U

#1450

NLST

```
*********************************
      NAM PAGER
*     02/25/86 V2.00
*     03/26/86 V2.10 FIX BUG IN CALC
*      PROGRAM
*      J & R ELECTRONICS
*      P.O.BOX 2572
*      COLUMBIA, MD 21045
*
*      J.W.JACKSON
*      COPYRIGHT 1985
*      ALL RIGHTS RESERVED
*********************************
* REPRODUCTION OF THIS PROGRAM*
*, IN WHOLE OR IN PART, IS     *
*EXPRESSLY PROHIBITED FOR USES*
*OTHER THAN EDUCATIONAL, OR    *
*FOR THE DEVELOPMENT OF SOFT-  *
*WARE THAT WILL BE USED        *
*EXCLUSIVELY ON J&R PRODUCTS.  *
*                              *
*********************************
* MAXBNK = MAXIMUM BANK # ;     *
*  3 FOR 256K, 7 FOR 512K.      *
* MAXPAG = 2*MAXBNK - 1;        *
*  7 FOR 256K, 15 FOR 512K.     *
* PAGMSK = UPPER PAGE MASK BIT*
*  4 FOR 256K, 8 FOR 512K.      *
* THE ARRANGEMENT IS THUS:      *
*===============================*
*       256K (MAXBNK = 3)       *
*===============================*
*             BANK              *
*  0   :   1   :   2   :   3  *
*===============================*
*      UPPER 32K PAGE #         *
*  4   :   5   :   6   :   7  *
*-------------------------------*
*      LOWER 32K PAGE #         *
*  0   :   1   :   2   :   3  *
*********************************
*===============================*
*       512K (MAXBNK = 7)       *
*===============================*
*             BANK              *
*0 : 1 : 2 : 3 : 4 : 5 : 6 : 7*
*===============================*
*      UPPER 32K PAGE #         *
*8 : 9 : 10: 11: 12: 13: 14:15*
*-------------------------------*
*      LOWER 32K PAGE #         *
```

wrong (+1)

Seems to work with 64K SW OK

```
*0 : 1 : 2 : 3 : 4 : 5 : 6 : 7*
*================================*
*                                *
**********************************
* THIS PROGRAM COPIES PAGE 0,    *
* BANK 0 TO PAGES 1 THRU MAXPG*
* OF THE       RAM TO ACHIEVE    *
*   PAGES OF 32K IN THE LOWER    *
* 32K OF THE COMPUTER MEMORY.    *
* THE UPPER 32K WILL BE ROM.     *
* THE COMMAND : PAGE N           *
* PAGE command added             *
* USE : PAGE n                   *
* WHERE n = 0 thru MAXPAG.       *
*  PAGE alone will not           *
* switch pages, but will         *
* print the active page.         *
*                                *
*  v1.01                         *
* PPOKE command added            *
* USE : PPOKE P,A,V              *
* WHERE P = page number          *
*       A = address              *
*       V = value                *
*                                *
* PPEEK command added            *
* USE : PPEEK P,A                *
* WHERE P = page number          *
*       A = address              *
*                                *
* if a 64k program is run ,      *
*Pages will be full banks of     *
*64k and Pages are 0-MAXBNK.    *
**********************************
STRPTR EQU $21 START OF BASIC POINTER
EOBPTR EQU $23 END   OF BASIC POINTER
LIMPTR EQU $27 END OF STRING POINTER
MEMEND EQU $74 END OF RAM POINTER
DEVNUM EQU $6F DEVICE NUMBER ADDRESS
INCH    EQU $A000
OUTCH   EQU $A002
GNCBIN  EQU $B70B GET NEXT CHR
BLATCH  EQU $FFC0 CPU BANK LATCH
VLATCH  EQU $FFD0 VDG BANK LATCH

 LIST
*********************************
*THIS IS TEMPORARY CODE FOR INITIALIZATION
       ORG $0E00
START  BRA START1 SKIP TABLES
DESTIN EQU $7800 WHERE THE PAGER WILL RESIDE WHEN MOVED
PDEST  FDB DESTIN PROGRAM DESTINATION FOR TRANSFER
* MAPTYP IS FOR FUTURE VERSION
MAPTYP FCB $00   IFNE. PAGES ARE 64K. ELSE 32K
```

page def changes

11

```
        * OFFSET IS CALC'D BY THE INSTALL ROUTINE
        OFFSET FDB $0000 OFFSET FROM HERE TO DEST

         NLST
+7      START1 PSHS D,U,X,Y,CC
               ORCC #$50
+05 ——————     LDD LIMPTR FOR CALC
               SUBD PDEST,PCR
+10 ——         BLT SAVSTK STACK UNDER US
               TFR S,D GET STACK
               SUBD PDEST,PCR
               CMPD #(CODFIN-CODE+$80) IN OUR SPACE?
