File created: 1-Jan-99 21:45:52 **{DSK}<disk>disk3>lispcore3.0>sources>LLBIGNUM.;2**

changes to: (FNS \INITBIGNUMS)

previous date: 19-Jan-93 10:44:45 {DSK}<disk>disk3>lispcore3.0>sources>LLBIGNUM.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;

```
(RPAQQ LLBIGNUMCOMS
       [(COMS (DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS BIGNUM))
              (INITRECORDS BIGNUM)
              (CONSTANTS \BIGNUM.THETA (\BIGNUM.BETA (EXPT 2 14))
                     (\BIGNUM.BETA1 (SUB1 \BIGNUM.BETA)))
              [DECLARE%: EVAL@COMPILE (ADDVARS (CHARACTERNAMES (INFINITY 8551]
              (ADDVARS (GLOBALVARS MIN.INTEGER MAX.INTEGER \BIG.0 \BIG.1)))
        (COMS                                                        ; entries
              (FNS \BIGNUM.COMPARE \BIGNUM.DIFFERENCE \BIGNUM.INTEGERLENGTH \BIGNUM.LOGAND \BIGNUM.LOGOR
                   \BIGNUM.LOGXOR \BIGNUM.PLUS \BIGNUM.LSH \BIGNUM.TIMES \BIGNUM.QUOTIENT \BIGNUM.REMAINDER
                   \BIGNUM.TO.FLOAT)
              (FNS FINITEP INFINITEP))
        (COMS                                                        ; internal functions
              (FNS \BIGNUM.TO.INT \BN.2TH \BN.ABS \BN.DIFFERENCE \BN.DIVIDE \BN.FLOAT \BN.IGNN BIGNUM.DEFPRINT
                   \BN.INTEGERLENGTH \BN.LOGAND \BN.LOGANDC2 \BN.LOGOR \BN.LOGXOR \BN.MINUS \BN.PLUS2 \BN.SIGN
                   \BN.TIMES2 \BN.COMPAREN \BN.D2TH \BN.FROM.FIXP \BN.ICANON \BN.IDIVIDE \BN.ISUM0 \BN.ISUM1
                   \BN.MADD \BN.TO.FIXP \BN.NZEROS \BN.QRS \BN.SIGN \BN.TH2B \BN.TH2D))
        (COMS (FNS \INITBIGNUMS)
                                                                     ; MAKERATIONAL needs work

              ;; needs work: MASK.1'S MASK.0'S BITTEST BITSET BITCLEAR LOGNOT LOADBYTE DEPOSITBYTE IMODLESSP IMODPLUS
              ;; IMODDIFFERENCE ROT

              (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\INITBIGNUMS])

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

[DATATYPE BIGNUM (ELEMENTS)
       (INIT (DEFPRINT 'BIGNUM 'BIGNUM.DEFPRINT]
)

(/DECLAREDATATYPE 'BIGNUM '(POINTER)
       ;; ---field descriptor list elided by lister---
       '2)

(DEFPRINT 'BIGNUM 'BIGNUM.DEFPRINT)
)

(/DECLAREDATATYPE 'BIGNUM '(POINTER)
       ;; ---field descriptor list elided by lister---
       '2)

(DEFPRINT 'BIGNUM 'BIGNUM.DEFPRINT)

(DECLARE%: EVAL@COMPILE

(RPAQQ \BIGNUM.THETA 10000)

(RPAQ \BIGNUM.BETA (EXPT 2 14))

(RPAQ \BIGNUM.BETA1 (SUB1 \BIGNUM.BETA))

(CONSTANTS \BIGNUM.THETA (\BIGNUM.BETA (EXPT 2 14))
       (\BIGNUM.BETA1 (SUB1 \BIGNUM.BETA)))
)

(DECLARE%: EVAL@COMPILE

(ADDTOVAR CHARACTERNAMES (INFINITY 8551))
)

(ADDTOVAR GLOBALVARS MIN.INTEGER MAX.INTEGER \BIG.0 \BIG.1)

;; entries

(DEFINEQ

(\BIGNUM.COMPARE
  [LAMBDA (X Y)                                                  (* lmm "15-Apr-85 17:36")
    (COND
       ((EQ X MIN.INTEGER)
        (COND
           ((EQ Y MIN.INTEGER)
            0)
           (T -1)))
       ((EQ X MAX.INTEGER)
        (COND
           ((EQ Y MAX.INTEGER)
            0)
           (T 1)))
       ((EQ Y MIN.INTEGER)
        1)
       ((EQ Y MAX.INTEGER)
        -1)
       (T (\BN.COMPAREN (\BN.FROM.FIXP X)
                 (\BN.FROM.FIXP Y])

(\BIGNUM.DIFFERENCE
  [LAMBDA (X Y)                                                  (* lmm "12-Apr-85 08:38")
    (\BN.TO.FIXP (\BN.DIFFERENCE (\BN.FROM.FIXP X)
                        (\BN.FROM.FIXP Y])

(\BIGNUM.INTEGERLENGTH
  [LAMBDA (X)                                                    (* lmm "12-Apr-85 08:01")
    (\BN.INTEGERLENGTH (\BN.FROM.FIXP X])

(\BIGNUM.LOGAND
  [LAMBDA (X Y)                                                  (* kbr%: "16-Sep-86 12:28")
    (COND
       ((OR (EQ X 0)
            (EQ Y 0))
        0)
       ((OR (INFINITEP X)
            (INFINITEP Y))
        (ERROR "Can't do logical operations with infinity"))
       [(LESSP Y 0)
        (COND
           [(LESSP X 0)
            (DIFFERENCE -1 (LOGOR (LOGNOT X)
                                  (LOGNOT Y]
           (T (\BN.TO.FIXP (\BN.LOGANDC2 (\BN.FROM.FIXP X)
                                 (\BN.FROM.FIXP (DIFFERENCE -1 Y]
       ((LESSP X 0)
        (\BIGNUM.LOGAND Y X))
       (T (\BN.TO.FIXP (\BN.LOGAND (\BN.FROM.FIXP X)
                              (\BN.FROM.FIXP Y])

(\BIGNUM.LOGOR
  [LAMBDA (X Y)                                                  (* kbr%: "16-Sep-86 12:29")
    (COND
       ((EQ X 0)
        Y)
       ((EQ Y 0)
        X)
       ((OR (INFINITEP X)
