

# hex

ROOT ROOT

PRG P7 P7 P7 P7 P2A B A D E1 E2

KB E E E E 2 G \* param: d=100 p.=12 param: temp=10 ampl=12 param: gliss=25 KB PLAY 1-2-3-4

Mon PIZZ ON suspended violin GOTO P2B BLOW IT GOTO P5 BLOW IT

Mus Mon +P2B acc.

ROOT

PRG P5 B D B C P6 E P6

KB 5 D 20 PLAY KB 1,2,3,4,5,6 PLAY KB ALL \* param: tempo=25 param: tempo=10

Mon GOTO P6 BLOW STRINGS GOTO VIOLIN PLAY

automatico accelerando STAPPENMOTOREN BALLET p long notes

ROOT ROOT STOKASTIC LETTERS ROOT

PRG P3 P2 B E D G

KB 3 param: E=2 P=25 H=100 BLOW P3 \* 8 B PLAY TYPE ALLOPHONES \* \* \* PLAY TYPE \* G param: 32 PLAY A-Z ↑ A-Z ↓

Mon viol Interaktief +vokas! glissandi

ROOT A Drumsolo STOCH. TUTTI TELTUTTI H P4

PRG P4 B G D H

KB 4 param: d=100 t=24 H=1 param: E1=1 E2=4 param: manual P3 TRIGGER Y/N

Mon (echo pizz) l'attacca! ritmisch ff martellato



# HEX 0889

T1000 HEX-versie - last update 08.09.1989 - Gent

DECLARE SUB INSTALL (DP%, CP%, SP%)

DECLARE SUB ROM (RP0%(), RP1%(), RP6%(), RP8%())

DEFINT A-V

CALL INSTALL(DP, CP, SP)

DIM RP1(&HFF): DIM RP8(&H2FF): DIM RP6(&H27F): DIM RP0(&H1FF)

DIM A(10): DIM B(10)

CALL ROM(RP0(), RP1(), RP6(), RP8())

MENU:

GOSUB BLANKALL

LOCATE 10, 10: PRINT "Welke Hex versie wil je horen ... : "

LOCATE 11, 10: PRINT " 1. - HEX- Concerto met Viool - Logos-Duo"

LOCATE 12, 10: PRINT " 2. - HEX- Automatische versie "

LOCATE 13, 10: PRINT " 3. - HEX- Debug mode for Research - Manual"

LOCATE 14, 10: PRINT " 9. - QUIT "

LOCATE 15, 10: INPUT " KEUZE CIJFER ? ", HM

IF HM = 9 THEN END

IF HM < 1 OR HM > 3 THEN GOTO MENU

GOSUB BLANKLINES

P1 = 96: P2 = P1 + 2: P3 = P2 + 2: P4 = P3 + 2

P5 = P4 + 2: P6 = P5 + 2: P7 = P6 + 2: P8 = P7 + 2

mid-poort initialiseren voor REX50

OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A\$ = "RAES": OUT SP, P1

OUT DP, 30: OUT SP, P1 + 1: A\$ = "RAES": OUT SP, P1

START\$ = TIME\$

LOCATE 1, 10: PRINT " H E X - 0889

T1000 versie Braslie"

LOCATE 2, 10: PRINT " \*\*\*\*\*

by:

LOCATE 3, 10: PRINT "

Godfried-Willem RAES"

LOCATE 4, 10: PRINT "

Logos - Duo Production "

LOCATE 5, 10: PRINT "

ROOTLOOP:

GOSUB BLANKLINES

LOCATE 8, 5: PRINT "ROOT-LOOP Nummer="; CY; " Speelsektie? ( 1 tot G)"

GOSUB TALK7

CY = CY + 1

ROOTTST:

K\$ = INKEY\$

IF K\$ = "\*" THEN

GOSUB ALLOFF

BND

BND IF

IF HM = 2 THEN K\$ = "F"

IF K\$ = "F" THEN HM = 2

IF K\$ = "" THEN GOTO ROOTTST

IF (K\$ < "1") OR (K\$ > "G") THEN GOTO ROOTTST

A = ASC(K\$)

LOCATE 9, 10: PRINT "Programma loopt nu naar sekte &H "; K\$

IF A > 57 AND A < 72 THEN B = 55 ELSE B = 48

K = A - B: IF K < 0 OR K > 16 THEN GOTO ROOTTST

GOSUB BLANKLINES

ON K GOSUB MIDI, VIOOL, MDOSC, DRUM, STRISPRI, STAPMOT, TALK7, TALK8, TUT, BUILD, BARS,  
STOCH, TELTUTTI,

COUNT, ROBOT, GENGLIS

GOTO ROOTLOOP

\*\*\*\*\*

1

```

MIDI:
Poort 1 - kaart 1 :                               Unidirectionele Midi-kaart voor REX50
IF HM = 1 THEN RETURN
GOSUB BLANKLINES
LOCATE 15, 10: PRINT "Midi-REX50 sturingsroutine - Stop met * -toets"
      OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
      OUT DP, 16: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
MITSTKY:
K$ = INKEY$
IF K$ = "*" THEN RETURN ELSE GOTO MISPEL:
MISPEL:
FOR J = 0 TO 24: A = INT(RND(1) * J + 47)
      OUT DP, 144: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
      A$ = "RAES"
      OUT DP, A: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
      A$ = "RAES"
      OUT DP, 60: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
SOUND 20000, RND(1) * 40
      OUT DP, 128: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
      A$ = "RAES"
      OUT DP, A: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
      A$ = "RAES"
      OUT DP, 60: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
NEXT J
GOTO MITSTKY:

```

2

```

VIOOL:
Poort 2 : printernaalddriver voor opgehangen viool
GOSUB BLANKLINES: OUT SP, P2
LOCATE 15, 10: PRINT " POORT "; K: "Viool-pizzicato-speelautomaat"
LOCATE 16, 10: PRINT "          KIES  A, B, C , D , E  of * to quit "
K2TST:
K$ = INKEY$: IF K$ = "" THEN GOTO K2TST
IF K$ = "*" THEN GOTO K2END
IF K$ = "A" OR K$ = "a" THEN GOTO P2A1
IF K$ = "B" OR K$ = "b" THEN GOTO P2A2
IF K$ = "C" OR K$ = "c" THEN GOTO P2A3
IF K$ = "D" OR K$ = "d" THEN GOTO P2A4
IF K$ = "E" OR K$ = "e" THEN GOSUB P2B ELSE GOTO VIOOL
GOTO K2TST

```

A

```

P2A1:
LOCATE 17, 10: PRINT "          Poort 2- routine A loopt nu... "
FOR I = 8 TO 0 STEP -2
  FOR B = 0 TO 15
    OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
    OUT DP, B: OUT SP, P2 + 1: OUT SP, P2
    SOUND 20000, (I + 1) * 2
  NEXT B
  OUT DP, 128: OUT SP, P2 + 1: OUT SP, P2
NEXT I
FOR I = 0 TO 6 STEP 2
  FOR B = 16 TO 0 STEP -1
    FOR A = 0 TO I
      OUT DP, B: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 2
    NEXT A
    OUT DP, B * 4: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, I + 1
  NEXT B
  OUT DP, 64: OUT SP, P2 + 1: OUT SP, P2
NEXT I

```

```

OUT DP, 32: OUT SP, P2 + 1: OUT SP, P2
SOUND 20000, 4
OUT DP, 16: OUT SP, P2 + 1: OUT SP, P2
SOUND 20000, 4
OUT DP, 0: OUT SP, P2 + 1: OUT SP, P2
LOCATE 17, 10: PRINT "      Einde routine A -   vicolkaart      "
GOSUB BLANKLINES
IF HM = 1 THEN GOTO P2A4 ELSE RETURN

```

B

```

P2A2:
LOCATE 17, 10: PRINT "      Routine B - vicolkaart "
LOCATE 18, 10: INPUT "DUUR ? (100)      ", A
  IF A = 0 THEN A = 100
LOCATE 19, 10: INPUT "PULSDUUR ? (12)   ", B
  IF B = 0 THEN B = 12
LOCATE 18, 10: PRINT "
LOCATE 19, 10: PRINT "
LUS:
FOR I = 0 TO A
  OUT DP, 1: OUT SP, P2 + 1: OUT SP, P2
  IF I = A \ 4 THEN
    OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
    A$ = "RAES": OUT DP, 30: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
    END IF
  IF I >= A \ 4 AND I <= A \ 3 THEN
    SOUND 20000, B \ 2: OUT DP, 2: OUT SP, P2 + 1: OUT SP, P2
    SOUND 20000, B \ 2: OUT DP, 1: OUT SP, P2 + 1: OUT SP, P2
    END IF
  IF I >= A \ 2 AND I <= 3 * A \ 4 THEN
    SOUND 20000, B \ 3: OUT DP, 4: OUT SP, P2 + 1: OUT SP, P2
    SOUND 20000, B \ 3: OUT DP, 2: OUT SP, P2 + 1: OUT SP, P2
    SOUND 20000, B \ 3: OUT DP, 1: OUT SP, P2 + 1: OUT SP, P2
    END IF
  IF I > (3 * A \ 4) THEN
    SOUND 20000, B \ 4: OUT DP, 8: OUT SP, P2 + 1: OUT SP, P2
    SOUND 20000, B \ 4: OUT DP, 4: OUT SP, P2 + 1: OUT SP, P2
    SOUND 20000, B \ 4: OUT DP, 2: OUT SP, P2 + 1: OUT SP, P2
    SOUND 20000, B \ 4: OUT DP, 1: OUT SP, P2 + 1: OUT SP, P2
    END IF
  IF I < (A \ 4) THEN
    SOUND 20000, B: OUT DP, 1: OUT SP, P2 + 1: OUT SP, P2
    END IF
NEXT I
OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
A$ = "RAES": OUT DP, 31: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
LOCATE 17, 10: PRINT "      Einde routine B vicol-kaart      "
LOCATE 18, 10: PRINT "
LOCATE 19, 10: PRINT "
IF HM = 1 THEN
  GOSUB BLANKLINES
  GOTO P2A1
ELSE
  RETURN
END IF

```

C

```

P2A3:
'stochastisch stuk begin vicol-kaart
FOR I = 15 TO 20
LOCATE 1, 10: PRINT "
NEXT I

```

```

LOCATE 17, 10: PRINT "Poort 2- C - Stochastisch spel op de viool"
LOCATE 18, 10: PRINT " Stop deze subroutine met de * toets ..."
P2A3LUS:
  J = 0
  FOR I = 0 TO 3
    A(I) = INT(2 * (RND(1) ^ 2)); J = J + (2 ^ I) * A(I)
    LOCATE 19, 10 + (I * 4): PRINT A(I);
  NEXT I
  OUT DP, J: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, RND(1) * 10
  FOR I = 0 TO 3: B(2) = B(2) + A(I): NEXT I
  IF B(2) = 1 THEN GOTO P2A3LUS
  SOUND 20000, RND(1) * 10
K$ = INKEY$
  IF K$ = "*" THEN
    IF HM = 1 THEN
      GOSUB BLANKLINES
      GOTO P2A2
    END IF
    IF HM = 3 THEN GOTO P2A3END
  ELSE
    GOTO P2A3LUS
  END IF
P2A3END:
  GOSUB BLANKLINES
  RETURN