               BGT SAVSTK STACK OVER US
               LBSR FIXMEM CLEAR STACK SPACE
+20    SAVSTK STS RETSTK,PCR

        *PRINT SIGN-ON MESSAGE ********************
               LEAX PMSG,PCR INDEX MESSAGE TEXT
               LBSR PSTRNG PRINT THE STRING

               LEAX PBOOT,PCR INDEX MESSAGE TEXT
               LBSR PSTRNG PRINT THE STRING
        *CALC MAXPAG & PAGMSK
               LDA MAXBNK,PCR
               LSLA 2*MAXBNK
               INCA  PLUS 1
               STA MAXPAG,PCR
               EORA MAXBNK,PCR
               STA PAGMSK,PCR

        * MOVE THE CODE TO DEST ADDDRESS
               LEAU CODE,PCR BASE ADDRESS NOW
               LDX PDEST,PCR WHERE IT GOES
               LDD PDEST,PCR FOR CALC
               STU OFFSET,PCR SO WE CANSUBTRACT
               SUBD OFFSET,PCR FIND DIFFERENCE
               STD OFFSET,PCR SAVE FOR CALCS

        *FIND EMPTY CMD TABLE
               LDU #$120 START OF TOKEN TABLES
               LDB #$80 TOKEN OFFSET
        FNDMTC LDA 0,U HOW MANY
               BEQ MTC EMPTY?
               ADDB 0,U CALC TOKEN
               LEAU $0A,U TEN BYTES PER TOKEN ENTRY
               BRA FNDMTC LOOK AGAIN

        MTC    STB NEWCMD,PCR SAVE NEW TOKEN VALUE
               ADDB #NUMCMD-1
               STB LASTCM,PCR SAVE LAST TOKEN VALUE
               STU NCMADD,PCR & NEW CMD TABLE ADDRESS
        *ADD PAGE COMMAND
               LDA #(CTBEND-CMDTAB)/2 =NUMBER OF ADDED COMMANDS
               STA 0,U
```

```
        LEAX CMDLST,PCR POINT TO CMD
        LDD OFFSET,PCR GET OFFSET
        LEAX D,X ADD OFFSET
        STX 1,U
        LEAX FNDCMD,PCR POINT TO EXEC
        LEAX D,X ADD OFFSET
        STX 3,U
        CLR $A,U NO MORE
        LDX #$B277 ERROR ADDRESS
        STX $B,U
        STX $D,U

*FIND EMPTY FUN TABLE
        LDU #$125 START OF FUNCTION TABLES
        LDB #$00 TOKEN OFFSET
FNDMTF  LDA 0,U HOW MANY
        BEQ MTF EMPTY?
        ADDB 0,U CALC TOKEN
        LEAU $0A,U TEN BYTES PER TOKEN ENTRY
        BRA FNDMTF LOOK AGAIN

MTF     LSLB #FUNCTIONS*2 = LAST TOKEN
        STB NEWFUN,PCR SAVE NEW TOKEN VALUE
        ADDB #(NUMFUN-1)*2
        STB LASTFN,PCR SAVE LAST TOKEN VALUE
        STU NFNADD,PCR & NEW CMD TABLE ADDRESS
*ADD FUNCTION
        LDA #(FTBEND-FUNTAB)/2 =NUMBER OF ADDED FUNCTIONS
        STA 0,U
        LEAX FUNLST,PCR POINT TO FUN
        LDD OFFSET,PCR GET OFFSET
        LEAX D,X ADD OFFSET
        STX 1,U
        LEAX FNDFUN,PCR POINT TO EXEC
        LEAX D,X ADD OFFSET
        STX 3,U

        CLR $A,U NO MORE
        LDX #$B277 ERROR ADDRESS
        STX $B,U
        STX $D,U

        LEAU CTBEND,PCR
        STU CODFIN,PCR
        LEAU CMDTAB,PCR INDEX COMMAND TABLE
FIXCTB  LDX 0,U GET POINTER
        LEAX D,X CALC NEW ADDRESS
        STX 0,U++ UPDATE IT
        CMPU CODFIN,PCR END OF TABLE?