            (INFINITEP Y))
        (ERROR "Can't do logical operations with infinity"))
       [(AND (GREATERP X 0)
             (GREATERP Y 0))
        (\BN.TO.FIXP (\BN.LOGOR (\BN.FROM.FIXP X)
                           (\BN.FROM.FIXP Y]
       (T
          (* stupid slow but maybe working definition. Problem is that logors of negatives are difficult in current representation)

          (DIFFERENCE -1 (LOGAND (DIFFERENCE -1 X)
                                 (DIFFERENCE -1 Y])

(\BIGNUM.LOGXOR
  [LAMBDA (X Y)                                                  (* kbr%: "16-Sep-86 12:29")
    (COND
       ((EQ X 0)
        Y)
       ((EQ Y 0)
        X)
       ((OR (INFINITEP X)
            (INFINITEP Y))
        (ERROR "Can't do logical operations with infinity"))
       [(LESSP X 0)
        (COND
           ((LESSP Y 0)
             (LOGXOR (DIFFERENCE -1 X)
                     (DIFFERENCE -1 Y)))
            (T                                                        (* stupid dumb but working definition)
               (\BIGNUM.DIFFERENCE (\BIGNUM.LOGOR X Y)
                       (\BIGNUM.LOGAND X Y]
        ((LESSP Y 0)
         (\BIGNUM.LOGXOR Y X))
        (T (\BN.TO.FIXP (\BN.LOGXOR (\BN.FROM.FIXP X)
                                (\BN.FROM.FIXP Y])

(\BIGNUM.PLUS
  [LAMBDA (X Y)                                                       (* lmm "12-Apr-85 08:03")
    (\BN.TO.FIXP (\BN.PLUS2 (\BN.FROM.FIXP X)
                       (\BN.FROM.FIXP Y])

(\BIGNUM.LSH
  [LAMBDA (X N)                                                       ; Edited 23-Feb-87 16:09 by jrb:
    (COND
       ((EQ X 0)
        0)
       ((EQ N MIN.INTEGER)
        (COND
           ((INFINITEP X)
            (ERROR "Can't shift infinity minus infinity places"))
           (T 0)))
       ((INFINITEP X)
        X)
       ((EQ N MAX.INTEGER)
        (COND
           ((EQ X 0)
            0)
           ((IGREATERP X 0)
            MAX.INTEGER)
           (T MIN.INTEGER)))
       [(IGEQ N 0)
        (SETQ X (\BN.FROM.FIXP X))                                    ; Don't smash original input
        [if (>= N 14)
            then (while (>= N 14) do (SETQ N (IDIFFERENCE N 14))
                                     (SETQ X (CONS 0 X]
        (\BN.TO.FIXP (\BN.TIMES2 X (\BN.FROM.FIXP (EXPT 2 N]
       [(IGREATERP X 0)
        (SETQ X (\BN.FROM.FIXP X))                                    ; Don't smash original input
        [if (<= N -14)
            then (while (<= N -14) do (SETQ N (IPLUS N 14))
                                      (SETQ X (CDR X]
        (\BIGNUM.QUOTIENT (create BIGNUM
                                  ELEMENTS _ X)
               (EXPT 2 (IMINUS N]
       (T ;; RIGHTSHIFT A NEGATIVE - result must be adjusted if not a bignum

          (SETQ X (MINUS (\BIGNUM.LSH (MINUS X)
                                  N)))
          (if (NOT (type? BIGNUM X))
              then (SETQ X (SUB1 X)))
          X])

(\BIGNUM.TIMES
  [LAMBDA (X Y)                                                       (* lmm "12-Apr-85 08:03")
    (\BN.TO.FIXP (\BN.TIMES2 (\BN.FROM.FIXP X)
                       (\BN.FROM.FIXP Y])

(\BIGNUM.QUOTIENT
  [LAMBDA (X Y)                                                       (* kbr%: "16-Sep-86 12:30")
    (COND
       ((EQ Y MAX.INTEGER)
        (COND
           ((INFINITEP X)
            (ERROR "Can't divide infinity by infinity"))
           (T 0)))
       ((EQ Y MIN.INTEGER)
        (COND
           ((INFINITEP X)
            (ERROR "Can't divide infinity by infinity"))
           (T 0)))
       ((EQ X MAX.INTEGER)
        (COND
           ((EQ Y 0)
            (ERROR "Can't divide infinity by 0"))
           ((IGREATERP Y 0)
            MAX.INTEGER)
           (T MIN.INTEGER)))
       ((EQ X MIN.INTEGER)
        (COND
           ((EQ Y 0)
            (ERROR "Can't divide infinity by 0"))
           ((IGREATERP Y 0)
            MIN.INTEGER)
           (T MAX.INTEGER)))
       (T (\BN.TO.FIXP (CAR (\BN.DIVIDE (\BN.FROM.FIXP X)
                                         (\BN.FROM.FIXP Y])

(\BIGNUM.REMAINDER
  [LAMBDA (X Y)                                             (* kbr%: "16-Sep-86 12:30")
    (COND
       ((OR (INFINITEP X)
            (INFINITEP Y))
        (ERROR "Can't take remainder with infinity"))
       (T (\BN.TO.FIXP (CDR (\BN.DIVIDE (\BN.FROM.FIXP X)
                                         (\BN.FROM.FIXP Y])

(\BIGNUM.TO.FLOAT