```

**D**

```

P2A4:
GOSUB BLANKLINES
LOCATE 17, 10: PRINT " Poort 2- D - Muziekje Moniek voor vioolkaart "
IF HM = 1 OR HM = 3 THEN
  LOCATE 18, 10: INPUT " speeltempo ? ( 10 ) = ", TMP
  ELSE
    TMP = 8 + (RND(1) * 10)
  END IF
  IF TMP < 1 THEN GOTO P2A4
  OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
  OUT DP, 32: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: OUT SP, P2
  FOR B = 0 TO 4
    FOR I = 0 TO 3
      OUT DP, 2 ^ I: OUT SP, P2 + 1: OUT SP, P2
      OUT DP, B: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, 50 \ TMP
      OUT DP, 32: OUT SP, P2 + 1: OUT SP, P2
    NEXT I
    OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
  FOR I = 0 TO 3
    OUT DP, 2 ^ I: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 25 \ TMP
  NEXT I
  OUT DP, 2 ^ (3 + B): OUT SP, P2 + 1: OUT SP, P2
  SOUND 20000, 100 \ TMP
NEXT B
  OUT SP, P4: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
  FOR I = 0 TO 3
    OUT DP, 2 ^ I: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 25 \ TMP
  NEXT I
  SOUND 20000, 50 \ TMP
  OUT DP, 2: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, 50 \ TMP
  OUT DP, 3: OUT SP, P4 + 1: OUT SP, P4
  OUT DP, 64: OUT SP, P2 + 1: OUT SP, P2
  FOR A = 1 TO 5

```

```

OUT DP, 8: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 250 \ TMP
OUT DP, A: OUT SP, P4 + 1: OUT SP, P4
NEXT A
  OUT DP, 128: OUT SP, P2 + 1: OUT SP, P2
FOR I = 0 TO 7
  OUT DP, 2 ^ I: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 25 \ TMP
NEXT I
  SOUND 20000, 250 \ TMP
OUT DP, 130: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 250 \ TMP
  OUT DP, 64: OUT SP, P4 + 1: OUT SP, P4
OUT DP, 68: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 250 \ TMP
  OUT DP, 4: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 250 \ TMP
FOR A = 0 TO 2
  FOR I = 0 TO 3
    OUT DP, 2 ^ I: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 25 \ TMP
  NEXT I
  OUT DP, A: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, 25 \ TMP
NEXT A
  OUT DP, 16: OUT SP, P2 + 1: OUT SP, P2
  SOUND 20000, 250 \ TMP
FOR I = 0 TO 10
  OUT DP, 1: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 10 \ TMP
NEXT I
  OUT DP, 255: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
  A$ = "RAES"
  OUT DP, 33: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: SOUND 20000, 125 \ TMP
  FOR I = 0 TO 10
    OUT DP, 8: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 10 \ TMP
  NEXT I
  SOUND 20000, 125 \ TMP
  FOR I = 0 TO 10
    OUT DP, 2: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 10 \ TMP
  NEXT I
  SOUND 20000, 500 \ TMP
FOR A = 0 TO 2
  OUT DP, 2 ^ (4 + A): OUT SP, P2 + 1: OUT SP, P2
  OUT DP, 128: OUT SP, P4 + 1: OUT SP, P4
  OUT DP, 1: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 25 \ TMP
  OUT DP, 2: OUT SP, P2 + 1: OUT SP, P2: SOUND 20000, 25 \ TMP
NEXT A
  FOR B = 1 TO 7
    OUT DP, 2 ^ B: OUT SP, P2 + 1: OUT SP, P2
    OUT DP, 2 ^ B: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, 10 \ TMP
  NEXT B
LOCATE 18, 10: PRINT "      Einde D-routine voor vioolkaart "
IF HM = 1 THEN
  GOSUB BLANKLINES
  GOSUB P2B
  END IF
K2END:
GOSUB BLANKLINES
RETURN

```

GOSUB BLANKLINES  
GOSUB P2B  
END IF



3

```

*****
MDOSC:                                'Poort 3 Multidimensionele oscillatorkaart
GOSUB BLANKLINES
IF HM = 1 OR HM = 3 THEN
LOCATE 15, 10: PRINT "POORT "; K; "                STOP met * "
LOCATE 16, 10: PRINT " Multidimensionele-Oscillator    ROUTINE K3R1"
LOCATE 17, 10: INPUT "Exponent? (2) ", C
LOCATE 18, 10: INPUT "Pause? (25) ", B
LOCATE 19, 10: INPUT "Hold ? (150) ", J
ELSE
C = 2: B = 10 + RND(1) * 15: J = 100 + RND(1) * 100
END IF

K3R1:
L = 0: A = 0
FOR I = 0 TO 7
A = A + 1: IF A > 220 THEN A = 0
OUT DP, I + A: OUT SP, P2 + 1: OUT SP, P2
OUT DP, I: OUT SP, P4 + 1: OUT SP, P4
SOUND 20000, RND(1) * (I + 2): B(I) = INT(2 * (RND(1) ^ C))
OUT DP, B(I): OUT SP, P4 + 1: OUT SP, P4
NEXT I
M = 0
FOR I = 0 TO 7: OUT DP, I: OUT SP, P2 + 1: OUT SP, P2: M = M + (2 ^ I) * B(I)
OUT DP, M: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, I
NEXT I
OUT SP, P3: OUT DP, M: OUT SP, P3 + 1: A$ = "RAES": OUT SP, P3
FOR I = 0 TO 7: LOCATE 20, 10 + (I * 4): PRINT B(I); " ";
OUT DP, I: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, I \ 2
NEXT I
P3HOLD:
HLD = INT(RND(1) * J) + 2: SOUND 20000, HLD
IF A > 32 AND RND(0) > RND(1) THEN
OUT SP, P5: OUT DP, 128: OUT SP, P5 + 1: A$ = "RAES": OUT SP, P5
SOUND 20000, 2
END IF
FOR I = 0 TO 7: L = L + B(I): NEXT I
IF L = 1 THEN GOTO K3R1
OUT SP, P3: OUT DP, 0: OUT SP, P3 + 1: A$ = "RAES": OUT SP, P3
T = INT(B * (RND(1)))
IF T > B \ 2 THEN
OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
A$ = "RAES": ' midi programm-change REX-50
OUT DP, 34: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
END IF
IF T <= B \ 2 THEN
OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
OUT DP, 33: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
END IF
LOCATE 21, 10: PRINT "Stilte op kaart 3 ="; T; " Units ";
SOUND 20000, T \ 2
FOR I = 0 TO T
OUT DP, I: OUT SP, P2 + 1: OUT SP, P2: SOUND*20000, RND(1) * (T \ 10)
NEXT I
SOUND 20000, T \ 2
OUT SP, P5: OUT DP, 0: OUT SP, P5 + 1: A$ = "RAES": OUT SP, P5
IF K$ = "F" OR HM = 2 THEN GOTO K3END
K$ = INKEY$: IF K$ = "*" OR HM = 2 THEN GOTO K3END ELSE GOTO K3R1
K3END:
GOSUB BLANKLINES
RETURN

```



4

```
DRUM: Poort 4 - elektromagneetdrumkaart
GOSUB BLANKLINES
LOCATE 15, 10: PRINT " POORT"; K; " Elektromagnetische drums "
KATST:
LOCATE 16, 10: PRINT " KIES A,B OF D ... of * to quit"
K$ = INKEY$: IF K$ = "" THEN GOTO KATST
IF K$ = "*" THEN GOTO K4END
IF K$ = "A" OR K$ = "a" THEN GOTO P4A
IF K$ = "B" OR K$ = "b" THEN GOTO P4B
IF K$ = "D" OR K$ = "d" THEN GOTO P4D ELSE GOTO KATST
```

A

```
P4A:
GOSUB BLANKLINES
LOCATE 15, 10: PRINT "Poort 4 - drumsnaarroutine A "
FOR B = 0 TO 255
OUT DP, B: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, 4
OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, 2
NEXT B
GOSUB BLANKLINES
RETURN
```

B

```
P4B:
GOSUB BLANKLINES
LOCATE 15, 10: PRINT "Poort 4 - drumsnaarroutine B "
IF HM = 1 OR HM = 3 THEN
LOCATE 16, 10: INPUT " DOOR ? ( 100 ) ", DUR
LOCATE 17, 10: INPUT " TEMPO ( > 12 ( groter cijfer is trager))24? ", T
LOCATE 18, 10: INPUT " HOLD-TIME ? ( > 1 ) ", H
IF DUR < 1 THEN DUR = 100
IF T < 12 THEN T = 24
IF H < 1 THEN H = 1
ELSE
DUR = 120: T = 12 + (RND(1) * 24): H = 2
END IF
FOR DC = 0 TO DUR
OUT DP, DUR: OUT SP, P2 + 1: OUT SP, P2
IF DC < DUR / 6 THEN
OUT DP, 1: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T
GOTO P4BLUS
END IF
IF DC >= DUR / 6 AND DC < DUR / 5 THEN
FOR E = 0 TO 1
OUT DP, 2 ^ E: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 2
NEXT E
GOTO P4BLUS
END IF
IF DC >= DUR / 5 AND DC < DUR / 4 THEN
FOR E = 1 TO 3
OUT DP, 2 ^ E: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 3
NEXT E
GOTO P4BLUS
END IF
IF DC >= DUR / 4 AND DC < DUR / 3 THEN
OUT SP, P1: OUT DP, 132: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
OUT DP, 36: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
FOR E = 2 TO 5
OUT DP, 2 ^ E: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 4
NEXT E
GOTO P4BLUS
END IF
```

```

IF DC >= DUR / 3 AND DC < (DUR * 2) / 5 THEN
  FOR E = 3 TO 7
    OUT DP, 2 ^ E: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 5
  NEXT E
  GOTO P4BLUS
      END IF
IF DC >= DUR * 2 / 5 AND DC < DUR / 2 THEN
  FOR E = 0 TO 5
    OUT DP, 2 ^ E: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 6
  NEXT E
  GOTO P4BLUS
      END IF
IF DC >= DUR / 2 AND DC < DUR * 3 / 5 THEN
  FOR E = 0 TO 6
    OUT DP, 2 ^ E: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 7
  NEXT E
  GOTO P4BLUS
      END IF
IF DC >= DUR * 3 / 5 AND DC < DUR * 2 / 3 THEN
  FOR E = 0 TO 7
    OUT DP, 2 ^ E: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 8
  NEXT E
  GOTO P4BLUS
      END IF
IF DC >= DUR * 2 / 3 AND DC < DUR * 3 / 4 THEN
  FOR E = 7 TO 0 STEP -1
    OUT DP, 2 ^ E: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 9
  NEXT E
  GOTO P4BLUS
      END IF
IF DC >= DUR * 3 / 4 AND DC < DUR * 4 / 5 THEN
  FOR E = 0 TO 7
    OUT DP, (2 ^ E) + 1: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 10
  NEXT E
  GOTO P4BLUS
      END IF
IF DC >= DUR * 4 / 5 AND DC < DUR * 5 / 6 THEN
  FOR E = 0 TO 7
    OUT DP, (2 ^ E) + 3: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 11
  NEXT E
  GOTO P4BLUS
      END IF
IF DC >= DUR * 5 / 6 THEN
  FOR E = 0 TO 7
    OUT DP, (2 ^ E) + 7: OUT SP, P4 + 1: OUT SP, P4: SOUND 20000, T \ 12
  NEXT E
      END IF
P4BLUS:
  NEXT DC
  OUT DE, 0
IF K$ = "F" OR HM > 1 THEN GOTO P4BEND
IF HM = 1 THEN GOTO STOCH:
P4BEND:
  GOSUB BLANKLINES
RETURN