        BNE FIXCTB


        LEAU FTBEND,PCR
        STU CODFIN.PCR
```

```
        LEAU FUNTAB,PCR INDEX FUNCTION TABLE
FIXFTB  LDX 0,U GET POINTER
        LEAX D,X CALC NEW ADDRESS
        STX 0,U++ UPDATE IT
        CMPU CODFIN,PCR END OF TABLE?
        BNE FIXFTB

        LEAX VDGHOK,PCR WHERE TOHOOK
        LEAX D,X CALC OFFSET
        CMPX #0168 HOOK FROM ADDRESS
* HOOKS ALREADY SET?
        BEQ MVCODE IFEQ, SKIP SETTING HOOKS

        LDX #$0168 POINT TO OUTCH HOOK
        LEAU RESUME+1,PCR RE-HOOK ADDRESS
        LDA -1,X GET OLD JMP
        STA -1,U SAVE FOR RE-HOOKING
        LDD 0,X GET OLD ADDRESS
        STD 0,U SAVE FOR RE-HOOKING

        LEAU VDGHOK,PCR ADDRESS OF NEW HOOK
        LDD OFFSET,PCR
        LEAU D,U ADD OFFSET
        LDA #$7E "JMP" OP-CODE
        STA -1,X SET HOOK
        STU 0,X ADDRESS

*MOVE THE CODE FROM LOAD ADDRESS TO PDEST
MVCODE  LDX PDEST,PCR INDEX THE DESTINATION
        LEAU CODFIN,PCR GET END OF CODE
        STU 0,U SAVE ADDDRESS
        LEAU CODE,PCR INDEX THE SOURCE

MVLOOP  LDA 0,U+ GET A BYTE
        STA 0,X+ PUT MOVE IT
        CMPU CODFIN,PCR FINISHED?
        BNE MVLOOP IFNE, KEEP MOVING

* COPY (0 TO )0 - )(MAXBNK-256 BYTES) ***************
CPYBUX  LDU #$0000 START OF LOWER RAM
        LDX #$8000 START OF UPPER RAM
        LDY #$FFC8 CPU ADDRESS -HOLDS LOW BANK ALWAYS 0
*COPY BANK DATA FROM U TO X
CPYB1   LDA 0,U+ GET DATA FROM (0
        STA $FFDF MAP TYPE 1 - ALL RAM
        LDB MAXBNK,PCR DO ALL BANKS
CPYB2   BITA B,Y SWITCH BANKS
        STA 0,X STORE TO )32K BANK
        BITA 0,Y BACK TO BANK 0
        DECB ALL DONE?
        BPL CPYB2 ALL DONE?

        LEAX 1,X NEXT ADDR
        CMPX #$FF00 END OF )32K RAM?
```

```
        BNE CPYB1 FINISH UP

        LEAU PTHI,PCR CALC ADDRESS
        JMP $8000,U IN UPPER RAM

*COPY )0 TO (1 - (MAXBNK ###############
PTHI    LDU #$8000 START OF UPPER RAM
        LDX #$0000 START OF LOWER RAM
        LDY #BLATCH WHOLE BANKS NOW THAT UPPER 32K IS READY
CPYB3   LDA 0,U+ GET DATA FROM )0
        STA $FFDF MAP TYPE 1 - ALL RAM
        LDB MAXBNK,PCR ALL BANKS
CPYB4   BITA B,Y SWITCH BANKS
        STA 0,X STORE TO (32K BANK
        BITA 0,Y BACK TO BANK 0
        DECB XNEXT BANK
        BPL CPYB4 ALL DONE?

        LEAX 1,X
        CMPU #$FF00 END OF UPPERRAM?
        BNE CPYB3 DONE?