  [LAMBDA (X)                                               (* lmm "12-Apr-85 08:06")
                                                            (* called by \FLOAT)

    (\BN.FLOAT (\BN.FROM.FIXP X])

)

(DEFINEQ

(FINITEP
  [LAMBDA (CL:NUMBER)                                       (* kbr%: "16-Sep-86 12:24")
    (NOT (OR (EQ CL:NUMBER MAX.INTEGER)
             (EQ CL:NUMBER MIN.INTEGER])

(INFINITEP
  [LAMBDA (CL:NUMBER)                                       (* kbr%: "16-Sep-86 12:25")
    (OR (EQ CL:NUMBER MAX.INTEGER)
        (EQ CL:NUMBER MIN.INTEGER])

)

;; internal functions

(DEFINEQ

(\BIGNUM.TO.INT
  [LAMBDA (X)                                               (* lmm " 9-Jan-86 15:30")
    (COND
       ((NULL (CDR X))
        (CAR X))
       (T (IPLUS (CAR X)
                 (ITIMES \BIGNUM.BETA (\BIGNUM.TO.INT (CDR X])

(\BN.2TH
  [LAMBDA (A)                                               (* lmm " 9-Jan-86 15:31")
    (PROG (L B)
          [while A do (PROGN (SETQ L (\BN.QRS A \BIGNUM.THETA))
                             (SETQ A (CAR L))
                             (SETQ B (CONS (CDR L)
                                           B]
          (RETURN B])

(\BN.ABS
  [LAMBDA (U)                                               (* lmm "20-JUL-84 02:00")
    (COND
       ((ILESSP (\BN.SIGN U)
               0)
        (\BN.MINUS U))
       (T U])

(\BN.DIFFERENCE
  [LAMBDA (U V)                                             (* lmm "20-JUL-84 01:33")
    (\BN.PLUS2 U (\BN.MINUS V])

(\BN.DIVIDE
  [LAMBDA (A B FLG)                                         (* lmm " 9-Jan-86 15:33")
    (PROG (M N K SA ST C D W E F B1 B2 A1 A2 A3 QHAT C1 R1 R2 U V Q X IP BIP L1 L2)
          [COND
             ((OR (NULL A)
                  (NULL B))
              (RETURN (CONS \BIG.0 A]
          (COND
             ((CDR B)
              (GO LL1)))
          (SETQ BIP (\BN.QRS A (CAR B)))
          [RETURN (CONS (CAR BIP)
                        (AND (NEQ FLG 'QUOTIENT)
                             (\BN.FROM.FIXP (CDR BIP]
      LL1 (SETQ M (FLENGTH A))
          (SETQ N (FLENGTH B))
          (SETQ K (IDIFFERENCE M N))
          [COND
             ((ILESSP K 0)
              (RETURN (CONS \BIG.0 A]
          (SETQ SA (\BN.SIGN A))
          (SETQ U B)
          (for i from 1 to (IDIFFERENCE N 1) do (SETQ U (CDR U)))
          (SETQ C (CAR U))
          (SETQ ST 1)
          [COND
             ((ILESSP C 0)
              (PROGN (SETQ ST -1)
                     (SETQ C (IMINUS C]
          (SETQ D (IQUOTIENT \BIGNUM.BETA (IPLUS C 1)))
          (SETQ W (ITIMES SA ST))
          [SETQ A (\BN.TIMES2 A (\BN.FROM.FIXP (ITIMES SA D]
          [SETQ B (\BN.TIMES2 B (\BN.FROM.FIXP (ITIMES ST D]
          (SETQ U A)
          (SETQ L1 NIL)
          [for I from 1 to (IPLUS K 1) do (PROGN (SETQ L1 (CONS U L1))
                                                 (SETQ U (CDR U]
          (SETQ L2 L1)
          (for I from 1 to (IDIFFERENCE N 2) do (SETQ L2 (CONS U L2))
                                                 (SETQ U (CDR U)))
          [COND
             ((NULL (CDR U))
              (RPLACD U (CONS 0 NIL]
          (SETQ U B)
          (for I from 1 to (IDIFFERENCE N 2) do (SETQ U (CDR U)))
          (SETQ B2 (CAR U))
          (SETQ U (CDR U))
          (SETQ B1 (CAR U))
      L10 (SETQ U (CAR L2))
          (SETQ A3 (CAR U))
          (SETQ U (CDR U))
          (SETQ A2 (CAR U))
          (SETQ U (CDR U))
          (SETQ A1 (CAR U))
          (SETQ U (CDR U))
          [COND
             ((IGEQ A1 B1)
              (SETQ QHAT \BIGNUM.BETA1))
             (T (SETQ QHAT (IQUOTIENT (IPLUS (ITIMES A1 \BIGNUM.BETA)
                                             A2)
                                      B1]
      L12 (SETQ IP (\BN.IDIVIDE (ITIMES QHAT B1)
                          \BIGNUM.BETA))
          (SETQ R1 (IDIFFERENCE A1 (CAR IP)))
          (SETQ R2 (IDIFFERENCE A2 (CDR IP)))
          [COND
             ((ILESSP R2 0)
              (PROGN (SETQ R2 (IPLUS R2 \BIGNUM.BETA))
                     (SETQ R1 (IDIFFERENCE R1 1]
          (COND
             ((IGREATERP R1 0)
              (GO L13)))
          (SETQ IP (\BN.IDIVIDE (ITIMES QHAT B2)
                          \BIGNUM.BETA))
          (SETQ R1 (IDIFFERENCE R2 (CAR IP)))
          (COND
             ((IGREATERP R1 0)
              (GO L13)))
          (SETQ R2 (IDIFFERENCE A3 (CDR IP)))
          [COND
             ((OR (ILESSP R1 0)
                  (ILESSP R2 0))