```

**D** **P4D:**  
GOSUB BLANKLINES  
LOCATE 15, 10: PRINT " Poort Nr4 - Drumsnaartjeskaart Subroutine 0 "  
LOCATE 16, 10: PRINT " MANUBEL te bespelen met de numerieke 1-6 TOETSEN "

```

LOCATE 17, 10: PRINT " Stop via de *-toets ... ( terug naar hoofdmenu) "
P4DTK:
K$ = INKEY$: IF K$ = "" THEN GOTO P4DTK
IF K$ = "*" THEN GOTO K4END
FOR I = 49 TO 56
  IF ASC(K$) = I THEN
    OUT DP, 2 ^ (I - 49): OUT SP, P4 + 1: OUT SP, P4
  END IF
NEXT I
SOUND 20000, 1: OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4
GOSUB BLANKLINES

```

```
GOTO P4DTK
```

```
K4END:
```

```
GOSUB BLANKLINES
```

```
RETURN
```

```
*****
```

5

```
STRISPRI:
```

```
GOSUB BLANKLINES
```

```
LOCATE 15, 10: PRINT " Stringbass-kaart en veertjes "
```

```
LOCATE 16, 10: PRINT " POORT 5 "
```

```
P5MEN:
```

```
LOCATE 17, 10: PRINT " Kies A,B,C of D ... (* = STOP !!!) "
```

```
P5KTST:
```

```
K$ = INKEY$: IF K$ = "" THEN GOTO P5KTST
```

```
IF K$ = "*" THEN GOTO K5END
```

```
IF K$ = "A" OR K$ = "a" THEN GOTO P5A
```

```
IF K$ = "B" OR K$ = "b" THEN GOTO P5B
```

```
IF K$ = "C" OR K$ = "c" THEN GOTO P5C
```

```
IF K$ = "D" OR K$ = "d" THEN GOTO P5D ELSE GOTO P5MEN
```

A

```
P5A:
```

```
LOCATE 17, 10: PRINT " Manueel -speel via cijfers van het toetsenbord - * = STOP "
```

```
P5AKTST:
```

```
K$ = INKEY$: IF K$ = "" THEN GOTO P5AKTST
```

```
IF K$ = "*" AND HM = 1 THEN GOTO STRISPRI: ' HEX-concerto
```

```
IF K$ = "*" AND HM = 3 THEN GOTO K5END: ' HEX - manueel & debug
```

```
FOR I = 48 TO 56
```

```
IF ASC(K$) = I THEN
```

```
OUT SP, P5: OUT DP, 2 ^ (I - 49): OUT SP, P5 + 1: OUT SP, P5
```

```
END IF
```

```
NEXT I
```

```
GOTO P5AKTST
```

B

```
P5B:
```

```
GOSUB BLANKLINES
```

```
LOCATE 17, 10: PRINT " Kaart 5 - Poort 5 - SUBROUTINE B - semi-automatisch "
```

```
LOCATE 18, 10: PRINT " elke ingedrukte toets levert een andere combinatie op... "
```

```
LOCATE 19, 10: PRINT " Bespeel het klavier ritmisch... Stop met * ! "
```

```
P5BLUS:
```

```
A = 2 ^ (RND(1) * 7): OUT SP, P5: OUT DP, A: OUT SP, P5 + 1: OUT SP, P5
```

```
P5BKTST:
```

```
K$ = INKEY$: IF K$ = "" THEN GOTO P5BKTST
```

```
IF K$ <> "*" THEN GOTO P5BLUS
```

```
IF HM = 1 THEN
```

```
OUT DP, 0: OUT SP, P5 + 1: OUT SP, P5
```

```
GOTO P5C
```

```
END IF
```

```
GOSUB BLANKLINES
```

```
RETURN
```

P5C:

GOSUB BLANKLINES

LOCATE 17, 10: PRINT " Kaart 5 - Subroutine C - volautomatisch"

IF HM = 1 OR HM = 3 THEN

LOCATE 18, 10: INPUT " afspeeltempo ( hoger=sneller) 4-50 ? (25) ", TMP

ELSE

TMP = 15 + (RND(1) \* 20)

END IF

J = -1

FOR A = 0 TO 7: J = J + 1

FOR B = RPO(&H37 - J) TO 0 STEP -5

I = INT(RND(1) \* 7): C = RPO(&H80 + (7 \* A) - 1 + I)

OUT SP, P5: OUT DP, C: OUT SP, P5 + 1: OUT SP, P5

SOUND 20000, RPO(&HE7 - J) \ TMP: OUT DP, 0: OUT SP, P5 + 1: OUT SP, P5

NEXT B

NEXT A

OUT DP, 0

GOTO K5END

P5D:

GOSUB BLANKLINES

LOCATE 11, 10: PRINT " Glissandi voor kaart 5 - lowest 6 bits "

LOCATE 12, 10: PRINT " Druk de \* toets voor overgang naar P5B of root"

LOCATE 13, 10: PRINT " Speel via klavier met de toetsen 1,2,3,4,5,6,7"

LOCATE 14, 10: INPUT " Hoeveel glissandi wil je ? (20) "; Q

N = 0

P5DGLISS:

K\$ = INKEY\$

IF K\$ = "" THEN GOTO P5DGLISS

IF K\$ = "\*" THEN GOTO P5DEND

IF K\$ < "1" OR K\$ > "6" THEN GOTO P5DGLISS

IF K\$ = "1" THEN M = 1

IF K\$ = "2" THEN M = 2

IF K\$ = "3" THEN M = 4

IF K\$ = "5" THEN M = 8

IF K\$ = "6" THEN M = 16

IF K\$ = "7" THEN M = 32

QT = RND(1) \* 5: OUT SP, P5: N = N + 1

IF N = Q THEN GOTO P5DEND

IF QT > 4 THEN GOTO P5DGLISDOWN ELSE GOTO P5DGLISUP

P5DGLISDOWN:

FOR C = 0 TO RND(1) \* 500

OUT DP, M: OUT SP, P5 + 1: OUT SP, P5

OUT DP, 0: OUT SP, P5 + 1: OUT SP, P5

NEXT C

FOR C = 0 TO (RND(1) \* ((Q ^ 2) - (N ^ 2))) + 20

OUT DP, M: OUT SP, P5 + 1: OUT SP, P5: FOR J = 0 TO C: NEXT J

OUT DP, 0: OUT SP, P5 + 1: OUT SP, P5: FOR J = 0 TO C: NEXT J

NEXT C

P5DGLISUP:

FOR C = (RND(1) \* ((Q ^ 2) - (N ^ 2)) + 20) TO 0 STEP -1

OUT DP, M: OUT SP, P5 + 1: OUT SP, P5: FOR J = 0 TO C: NEXT J

OUT DP, 0: OUT SP, P5 + 1: OUT SP, P5: FOR J = 0 TO C: NEXT J

NEXT C

FOR C = 0 TO (100 + (RND(1) \* 500))

OUT DP, M: OUT SP, P5 + 1: OUT SP, P5

OUT DP, 0: OUT SP, P5 + 1: OUT SP, P5

NEXT C

GOTO P5DGLISS

P5DEND:

```
OUT DP, 0: OUT SP, P5
K$ = INKEY$: K$ = ""
IF HM = 1 THEN GOTO P5B
```

```
K$END:
```

```
GOSUB BLANKLINES
```

```
IF HM = 1 THEN GOTO STAPMOT: ' for concerto
```

```
RETURN
```

6

```
STAPMOT:
```

```
GOSUB BLANKLINES
```

```
LOCATE 15, 10: PRINT " Stappen-motoren kaart "
```

```
LOCATE 16, 10: PRINT " Poort 6 "
```

```
OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
```

```
OUT DP, 16: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
```

```
P6MEN:
```

```
IF HM = 3 THEN
```

```
LOCATE 17, 10: PRINT " Kies A, B, C, D, E, F ( F voor concert-versie) "
```

```
LOCATE 18, 10: PRINT " routine verlaten kan via de * toets ..."
```

```
P6KTST:
```

```
K$ = INKEY$
```

```
IF K$ = "" THEN GOTO P6KTST
```

```
IF K$ = "*" THEN GOTO K$END
```

```
IF K$ = "F" OR K$ = "f" THEN GOTO P6F: 'Koncertversie-keuze ex 6800
```

```
ELSE
```

```
IF HM = 1 THEN K$ = "F"
```

```
IF HM = 1 THEN GOTO P6F
```

```
IF HM = 2 THEN K$ = "A": A = &H0: B = 7: 'Automaat-versie
```

```
END IF
```

```
P6PARAM:
```

```
IF HM = 3 OR HM = 1 THEN
```

```
LOCATE 19, 10: INPUT " Tempo ? ( Hoe groter hoe sneller (0=QUIT) (10) )", J
```

```
IF J = 0 THEN GOTO P6FEND
```

```
LOCATE 20, 10: INPUT " Time ? ", D
```

```
IF K$ = "A" OR K$ = "a" THEN A = &H0: B = 23
```

```
IF K$ = "B" OR K$ = "b" THEN A = &H18: B = 23
```

```
IF K$ = "C" OR K$ = "c" THEN A = &H30: B = 15
```

```
IF K$ = "D" OR K$ = "d" THEN A = &H40: B = 23
```

```
IF K$ = "E" OR K$ = "e" THEN A = &H58: B = 39
```

```
ELSE
```

```
IF HM = 2 THEN J = 15 + (RND(1) * 10)
```

```
IF HM = 2 THEN D = 10 + (RND(1) * 10)
```

```
END IF
```

```
GOSUB P6DOIT
```

```
IF HM = 3 THEN GOTO STAPMOT ELSE GOTO K$END
```

