

File created: 28-Sep-91 12:55:09 {PELE:MV:ENVOS}<LISPUSERS>MEDLEY>PLAY.;2

changes to: (VARS PLAYCOMS DEMO.MELODY DEMO.TUNE)  
(RECORDS MELODY PASSAGE BEEP)  
(FNS PLAY.TRANSCRIBE)

previous date: 11-Jul-86 13:55:55 {PELE:MV:ENVOS}<LISPUSERS>MEDLEY>PLAY.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
:: Copyright (c) 1986, 1991 by Xerox Corporation. All rights reserved.

```
(RPAQQ PLAYCOMS
  (
    (COMS
      (CONSTANTS (PLAY.ROOM 10000)
        (PLAY.TOY 32)
        (PLAY.FREQA 1760)
        (PLAY.SILENT NIL))
      (INITVARS (PLAY.OCTAVE NIL)
        (PLAY.DURATION NIL)
        (PLAY.BREAK NIL)
        (PLAY.SHARP NIL)
        (PLAY.FLAT NIL)
        (PLAY.TWELFTHS NIL)
        (PLAY.POWERS NIL))
      (RECORDS MELODY PASSAGE BEEP)
      (FNS PLAY.RESTART PLAY.NOTES PLAY.MELODY PLAY.MELODY1 PLAY.OCTAVE PLAY.TEMPO PLAY.KEY PLAY.REPEAT
        PLAY.PASSAGE PLAY.NOTE PLAY.NOTE1))
    (COMS
      (INITVARS (PLAY.KEYBOARD.ALIST NIL)
        (PLAY.TRANSCRIBE.ALIST NIL))
      (FNS PLAY.KEYBOARD PLAY.TRANSCRIBE PLAY.ADJUST.TEMPO PLAY.ADJUST.PITCH))
    (COMS
      (INITVARS (PLAY.MESA.OCTAVE NIL)
        (PLAY.MESA.TRIPLE NIL)
        (PLAY.MESA.DURATION NIL)
        (PLAY.MESA.BREAK NIL))
      (FNS PLAY.MESA PLAY.MESA.NOTE PLAY.MESA.NOTE1))
    (COMS
      (VARS DEMO.MELODY DEMO.TUNE)
      (FNS PLAY.DEMO))
    (P (PLAY.RESTART)))
    ; PLAY -- By Kelly Roach. *
    ; PLAYLISP *
    ; PLAYKBD *
    ; PLAYMESA *
    ; DEMO *
```

:: PLAY -- By Kelly Roach. \*

:: PLAYLISP \*

(DECLARE%: EVAL@COMPILE

(RPAQQ **PLAY.ROOM** 10000)

(RPAQQ **PLAY.TOY** 32)

(RPAQQ **PLAY.FREQA** 1760)

(RPAQQ **PLAY.SILENT** NIL)

(CONSTANTS (PLAY.ROOM 10000)
 (PLAY.TOY 32)
 (PLAY.FREQA 1760)
 (PLAY.SILENT NIL))
)

(RPAQ? **PLAY.OCTAVE** NIL)

(RPAQ? **PLAY.DURATION** NIL)

(RPAQ? **PLAY.BREAK** NIL)

(RPAQ? **PLAY.SHARP** NIL)

(RPAQ? **PLAY.FLAT** NIL)

(RPAQ? **PLAY.TWELFTHS** NIL)

(RPAQ? **PLAY.POWERS** NIL)

(DECLARE%: EVAL@COMPILE

(TYPE RECORD MELODY (TEMPO KEY METER BEAT PASSAGES))

(RECORD PASSAGE (REPEATS MEASURES))

(RECORD BEEP (FREQ . DURATION))

)