              (PROGN (SETQ QHAT (IDIFFERENCE QHAT 1))
                     (GO L12]
      L13 (SETQ U (CAR L1))
          (SETQ V B)
          (SETQ C1 0)
      L14 (SETQ E (IMINUS QHAT))
          (SETQ IP (\BN.IDIVIDE (ITIMES E (CAR V))
                          \BIGNUM.BETA))
          (SETQ V (CDR V))
          (SETQ E (CAR IP))
          (SETQ A1 (CAR U))
          (SETQ IP (\BN.IDIVIDE (IPLUS C1 (IPLUS A1 (CDR IP)))
                          \BIGNUM.BETA))
          (SETQ A1 (CDR IP))
```

```
          (SETQ C1 (CAR IP))
          [COND
             ((ILESSP A1 0)
              (PROGN (SETQ A1 (IPLUS A1 \BIGNUM.BETA))
                     (SETQ C1 (IDIFFERENCE C1 1]
          (SETQ C1 (IPLUS C1 E))
          (RPLACA U A1)
          (SETQ X U)
          (SETQ U (CDR U))
          (COND
             (V (GO L14)))
          (SETQ A1 (IPLUS (CAR U)
                          C1))
          (SETQ U (CDR U))
          (RPLACD X \BIG.0)
          (COND
             ((EQ A1 0)
              (GO L17)))
          (SETQ U (CAR L1))
          (SETQ V B)
          (SETQ C1 0)
          (SETQ QHAT (IDIFFERENCE QHAT 1))
      L16 (SETQ A1 (CAR U))
          (SETQ B1 (CAR V))
          (SETQ V (CDR V))
          (SETQ IP (\BN.IDIVIDE (IPLUS C1 (IPLUS A1 B1))
                          \BIGNUM.BETA))
          (RPLACA U (CDR IP))
          (SETQ U (CDR U))
          (COND
             (V (GO L16)))
      L17 [COND
             ((OR (NEQ QHAT 0)
                  Q)
              (SETQ Q (CONS (ITIMES W QHAT)
                            Q]
          (SETQ L1 (CDR L1))
          (SETQ U (CAR L2))
          (SETQ L2 (CDR L2))
          (COND
             (L1 (GO L10)))
          (RETURN (CONS Q (AND (NEQ FLG 'QUOTIENT)
                               (CAR (\BN.QRS A (ITIMES SA D])
```

```
(\BN.FLOAT
  [LAMBDA (X)                                                   (* kbr%: "16-Sep-86 12:21")
    (COND
       ((NULL X)
        0.0)
       [(LISTP X)
        (FPLUS (CAR X)
               (FTIMES \BIGNUM.BETA (\BN.FLOAT (CDR X]
       ((OR (EQ X 'MAX.INTEGER)
            (EQ X 'MIN.INTEGER))

          (* KBR%: After some consideration, I've decided that it would be best that rational infinities and floating point infinities be
          kept distinct in the same way that we consider 1 distinct from 1.0.
          This is an admission that the systems of Lisp rationals and Lisp floating point numbers are two disjoint sets of Lisp
          expressions. The semantics of these expressions--what they denote--is slightly more than the rational numbers we attach to
          them. These expressions should be viewed as denoting a pair consisting of a rational number and an atom recording the
          type of the expression. A Lisp rational X denotes the pair (X CL:RATIONAL) and a Lisp floating point number X denotes
          (X FLOAT) in our mind. The FLOAT operation is an injection that changes an expression denoting a pair
          (X CL:RATIONAL) into an expression denoting a pair (X FLOAT)%.
          Arithmetic on these expressions is typed arithmetic with rounding of the results in the case of FLOATPs according to IEEE
          spec. Let's suppose that MAX.INTEGER and MAX.FLOAT are Lisp expressions denoting the respective rational and floating
          point infinities. (We now know with the advent of Common Lisp that MAX.INTEGER and MIN.INTEGER should have been
          called MAX.RATIONAL and MIN.RATIONAL) Then rules of floating point coercion can continue to make sense%:
          (EQL (/ MAX.INTEGER) 0) (EQL (/ MAX.FLOAT) 0.0) (EQL (EXPT 2 MAX.INTEGER) MAX.INTEGER)
          (EQL (EXPT 2.0 MAX.INTEGER) MAX.FLOAT) (EQL (EXPT 2 MAX.FLOAT) MAX.FLOAT)
          (EQL (EXPT 2.0 MAX.FLOAT) MAX.FLOAT) etc. But if we tried to make rational infinities and floating point infinities
          identical, then we would have to arbitrarily decide in an unnatural way whether
          (EQL (/ MAX.INTEGER) 0) or (EQL (/ MAX.FLOAT) 0.0) is true, etc.
          Recommendation%: Currently Xerox Lisp does not support floating point infinities.
          Larry Masinter added rational infinities. If there is a desire to add floating point infinities at some point in the future, then I
          recommend that rational and floating point infinities be kept distinct.
          *)

          (* * Error because Xerox Lisp does not support floating point infinities
          (at this time)%. *)

        (ERROR "Can't float integer infinity."))