A-E

```
P6F:
```

```
C = 0
```

```
GOSUB BLANKLINES
```

```
pitch shift programma naar REX 50
```

```
OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
```

```
A$ = "RAES": OUT DP, 16: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
```

```
LOCATE 21, 10: PRINT "Poort-nummer 6 - Stepping Motors - Routine F "
```

```
LOCATE 22, 10: INPUT "Afspeeltempo ? (hoger is sneller- 0=QUIT ) bvb.10 ", TMP
```

```
IF TMP = 0 THEN GOTO P6FEND
```

```
IF TMP < 1 THEN TMP = 1
```

```
TMP = TMP MOD 50
```

```
FOR J = 0 TO 255
```

```
OUT SP, P6
```

```
OUT DP, P6(&H100 + J): OUT SP, P6 + 1: A$ = "RAES": OUT SP, P6
```

```

TP = (RPO(&H16F - (C \ 8))) \ TMP: IF TP < 2 THEN TP = 1
SOUND 20000, TP: C = C + 1

NEXT J
LOCATE 23, 10: PRINT " Einde eerste motorendans - start viool pitch-shifts "
FOR I = 0 TO 31
    ,
    midi note on REX-50 - notenreeks staat in array
OUT SP, P1: OUT DP, 144: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
A$ = "RAES": OUT DP, RPI(&H90 + I): OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
A$ = "RAES": OUT DP, 127: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
FOR J = &HC8 TO &HCF
    ,
    motor-beweging
OUT SP, P6: OUT DP, RPI(J): A$ = "RAES": OUT SP, P6 + 1: A$ = "RAES": OUT SP, P6
TP = RPO(&H16F - I) \ TMP: IF TP < 2 THEN TP = 1
SOUND 20000, TP
NEXT J
    ,
    midi note off REX-50
OUT SP, P1: OUT DP, 128: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
A$ = "RAES": OUT DP, RPI(&H90 + I): OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
A$ = "RAES": OUT DP, 127: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
A$ = "RAES"
NEXT I
FOR I = &HB0 TO &HB7: ' midi info na te zien in eprom dump !
OUT SP, P1: OUT DP, RPI(I): OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
NEXT I
FOR I = 0 TO 11
    FOR A = &HB8 TO &HB8 + 5
        OUT SP, P1: OUT DP, RPI(A): OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
        SOUND 20000, 1
    NEXT A
    FOR A = &HE0 TO &HE7
        OUT SP, P6: A$ = "RAES"
        OUT DP, RPI(A): A$ = "RAES": OUT SP, P6 + 1: A$ = "RAES": OUT SP, P6
        SOUND 20000, (20 \ TMP)
        OUT DP, RND(1) * 16: OUT SP, P2 + 1: OUT SP, P2: A$ = "RAES"
    NEXT A
    SOUND 20000, RPO(&HF4 - I) \ TMP: 'reeks 1 lookup table
    FOR A = &HE0 TO &HE7
        OUT SP, P6: OUT DP, RPI(A): A$ = "RAES": OUT SP, P6 + 1: A$ = "RAES": OUT SP, P6: A$ = "RAES"
        OUT SP, P3: OUT DP, 2 ^ (RND(1) * 7): A$ = "RAES": OUT SP, P3 + 1: OUT SP, P3
        SOUND 20000, (20 \ TMP)
    NEXT A
    OUT SP, P3: OUT DP, 0: OUT SP, P3 + 1: OUT SP, P3: A$ = "RAES"
NEXT I
    OUT SP, P3: OUT DP, 33: OUT SP, P3 + 1: OUT SP, P3: OUT SP, P1
    OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
    OUT DP, 33: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
P6FEND:
GOSUB BLANKLINES
RETURN
P6DOIT:
FOR X = 0 TO D
    FOR I = A TO A + B
        OUT SP, P6: OUT DP, RPI(I): A$ = "RAES": OUT SP, P6 + 1: A$ = "RAES": OUT SP, P6
        SOUND 20000, 256 \ J
    NEXT I
NEXT X
NEXT X
RETURN
KGEND:
GOSUB BLANKLINES
RETURN

```

A ← E →

7

```
*****  
TALK7:  
'ROOT-COUNTER   spreekt MUSIC en dan een getal uit via p8 en p7  
IF CY = 0 THEN  
    OUT SP, P8:                                     'speak hex  
    FOR I = 0 TO 7: OUT DP, RP8(&H1E8 + I): OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8  
        WHILE INP(CP) \ 128 = 0: WEND  
    NEXT I  
    END IF  
OUT SP, P8  
IF CY > 0 THEN  
    IF CY = 2 THEN  
        'speak start  
        FOR I = 0 TO 7: OUT DP, RP8(&H208 + I): OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8  
            WHILE INP(CP) \ 128 = 0: WEND  
        NEXT I  
        END IF  
    IF CY > 3 THEN  
        FOR I = 0 TO 7  
            OUT DP, RP8(&H228 + I): OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8  
                WHILE INP(CP) \ 128 = 0: WEND  
        NEXT I  
        END IF  
        OUT DP, 0: OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8  
OUT SP, P7  
IF CY < 21 THEN  
    OUT DP, CY: OUT SP, P7 + 1: OUT SP, P7  
    GOTO K7END  
END IF  
IF CY > 20 AND CY < 30 THEN  
    OUT DP, 20: OUT SP, P7 + 1: OUT SP, P7  
    SOUND 20000, 2  
    OUT DP, CY - 20: OUT SP, P7 + 1: OUT SP, P7  
END IF  
IF CY >= 30 AND CY < 40 THEN  
    OUT DP, 21: OUT SP, P7 + 1: OUT SP, P7  
    SOUND 20000, 2  
    OUT DP, CY - 29: OUT SP, P7 + 1: OUT SP, P7  
END IF  
IF CY >= 40 AND CY < 50 THEN  
    OUT DP, 22: OUT SP, P7 + 1: OUT SP, P7  
    SOUND 20000, 2  
    OUT DP, CY - 28: OUT SP, P7 + 1: OUT SP, P7  
END IF  
IF CY >= 50 THEN  
    OUT DP, 23: OUT SP, P7 + 1: OUT SP, P7  
    SOUND 20000, 2  
    OUT DP, CY - 27: OUT SP, P7 + 1: OUT SP, P7  
END IF  
END IF  
K7END:  
RETURN
```

8

direct PBissen?

```

*****
TALK8:
  OUT SP, P8: 'P8
  GOSUB BLANKLINES
  LOCATE 15, 10: PRINT "POORT"; K: " Speech-card nr.8 - Allophones "
  LOCATE 16, 10: PRINT "          KIES A,B,C,D "

TK8:
  K$ = INKEY$: IF K$ = "" THEN GOTO TK8
  IF K$ = "*" THEN RETURN
  IF K$ = "A" OR K$ = "a" THEN GOTO K8R1
  IF K$ = "B" OR K$ = "b" THEN GOTO K8R2
  IF K$ = "C" OR K$ = "c" THEN GOTO K8R3
  IF K$ = "D" OR K$ = "d" THEN GOTO K8R4 ELSE GOTO TALK8:

```

A

```

K8R1:
  LOCATE 17, 10: PRINT " Kaart 8 spreekt het omgekeerde alfabet uit..."
  FOR I = &H22P TO &H110 STEP -1
    OUT DP, RP8(I): OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
    WHILE INP(CP) \ 128 = 0: WEND
  NEXT I
  OUT DP, 2: OUT SP, P8 + 1: OUT SP, P8
  GOSUB BLANKLINES
  RETURN

```

B

```

K8R2:
  LOCATE 17, 10: PRINT " A L L O P H O N E   S P E A C H   - P L A Y   "
                                     'midi - rex50 - programma nr. 35
  OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
  A$ = "RAES"
  OUT DP, 34: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
  A$ = "RAES"
  LOCATE 18, 10: PRINT "          * = STOP "

TKP8R2:
  K$ = INKEY$
  IF K$ = "" THEN GOTO TKP8R2
  IF K$ = "*" THEN GOTO K8R2END
  A = ASC(K$)
  IF A < 48 THEN GOTO TKP8R2
  OUT DP, ((A - 48) MOD 64): OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
  WHILE INP(CP) \ 128 = 0: WEND
  GOTO TKP8R2

K8R2END:
  OUT DP, 2: OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
  WHILE INP(CP) \ 128 = 0: WEND
  IF HM = 1 THEN GOTO K8R3 ELSE RETURN
                                     ' gaat na * direkt over naar automatisch alfabet ...

```

C

```

K8R3:
  LOCATE 17, 10: PRINT " Auto- Alfabet   Routine   C "
  LOCATE 18, 10: PRINT "          Stop met * "

TKP8R3:
  K$ = INKEY$
  IF K$ = "*" THEN GOTO K8R3END
  B = &H110 + (8 * INT((RND(1) * 50)))
  FOR I = 0 TO 7
    OUT DP, RP8(B + I): OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
    WHILE INP(CP) \ 128 = 0: WEND
  NEXT I
  OUT DP, 1: OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
  GOTO TKP8R3

```



KSR3END:

```

IF HM = 1 THEN
    K$ = "": GOTO KSR4
ELSE
    RETURN
END IF

```

KSR4:

```

LOCATE 17, 10: PRINT "      Talking Keyboard"
LOCATE 18, 10: PRINT "      letters en cijfers worden uitgesproken"
LOCATE 19, 10: PRINT "      hoofdletters zijn gehele woorden"
LOCATE 20, 10: PRINT "      Stop met *
OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
A$ = "RAES"
OUT DP, 35: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1

```

Midi

OUT SP, P1

P1 - P8 = 3 bit changes!

TKP8R4:

```

K$ = INKEY$
IF K$ = "*" THEN GOTO KSR3END
IF K$ = "" THEN GOTO TKP8R4
IF K$ >= "0" AND K$ <= "9" THEN
    OUT DP, ASC(K$) - 48: OUT SP, P7 + 1: OUT SP, P7
    SOUND 20000, 1
    GOTO TKP8R4

```

END IF

```

IF K$ >= "a" AND K$ <= "z" THEN
    B = &H110 + (8 * (ASC(K$) - 97)):
    GOTO KSR4LUS

```

'lower case

END IF

```

IF K$ >= "A" AND K$ <= "_" THEN
    IF K$ = "H" THEN AD = &H1E8: S = 7: ' hex
    IF K$ = "Y" THEN AD = &H1F0: S = 7: ' yes
    IF K$ = "N" THEN AD = &H1F8: S = 7: ' no
    IF K$ = "Z" THEN AD = &H200: S = 7: ' stop
    IF K$ = "A" THEN AD = &H208: S = 7: ' start
    IF K$ = "K" THEN AD = &H210: S = 7: ' switch
    IF K$ = "F" THEN AD = &H218: S = 7: ' freeze
    IF K$ = "L" THEN AD = &H220: S = 7: ' logos
    IF K$ = "M" THEN AD = &H228: S = 7: ' music
    IF K$ = "B" THEN AD = &H230: S = 15: 'emotional error
    IF K$ = "E" THEN AD = &H238: S = 7: ' error
    IF K$ = "C" THEN AD = &H240: S = 15: 'engagement
    IF K$ = "S" THEN AD = &H250: S = 12: 'systems
    IF K$ = "D" THEN AD = &H260: S = 8: ' robots
    IF K$ = "G" THEN AD = &H1F0: S = 7: ' genius
    IF K$ = "I" THEN AD = &H2A8: S = 7: ' little
    IF K$ = "J" THEN AD = &H238: S = 7: ' memory
    IF K$ = "O" THEN AD = &H2A0: S = 4: ' key
    IF K$ = "P" THEN AD = &H2B0: S = 7: ' sweat
    IF K$ = "Q" THEN AD = &H2C0: S = 12: 'sincerely
    IF K$ = "R" THEN AD = &H2B8: S = 7: ' ready
    IF K$ = "T" THEN AD = &H268: S = 15: 'time - talking
    IF K$ = "U" THEN AD = &H2E0: S = 15: 'sensitive
    IF K$ = "V" THEN AD = &H280: S = 7: ' speak
    IF K$ = "W" THEN AD = &H278: S = 7: ' whalers
    IF K$ = "X" THEN AD = &H288: S = 7: ' pinning
    IF K$ = "I" THEN AD = &H2F0: S = 15: 'fast - fuck
    IF K$ = "\" THEN AD = &H2D0: S = 7: ' score
    IF K$ = "J" THEN AD = &H2D8: S = 7: ' plus
    IF K$ = "" THEN AD = &H2F8: S = 7: ' fuck
    IF K$ = "_" THEN AD = &H270: S = 7: ' talking