        LEAU XRET,PCR CALC ADDRESS
        JMP $8000,U IN LOW RAM
XRET    STA $FFDE BACK TO ROM

*SWITCH 32K PAGE AND COPY LAST 256 BYTES $7F00-$7FFF
* COPIES (0 $7F00-$7FFF TO (1 -( MAXBNK (PAGE BIT = 0)
*   AND )0 - )MAXBNK  (PAGE BIT = 1)
        LDU #$7F00 GET REST OF LOWER 32K
        LDY #BLATCH

X06     LDB MAXBNK,PCR
X05     BITA 0,Y SWITCH TO BANK 0
        LDA 0,U GET DATA
        BITA B,Y SWITCH TO BANK
        STA $FFD5 SWITCH PAGE 1 32K
        STA 0,U STORE DATA

        STA $FFD4 SWITCH PAGE 0 32K
        STA 0,U STORE DATA
        DECB NEXT BANK
        BPL X05 DONE?

        LEAU 1,U NEXT ADDRESS
        CMPU #$8000 END OF LOWER32K?
        BNE X06

*CALCULATE TRANSFERRED PAGE ADDRESS
        LEAX PAGID,PCR IDENT LOCATION
        LDD OFFSET,PCR GET OFFSET
        LEAX D,X CALC NEW ADDRESS

*PUT PAGE IDENTIFICATION IN PAGES OF 32K
```

```
* Y IS $FFC0 FROM PREVIOUS ROUTINE
        LDB MAXBNK,PCR
PUTID   BITA B,Y SWITCH BANKS
        STB 0,X ID PAGES 0-MAXBNK
        STA $FFD5 SWITCH PAGE 1
        ORB PAGMSK,PCR HIGH PAGE OFFSET
        STB 0,X ID UPPER PAGES

        ANDB MAXBNK,PCR LOW ID MASK
        STA $FFD4 SWITCH PAGE 0
        DECB
        BPL PUTID PUT ID IN PAGES

        LDS RETSTK,PCR
        LEAX PDOC,PCR INDEX MESSAGE TEXT
        LBSR PSTRNG PRINT THE STRING
        PULS D,U,X,Y,CC
        RTS

FIXMEM EXG U,S SAVE STACK IN U
        LDS PDEST,PCR
        LEAS -1,S
        LDD LIMPTR
        SUBD STRPTR CLEAR HOW MANY?
        STS LIMPTR
        STD 0,S
        TFR S,D GET STACK
        SUBD 0,S MOVE DOWN
        TFR D,S CLEAR STRINGS
        STS STRPTR
        LEAS -5,S STACK SET
        EXG U,S RESTORE STACK

        LDB #$20 MOVE 32 BYTES
STK     LDA B,S FROM OLD STACK LOC
        STA B,U TO NEW STACK LOC
        DECB
        BPL STK
        EXG U,S NEW STACK

FIXRET RTS
*****************************************

PSTRNG LDA 0,X+ GET CHAR
        BEQ PSTRN1 0 TERMINATES
        JSR [OUTCH]
        BRA PSTRNG ANOTHER
PSTRN1 RTS

PMSG    FCB $0D,$0D,$0D,$0D,$0D
        FCC '  ******************************',$0D
        FCC '  *       PAGER    PROGRAM     *',$0D
        FCC '  *              V2.00          *',$0D
        FCC '  *       J & R ELECTRONICS     *',$0D
```

```
        FCC ' *       P.O.BOX 2572          *',$0D
        FCC ' *                             *',$0D
        FCC ' *       J. W. JACKSON         *',$0D
        FCC ' *       COPYRIGHT 1985        *',$0D
        FCC ' *       ALL RIGHTS RESERVED *',$0D
        FCC ' ***********************************',$0D
        FCB $0D,0

PBOOT   FCB $0D
        FCC ' ..Booting PAGER..',$0D,0
PDOC    FCB $0D,$0D,$0D,$0D
        FCC ' ***********************************',$0D
        FCC ' Commands added:',$0D,$0D
        FCC '      PAGE page',$0D
        FCC '      PPOKE page,address,data',$0D
        FCC '      PPEEK page,(address)',$0D,$0D
        FCC ' USE : PAGE',$0D
        FCC ' TO DISPLAY THE ACTIVE PAGE# ',$0D
        FCC ' ***********************************',$0D
        FCB $0D,$0D,0
        FCC '****' FILL SPACE

******* END OF THE INSTALLATION CODE ********
 LIST
*       ORG [PDEST,PCR]
*THIS IS THE PERMANENT CODE THAT MUST REMAIN IN
* FOR THE PAGE COMMAND
* A=TOKEN B= ?