       (T (SHOULDNT])
```

```
(\BN.IGNN
  [LAMBDA (U)                                                   (* lmm " 9-Jan-86 15:30")
```

```
    (COND
       ((NULL U)
        NIL)
       ((ILESSP U \BIGNUM.BETA)
        (LIST U))
       (T (PROG (Y)
                (SETQ Y (IQUOTIENT U \BIGNUM.BETA))
                (SETQ U (IDIFFERENCE U (ITIMES Y \BIGNUM.BETA)))
                (RETURN (CONS U (\BN.FROM.FIXP Y])

(BIGNUM.DEFPRINT
  [LAMBDA (BIGN STREAM)                                          (* kbr%: "16-Sep-86 12:31")
    (COND
       [(INFINITEP BIGN)                                         (* Distinguished integers smaller/larger than any others.
                                                                 Print using "evaluate at read time" syntax)
        (CONS (CONCAT (CHARACTER (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
                     ".")
              (COND
                 ((EQ BIGN MIN.INTEGER)
                  'MIN.INTEGER)
                 (T 'MAX.INTEGER]
       (T (LET* ((RADIX (\CHECKRADIX *PRINT-BASE*))
                 [TH (SELECTQ RADIX
                          (10 10000)
                          (8 4096)
                          (bind (TH _ RADIX)
                               NEWTH while (LEQ (SETQ NEWTH (TIMES TH RADIX))
                                                \BIGNUM.BETA)
                             do (SETQ TH NEWTH) finally (RETURN TH]
                 (CHARS (\BN.TH2D (bind (ELS _ (fetch (BIGNUM ELEMENTS) of BIGN))
                                         L B while ELS do (SETQ L (\BN.QRS ELS TH))
                                                             (SETQ ELS (CAR L))
                                                             (SETQ B (CONS (CDR L)
                                                                           B))
                                      finally (RETURN B))
                               RADIX TH)))
                [COND
                   (*PRINT-RADIX*                                (* need radix qualifier)
                          (COND
                             ((AND (EQ RADIX 8)
                                   (NOT (fetch (READTABLEP COMMONLISP) of *READTABLE*)))
                              (NCONC1 CHARS (CHARCODE Q)))
                             (T [push CHARS (SELECTQ RADIX
                                                  (8 (CHARCODE o))
                                                  (16 (CHARCODE x))
                                                  (2 (CHARCODE b))
                                                  (PROGN (push CHARS (CHARCODE r))
                                                         [COND
                                                            ((IGEQ RADIX 10)
                                                             (push CHARS (IPLUS (CHARCODE 0)
                                                                                (IMOD RADIX 10)))
                                                             (SETQ RADIX (IQUOTIENT RADIX 10]
                                                         (IPLUS RADIX (CHARCODE 0]
                                 (push CHARS (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*]
                (.SPACECHECK. STREAM (LENGTH CHARS))
                (for C in CHARS do (\OUTCHAR STREAM C))          (* Return T to show we have done it ourselves)
                T])

(\BN.INTEGERLENGTH
  [LAMBDA (X)                                                    (* kbr%: "16-Sep-86 12:31")
    (COND
       ((NULL X)
        0)
       [(LISTP X)
        (COND
           [(CDR X)
            (IPLUS (CONSTANT (INTEGERLENGTH (SUB1 \BIGNUM.BETA)))
                   (\BN.INTEGERLENGTH (CDR X]
           (T (INTEGERLENGTH (CAR X]
       ((INFINITEP X)
        MAX.INTEGER)
       (T (SHOULDNT])

(\BN.LOGAND
  [LAMBDA (B1 B2)                                                (* lmm "20-Jul-84 11:13")
    (COND
       ((NULL B1)
        NIL)
       ((NULL B2)
        NIL)
       (T (PROG (B)
                (SETQ B (\BN.LOGAND (CDR B1)
                               (CDR B2)))
                (SETQ B1 (LOGAND (CAR B1)
                                (CAR B2)))
                (COND
                   ((AND (NULL B)
                         (EQ B1 0))
                    (RETURN B)))
                (RETURN (CONS B1 B])

(\BN.LOGANDC2
  [LAMBDA (B1 B2)                                                  (* lmm "14-May-86 10:47")
    (COND
       ((NULL B1)
        NIL)
       ((NULL B2)
        B1)
       (T (PROG (B)
                (SETQ B (\BN.LOGANDC2 (CDR B1)
                               (CDR B2)))
                [SETQ B1 (LOGAND (CAR B1)
                                (DIFFERENCE -1 (CAR B2]
                (COND
                   ((AND (NULL B)
                         (EQ B1 0))
                    (RETURN B)))
                (RETURN (CONS B1 B])

(\BN.LOGOR
  [LAMBDA (B1 B2)                                                  (* lmm "21-JUL-84 23:57")
    (COND
       ((NULL B1)
        B2)
       ((NULL B2)
        B1)
       (T (CONS (LOGOR (CAR B1)
                       (CAR B2))
                (\BN.LOGOR (CDR B1)
                       (CDR B2])

(\BN.LOGXOR
  [LAMBDA (B1 B2)                                                  (* lmm "21-JUL-84 23:59")