(DEFINEQ

**(PLAY.RESTART**

```

[LAMBDA NIL (* kbr%: " 8-Feb-84 10:47")
 (PROG NIL (* PLAY. *)
  (SETQ PLAY.TWELFTHS (ARRAY 7 'FIXP 0 0))
  (for I from 0 to 6 as J in '(12 14 3 5 7 8 10) do (SETA PLAY.TWELFTHS I J))
  (SETQ PLAY.POWERS (ARRAY 12 'FIXP 0 0))
  (for I from 0 to 11 do (SETA PLAY.POWERS I (FIXR (FTIMES (EXPT 2.0 (FQUOTIENT (FLOAT I)
                                                                              12.0))
                                                    (FLOAT PLAY.ROOM]
                                                    (* Global Environment. *)
  (PLAY.OCTAVE 8)
  (PLAY.TEMPO 'MODERATE)
  (PLAY.KEY 'CMAJOR) (* KEYBOARD. *)
  [SETQ PLAY.KEYBOARD.ALIST
   (for CODE in (CHCON "ASDFGHJKL;'_WRTUIO[]") as NOTE
    in '(<a <b c d e f g a b >c >d >e <a# c# d# f# g# a# >c# >d#)
    collect (CONS CODE (fetch (BEEP FREQ) of (CAR (PLAY.NOTE NOTE]
  (SETQ PLAY.TRANScribe.ALIST
   (for BUCKET in PLAY.KEYBOARD.ALIST as NOTE
    in '(<a <b c d e f g a b >c >d >e <a# c# d# f# g# a# >c# >d#) collect (CONS (CDR BUCKET)
                                                                              NOTE])

```

**(PLAY.NOTES**

```

[LAMBDA (NOTES) (* kbr%: " 8-Feb-84 10:47")
 (* Converts series of NOTES into a TUNE suitable for playing by
  PLAYTUNE. *)
  (FOR NOTE IN NOTES JOIN (PLAY.NOTE NOTE])

```

**(PLAY.MELODY**

```

[LAMBDA (MELODY) (* kbr%: " 8-Feb-84 10:47")
 (* Converts a MELODY into a TUNE suitable for playing by
  PLAYTUNE. *)
  (PROG (TUNE)
   (COND
    ((NOT (TYPE? MELODY MELODY))
     (printout T "Illegal MELODY " MELODY " ignored")
     (RETURN NIL)))
   (PLAY.OCTAVE 8)
   (PLAY.TEMPO (fetch (MELODY TEMPO) of MELODY)
               (fetch (MELODY BEAT) of MELODY))
   (PLAY.KEY (fetch (MELODY KEY) of MELODY))
   (SETQ TUNE (PLAY.MELODY1 MELODY))
   (RETURN TUNE])

```

**(PLAY.MELODY1**

```

[LAMBDA (MELODY) (* kbr%: " 8-Feb-84 10:47")
 (PROG (MAXREPEAT TUNE) (* Calc MAXREPEAT. *)
  (SETQ MAXREPEAT (MIN.FIXP)
  [FOR PASSAGE IN (fetch (MELODY PASSAGES) of MELODY) DO (SETQ MAXREPEAT
   (IMAX MAXREPEAT (APPLY 'IMAX
                          (fetch (PASSAGE REPEATS)
                                  of PASSAGE]
   (* Calc TUNE. *)
  (SETQ TUNE (FOR REPEAT FROM 1 TO MAXREPEAT JOIN (PLAY.REPEAT MELODY REPEAT)))
  (RETURN TUNE])

```

**(PLAY.OCTAVE**

```

[LAMBDA (OCTAVE) (* kbr%: " 8-Feb-84 10:47")
 (PROG NIL
  (SETQ PLAY.OCTAVE OCTAVE])

```

**(PLAY.TEMPO**

```

[LAMBDA (TEMPO BEAT) (* kbr%: " 8-Feb-84 10:47")
 (* Establish TEMPO = beats per minute.
  *)
  (COND
   ((NULL BEAT)
    (SETQ BEAT 4)))
  (PROG NIL
   (SETQ TEMPO (SELECTQ TEMPO
                        ((ALLEGRO FAST)
                         120)
                        ((MODERATO MODERATE NIL)
                         90)

```

```

      ((ADAGIO SLOW)
       60)
      TEMPO)
    (SETQ PLAY.DURATION (IQUOTIENT (ITIMES 600000 BEAT)
                                   (ITIMES TEMPO 2]))

```

**PLAY.KEY**

[LAMBDA (KEY)