* U= PTR TO NEXT CHAR IN BUFFER
* Y= PTR TO NEXT KEYWORD


FNDCMD BRA BB1
*
CODE EQU FNDCMD START OF CODE TRANSFER
MAXBNK FCB $03 256K=3, 512K=7
MAXPAG FCB $07 INSTALL CALCS THIS
PAGMSK FCB $04 " "
PAGE   FCB $00
PAGID  FCB $00
NEWCMD FCB $00 NEW CMD TOKEN VALUE
NCMADD  FDB $0000 NEW TOKEN TABLE ADDRESS
LASTCM FCB $00 LAST CMD TOKEN VALUE
NEWFUN FCB $00 NEW FUN TOKEN VALUE
NFNADD  FDB $0000 NEW TOKEN TABLE ADDRESS
LASTFN FCB $00 LAST FUN TOKEN VALUE
RETSTK FDB $0000

 NLST
BB1     CMPA LASTCM,PCR TOO HIGH?
        BHI NOTFND
        SUBA NEWCMD,PCR GET TOKEN OFFSET
        ASLA TABLE OF TWO'S
        LEAU CMDTAB,PCR COMMAND TABLE
```

```
        JMP [A,U] EXECUTE

NOTFND  LDU NCMADD,PCR
        JMP [11,U] ERROR ROUTINE

CMDLST  FCC 'PAG','E+$80
        FCC 'PPOK','E+$80
CMDTAB  FDB PAGEXC
        FDB POKEPG
CTBEND  EQU * COMMAND TABLE END
* FOR THE FUNCTIONS
* A=NXTCH B=SECOND BYTE OF TOKN
* U= PTR TO NEXT CHAR IN BUFFER
* Y= PTR TO NEXT KEYWORD
*
FNDFUN  CMPB LASTFN,PCR TOO HIGH?
        BHI NOTFND
        SUBB NEWFUN,PCR GET TOKEN OFFSET
        ASLB TABLE OF TWO'S
        LEAU FUNTAB,PCR COMMAND TABLE
        JMP [B,U] EXECUTE

FUNLST  FCC 'PPEE','K+$80
FUNTAB  FDB PEEKPG

*
FTBEND  EQU * FUNCTION TABLE END
*
NUMCMD  EQU (CTBEND-CMDTAB)/2 =NUMBER OF ADDED COMMANDS
NUMFUN  EQU (FTBEND-FUNTAB)/2 =NUMBER OF ADDED FUNCTIONS
PAGEXC  JSR $9F GET NEXT CHAR
        TSTA NO ARGUMENT?
        BEQ SHOPAG IF NONE, PRINT PAGE #

        JSR GNCBIN GET PAGE #
        CMPB MAXPAG,PCR PAGE # TOO HIGH?
        BHI NOTFND ERROR IN PAGE #
        STB PAGE,PCR STORE PAGE #
        BRA SWITCH GO TO IT

SHOPAG  LDA PAGID,PCR WHAT PAGE ARE WE IN?
        CMPA #$09 NUMERIC?
        BLS SHOX1 PRINT IT

        SUBA #$0A ALPHA $0A-$0F
        PSHS A SAVE
        LDA #$01 TENS DIGIT
        BSR SHOX1
        PULS A ONES DIGIT

SHOX1   ORA #$30 BINARY TO ASCII
        JSR [OUTCH] PRINT PAGE #
FNDRET  RTS
```

```
SWITCH PSHS D,U,X,Y,CC SAVE THIS PAGE'S REGISTERS
       ORCC #$50
       STS RETSTK,PCR SAVE THIS PAGE'S STACK
       LDY #BLATCH

       LDA PAGE,PCR WHERE TO GO
       STA $FFD2 SCREEN LOW
       STA $FFD4 PAGE BIT 0
       BITA PAGMSK,PCR IS THIS AN UPPER 32K?