    (COND
       ((NULL B1)
        B2)
       ((NULL B2)
        B1)
       (T (CONS (LOGXOR (CAR B1)
                       (CAR B2))
                (\BN.LOGXOR (CDR B1)
                       (CDR B2])

(\BN.MINUS
  [LAMBDA (U)                                                      (* kbr%: "11-Sep-86 15:00")
    (COND
       ((NULL U)
        NIL)
       [(LISTP U)
        (CONS (IMINUS (CAR U))
              (\BN.MINUS (CDR U]
       ((EQ U 'MAX.INTEGER)
        'MIN.INTEGER)
       ((EQ U 'MIN.INTEGER)
        'MAX.INTEGER)
       (T (SHOULDNT])

(\BN.PLUS2
  [LAMBDA (U V)                                                    (* kbr%: "11-Sep-86 15:26")
    (COND
       ((NULL U)
        V)
       ((NULL V)
        U)
       [(AND (LISTP U)
             (LISTP V))
        (PROG (L)
              (SETQ L (IDIFFERENCE (FLENGTH U)
                             (FLENGTH V)))
              [COND
                 [(ILESSP L 0)
                  (SETQ U (APPEND U (\BN.NZEROS (IDIFFERENCE 0 L]
                 ((IGREATERP L 0)
                  (SETQ V (APPEND V (\BN.NZEROS L]
              (RETURN (COND
                         ((EQ (\BN.SIGN U)
                              (\BN.SIGN V))
                          (\BN.ISUM0 U V))
                         (T (\BN.ISUM1 U V]
       ((EQ U 'MAX.INTEGER)
        (COND
           ((EQ V 'MIN.INTEGER)
            (ERROR "Can't add plus infinity to minus infinity"))
           (T U)))
       ((EQ U 'MIN.INTEGER)
        (COND
           ((EQ V 'MAX.INTEGER)
            (ERROR "Can't add plus infinity to minus infinity"))
           (T U)))
       (T V])
```

```
(\BN.SIGN
  [LAMBDA (U)                                                     (* kbr%: "11-Sep-86 15:22")
    (COND
       [(ATOM U)
        (COND
           ((NULL U)
            0)
           ((EQ U 'MAX.INTEGER)
            1)
           ((EQ U 'MIN.INTEGER)
            -1)
           (T (SHOULDNT]
       ((IGREATERP (CAR U)
               0)
        1)
       ((ILESSP (CAR U)
               0)
        -1)
       (T (\BN.SIGN (CDR U])
```

```
(\BN.TIMES2
  [LAMBDA (U V)                                                   (* kbr%: "11-Sep-86 15:19")
    (PROG (TAIL U1 W W1 W2 L C AP BP)
          [COND
             [(NULL U)
              (COND
                 ((OR (EQ V 'MAX.INTEGER)
                      (EQ V 'MIN.INTEGER))
                  (ERROR "Can't multiply infinity and zero."))
                 (T (RETURN NIL]
             [(EQ U 'MAX.INTEGER)
              (COND
                 ((NULL V)
                  (ERROR "Can't multiply infinity and zero."))
                 ((EQ (\BN.SIGN V)
                      1)
                  (RETURN U))
                 (T (RETURN 'MIN.INTEGER]
             ((EQ U 'MIN.INTEGER)
              (COND
                 ((NULL V)
                  (ERROR "Can't multiply infinity and zero."))
                 ((EQ (\BN.SIGN V)
                      1)
                  (RETURN U))
                 (T (RETURN 'MAX.INTEGER]
          (SETQ TAIL (LIST 0 0))
          (SETQ L (IPLUS (FLENGTH U)
                         (IDIFFERENCE (FLENGTH V)
                                2)))
          (SETQ W TAIL)
          (for I from 1 to L do (SETQ W (CONS 0 W)))
          (SETQ W1 W)
      A   (SETQ U1 U)
          (SETQ W2 W1)
          (SETQ C 0)
      B   (SETQ AP (\BN.IDIVIDE (ITIMES (CAR U1)
                                        (CAR V))
                          \BIGNUM.BETA))
          (SETQ BP (\BN.IDIVIDE (IPLUS (CAR W2)
                                       (IPLUS (CDR AP)
                                              C))
                          \BIGNUM.BETA))
          (RPLACA W2 (CDR BP))
          (SETQ C (IPLUS (CAR AP)
                         (CAR BP)))
          (SETQ W2 (CDR W2))
          (SETQ U1 (CDR U1))
          (COND
             (U1 (GO B)))
          (RPLACA W2 C)
          (SETQ W1 (CDR W1))
          (SETQ V (CDR V))
          (COND
             (V (GO A)))
          (COND
             ((EQ C 0)
              (RPLACD TAIL NIL)))
          (RETURN W])

(\BN.COMPAREN
  [LAMBDA (U V)                                                 (* lmm "12-Apr-85 08:33")
    (PROG ((SU 0)
           (SV 0)
           (ST 0)
           (S 0))
          [COND
             [(EQ (SETQ SU (\BN.SIGN U))
                  0)
              (RETURN (IMINUS (\BN.SIGN V]
             ((EQ (SETQ SV (\BN.SIGN V))
                  0)
              (RETURN SU))
             ((NEQ (SETQ S (IDIFFERENCE SU SV))
                   0)
              (RETURN (COND
                         ((IGREATERP S 0)
                          1)
                         ((ILESSP S 0)
                          -1)
                         (T (SHOULDNT]
      A   (COND
             ((NEQ (SETQ ST (IDIFFERENCE (CAR U)
                                   (CAR V)))
                   0)
              (SETQ S ST)))
          (SETQ V (CDR V))
          (SETQ U (CDR U))
          (COND
             [(NULL U)
              (RETURN (COND
                         (V (IMINUS SU))
                         (T (COND
                               ((IGREATERP S 0)
                                1)
                               ((ILESSP S 0)
                                -1)