(\* kbr%: "11-Jul-86 12:59")  
(\* Establish KEY signature. \*)

```

  (PROG NIL
   [SETQ KEY (COND
    ((LISTP KEY)
     KEY)
    (T (SELECTQ KEY
      (CMAJOR '(%#))
      (GMAJOR '(%# F))
      (DMAJOR '(%# F C))
      (AMAJOR '(%# F C G))
      (EMAJOR '(%# F C G D))
      (BMAJOR '(%# F C G D A))
      (F#MAJOR '(%# F C G D A E))
      (C#MAJOR '(%# F C G D A E B))
      (FMAJOR '(@ B))
      (B@MAJOR '(@ B E))
      (E@MAJOR '(@ B E A))
      (A@MAJOR '(@ B E A D))
      (D@MAJOR '(@ B E A D G))
      (G@MAJOR '(@ B E A D G C))
      (C@MAJOR '(@ B E A D G C F))
      NIL]
    (SELECTQ (CAR KEY)
      (%# (SETQ PLAY.SHARP (CDR KEY))
          (SETQ PLAY.FLAT NIL))
      (@ (SETQ PLAY.SHARP NIL)
         (SETQ PLAY.FLAT (CDR KEY)))
      (PROGN (SETQ PLAY.SHARP NIL)
             (SETQ PLAY.FLAT NIL)))
    (PROGN (SETQ PLAY.SHARP (for LETTER in PLAY.SHARP collect (CHCON1 LETTER)))
           (SETQ PLAY.FLAT (for LETTER in PLAY.FLAT collect (CHCON1 LETTER)))))

```

**PLAY.REPEAT**

[LAMBDA (MELODY REPEAT)

(\* kbr%: " 8-Feb-84 10:47")  
(\* Return TUNE for this REPEAT of MELODY.  
\*)

```

  (FOR PASSAGE IN (fetch (MELODY PASSAGES) of MELODY) WHEN (MEMB REPEAT (fetch (PASSAGE REPEATS) of PASSAGE))
   JOIN (PLAY.PASSAGE PASSAGE])

```

**PLAY.PASSAGE**

[LAMBDA (PASSAGE)

(\* kbr%: " 8-Feb-84 10:47")  
(\* Return TUNE for PASSAGE. \*)

```

  (FOR MEASURE IN (fetch (PASSAGE MEASURES) of PASSAGE) JOIN (PLAY.NOTES MEASURE])

```

**PLAY.NOTE**

[LAMBDA (NOTE)

(\* kbr%: "11-Jul-86 12:56")  
(\* Return TUNE for NOTE. \*)

```

  (PROG (LETTER OCTAVE DURATION BREAK ACCIDENTAL DOTS TUNE)
   (SETQ OCTAVE PLAY.OCTAVE)
   (SETQ DURATION PLAY.DURATION)
   (SETQ BREAK PLAY.ROOM)
   (SETQ DOTS 0)
   [for CODE in (CHCON NOTE)
    do (SELCHARQ CODE
      ((A B C D E F G R a b c d e f g r)
       (SETQ LETTER CODE))
      ("<" (SETQ OCTAVE (ITIMES OCTAVE 2)))
      (">" (SETQ OCTAVE (IQUOTIENT OCTAVE 2)))
      ("x" (SETQ DURATION (ITIMES DURATION 2)))
      ("/" "/" (SETQ DURATION (IQUOTIENT DURATION 2)))
      ("1" "2" "3" "4" "5" "6" "7" "8" "9")
       [SETQ DURATION (IQUOTIENT (ITIMES (CHARACTER CODE)
                                     DURATION)
                               (ADD1 (CHARACTER CODE)))]
      ("+" (SETQ DOTS (ADD1 DOTS)))
      ("^" (SETQ BREAK (ITIMES BREAK 2)))
      ("_" (SETQ BREAK (IQUOTIENT BREAK 2)))
      "@" [COND
        (NULL ACCIDENTAL)
        (SETQ ACCIDENTAL -1))
        (T (SETQ ACCIDENTAL (SUB1 ACCIDENTAL))]
      "#" [COND
        (NULL ACCIDENTAL)
        (SETQ ACCIDENTAL 1))
        (T (SETQ ACCIDENTAL (ADD1 ACCIDENTAL))]