```

```
GOSUB TEXT
GOTO TKP8R4
ELSE
GOTO TKP8R4
END IF
K8R4LUS:
S = 7: AD = B: GOSUB TEXT
GOTO TKP8R4

K8END:
GOSUB BLANKLINES
RETURN
```

9

TUT:

GOSUB BLANKLINES

LOCATE 15, 10: PRINT "

T U T T I - T E S T "

LOCATE 16, 10: PRINT "

F I N A L E "

FOR B = 0 TO 255

OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A\$ = "RAES": OUT SP, P1

A\$ = "RAES": OUT DP, 58 + (B \ 16): OUT SP, P1 + 1: A\$ = "RAES": OUT SP, P1: A\$ = "RAES"

OUT SP, P2: OUT DP, B: OUT SP, P2 + 1: OUT SP, P2

OUT SP, P3: OUT DP, B: OUT SP, P3 + 1: OUT SP, P3: A\$ = "RAES"

OUT SP, P4: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4: A\$ = "RAES"

OUT SP, P5: OUT DP, B: OUT SP, P5 + 1: OUT SP, P5: A\$ = "RAES"

OUT SP, P6: A\$ = "RAES": OUT DP, (RPG(&H100 + B)): A\$ = "RAES": OUT SP, P6 + 1: A\$ = "RAES": OUT SP, P6

SOUND 20000, 2

OUT SP, P7: OUT DP, B \ 8: OUT SP, P7 + 1: OUT SP, P7: A\$ = "RAES"

OUT SP, P8: OUT DP, B \ 4: OUT SP, P8 + 1: OUT SP, P8: A\$ = "RAES"

SOUND 20000, 1

FOR A = P8 TO P2 STEP -1

IF A = P6 OR A = P6 + 1 THEN

OUT SP, P6: A\$ = "RAES": OUT DP, (RPG(&H100 + B)): A\$ = "RAES": OUT SP, A: A\$ = "RAES"

ELSE

OUT DP, 0: A\$ = "RAES": OUT SP, A: A\$ = "RAES"

END IF

NEXT A

NEXT B

SOUND 20000, 18

FOR I = 0 TO 7: ' spreekt stop ?

OUT SP, P8: OUT DP, RPG(&H200 + I): OUT SP, P8 + 1: A\$ = "RAES": OUT SP, P8

WHILE INP(CP) \ 128 = 0: WEND

NEXT I

SOUND 20000, 18

'Motor laat snaren los

OUT SP, P6: OUT DP, 0: OUT SP, P6 + 1: OUT DP, 0: A\$ = "RAES": OUT SP, P6

GOSUB ALLOFF

IF HM = 1 THEN

OUT DP, 255

LOCATE 20, 10: PRINT START\$, " Begintijdstip van deze HEX-uitvoering"

LOCATE 21, 10: PRINT TIME\$, " Einde-tijdstip van deze HEX-uitvoering"

LOCATE 22, 10: PRINT " a p p l a u s ??? "

LOCATE 25, 10: PRINT " HEX - 0689 composed by Godfried-Willem RAES - Logos-Duo performance"

END

ELSE

OUT DP, 255

RETURN

END IF

A

BUILD:

```

GOSUB BLANKLINES
LOCATE 15, 10: PRINT " Subroutine &H A - Kombinatie 2 - opbouw kaarten "
OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
OUT DP, 56: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
LOCATE 16, 10: OUT SP, P4: PRINT " Poort 4 - solo voor drum-snaartjes "
FOR I = 0 TO 12
  FOR J = 0 TO 7
    SOUND 20000, 2: OUT DP, 2 ^ J: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, RND(1) * 255: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4
  NEXT J
NEXT I
LOCATE 16, 10: PRINT " 2 KAARTEN: drumsnaartjes en viool "
FOR I = 0 TO 4
  FOR J = 0 TO 7
    A = 2 ^ I: B = (2 ^ J) + A: OUT SP, P4
    SOUND 20000, 2: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
    A = B \ 16: OUT SP, P2: OUT DP, A: OUT SP, P2 + 1: OUT SP, P2
  NEXT J
NEXT I
LOCATE 16, 10: PRINT " 2 KAARTEN: P2A + P2B en drumsnaartjes P4 "
FOR I = 5 TO 7
  FOR J = 0 TO 7
    A = 2 ^ I: B = (2 ^ J) + A: OUT SP, P4
    SOUND 20000, 2: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4
    OUT SP, P2 + 1: OUT SP, P2: OUT SP, P4
    SOUND 20000, 2: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
    SOUND 20000, 2: OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4
    OUT SP, P2: OUT DP, B: OUT SP, P2 + 1: OUT SP, P2
  NEXT J
NEXT I
LOCATE 16, 10: PRINT " 3 KAARTEN: viool - speechkaart 8 en drumsnaartjes "
SOUND 20000, 3: OUT SP, P3: OUT DP, 0: OUT SP, P3 + 1: OUT SP, P3
FOR I = 0 TO 4
  FOR J = 4 TO 8
    FOR A = 0 TO 7: B = ((2 ^ I) + (2 ^ J) + (2 ^ A)) MOD 256
      OUT SP, P4: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4
      OUT SP, P2: OUT DP, B \ 16: OUT SP, P2 + 1: OUT SP, P2: A$ = "RAES"
      IF I > 2 THEN
        OUT DP, ((2 ^ (J - 1)) + (2 ^ I)) MOD 256: OUT SP, P2 + 1: OUT SP, P2: A$ = "RAES"
      END IF
      OUT SP, P8: OUT DP, B \ 4: OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8: CC = 0
    NEXT A
  NEXT J
  WHILE INP(CP) \ 128 = 0: CC = CC + 1: WEND
  FOR TP = 0 TO (200 - (CC MOD 200)): NEXT TP
  TP = 0: CC = 0: SOUND 20000, 3
NEXT I
NEXT I
LOCATE 16, 10: PRINT " 4 KAARTEN: drums + viool + speech 8 + md-osc "
FOR B = 255 TO 0 STEP -1

```

```

OUT SP, P4: OUT DP, B: OUT SP, P4 + 1: OUT SP, P4: OUT SP, P2
OUT DP, B: OUT SP, P2 + 1: OUT SP, P2: OUT SP, P8
OUT DP, B \ 4: OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8: CC = 0
    WHILE INP(CP) \ 128 = 0: CC = CC + 1: WEND
    FOR TP = 0 TO (200 - (CC MOD 200)): NEXT TP
OUT SP, P3: OUT DP, 2 ^ (B MOD 8): A$ = "RAES": OUT SP, P3 + 1: A$ = "RAES": OUT SP, P3: OUT SP, P4
SOUND 20000, 1
OUT DP, 0: A$ = "RAES": OUT SP, P4 + 1: OUT SP, P4: OUT SP, P3: A$ = "RAES"
SOUND 20000, 1
OUT DP, 0: OUT SP, P3 + 1: A$ = "RAES": OUT SP, P3: A$ = "RAES"
NEXT B
X = 0
    OUT SP, P1: OUT DP, 192: A$ = "RAES": OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
    OUT DP, 58: A$ = "RAES": OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1

LOCATE 16, 10: PRINT "                van 5 naar 6 KAARTEN                "
'in geval Hex automatisch loopt geldt volgende zin:
IF K$ = "F" OR HM = 2 THEN RETURN

LOCATE 17, 10: PRINT "    Stop door eender welke toets in de drukken    "
KILOOP:
    X = X + 1: OUT SP, P8: OUT DP, 0: OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
    WHILE INP(CP) \ 128 = 0: WEND
    SOUND 20000, 2: OUT SP, P2: OUT DP, (RND(1) * 127) * 2: OUT SP, P2 + 1: OUT SP, P2: OUT SP, P3
    OUT DP, RND(1) * 255: OUT SP, P3 + 1: A$ = "RAES": OUT SP, P3: OUT SP, P4: A$ = "RAES"
    OUT DP, RND(1) * 255: OUT SP, P4 + 1: OUT SP, P4: OUT SP, P7
    OUT DP, RND(1) * 32: OUT SP, P7 + 1: A$ = "RAES": OUT SP, P8
    OUT DP, RND(1) * 63: OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
IF X > 50 THEN
    OUT SP, P5: OUT DP, RND(1) * 255: OUT SP, P5 + 1: A$ = "RAES": OUT SP, P5
END IF
IF X = 32 THEN
    OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
    OUT DP, 57: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
END IF
K$ = INKEY$: IF K$ = "" THEN GOTO KILOOP
GOSUB BLANKLINES
RETURN

```

B

RAES:

```
*****
GOSUB BLANKLINES
LOCATE 15, 10: PRINT " Input routine Interaktief ! "
'hier komt een routine die van beide input poorten gebruikt maakt
LEES:
OUT DP, 255: A$ = "RAES"
OUT SP, 168: 'set bit 7 and 5 to enable port &H378 as input port
  I = 0
  WHILE ((NOT (INP(CP))) AND 8) AND I = 0: '1 als strobe op pin 15, bit 3, 0 is.
    B(1) = INP(DP): I = I + 1
  WEND
  I = 0
  WHILE ((NOT (INP(CP))) AND 16) AND I = 0: ' test strobe op pin 13, bit 4
    B(2) = INP(DP): I = I + 1
  WEND
IF B(1) < 255 THEN
LOCATE 10, 15: PRINT "Input byte van poort 1="; B(1)
  END IF
IF B(2) < 255 THEN
LOCATE 11, 15: PRINT "Input byte van poort 2="; B(2)
  END IF
OUT SP, P1: 'reset bit 7 to make port &H378 output again
IF B(1) < 255 AND B(1) > 0 THEN
  FOR I = 0 TO 100
    OUT DP, B(1): OUT SP, P4 + 1: OUT SP, P4
    FOR J = 0 TO L: NEXT J
    OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4
  NEXT I
END IF
IF B(2) < 255 THEN
  OUT DP, B(2): OUT SP, P5 + 1: OUT SP, P5
END IF
WHILE INKEY$ = ""
  GOTO LEES:
WEND
GOSUB BLANKLINES
RETURN
```