                               (T 0]
             (V (GO A))
             (T (RETURN SU])

(\BN.D2TH
  [LAMBDA (U)                                                   (* lmm " 9-Jan-86 15:31")
    (PROG (B S V BI M AI)
          (COND
             ((NULL U)
              (RETURN B)))
          [COND
             [(OR (EQ (CAR U)
                      '+)
                  (EQ (CAR U)
                      '-))
              (PROGN (SETQ S (CAR U))
                     (SETQ U (CDR U]
             (T (SETQ S '+]
          (COND
             ((NULL U)
              (RETURN B)))
          (SETQ U (SETQ V (REVERSE U)))
      L2  (SETQ BI 0)
          (SETQ M 1)
          [while (AND U (ILESSP M \BIGNUM.THETA)) do (PROGN (SETQ AI (CAR U))
                                                           (SETQ U (CDR U))
                                                           (SETQ BI (IPLUS (ITIMES AI M)
                                                                           BI))
                                                           (SETQ M (ITIMES 10 M]
          [COND
             ((EQ S '-)
              (SETQ BI (IMINUS BI]
          (SETQ B (CONS BI B))
          (COND
             (U (GO L2)))
          (RETURN B])
```

```
(\BN.FROM.FIXP
  [LAMBDA (U)                                                    (* kbr%: "11-Sep-86 14:54")
    (COND
       ((type? BIGNUM U)
        (fetch (BIGNUM ELEMENTS) of U))
       ((OR (NULL U)
            (EQ U 0))
        NIL)
       ((LISTP U)
        U)
       [(ILESSP U 0)
        (COND
           ((EQUAL U MIN.FIXP)
            (\BN.DIFFERENCE (\BN.FROM.FIXP (IPLUS U \BIGNUM.THETA))
                            (\BN.FROM.FIXP \BIGNUM.THETA)))
           (T (\BN.MINUS (\BN.IGNN (IMINUS U]
       (T (\BN.IGNN U])

(\BN.ICANON
  [LAMBDA (U SIGN)                                               (* jrb%: " 6-Nov-86 15:30")
    (PROG ((U0 U)
           U1
           (CARRY 0)
           B)
      A   (SETQ B (IPLUS (CAR U)
                         CARRY))
          (SETQ CARRY (COND
                         ((AND (IGREATERP SIGN 0)
                               (ILESSP B 0))
                          -1)
                         ((AND (ILESSP SIGN 0)
                               (IGREATERP B 0))
                          1)
                         (T 0)))
          (SETQ B (IDIFFERENCE B (ITIMES CARRY \BIGNUM.BETA)))
          (RPLACA U B)                                           (* "U1 points to the high-order non-zero bignum node")
          (COND
             ((NEQ B 0)
              (SETQ U1 U)))
      B   [COND
             ((CDR U)
              (SETQ U (CDR U))
              (GO A))
             (T                                                  (* "If U1 is not eq to U here, we have high-order zero nodes in
                                                                 this bignum")

                (CL:IF (NEQ U1 U)
                       (RPLACD U1 NIL]
          (RETURN U0])

(\BN.IDIVIDE
  [LAMBDA (A B)                                                  (* lmm "20-JUL-84 01:37")
    (CONS (IQUOTIENT A B)
          (IREMAINDER A B])

(\BN.ISUM0
  [LAMBDA (U V)                                                  (* lmm " 9-Jan-86 15:30")
    (PROG ((CARRY 0)
           RES BP)
      A   (SETQ BP (\BN.IDIVIDE (IPLUS (CAR U)
                                       (IPLUS (CAR V)
                                              CARRY))
                          \BIGNUM.BETA))
          (SETQ CARRY (CAR BP))
          (SETQ RES (CONS (CDR BP)
                          RES))
          (SETQ U (CDR U))
          (SETQ V (CDR V))
          (COND
             (V (GO A)))
          [COND
             ((NEQ CARRY 0)
              (SETQ RES (CONS CARRY RES]
          (RETURN (REVERSE RES])

(\BN.ISUM1
  [LAMBDA (U V)                                                  (* lmm "20-JUL-84 02:22")
    (PROG (C S RES)
          (SETQ C 0)
          (SETQ S 0)
      A   (SETQ C (IPLUS (CAR U)
                         (CAR V)))
          (COND
             ((NEQ C 0)
              (SETQ S C)))
          (SETQ RES (CONS C RES))
          (SETQ U (CDR U))
          (SETQ V (CDR V))
          (COND
             (V (GO A)))
          (RETURN (COND
                     ((EQ S 0)
                      NIL)
                     (T (\BN.ICANON (DREVERSE RES)
                               (COND
                                  ((ILESSP S 0)
                                   -1)
                                  (T 1])
```

```
(\BN.MADD
  [LAMBDA (A B C)                                               (* lmm " 9-Jan-86 15:30")
    (PROG (H TT TTT IP IPP)
          (SETQ TT A)
          (SETQ H 0)
      L2  (SETQ IP (\BN.IDIVIDE (ITIMES B (CAR TT))
                          \BIGNUM.BETA))
          (SETQ IPP (\BN.IDIVIDE (IPLUS C (IPLUS (CDR IP)
                                                 H))
                           \BIGNUM.BETA))
          (RPLACA TT (CDR IPP))
          (SETQ H (CAR IP))
          (SETQ C (CAR IPP))
          (SETQ TTT TT)
          (SETQ TT (CDR TT))
          (COND
             (TT (GO L2)))
          (SETQ C (IPLUS C H))
          (COND
             ((EQ C 0)
              (RETURN A)))
          (RPLACD TTT (CONS C (CDR TTT)))
          (RETURN A])
```

```