```

```

      ( (N n)
        (SETQ ACCIDENTAL 0))
      (PROGN (printout T "Illegal note " NOTE " ignored.")
              (RETURN NIL])
[COND
  ((NULL ACCIDENTAL)
   (COND
    ((MEMB LETTER PLAY.SHARP)
     (SETQ ACCIDENTAL 1))
    ((MEMB LETTER PLAY.FLAT)
     (SETQ ACCIDENTAL -1))
    (T (SETQ ACCIDENTAL 0))
   [SELECTQ DOTS
    (0 (* Do nothing. *))
    (1 (SETQ DURATION (IQUOTIENT (ITIMES 3 DURATION)
                                  2)))
    (2 (SETQ DURATION (IQUOTIENT (ITIMES 7 DURATION)
                                  4)))
    (SETQ DURATION (IQUOTIENT (ITIMES (SUB1 (EXPT 2 (ADD1 DOTS)))
                                DURATION)
                              (EXPT 2 DOTS] (* Compute DURATION & BREAK.
*))
(SETQ BREAK (IQUOTIENT (ITIMES 3 BREAK DURATION)
                       (ITIMES 8 PLAY.ROOM)))
(SELCHARQ LETTER
  ((A B C D E F G R)
   (SETQ BREAK 0))
  (r (SETQ BREAK 0)
     (SETQ LETTER (CHARCODE R)))
  ((a b c d e f g)
   (SETQ BREAK (IMIN BREAK DURATION))
   (SETQ DURATION (IDIFFERENCE DURATION BREAK))
   [SETQ LETTER (IPLUS LETTER (IDIFFERENCE (CHARCODE A)
                                             (CHARCODE a))]
   (SHOULDNT) (* Compute TUNE. *)
(SETQ TUNE (PLAY.NOTE1 OCTAVE LETTER ACCIDENTAL DURATION BREAK))
(RETURN TUNE])

```

(PLAY.NOTE1

```

[LAMBDA (OCTAVE LETTER ACCIDENTAL DURATION BREAK) (* kbr%: " 8-Feb-84 10:47")
  (PROG (FREQ TWELFTH TUNE) (* Compute FREQ. *)
    [COND
      ((IEQP LETTER (CHARCODE R)) (* Rest. *)
       (SETQ FREQ PLAY.SILENT))
      (T (* Compute TWELFTH. *)
         [SETQ TWELFTH (ELT PLAY.TWELFTHS (IDIFFERENCE LETTER (CHARCODE A)
                                                             (CHARCODE a))
                                         (IDIFFERENCE DURATION BREAK))
              (SETQ TWELFTH (IPLUS TWELFTH ACCIDENTAL))
         [COND
           [(IGEQ TWELFTH 12)
            (while (IGEQ TWELFTH 12) do (SETQ TWELFTH (IDIFFERENCE TWELFTH 12))
                  (SETQ OCTAVE (IQUOTIENT OCTAVE 2))
            ((ILESSP TWELFTH 0)
             (WHILE (ILESSP TWELFTH 0) DO (SETQ TWELFTH (IPLUS TWELFTH 12))
                                           (SETQ OCTAVE (ITIMES OCTAVE 2))

```

(\* Constant PLAY.TOY is to help reduce round off error. Adding (IQUOTIENT OCTAVE 2) then dividing by OCTAVE amounts to dividing by OCTAVE then adding one-half, but less round off error. \*)

```

      (SETQ FREQ (ITIMES PLAY.TOY PLAY.FREQA))
      (SETQ OCTAVE (ITIMES PLAY.TOY OCTAVE))
      (SETQ FREQ (IQUOTIENT (ITIMES FREQ (ELT PLAY.POWERS TWELFTH)
                                     PLAY.ROOM))
                            (IPLUS FREQ (IQUOTIENT OCTAVE 2))
                            (* Compute TUNE. *)
      [COND
        ((NOT (ZEROP BREAK))
         (push TUNE (create BEEP
                            FREQ _ PLAY.SILENT
                            DURATION _ BREAK]
        [COND
          ((NOT (ZEROP DURATION))
           (push TUNE (create BEEP
                              FREQ _ FREQ
                              DURATION _ DURATION] (* Okey Dokey. *)
      (RETURN TUNE])
)

```

:: PLAYKBD \*

(RPAQ? PLAY.KEYBOARD.ALIST NIL)

(RPAQ? PLAY.TRANSSCRIBE.ALIST NIL)