       BEQ S01 IFNE, IT'S >32K

       STA $FFD3 SCREEN HI IS AT $8400
       STA $FFD5 PAGE BIT= 1

S01    ANDA MAXBNK,PCR BANK # OF PAGE
       BITA A,Y SWITCH TO IT
       LDY #VLATCH VIDEO LATCH ADDRESS
       BITA A,Y SWITCH TO VIDEO IN THAT BANK

       LDS RETSTK,PCR GET PAGE'S STACK
       PULS D,U,X,Y,CC RESTORE PAGE'S REGISTERS
       RTS

POKEPG JSR $9F GET NEXT CHAR
       JSR $B70B GET NEXT BIN CHAR IN B
       CMPB MAXPAG,PCR TOO HIGH?
       LBHI NOTFND
       STB PAGE,PCR WHERE TO
*NOW DO POKE
       JSR $B26D PARSE COMMA
       JSR $B734 GET ADDRESS ARGUMENT IN $2B, DATA IN REG. B
       LDX $2B GET POKE ADDRESS

*POKE DATA IN B TO ADDRESS AT X
POKB2X LDA PAGE,PCR WHERE TO
       LDU PAGID,PCR UH=FROM UL=IRRELEVANT
       LDY #BLATCH
       STA $FFD4 PAGE 0 THIS BANK
       BITA PAGMSK,PCR IS THIS AN UPPER 32K?
       BEQ PA1 IFNE, IT'S >32K
       STA $FFD5 PAGE BIT= 1

PA1    ANDA MAXBNK,PCR GET BANK # OF PAGE
       BITA A,Y SWITCH TO IT
       STB 0,X POKE IT
       EXG D,U A=PAGE FROM
       STA $FFD4 PAGE 0 THIS BANK
       BITA PAGMSK,PCR IS THIS AN UPPER 32K?
       BEQ PA2 IFNE, IT'S >32K
       STA $FFD5 PAGE BIT= 1

PA2    ANDA MAXBNK,PCR GET BANK # OF PAGE
       BITA A,Y SWITCH TO IT
```

```
        RTS

PEEKPB  JSR $B70B GET NXT CHAR BIN
        CMPB MAXPAG,PCR MAX PAGE?
        LBHI NOTFND
        STB PAGE,PCR WHERE TO
        JSR $B26D PARSE COMMA
        JSR $B73D GET ADDRESS ARGUMENT IN X
        BSR PEKX2B DO PEEK
        JMP $B4F3 LIKE THE (PEEK) COMMAND

*PEEK ADDRESS IN X TO DATA IN B
PEKX2B  LDA PAGE,PCR WHERE TO
        LDU PAGID,PCR UH=FROM UL=IRRELEVANT
        LDY #BLATCH
        STA $FFD4 PAGE 0 THIS BANK
        BITA PAGMSK,PCR IS THIS AN UPPER 32K?
        BEQ PE1 IFNE, IT'S >32K
        STA $FFD5 PAGE BIT= 1

PE1     ANDA MAXBNK,PCR GET BANK # OF PAGE
        BITA A,Y SWITCH TO IT
        LDB 0,X PEEK IT
        EXG D,U A=PAGE FROM
        STA $FFD4 PAGE 0 THIS BANK
        BITA PAGMSK,PCR IS THIS AN UPPER 32K?
        BEQ PE2 IFNE, IT'S >32K
        STA $FFD5 PAGE BIT= 1

PE2     ANDA MAXBNK,PCR GET BANK # OF PAGE
        BITA A,Y SWITCH TO IT
        EXG D,U GET DATA BACK
        RTS
*********************************
VDGHOK  TST DEVNUM TEST OUTPUT DEVICEI
        BEQ VFIX IFEQ, IS SCREEN (FIX IT)
RESUME  JMP $FFFF RE-HOOK OUTPUT ROUTINE

VFIX    PSHS A
        LDA PAGID,PCR WHAT IS PAGE #?
        BITA PAGMSK,PCR IS AN UPPER 32K?