STOCH

```
*****
GOSUB BLANKLINES
LOCATE 15, 10: PRINT " Stochastisch Tutti "
OUT SP, P1
OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
OUT DP, 37: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
LOCATE 16, 10: INPUT "START-EXPONENT (1) ? ", A
LOCATE 17, 10: INPUT "STOP-EXPONENT (4) ? ", B
LOCATE 18, 10: PRINT TIMES
IF A = B THEN GOTO STOCHEND
' beide getallen gelijk maken om de routine over te schrikkelen
IF A <= 0 THEN A = 1: IF B <= 0 THEN B = 3
IF A > 6 OR B > 7 THEN GOTO STOCH:
D = 0
STOCHSTART:
FOR Z! = B TO A STEP -.05
FOR I = 0 TO 7
A(I) = INT(2 * (RND(CY) ^ Z!))
LOCATE 20, 10 + (I * 4): PRINT A(I);
NEXT I
M = 0
FOR I = 0 TO 7
M = INT(M + (2 ^ I) * A(I)): D = D + 1
IF I = 1 AND A(I) = 1 THEN
FOR C = 0 TO M + 1
SOUND 20000, RND(1) * 2: OUT SP, P2: OUT DP, (RND(M) * 255): OUT SP, P2 + 1: OUT SP, P2
SOUND 20000, RND(1) * 2: OUT DP, 0: OUT SP, P2 + 1: OUT SP, P2
NEXT C
END IF
IF I = 2 AND A(I) = 1 THEN
OUT SP, P3
FOR C = ((M + 1) * (RND(1) * 40)) TO 0 STEP -1
OUT DP, 0: OUT SP, P3 + 1: OUT SP, P3
FOR Y = 0 TO C: NEXT Y
OUT DP, M: OUT SP, P3 + 1: OUT SP, P3
FOR Y = 0 TO C: NEXT Y
NEXT C
OUT SP, P3: OUT DP, M: OUT SP, P3 + 1: A$ = "RAES": OUT SP, P3
SOUND 20000, RND(1) * 50
END IF
IF I = 2 AND A(I) = 0 THEN
QT = INT(RND(D) * 10)
IF QT < 3 THEN
OUT SP, P3: OUT DP, 0: OUT SP, P3 + 1: A$ = "RAES": OUT SP, P3
SOUND 20000, QT
GOTO STOCHML: ' 12960
END IF
END IF
IF I = 3 AND A(I) = 1 THEN
QT = RND(1) * 10: OUT SP, P4
IF QT > 5 THEN GOTO GLISDOWN ELSE GOTO GLISUP
GLISDOWN:
FOR C = 0 TO ((M + 1) * (RND(1) * 30))
OUT DP, M: OUT SP, P4 + 1: OUT SP, P4: FOR J = 0 TO C: NEXT J
OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4: FOR J = 0 TO C: NEXT J
NEXT C
GLISUP:
FOR C = ((M + 1) * (RND(1) * 30)) TO 0 STEP -1
OUT DP, M: OUT SP, P4 + 1: OUT SP, P4: FOR J = 0 TO C: NEXT J
OUT DP, 0: OUT SP, P4 + 1: OUT SP, P4: FOR J = 0 TO C: NEXT J
```

```

                NEXT C
END IF
IF I = 4 AND A(I) = 1 THEN
    OUT SP, P5: OUT DP, M: OUT SP, P5 + 1: OUT SP, P5
    SOUND 20000, 6: OUT DP, 0: OUT SP, P5 + 1: OUT SP, P5
    END IF
IF I = 5 AND A(I) = 1 AND ((RND(1) * 10) < 5) THEN
    OUT SP, P6: OUT DP, RP6(&H100 + M): OUT SP, P6 + 1: A$ = "RAES": OUT SP, P6
    SOUND 20000, 3
    END IF
IF I = 6 AND A(I) = 1 THEN
    OUT SP, P7: OUT DP, RND(CY + M) * 31: OUT SP, P7 + 1: OUT SP, P7
    SOUND 20000, RND(1) * 4
    END IF
IF I = 7 AND A(I) = 1 THEN
    B = &H110 + (8 * INT(RND(1) * 61)): OUT SP, P8
    FOR J = 0 TO 7
        OUT DP, RP8(B + J): OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
        WHILE INP(CP) \ 128 = 0: WEND
    NEXT J
    OUT DP, 2: OUT SP, P8 + 1: OUT SP, P8
    WHILE INP(CP) \ 128 = 0: WEND
    END IF
STOCHMI:
IF I = 0 AND A(I) = 1 THEN
    OUT SP, P1: OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1: A$ = "RAES"
    OUT DP, (37 + INT(RND(1) * 20)): OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
    END IF
NEXT I
NEXT Z!
STOCHEND:
GOSUB BLANKLINES
    IF HM = 1 THEN GOTO TELTUTTE: '
RETURN

```

koncertversie



D

TELTUTTI:

```

Teltutti 2 - subroutine &H C *****
GOSUB BLANKLINES
  IF K$ = "F" THEN RETURN
  OUT SP, P1
  OUT DP, 192: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
  A$ = "RAES": N = -1
  OUT DP, 56: OUT SP, P1 + 1: A$ = "RAES": OUT SP, P1
  LOCATE 15, 10: PRINT " T E L T U T T I    2    ";

```

CBEGIN:

```

  N = N + 1
  SOUND 20000, 2
  FOR E = 7 TO 0 STEP -1
    FOR P = P2 TO P8 STEP 2
      BY = (2 ^ E) + N
      IF P = P6 THEN
        OUT SP, P6
        FOR I = 0 TO 7
          OUT SP, P6
          OUT DP, RP6(&H50 + I): OUT SP, P6 + 1: A$ = "RAES": OUT SP, P6
          SOUND 20000, 1
        NEXT I
        END IF
      IF P = P7 THEN BY = BY MOD 33
      IF P = P8 THEN BY = BY MOD 64
    IF P <> P6 THEN
      OUT SP, P
      OUT DP, BY: OUT SP, P + 1: A$ = "RAES": OUT SP, P
      END IF
      IF P = P8 THEN
        WHILE INP(CP) \ 128 = 0: WEND
        END IF
        T = 5 - N: IF T < 1 THEN T = 1:
        SOUND 20000, T
      NEXT P

```

"T was 0

NEXT E

REPEAT:

```

  LOCATE 16, 10: PRINT "

```

Repeat? Y/N";

TELTST:

```

  K$ = INKEY$
  IF K$ = "" THEN GOTO TELTST
  IF K$ = "*" THEN RETURN
  IF K$ = "Y" OR K$ = "y" THEN GOTO CBEGIN
  IF K$ = "N" OR K$ = "n" THEN
    OUT DP, 0: OUT SP, P8 + 1: A$ = "RAES": OUT SP, P8
    IF HM = 1 THEN GOTO BUILD:
    IF HM > 1 THEN RETURN ELSE GOTO REPEAT:
  END IF

```

\*\*\*\*\*

COUNT:

'dummy routine E voor counter

```

  LOCATE 15, 10: PRINT " COUNT-DOWN DUMMY ROUTINE "

```

```

  SOUND 20000, 2

```

```

  GOSUB BLANKLINES:

```

RETURN

\*\*\*\*\*

E

F

# ROBOT

ROBOT

```
GOSUB BLANKLINES
LOCATE 15, 10: PRINT "  H E X  A U T O M A A T
CY = CY + 1: GOSUB P2A4
FOR J = 0 TO 10
  GOSUB MDOSC
NEXT J
GOSUB P4B
GOSUB TALK7
GOSUB P5C
GOSUB BULLD
A = 1: B = (RND(1) * 4) + 1: GOSUB STOCHSTART
  OUT DP, 0: OUT SP, P3 + 1: OUT SP, P3
  OUT DP, 0: OUT SP, P6 + 1: OUT SP, P6
  OUT DP, 0: OUT SP, P5 + 1: OUT SP, P5
  SOUND 20000, 4
GOSUB K8R1
GOSUB STAPMOT
GOSUB P4A
C = 2 ^ (7 * RND(1))
  OUT DP, C: OUT SP, P3 + 1: OUT SP, P3
  OUT DP, C: OUT SP, P5 + 1: OUT SP, P5
  OUT DP, C: OUT SP, P4 + 1: OUT SP, P4
  SOUND 20000, RND(1) * 60
  OUT DP, 0: OUT SP, P3 + 1: OUT SP, P3
J = 50: D = 20: A = &H8: B = 7: GOSUB P6DOT
IF HM = 2 THEN GOTO ROBOT
RETURN
```

**GENGLIS:**

```

GOSUB BLANKLINES
LOCATE 10, 10: PRINT " Algemene glissando routine met alle glisskaarten "
LOCATE 11, 10: PRINT "  TOETSSEN  A,B,C,D,E,F,G,H  zijn DRUMKAART  "
LOCATE 12, 10: PRINT "                    L,J,K,L,M,N,O,P  zijn MD-OSCILLATOR  "
LOCATE 13, 10: PRINT "                    Q,R,S,T            is kaart P2B      "
LOCATE 14, 10: PRINT "                    U,V,W,X,Y,Z      is kaart P5        "
LOCATE 15, 10: PRINT "                    *                  is RETURN to ROOT-MENU"
LOCATE 16, 10: PRINT "  UPPERCASE = stijgend  LOWERCASE = dalend  "
LOCATE 18, 10: INPUT "  Aantal glissandi ? (32) "; Q

```

```

N = 0
' P2 - bits 4-7 - P3 - all bits  P4- all bits - P5 bits 0-5
' totaal aantal bits 26 - controle door keyboard letters
GGKTST:
K$ = INKEY$
IF K$ = "" THEN GOTO GGKTST
IF K$ = "*" THEN RETURN
A = ASC(K$)
IF A > 96 AND A < 123 THEN
    A = A - 32
    QT = 1
ELSE
    QT = 2
END IF

```

```

IF (A < 65) OR (A > 90) THEN GOTO GGKTST ELSE M = A - 65: ' M is 0 - 25
IF M < 8 THEN P = P4: B = 2 ^ M: ' 1-128
IF (M > 7) AND (M < 16) THEN P = P3: B = 2 ^ (M - 8): ' 1-128
IF M > 15 AND M < 20 THEN P = P2: B = (2 ^ (M - 16)) * 16: ' 16-128
IF M > 19 AND M < 26 THEN P = P5: B = (2 ^ (M - 20)): ' 1-32
OUT SP, P: N = N + 1
IF N = Q THEN GOTO GGEND
IF QT = 1 THEN GOTO GGDOWN ELSE GOTO GGUP

```

```

GGDOWN:
FOR C = 0 TO ((RPO(&H150 + N)) * 20): ' look-up table just intonation
    OUT DP, B: OUT SP, P + 1: OUT SP, P: A$ = "D"
    OUT DP, 0: OUT SP, P + 1: OUT SP, P: A$ = "D"
NEXT C
FOR C = 0 TO (RND(1) * ((Q ^ 2) - (N ^ 2))) + 40
    OUT DP, B: OUT SP, P + 1: OUT SP, P: FOR J = 0 TO C: NEXT J
    OUT DP, 0: OUT SP, P + 1: OUT SP, P: FOR J = 0 TO C: NEXT J
NEXT C

```

*konta maken. 8 nu: 10*

```

GOTO GGKTST
GGUP:
FOR C = (RND(1) * ((Q ^ 2) - (N ^ 2))) + 40 TO 0 STEP -1
    OUT DP, B: OUT SP, P + 1: OUT SP, P: FOR J = 0 TO C: NEXT J
    OUT DP, 0: OUT SP, P + 1: OUT SP, P: FOR J = 0 TO C: NEXT J
NEXT C
FOR C = 0 TO ((RPO(&H150 + N)) * 20)
    OUT DP, B: OUT SP, P + 1: OUT SP, P: A$ = "U"
    OUT DP, 0: OUT SP, P + 1: OUT SP, P: A$ = "U"
NEXT C

```

*8 nu: 10*

```

GOTO GGKTST
GGEND:
OUT DP, 0: OUT SP, P
GOSUB BLANKLINES
RETURN

```

\*\*\*\*\* ALGEMENE SUBROUTINES \*\*\*\*\*

BLANKLINES:  
FOR BLANK = 7 TO 24  
LOCATE BLANK, 1: PRINT "  
NEXT BLANK  
RETURN

BLANKALL:  
FOR BLANK = 1 TO 24  
LOCATE BLANK, 1: PRINT "  
NEXT BLANK  
RETURN

ALLOFF:  
OUT DP, 0  
OUT SP, P2 + 1: A\$ = "RAES": OUT SP, P2: A\$ = "RAES": OUT DP, 0  
OUT SP, P3 + 1: A\$ = "RAES": OUT SP, P3: A\$ = "RAES": OUT DP, 0  
OUT SP, P4 + 1: A\$ = "RAES": OUT SP, P4: A\$ = "RAES": OUT DP, 0  
OUT SP, P5 + 1: A\$ = "RAES": OUT SP, P5: A\$ = "RAES": OUT DP, 0  
OUT SP, P6 + 1: A\$ = "RAES": OUT SP, P6: A\$ = "RAES": OUT DP, 0  
OUT SP, P8 + 1: A\$ = "RAES": OUT SP, P8: A\$ = "RAES": OUT DP, 0  
RETURN