(DEFINEQ

**(PLAY.KEYBOARD**

```

[LAMBDA NIL
  (PROG (KEY FREQ TUNE CLOCK OLDCLOCK OCTAVE)
    (SETQ OLDCLOCK (CLOCK 0))
    (SETQ OCTAVE PLAY.OCTAVE)
    [do
      (SETQ KEY (\GETKEY))
      (SELCHARQ KEY
        (" " (RETURN))
        (">" (SETQ OCTAVE (ITIMES OCTAVE 2)))
        ("<" (SETQ OCTAVE (IQUOTIENT OCTAVE 2))))
      (PROGN
        (SETQ CLOCK (CLOCK 0))
        (push TUNE (create BEEP
          FREQ _ PLAY.SILENT
          DURATION _ (IDIFFERENCE CLOCK OLDCLOCK)))
        (SETQ OLDCLOCK CLOCK)
        (push TUNE (create BEEP
          FREQ _ (CDR (ASSOC KEY PLAY.KEYBOARD.ALIST)))
          DURATION _ (IQUOTIENT (ITIMES FREQ OCTAVE)
            PLAY.OCTAVE))
        (COND
          (FREQ (BEEPON FREQ)
            (while (KEYDOWNP (CHARACTER KEY)) do
              (BEEPOFF)))
          (SETQ CLOCK (CLOCK 0))
          (push TUNE (create BEEP
            FREQ _ FREQ
            DURATION _ (IDIFFERENCE CLOCK OLDCLOCK)))
          (SETQ OLDCLOCK CLOCK)
          (SETQ TUNE (DREVERSE TUNE))
          (RETURN TUNE]))
    ]
  )
]

```

(\* kbr%: " 5-Feb-84 19:55")  
 (\* Get note. \*)  
 (\* Record rest. \*)

**(PLAY.TRANSCRIBE**

```

[LAMBDA (TUNE)
  (for BEEP in TUNE when (NOT (EQ (fetch (BEEP FREQ) of BEEP)
    PLAY.SILENT))
    collect (CDR (ASSOC (fetch (BEEP FREQ) of BEEP)
      PLAY.TRANSCRIBE.ALIST]))
]

```

; Edited 28-Sep-91 12:54 by jds  
 (\* Transcribe TUNE into NOTES.  
 \*)

**(PLAY.ADJUST.TEMPO**

```

[LAMBDA (TUNE FACTOR)
  (PROG (ANSWER)
    (SETQ FACTOR (FIXR (FTIMES (FLOAT FACTOR)
      PLAY.ROOM)))
    [SETQ ANSWER (for BEEP in TUNE collect (create BEEP
      FREQ _ (fetch (BEEP FREQ) of BEEP)
      DURATION _ (IQUOTIENT (ITIMES FACTOR (fetch (BEEP DURATION)
        of BEEP))
        PLAY.ROOM)]
    (RETURN ANSWER])
]

```

(\* kbr%: " 5-Feb-84 19:55")  
 (\* Adjust tempo of TUNE by FACTOR.  
 \*)

**(PLAY.ADJUST.PITCH**

```

[LAMBDA (TUNE SEMITONES)
  (PROG (FACTOR ANSWER)
    (SETQ FACTOR (FIXR (FTIMES (EXPT 2.0 (FQUOTIENT (FLOAT SEMITONES)
      12.0))
      PLAY.ROOM)))
    [SETQ ANSWER (for BEEP in TUNE collect (create BEEP
      FREQ _ (COND
        ((EQ (fetch (BEEP FREQ) of BEEP)
          PLAY.SILENT)
          PLAY.SILENT)
        (T (IQUOTIENT (ITIMES FACTOR (fetch (BEEP FREQ)
          of BEEP))
          PLAY.ROOM)))
      DURATION _ (fetch (BEEP DURATION) of BEEP)]
    (RETURN ANSWER])
]

```

(\* kbr%: " 5-Feb-84 19:55")  
 (\* Adjust pitch of TUNE by number of SEMITONES.  
 \*)

:: PLAYMESA \*

(RPAQ? **PLAY.MESA.OCTAVE** NIL)

(RPAQ? **PLAY.MESA.TRIPLE** NIL)

(RPAQ? PLAY.MESA.DURATION NIL)

(RPAQ? PLAY.MESA.BREAK NIL)

(DEFINEQ

**(PLAY.MESA**

```
[LAMBDA (STRING DURATION) (* kbr%: " 5-Feb-84 19:55")
  (COND
    ((NULL DURATION)
     (SETQ DURATION 0))
    (PROG (STREAM NOTES NOTE)
      (SETQ PLAY.MESA.NOTES NIL)
      (SETQ PLAY.MESA.OCTAVE 0)
      (SETQ PLAY.MESA.TRIPLE NIL)
      (SETQ PLAY.MESA.DURATION DURATION)
      (SETQ PLAY.MESA.BREAK 0)
      (SETQ STREAM (OPENSTRINGSTREAM STRING))
      (SETQ NOTES (while (AND (NOT (EOFP STREAM))
                              (SETQ NOTE (PLAY.MESA.NOTE STREAM)))
                          collect NOTE))
      (RETURN NOTES])
```