        PULS A
        BEQ RESUME IFNE, NO FIX (LOWER 32K)
        PSHS X,B,A
        LDX #$FFC8 SAM VDG OFFSET ADDRESS
        STA $0B,X SET $8000 OFFSET BIT
        JMP $95B3 RE-ENTER ECB'S SCREEN ROUTINE
CODFIN  FDB $0000 WILL HAVE THIS ADDRESS HERE
        FCC 'END'

 LIST

ZEND EQU *-1
ZSIZ EQU CODFIN-CODE SIZE OF ACTUAL PAGER
```

END

```
61000 ' RAMDSKUT/BAS V2.0 ********************
61010 ' FOR BANKRDSK VERSION 2.00              *
61020 ' RAMDSKUT RAM DISK UTILITY SUB-ROUTINE
61030 REM  J & R ELECTRONICS
61040 REM - J.W. JACKSON
61050 REM  (C) 1985 ALL RIGHTS RESERVED
61060 REM BANKRDSK SHOULD HAVE BEEN LOADM'D AND EXEC'D FIRST
61070 REM PD = PDEST OF BANKRDSK
61080 REM RD$= DRIVE TOGGLE FLAG "A" OR "B"
61090 REM RD = RAM DRIVE ASSIGNMENT #
61100 REM MT = MAXIMUM TRACK NUMBER
61110 REM RF = READ FLAG, IF NON-ZERO READ DISK ELSE READ RAM
61120 REM DK = DSKCON ROUTINE
61130 REM  &HEA = OP CODE &HEB=DRIVE &HEC=TRACK &HED=SECTOR
61140 REM
61150 RD$="A"'START AT RAM DRIVE A
61160 CLS'*****************************************
61170 PD=&HFD00' RAM DRIVE PRGRAM START
61180 MB=PEEK(PD+08)' GET MAXIMUM BANK  #
61190 MX=PEEK(PD+09)' GET MAXIMUM DRIVE #
61200 RA=PEEK(PD+15)' GET NUMBER OF RAM DRIVE A
61210 RB=PEEK(PD+20)' GET NUMBER OF RAM DRIVE B
61220 IF RD$="A" THEN RD=RA ELSE RD=RB
61230 TA=PEEK(PD+16)' TRACKS A
61240 TB=PEEK(PD+21)' TRACKS B
61250 IF RD$="A" THEN MT=TA ELSE MT=TB
61260 RF=PD+12' READ FLAG LOCATION
61270 DK=PEEK(&HC004)*256+PEEK(&HC005)' DSKCON
61280 '************ MENU *******************************
61290 PRINT"  BANKRDSK UTILITY MENU  "
61300 PRINT"  *** RAM DRIVE "; RD$" *** "
61310 IF RD > MX THEN PRINT " drive is disabled "
61320 PRINT" 0. TOGGLE DRIVE "
61330 PRINT" 1. RAM DRIVE # "; RD
61340 PRINT" 2. MAXIMUM TRACK # "; MT
61350 PRINT" 3. LOAD RAM DRIVE FROM A DISK "
61360 PRINT" 4. RETURN          "
61370 PRINT: PRINT" SELECT AN OPTION (0-4) "; : GOSUB61710
61380 ZZ=ASC(ZZ$)-48: IF ZZ<0 OR ZZ>4 THEN 61160
61390 ON ZZ+1 GOTO 61410,61480,61530,61580,61700
61400 '*********** TOGGLE RAM DRIVE ******************
61410 IF MB>3 THEN 61450 ELSE PRINT" can't toggle drive to B"
61420 PRINT" must have 512K and MAXBNK = 7 "
61430 PRINT"    PRESS A KEY ........."; 
61440 GOSUB 61710: GOTO61460
61450 IF RD$="A"THEN RD$="B" ELSE RD$="A"
61460 GOTO61160
61470 '*********** SET RAM DRIVE #  ******************
61480 PRINT" SELECT RAM DRIVE # (0-3) "; : GOSUB61710
61490 ZZ=ASC(ZZ$)-48: IF ZZ<0 OR ZZ>3 THEN 61160
61500 RD=ZZ: IF RD$="A" THEN POKE PD+15,RD ELSE POKE PD+20,RD
61510 GOTO61160
61520 '*********** SET MAX TRACK #  ******************
61530 INPUT" SELECT MAXIMUM TRACK NUMBER (34-41) "; ZZ$
61540 ZZ=VAL(ZZ$): IF ZZ<0 OR ZZ>41 THEN 61160
61550 MT=ZZ: IF RD$="A" THEN POKE PD+16,MT ELSE POKE PD+21,MT
61560 GOTO61160
61570 '*********** LOAD RAM DRIVE   ******************
61580 PRINT" SELECT LOAD DRIVE # (0-3) "; : GOSUB61710
61590 ZZ=ASC(ZZ$)-48: IF ZZ<0 OR ZZ>MX THEN 61160
```

```
61610 FOR TK=0 TO MT:POKE &HEC,TK
61620 PRINT@380,""
61630 PRINTUSING"READING TRACK ##, ON DRIVE ##";TK;LD
61640 PRINTUSING"WRITING TRACK ##, ON RAM DRIVE ##";TK;RD
61650 FOR SE=1TO18:POKE &HED,SE
61660 POKE RF,255:POKE &HEA,2:POKE &HEB,LD:EXEC DK' READ
61670 POKE RF,0  :POKE &HEA,3:POKE &HEB,RD:EXEC DK'WRITE
61680 NEXT SE,TK:GOTO61160
61690 '********** RETURN FROM S.R. ********************
61700 RETURN
61710 ZZ$=INKEY$:IF ZZ$=""THEN61710ELSE PRINTZZ$:RETURN
61720 ' END RAMDSKUT
```