(\BN.TO.FIXP
  [LAMBDA (X)                                                   (* kbr%: "11-Sep-86 14:58")
    (COND
       [(LISTP X)
        (COND
           ((OR (EQ (\BN.COMPAREN X (CONSTANT (\BN.FROM.FIXP MAX.FIXP)))
                    1)
                (EQ (\BN.COMPAREN X (CONSTANT (\BN.FROM.FIXP MIN.FIXP)))
                   -1))
            (create BIGNUM
                   ELEMENTS _ X))
           (T (\BIGNUM.TO.INT X]
       ((NULL X)
        0)
       ((EQ X 'MAX.INTEGER)
        MAX.INTEGER)
       ((EQ X 'MIN.INTEGER)
        MIN.INTEGER)
       (T (SHOULDNT])
```

```
(\BN.NZEROS
  [LAMBDA (N)                                                   (* lmm "20-JUL-84 02:30")
    (for I from 1 to N collect 0])
```

```
(\BN.QRS
  [LAMBDA (B I)                                                 (* lmm " 9-Jan-86 15:30")
    (PROG (D CP C1 C2)
          [COND
             ((NULL B)
              (RETURN (CONS B 0]
          (COND
             ((EQ I 0)
              (ERROR " QRS DIV BY 0 ")))
          (SETQ B (REVERSE B))
          (SETQ C1 0)
      A   (SETQ C2 (CAR B))
          (SETQ CP (\BN.IDIVIDE (IPLUS (ITIMES C1 \BIGNUM.BETA)
                                     C2)
                          I))
          [COND
             ((OR D (NOT (EQ (CAR CP)
                             0)))
              (SETQ D (CONS (CAR CP)
                            D]
          (SETQ B (CDR B))
          (SETQ C1 (CDR CP))
          [COND
             ((NULL B)
              (RETURN (CONS D C1]
          (GO A])

(\BN.SIGN
  [LAMBDA (U)                                                    (* kbr%: "11-Sep-86 15:22")
    (COND
       [(ATOM U)
        (COND
           ((NULL U)
            0)
           ((EQ U 'MAX.INTEGER)
            1)
           ((EQ U 'MIN.INTEGER)
            -1)
           (T (SHOULDNT]
       ((IGREATERP (CAR U)
               0)
        1)
       ((ILESSP (CAR U)
               0)
        -1)
       (T (\BN.SIGN (CDR U])

(\BN.TH2B
  [LAMBDA (U)                                                    (* lmm " 9-Jan-86 15:31")
    (PROG (AI B)
          (COND
             ((NULL U)
              (RETURN B)))
          (SETQ AI (CAR U))
          (SETQ U (CDR U))
          (SETQ B (CONS AI B))
          [while U do (PROGN (SETQ AI (CAR U))
                             (SETQ U (CDR U))
                             (SETQ B (\BN.MADD B \BIGNUM.THETA AI]
          (RETURN B])

(\BN.TH2D
  [LAMBDA (A RADIX TH)                                           (* bvm%: "15-Apr-86 14:28")

          (* * A is a list of integers obtained by repeatedly dividing some bignum by TH, which is a power of RADIX, hopefully chosen
          to keep the integers small. The elements of A concatenated thus make up the print name of the bignum in the indicated
          RADIX. Convert the list to a series of character codes by computing the print names of each subpart)

    (OR RADIX (SETQ RADIX 10))
    (COND
       ((NULL A)
        (LIST (CHARCODE 0)))
       (T (for AI in A bind (MAXFACTOR _ (IQUOTIENT TH RADIX))
                          DIGIT RESULT
             do [for (M _ MAXFACTOR) by (IQUOTIENT M RADIX) repeatuntil (EQ M 1)
                   do (SETQ DIGIT (IQUOTIENT AI M))
                      (SETQ AI (IDIFFERENCE AI (ITIMES DIGIT M)))
                      (COND
                         ((OR RESULT (NEQ DIGIT 0))
                          (push RESULT (COND
                                          ((GEQ (SETQ DIGIT (ABS DIGIT))
                                                10)              (* Use alphabetics for digits greater than 9)
                                           (IPLUS (IDIFFERENCE DIGIT 10)
                                                  (CHARCODE A)))
                                          (T (IPLUS DIGIT (CHARCODE 0]
             finally (RETURN (COND
                                ((IGREATERP (CAR A)
                                        0)
                                 (REVERSE RESULT))
                                (T                               (* Negative bignum)
                                   (CONS (CHARCODE -)
                                         (REVERSE RESULT])
)

(DEFINEQ

(\INITBIGNUMS
  [LAMBDA NIL                                                    (* JDS "1-JAN-99 22:00")

    ;; Initialize the BIGNUM datatype.

    ;; First, set up the type info so that newly created BIGNUM pages are correct.

    (\SETTYPEMASK (\TYPENUMBERFROMNAME 'BIGNUM)
           (LOGOR \TT.FIXP \TT.NUMBERP \TT.ATOM))
    ;; Now create some initial bignums for later use:

    (SETQ \BIG.0 (\BN.FROM.FIXP 0))                          ; BIGNUM of 0
    (SETQ \BIG.1 (\BN.FROM.FIXP 1))                          ; BIGNUM of 1
    (SETQ MIN.INTEGER (create BIGNUM
                             ELEMENTS _ 'MIN.INTEGER))
    (SETQ MAX.INTEGER (create BIGNUM
                             ELEMENTS _ 'MAX.INTEGER])
)

;; MAKERATIONAL needs work

;; needs work: MASK.1'S MASK.0'S BITTEST BITSET BITCLEAR LOGNOT LOADBYTE DEPOSITBYTE IMODLESSP IMODPLUS
;; IMODDIFFERENCE ROT

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\INITBIGNUMS)
)

(PUTPROPS LLBIGNUM COPYRIGHT ("Venue & Xerox Corporation" 1985 1986 1987 1990 1993 1999))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

RECORD INDEX