P8

TEXT:  
OUT SP, P8: IF S = 0 THEN S = 7  
FOR I = 0 TO S  
OUT DP, RP0(AD + I): OUT SP, P8 + 1: A\$ = "RAES": OUT SP, P8  
WHILE INP(CP) \ 128 = 0: WEND  
NEXT I  
RETURN

P2E

P2B:  
FOR I = 10 TO 23  
LOCATE I, 10: PRINT "  
NEXT I  
LOCATE 10, 10: PRINT " Solo-routine voor kaart P2B  
LOCATE 11, 10: INPUT " DUUR ? (12) ", TMP  
IF TMP < 0 THEN TMP = 0  
FOR J = TMP TO 0 STEP -1  
FOR I = 4 TO 7  
E = 2 ^ I: OUT DP, E: OUT SP, P2 + 1: OUT SP, P2  
SOUND 20000, J  
NEXT I  
NEXT J  
LOCATE 11, 10: PRINT " Glissando routine voor kaart P2 B  
LOCATE 12, 10: PRINT " DRUK DE \* TOETS VOOR RETURN NAAR ROOT-MENU  
LOCATE 13, 10: PRINT " Speel via klavier met de toetsen 1,2,3,4  
LOCATE 14, 10: INPUT " Hoeveel glissandi wil je ? (25) "; Q  
N = 0  
P2BGLISS:  
K\$ = INKEY\$  
IF K\$ = "" THEN GOTO P2BGLISS  
IF K\$ = "\*" THEN GOTO P2BEEND  
IF K\$ < "1" OR K\$ > "4" THEN GOTO P2BGLISS  
IF K\$ = "1" THEN M = 16  
IF K\$ = "2" THEN M = 32  
IF K\$ = "3" THEN M = 64  
IF K\$ = "4" THEN M = 128  
QT = RND(1) \* 10: OUT SP, P2: N = N + 1  
IF N = Q THEN GOTO P2BEEND

```

IF QT > 5 THEN GOTO P2BGLISDOWN ELSE GOTO P2BGLISUP
P2BGLISDOWN:
FOR C = 0 TO (RND(1) * ((Q ^ 2) - (N ^ 2))) + 20
  OUT DP, M: OUT SP, P2 + 1: OUT SP, P2: FOR J = 0 TO C: NEXT J
  OUT DP, 0: OUT SP, P2 + 1: OUT SP, P2: FOR J = 0 TO C: NEXT J
NEXT C
P2BGLISUP:
FOR C = (RND(1) * ((Q ^ 2) - (N ^ 2)) + 20) TO 0 STEP -1
  OUT DP, M: OUT SP, P2 + 1: OUT SP, P2: FOR J = 0 TO C: NEXT J
  OUT DP, 0: OUT SP, P2 + 1: OUT SP, P2: FOR J = 0 TO C: NEXT J
NEXT C
GOTO P2BGLISS
P2BEEND:
OUT DP, 0: OUT SP, P2 + 1: OUT SP, P2
K$ = INKEY$: K$ = "": OUT SP, P5: ' empty buffer
RETURN

```

# REX 50 data

```
'data &H00
DATA &H81, &H39, &H23, &H02, &H8B, &H09, &H84, &H0F, &H39, &HC4, &H0F, &HCA, &H30, &HC1, &H39, &H23 : 'HX20 chip
DATA &H02, &HCB, &H07, &H39, &H96, &H11, &H2A, &HFC, &H96, &H12, &H39, &H36, &H96, &H11, &H85, &H20 : 'HX26 Chip
DATA &H27, &HFA, &HD7, &H13, &H32, &H39, &H37, &H54, &H54, &H54, &H54, &H8D, &HDC, &H8D, &HEC, &H33 : 'HX20 Chip
DATA &HBD, &HE0, &H09, &HBD, &HE0, &H1B, &H39, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00 : 'HX20 Chip
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF
'&H80 - HEADER
DATA &H4D, &H49, &H44, &H49, &H2D, &H44, &H41, &H54, &H41, &H2D, &H52, &H45, &H58, &H35, &H30, &H3D
'&H90 - hier beginnen de eigenlijke midi-codes voor de REX-50
DATA &H3C, &H3D, &H3B, &H3E, &H3A, &H3F, &H39, &H40, &H38, &H41, &H37, &H42, &H36, &H43, &H35, &H45
'&HA0
DATA &H34, &H46, &H33, &H47, &H32, &H48, &H31, &H49, &H30, &H4A, &H2F, &H47, &H2F, &H47, &H2F, &H47, &H2F, &H47
'&HE0
DATA &HC0, &H10, &H90, &H3C, &H3C, &H80, &H3C, &H3C, &H90, &H2F, &H3C, &H80, &H2F, &H3C, &H90, &H32
'&HC0
DATA &H3C, &H80, &H32, &H3C, &H90, &H35, &H3C, &H80, &H35, &H3C, &H90, &H38, &H3C, &H80, &H36, &H3C
'&HD0
DATA &H90, &H3B, &H3C, &H80, &H3E, &H3C, &H90, &H3E, &H3C, &H80, &H3E, &H3C, &H90, &H41, &H3C, &H30
'&HE0
DATA &H41, &H3C, &H90, &H44, &H3C, &H80, &H44, &H3C, &H90, &H47, &H3C, &H80, &H47, &H3C, &H90, &H46
'&HF0
DATA &H3C, &H80, &H46, &H3C, &H90, &H43, &H3C, &H80, &H43, &H3C, &H90, &H40, &H3C, &H80, &H40, &H3C
```

'Speech Card data 4000:1500

'oud HX20 eepromadresgebied vanaf &H7500 tot &H77FF

'&H00

DATA &H44, &H41, &H54, &H41, &H5F, &H46, &H4F, &H52, &H5F, &H53, &H50, &H45, &H41, &H43, &H46, &H5F  
DATA &H43, &H41, &H52, &H44, &H53, &H5F, &H4F, &H4E, &H5F, &H50, &H4F, &H52, &H54, &H53, &H5F, &H5F  
DATA &H50, &H37, &H26, &H50, &H38, &H5F, &H2A, &H2A, &H2A, &H2A, &H2A, &H2A, &H2A, &H2A, &H2A, &H2A  
DATA &H46, &H45, &H58, &H20, &HFF, &HFF, &HFF, &HFF, &H44, &H41, &H54, &H41, &HFF, &HFF, &HFF, &HFF  
DATA &H14, &H3F, &H37, &H21, &H13, &H07, &H0A, &H14, &H13, &H0A, &H2A, &H07, &H07, &H07, &H35, &H09  
DATA &H2A, &H3B, &H07, &H0D, &H31, &H23, &H21, &H07, &H2E, &H2B, &H04, &H13, &H37, &H13, &H04, &H07  
DATA &H13, &H01, &H06, &H07, &H07, &H07, &H07, &H07, &H04, &H13, &H31, &H04, &H07, &H13, &H1F, &H13  
DATA &H0F, &H07, &H06, &H13, &H04, &H04, &H13, &H04, &H04, &H23, &H04, &H02, &H04, &H14, &H14, &H3E

'&H80

DATA &H10, &H0B, &H04, &H04, &H1F, &H04, &H37, &H04, &H04, &H04, &H01, &H02, &H04, &H04, &H04, &H04  
DATA &H04, &H04, &H04, &H28, &H04, &H32, &H04, &H04, &H04, &H04, &H04, &H04, &H04, &H04, &H03  
DATA &H37, &H04, &H04, &H04, &H3F, &H29, &H04, &H04, &H00, &H03, &H04, &H00, &H00, &H04, &H00, &H04  
DATA &H03, &H04, &H03, &H04, &H04, &H04, &H00, &H00, &H03, &H02, &H04, &H02, &H02, &H00, &H3E, &H37  
DATA &H00, &H00, &H00, &H02, &H00, &H00, &H00, &H04, &H00, &H03, &H00, &H00, &H00, &H03, &H03, &H04  
DATA &H00, &H00, &H00, &H00, &H04, &H00, &H00, &H00, &H31, &H37, &H00, &H00, &H00, &H00, &H00, &H00  
DATA &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00  
DATA &H00, &H00, &H1F, &H04, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00