**(PLAY.MESA.NOTE**

```
[LAMBDA (STREAM) (* kbr%: " 5-Feb-84 19:55")
  (* Read mesa note then add lisp note to PLAY.MESA.NOTES. *)
  (* Read mesa note. *)
  (PROG (LETTER SHARP DURATION DOT CODE NOTE)
    (SETQ SHARP 0)
    (SETQ DURATION PLAY.MESA.DURATION)
    (SETQ DOT 0)
    (while [AND (NOT (EOFP STREAM))
                (OR (NULL LETTER)
                    (MEMB (\PEEKBIN STREAM)
                          '(%# -])
                do (SETQ CODE (\BIN STREAM))
                  (SELCHARQ CODE
                    ((A B C D E F G a b c d e f g)
                     (SETQ LETTER CODE))
                    (% (SETQ LETTER (CHARCODE r)))
                    ("#" (SETQ SHARP (ADD1 SHARP)))
                    (">" (SETQ PLAY.MESA.OCTAVE (ADD1 PLAY.MESA.OCTAVE)))
                    ("<" (SETQ PLAY.MESA.OCTAVE (SUB1 PLAY.MESA.OCTAVE)))
                    ("/" "/" (SETQ PLAY.MESA.DURATION (SUB1 PLAY.MESA.DURATION))
                     (SETQ DURATION (SUB1 DURATION)))
                    "*" (SETQ PLAY.MESA.DURATION (ADD1 PLAY.MESA.DURATION))
                     (SETQ DURATION (ADD1 DURATION)))
                    "^" (SETQ PLAY.MESA.BREAK (ADD1 PLAY.MESA.BREAK)))
                    "-" (SETQ PLAY.MESA.BREAK (SUB1 PLAY.MESA.BREAK)))
                    "+" (SETQ DOT (ADD1 DOT))
                    "-" (SETQ DURATION (SUB1 DURATION)))
                    (" (" (SETQ PLAY.MESA.TRIPLE T))
                    (")" (SETQ PLAY.MESA.TRIPLE NIL)))
                    (* Do nothing. *)
                    (* Calc Lisp note. *)
    [COND
      (LETTER (SETQ NOTE (PLAY.MESA.NOTE1 LETTER SHARP DURATION DOT)
                        (* Okey Dokey. *)
    (RETURN NOTE])
```

**(PLAY.MESA.NOTE1**

```
[LAMBDA (LETTER SHARP DURATION DOT) (* kbr%: " 5-Feb-84 19:55")
  (PROG (NOTE)
    [SETQ NOTE (PACKC (\QUOTE ([\COMMAAT (COND
      ((NOT (IEQP LETTER (CHARCODE r)))
       (COND
         ((IGREATERP PLAY.MESA.OCTAVE 0)
          (FOR I FROM 1 TO PLAY.MESA.OCTAVE
              COLLECT (CHARCODE >)))
         ((ILESSP PLAY.MESA.OCTAVE 0)
          (FOR I FROM 1 TO (IMINUS PLAY.MESA.OCTAVE)
              COLLECT (CHARCODE <])
      (\COMMA LETTER)
      (\COMMAAT (FOR I FROM 1 TO SHARP COLLECT (CHARCODE %#)))
      [\COMMAAT (COND
        (PLAY.MESA.TRIPLE (LIST (CHARCODE "3")
      [\COMMAAT (COND
        ((IGREATERP DURATION 0)
         (FOR I FROM 1 TO DURATION COLLECT (CHARCODE x)))
        ((ILESSP DURATION 0)
         (FOR I FROM 1 TO (IMINUS DURATION)
             COLLECT (CHARCODE /]
      (\COMMAAT (FOR I FROM 1 TO DOT COLLECT (CHARCODE +]
    (RETURN NOTE])
```

)

