467, 1490, 1495, 1500, 1505, 1570, 1584, 1627, 1643, 1650, 1659, 1668, 1731, 1732, 1762

S

\saved@endqstest 2113, 2120
\saved@qstest 2112, 2115
\SavedNumexpr 1874, 1880, 1885, 1889, 1893, 1896
\space 1830, 1831, 1839, 1890, 2099
\StartTime 2102, 2116
\StopTime 2107, 2119
\strip@pt 2099
\SummaryTime . 2094, 2096, 2109, 2123

T

\Test 1843, 1866, 1917, 1922, 1924, 1927, 1930, 1933, 1936, 1939, 1942, 1944, 1952, 1955, 1963, 1966, 1974, 1979, 1987, 1998, 2003, 2014, 2016, 2023, 2025, 2032, 2034, 2042, 2050, 2053, 2056, 2060, 2062, 2068, 2082
\TestAbs 1926, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163
\TestAdd 1973, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312
\TestArg 1910, 1911, 1913, 1914
\TestCmp 1938, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236
\TestCount 1909, 1913, 1914, 1915
\TestDec . . . 1962, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283

\TestDiv 2055, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582
\TestDivBig 2059, 2583
\TestError 2648, 2657, 2658, 2659, 2660, 2661
\TestExch 1922, 2045
\TestFac 2044, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433
\TestFacBig 2049, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443
\TestInc 1951, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266
\TestInv 1923, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144
\TestMax 1935, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208
\TestMin 1932, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197
\TestMod . . 2067, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645
\TestMul . . . 2031, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403
\TestOdd 1941, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248
\TestPow 2052, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498
\TestResult 1892, 1918
\TestResultTwoExpansions . 1899, 1919
\TestSgn 1929, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183
\TestShl 2013, 2343, 2344, 2345, 2346, 2347
\TestShr 2022, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376
\TestSpaceAtEnd 1884, 1920
\TestSqr 2041, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417
\TestSub . . . 1997, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339
\TestTeXDivide 1912, 2057
\TestTime 2095, 2108, 2109, 2110
\the 77, 78, 79, 80, 81, 82, 83, 84, 97, 473, 531, 679, 699, 788, 803, 866, 870, 927, 934, 943, 950, 1001, 1008, 1810, 1830, 1831, 1915, 2046
\TimeDescription . . . 2103, 2106, 2110
\TMP@EnsureCode 94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116
\TMP@RequirePackage 141, 147
\tracingmacros 2240
\typeout 2098

U

\UNDEFINED . . 1875, 1886, 1894, 1902
\usepackage 1879, 1881

W

\write 23, 52

X

\x 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 135, 138, 1943, 1949, 1954, 1959, 1965, 1970, 1978, 1984, 1986, 1992, 2002, 2008, 2015, 2020, 2024, 2029, 2033, 2039, 2061, 2081, 2087

Z

\z@ 2096, 2142, 2180, 2184, 2196, 2207, 2233