'&H100

DATA &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00, &H00

'A

B

DATA &H00, &H14, &H04, &H00, &H00, &H00, &H00, &H00, &H00, &H3F, &H13, &H00, &H00, &H00, &H00, &H00

'C

D

DATA &H00, &H37, &H37, &H13, &H00, &H00, &H00, &H00, &H00, &H21, &H13, &H00, &H00, &H00, &H00, &H00

'E

F

DATA &H00, &H13, &H00, &H00, &H00, &H00, &H00, &H00, &H07, &H07, &H23, &H23, &H00, &H00, &H00

'G

H

DATA &H00, &H0A, &H13, &H00, &H00, &H00, &H00, &H00, &H14, &H00, &H01, &H32, &H00, &H00, &H00

'I

J

DATA &H00, &H18, &H06, &H00, &H00, &H00, &H00, &H00, &H0A, &H07, &H14, &H00, &H00, &H00, &H00, &H00

'K

L

DATA &H00, &H2A, &H07, &H14, &H00, &H00, &H00, &H00, &H00, &H07, &H07, &H3E, &H00, &H00, &H00, &H00

'M

N

DATA &H00, &H07, &H07, &H10, &H00, &H00, &H00, &H00, &H00, &H07, &H07, &H0B, &H00, &H00, &H00, &H00

'&H180

P

DATA &H00, &H35, &H00, &H00, &H00, &H00, &H00, &H00, &H09, &H13, &H00, &H00, &H00, &H00, &H00

'Q

R

DATA &H00, &H2A, &H31, &H1F, &H00, &H00, &H00, &H00, &H3B, &H00, &H00, &H00, &H00, &H00, &H00

'S

T

DATA &H00, &H07, &H07, &H37, &H37, &H00, &H00, &H00, &H00, &H0D, &H13, &H00, &H00, &H00, &H00, &H00

'U

V

DATA &H00, &H31, &H1F, &H00, &H00, &H00, &H00, &H00, &H23, &H13, &H00, &H00, &H00, &H00, &H00

'W

X

DATA &H21, &H0F, &H01, &H3F, &H3B, &H31, &H1F, &H00, &H00, &H07, &H07, &H01, &H29, &H37, &H37, &H00

'Y

Z

DATA &H00, &H2E, &H06, &H00, &H00, &H00, &H00, &H00, &H00, &H2B, &H13, &H00, &H00, &H00, &H00, &H00

'GENIUS

HEX

DATA 10 , 19 , 56 , 12 , 40 , 55 , 55 , 0 , &H1B, &H07, &H07, &H01, &H29, &H37, &H37, &H00

'YES

NO

DATA &H31, &H07, &H07, &H37, &H37, &H00, &H00, &H00, &H38, &H0F, &H35, &H00, &H00, &H00, &H00, &H00

'STOP &H200

START

DATA &H37, &H37, &H11, &H18, &H18, &H01, &H09, &H00, &H37, &H37, &H02, &H0D, &H3B, &H02, &H0D, &H00

'SWITCH

FREEZE

DATA 55 , 55 , &H30, &H0C, &H0C, &H02, &H32, &H00, &H23, &H23, &H0B, &H13, &H2B, &H00, &H00, &H00

'LOGOS

MUSIC

DATA 45 , 23 , 15 , 41 , 23 , 55 , 55 , 0 , &H10, &H31, &H1F, &H2B, &H0C, &H00, &H29, &H00

```

EMOTIONAL          &H230          ERROR
DATA 19 , 16 , 53 , 37 , 15 , 11 , 15 , 62 , 5 , 7 , 7 , 39 , 58 , 0 , 0 , 0
'ENGAGEMENT      &H240
DATA 7 , 7 , 0 , 11 , 36 , 20 , 2 , 10 , 16 , 7 , 7 , 11 , 2 , 3 , 13 , 2
'SYSTEMS        &H250
DATA 55 , 55 , 12 , 12 , 55 , 55 , 3 , 13 , 7 , 16 , 43 , 0 , 0 , 0 , 0 , 0
'ROBOTS &H260          TIME
DATA 14 , 53 , 1 , 63 , 24 , 2 , 17 , 55 , 0 , 13 , 24 , 6 , 16 , 0 , 0 , 0
'TALKING &H270          WHALERS
DATA 13 , 23 , 23 , 3 , 42 , 12 , 44 , 2 , 46 , 20 , 45 , 51 , 43 , 0 , 0 , 0
'SPEAK          (&H290)          PINNING
DATA 55 , 55 , 3 , 19 , 3 , 41 , 0 , 0 , 9 , 12 , 12 , 11 , 12 , 44 , 0 , 0
'NIP &H290          MEMORY
DATA 11 , 12 , 12 , 2 , 3 , 9 , 0 , 0 , 16 , 7 , 7 , 16 , 52 , 19 , 0 , 0
'KEY &H2A0          LITTLE
DATA 0 , 42 , 19 , 0 , 0 , 0 , 0 , 0 , 45 , 12 , 12 , 12 , 3 , 13 , 62 , 0
'SWEAT &H2B0          READY
DATA 55 , 55 , 46 , 7 , 7 , 3 , 13 , 0 , 14 , 7 , 7 , 1 , 33 , 19 , 0 , 0
'SINCERELY &H2C0
DATA 55 , 55 , 12 , 12 , 11 , 55 , 55 , 60 , 45 , 19 , 0 , 0 , 0 , 0 , 0 , 0
'SCORE &H2D0          PLUS
DATA 55 , 55 , 3 , 8 , 58 , 0 , 0 , 0 , 9 , 45 , 15 , 15 , 55 , 55 , 0 , 0
'SENSITIVE &H2E0
DATA 55 , 55 , 7 , 7 , 11 , 55 , 55 , 12 , 2 , 3 , 13 , 12 , 35 , 0 , 0 , 0
'FAST &H2F0          FUCK
DATA 40 , 26 , 55 , 55 , 13 , 0 , 0 , 0 , 40 , 15 , 1 , 8 , 1 , 0 , 0 , 0
'einde speech data voor kaart 8
=====

```





'data vanaf 4000:1A80 in eprom-dump -LOOKUP-tables dit was is in de HX20 versie het adresgebied van &H7A80

tot &H7C6F  
'&H00 ascii-label KWART-SINUS\*255-  
DATA &H4B, &H57, &H41, &H52, &H54, &H2D, &H53, &H49, &H4E, &H55, &H53, &H2A, &H32, &H35, &H2D  
'sinus  
DATA &H00, &H1A, &H35, &H4E, &H67, &H7F, &H95, &HAA, &HBD, &HCE, &HDC, &HE8, &HF2, &HF9, &HFD, &HFF  
'cosinus  
DATA &HFF, &HFD, &HF9, &HF2, &HE8, &HDC, &HCE, &HBD, &HAA, &H95, &H7F, &H67, &H4E, &H35, &H1A, &H00  
'tangens  
DATA &H00, &H1A, &H36, &H52, &H71, &H93, &HB9, &HE5, &HE5, &HB9, &H93, &H71, &H52, &H36, &H1A, &H00  
'ascii-label EINDE...GONIO...  
DATA &H45, &H49, &H4E, &H44, &H45, &HFF, &HFF, &HFF, &H47, &H4F, &H4E, &H49, &H4F, &HFF, &HFF, &HFF  
'STICHTING--LOGOS  
DATA &H53, &H54, &H49, &H43, &H48, &H54, &H49, &H4E, &H47, &H2D, &H2D, &H4C, &H4F, &H47, &H4F, &H53  
'Godfried-Willem.  
DATA &H47, &H6F, &H64, &H66, &H72, &H69, &H65, &H64, &H2D, &H57, &H69, &H6C, &H6C, &H65, &H6D, &HFF  
'RAES...GENT1988  
DATA &H52, &H41, &H45, &H53, &HFF, &HFF, &HFF, &HFF, &H47, &H45, &H4E, &H54, &H31, &H39, &H38, &H38  
'&H80 E12-reeks van 10 tot 180  
DATA &H0A, &H0C, &H0F, &H12, &H16, &H1B, &H21, &H27, &H2F, &H38, &H44, &H52, &H64, &H78, &H96, &HB4  
'1 byte (255) verdeeld in 12 gelijkgetemperde intervallen  
DATA &H01, &H02, &H03, &H04, &H0A, &H10, &H19, &H28, &H40, &H65, &HA1, &HFF, &H00, &H00, &H00, &H00  
'bit rotaties 1 bit doorschuiven 2 bits doorschuiven  
DATA &H01, &H02, &H04, &H08, &H10, &H20, &H40, &H80, &H03, &H06, &H0C, &H18, &H30, &H60, &HC0, &H81  
'3 bits doorschuiven 4 bits doorschuiven  
DATA &H07, &H0E, &H1C, &H38, &H70, &HE0, &HC1, &H83, &H0F, &H1E, &H3C, &H78, &HFO, &HE1, &HC3, &H87  
'5 bits doorschuiven 6 bits doorschuiven  
DATA &H1F, &H3E, &H7C, &HF8, &HF1, &HE3, &HC7, &H8F, &H3F, &H7E, &HFC, &HF9, &HF3, &HE7, &HCF, &H9F  
'7 bits doorschuiven just intonation 128-255 verdeling  
DATA &HFE, &HFD, &HFB, &HF7, &HEF, &HDF, &HBF, &H7F, &H80, &H90, &HAD, &HAB, &HCO, &HD5, &HE6, &HFF  
'25-192 in kwarten reeks 1  
DATA &H19, &H22, &H2E, &H3D, &H51, &H6C, &H90, &HCO, &H01, &H01, &H02, &H03, &H05, &H08, &H0D, &H15  
' reeks 2  
DATA &H22, &H37, &H59, &H90, &HE9, &H00, &H00, &H00, &H01, &H01, &H02, &H04, &H07, &H0D, &H18, &H2C  
'&H100 reeks 3  
DATA &H51, &H95, &H00, &H00, &H00, &H00, &H00, &H00, &H01, &H01, &H02, &H04, &H08, &H0F, &H1D, &H38  
' reeks 4  
DATA &H63, &HD0, &H00, &H00, &H00, &H00, &H00, &H00, &H01, &H01, &H02, &H04, &H08, &H10, &H1F, &H3D  
' reeks 5  
DATA &H78, &HEC, &H00, &H00, &H00, &H00, &H00, &H00, &H01, &H01, &H02, &H04, &H08, &H10, &H20, &H40  
' reeks 6  
DATA &H80, &HFF, &H00, &H00, &H00, &H00, &H00, &H00, &H01, &H03, &H06, &H0A, &H0F, &H15, &H1C, &H24  
'  
DATA &H2D, &H37, &H42, &H4E, &H5B, &H69, &H78, &H88, &H99, &HAB, &HBE, &HD2, &HE7, &HFD, &H00, &H00  
'major scale of just intonation 32 bytes  
DATA &H01, &H02, &H04, &H08, &H10, &H12, &H14, &H16, &H19, &H1B, &H1F, &H21, &H25, &H29, &H2C, &H32  
DATA &H37, &H3E, &H42, &H4A, &H52, &H58, &H63, &H6E, &H7C, &H84, &H94, &HA5, &HB0, &HC6, &HDC, &HF7  
'ascii-label EINDE\_LOOKUP...  
DATA &H45, &H49, &H4E, &H44, &H45, &H5F, &H4C, &H4F, &H4E, &H4B, &H55, &H50, &H00, &H00, &H00, &H00  
'&H180 vrij voor expansie  
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF  
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF  
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF  
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF  
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF  
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF  
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF  
DATA &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF, &HFF  
'einde data voor RAM-look-up table laatste plaats = &H1FF

### SUB INSTALL (DP, CP, SP)

```
CLS
DP = &H378: CP = DP + 1: SP = DP + 2
'bit 7 van CP is omgekeerde logica !!! - elektrisch 1 (busy) = logisch 0
'reset all cards:
OUT DP, 0
FOR I = 96 TO 110 STEP 2: OUT SP, I: OUT SP, I + 1: OUT SP, I: NEXT I
LOCATE 10, 10: PRINT " Opgelet : deze software werkt alleen met de erbij "
LOCATE 11, 10: PRINT "          horende hardware ! "
LOCATE 12, 10: PRINT " De aanwezigheid van speech-card 3 is essentieel "
LOCATE 13, 10: PRINT " bovendien dient voor deze kaart een centronics kabel"
LOCATE 14, 10: PRINT " gebruikt te worden met alle aders verbonden !!! "
LOCATE 16, 10: PRINT " Voedingsspanningen : "
LOCATE 17, 10: PRINT "          - P1 : Midi-kaart   5 Volt "
LOCATE 18, 10: PRINT "          - P2A:Printernaalden 5 Volt via wide flat "
LOCATE 19, 10: PRINT "          - P2B:staalnaalden  5 Volt via wide flat "
LOCATE 20, 10: PRINT "          - P3 : MD-oscillator 5 Volt "
LOCATE 21, 10: PRINT "          - P4 : drums          5 Volt via wide flat "
LOCATE 22, 10: PRINT "          - P5 StringSpring   5 Volt "
LOCATE 23, 10: PRINT "          - P6, steppermotors 12 Volt digital "
LOCATE 24, 10: PRINT "          - P7,P8:speech cards 12/5 Volt analog "
LOCATE 25, 10: PRINT " H E X 08/89 Toshiba T1000 versie "; DATE$; " "; TIME$
END SUB
```

### SUB ROM (RP0(), RP1(), RP6(), RP8())

```
'Eprom data corresponds to HX20 - HEX-Eprom
'data rex midi vanaf &H7000
FOR I = 0 TO &HFF
  READ RP1(I)
NEXT I
FOR I = 0 TO &H2FF
  READ RP6(I)
NEXT I
FOR I = 0 TO &H27F
  READ RP6(I)
NEXT I
FOR I = 0 TO &H1FF
  READ RP0(I)
NEXT I
END SUB
```