:: DEMO \*

(RPAQQ DEMO.MELODY

```
[MELODY MODERATE (%# F)
  4 4 ((1)
    ((b/+ >c//)
     (>Dx >d/+ b// a/+ b//)
     (gx+ b/+ b//)
     (A/+ G// E e/+ e// b/+ b//)
     (ax+ b/+ >c//)
     (>Dx >d/+ b// a/+ b//)
     (gx+ b/+ b//)
     (A/+ G// E e/+ e// b/+ b//)
     (ax+ b/ >c//)
     (>Dx >d/+ b// a/+ b//)
     (gx+ b/+ b//)
     (A/+ G// E e/+ e// b/+ b//)
     (ax+ b/ >c//+)
     (>Dx >d/+ b// a/+ b//)
     (gx+ b/+ b/)
     (a/+ g// ex b/+ b//)
     (axx)))
    ((1 2 3)
     ((>d >d >d >d/+ >e//)
      (a/+ a// a/+ a// ax)
      (a/+ a// a/+ a// ax)
      (g/+ g// g/+ g// gx)
      (>d >d >d >d/+ >e//)
      (a/+ a// a/+ a// ax)
      (a/+ a// a/+ a// ax)
      (g/+ g// g/+ g// g)))
    ((1 2)
     ((b/+ >c//)
      (>Dx >d/+ b// a/+ b/)
      (gx+ b/+ b//)
      (a/+ g// ex b/+ b//)
      (ax+ b/+ >c/)
      (>Dx >d/+ b// a/+ b//)))
    ((1)
     ((gx+ d3/ e3/ f3/)
      (g/ B// A#// b/ B// A#// b <d3/ <e3/ <f3/)
      (<g <f <e <d)))
    ((2)
     ((gx+ b/+ b//)
      (A/+ g// ex b/+ b//)
      (axx])
```

(RPAQQ DEMO.TUNE

- ( (494 . 6250)
- (NIL . 3749)
- (523 . 2084)
- (NIL . 1249)
- (587 . 266666)
- (587 . 6250)
- (NIL . 3749)
- (494 . 2084)
- (NIL . 1249)
- (440 . 6250)
- (NIL . 3749)
- (494 . 2084)
- (NIL . 1249)
- (392 . 25000)
- (NIL . 14999)
- (494 . 6250)
- (NIL . 3749)
- (494 . 2084)
- (NIL . 1249)
- (440 . 9999)
- (392 . 3333)
- (330 . 13333)
- (330 . 6250)
- (NIL . 3749)
- (330 . 2084)
- (NIL . 1249)
- (494 . 6250)
- (NIL . 3749)
- (494 . 2084)
- (NIL . 1249)
- (440 . 25000)
- (NIL . 14999)
- (494 . 6250)
- (NIL . 3749)
- (523 . 2084)
- (NIL . 1249)
- (587 . 266666)

(587 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(392 . 25000)  
(NIL . 14999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 9999)  
(392 . 3333)  
(330 . 13333)  
(330 . 6250)  
(NIL . 3749)  
(330 . 2084)  
(NIL . 1249)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 25000)  
(NIL . 14999)  
(494 . 4167)  
(NIL . 2499)  
(523 . 2084)  
(NIL . 1249)  
(587 . 26666)  
(587 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(392 . 25000)  
(NIL . 14999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 9999)  
(392 . 3333)  
(330 . 13333)  
(330 . 6250)  
(NIL . 3749)  
(330 . 2084)  
(NIL . 1249)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 25000)  
(NIL . 14999)  
(494 . 4167)  
(NIL . 2499)  
(523 . 3125)  
(NIL . 1874)  
(587 . 26666)  
(587 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(392 . 25000)  
(NIL . 14999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 4167)  
(NIL . 2499)  
(440 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(330 . 16667)  
(NIL . 9999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)



(NIL . 1249)  
(440 . 33333)  
(NIL . 19999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 6250)  
(NIL . 3749)  
(659 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 16667)  
(NIL . 9999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 6250)  
(NIL . 3749)  
(659 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 8334)  
(NIL . 4999)  
(494 . 6250)  
(NIL . 3749)  
(523 . 2084)  
(NIL . 1249)

(587 . 26666)  
(587 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(494 . 4167)  
(NIL . 2499)  
(392 . 25000)  
(NIL . 14999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(330 . 16667)  
(NIL . 9999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 25000)  
(NIL . 14999)  
(494 . 6250)  
(NIL . 3749)  
(523 . 4167)  
(NIL . 2499)  
(587 . 26666)  
(587 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(392 . 25000)  
(NIL . 14999)  
(294 . 3125)  
(NIL . 1874)  
(330 . 3125)  
(NIL . 1874)  
(349 . 3125)  
(NIL . 1874)  
(392 . 4167)  
(NIL . 2499)  
(494 . 3333)  
(466 . 3333)  
(494 . 4167)  
(NIL . 2499)  
(494 . 3333)  
(466 . 3333)  
(494 . 8334)  
(NIL . 4999)  
(147 . 3125)  
(NIL . 1874)  
(165 . 3125)  
(NIL . 1874)  
(175 . 3125)  
(NIL . 1874)  
(196 . 8334)  
(NIL . 4999)  
(175 . 8334)  
(NIL . 4999)  
(165 . 8334)  
(NIL . 4999)  
(147 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 6250)  
(NIL . 3749)  
(659 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)

(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 16667)  
(NIL . 9999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 6250)  
(NIL . 3749)  
(659 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 8334)  
(NIL . 4999)  
(494 . 6250)  
(NIL . 3749)  
(523 . 2084)  
(NIL . 1249)  
(587 . 26666)  
(587 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(494 . 4167)  
(NIL . 2499)  
(392 . 25000)  
(NIL . 14999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(392 . 2084)

(NIL . 1249)  
(330 . 16667)  
(NIL . 9999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 25000)  
(NIL . 14999)  
(494 . 6250)  
(NIL . 3749)  
(523 . 4167)  
(NIL . 2499)  
(587 . 26666)  
(587 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(392 . 25000)  
(NIL . 14999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 9999)  
(392 . 2084)  
(NIL . 1249)  
(330 . 16667)  
(NIL . 9999)  
(494 . 6250)  
(NIL . 3749)  
(494 . 2084)  
(NIL . 1249)  
(440 . 33333)  
(NIL . 19999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 6250)  
(NIL . 3749)  
(659 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 16667)  
(NIL . 9999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 8334)  
(NIL . 4999)  
(587 . 6250)  
(NIL . 3749)

(659 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 6250)  
(NIL . 3749)  
(440 . 2084)  
(NIL . 1249)  
(440 . 16667)  
(NIL . 9999)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 6250)  
(NIL . 3749)  
(392 . 2084)  
(NIL . 1249)  
(392 . 8334)  
(NIL . 4999))

(DEFINEQ

(PLAY.DEMO

[LAMBDA NIL  
(PLAYTUNE DEMO.TUNE)]

(\* kbr%: " 5-Feb-84 21:14")

)

(PLAY.RESTART)

(PUTPROPS PLAY COPYRIGHT ("Xerox Corporation" 1986 1991))

---

**FUNCTION INDEX**

PLAY.ADJUST.PITCH .5	PLAY.KEYBOARD .....5	PLAY.MESA.NOTE ....6	PLAY.NOTES .....2	PLAY.RESTART .....2
PLAY.ADJUST.TEMPO .5	PLAY.MELODY .....2	PLAY.MESA.NOTE1 ...6	PLAY.OCTAVE .....2	PLAY.TEMPO .....2
PLAY.DEMO .....13	PLAY.MELODY1 .....2	PLAY.NOTE .....3	PLAY.PASSAGE .....3	PLAY.TRANScribe ...5
PLAY.KEY .....3	PLAY.MESA .....6	PLAY.NOTE1 .....4	PLAY.REPEAT .....3	

---

**VARIABLE INDEX**

DEMO.MELODY .....7	PLAY.FLAT .....1	PLAY.MESA.OCTAVE .....5	PLAY.SHARP .....1
DEMO.TUNE .....7	PLAY.KEYBOARD.ALIST .....4	PLAY.MESA.TRIPLE .....5	PLAY.TRANScribe.ALIST ...4
PLAY.BREAK .....1	PLAY.MESA.BREAK .....6	PLAY.OCTAVE .....1	PLAY.TWELFTHS .....1
PLAY.DURATION .....1	PLAY.MESA.DURATION .....6	PLAY.POWERS .....1	

---

**CONSTANT INDEX**

PLAY.FREQA .....1	PLAY.ROOM .....1	PLAY.SILENT .....1	PLAY.TOY .....1
-------------------	------------------	--------------------	-----------------

---

**RECORD INDEX**

BEEP .....2	MELODY .....1	PASSAGE .....2
-------------	---------------	----------------

---