```
10 REM RDCUSTMZ/BAS V1.00
20 REM PROGRAM TO CUSTOMIZE "BANKRDSK/BIN"
30 REM  FROM MENU SELECTIONS.
40 REM  BY RICHARD FLATHMANN
50 '*****************************
60 'J&R PROVIDES THIS PROGRAM ON AN "AS IS" BASIS, AND MAKES
70 'NO CLAIMS AS TO ITS ACCURACY. J&R IS NOT RESPONSIBLE FOR
80 'ANY EFFECT THE USE OF THIS PROGRAM WILL HAVE ON YOUR  SYSTEM.
90 '*****************************
100 CLS
110 PRINT" ***********************************"
120 PRINT" *    bankrdsk mod utility      *"
130 PRINT" *              by              *"
140 PRINT" *      Richard Flathmann       *"
150 PRINT" ***********************************"
160 FOR I=0TO1000:NEXT
170 N$="N/A"
180 PRINT:PRINT:PRINT"    LOADING BANKRDSK ......."
190 LOADM"BANKRDSK"
200 CLS
210 PRINT"  1) 256K   or   2) 512K":GOSUB480:IF A=2 THEN POKE &H7008,7
220 PRINT:PRINT"  1) SLOW   or   2) FAST FORMAT":GOSUB480:IF A=2 THEN POKE &H6FF
3,255
230 PRINT:PRINT"  1) MENU   or   2) BYPASS":GOSUB480:IF A=2 THEN POKE &H6FF4,255
:POKE &H6FF5,255
240 IF PEEK(&H7008)=7 THEN PRINT:PRINT"LOAD RAMDSK (A) FROM DRIVE :";:GOSUB500:P
RINTA:POKE &H6FF6,A
250 IF PEEK(&H7008)=7 THEN PRINT"LOAD RAMDSK (B) FROM DRIVE :";:GOSUB500:PRINTA:
POKE &H6FF7,A
260 IF PEEK(&H7008)=7 THEN PRINT:PRINT"RAMDSK (A) = DRIVE :";:GOSUB500:PRINTA:PO
KE &H700F,A
270 IF PEEK(&H7008)=7 THEN PRINT"RAMDSK (B) = DRIVE :";:GOSUB500:PRINTA:POKE &H7
014,A
280 IF PEEK(&H7008)=7 THEN PRINT:PRINT"RAMDSK(A)= 1)35TRKS or 2)40TRKS";:GOSUB48
0:PRINTA:IF A=2 THEN POKE&H7010,39 ELSE POKE&H7010,34
290 IF PEEK(&H7008)=7 THEN PRINT"RAMDSK(B)= 1)35TRKS or 2)40TRKS";:GOSUB480:PRIN
TA:IF A=2 THEN POKE&H7015,39 ELSE POKE&H7015,34
300 CLS:IF PEEK(&H7008)=7 THEN PRINT"    512K" ELSE PRINT "    256K"
310 IF PEEK(&H6FF3)=255 THENPRINT"FAST"; ELSE PRINT "SLOW";
320 PRINT" INITIALIZE"
330 IF PEEK(&H6FF4)=255 THEN PRINT"AUTO"; ELSE PRINT "MENU";
340 PRINT" INITIALIZE"
350 A=PEEK(&H6FF6):PRINT"RAMDSK(A) LOADS FROM DRIVE :";:IF A=255 THEN PRINT N$ E
LSE PRINT A
360 A=PEEK(&H6FF7):PRINT"RAMDSK(B) LOADS FROM DRIVE :";:IF A=255 THEN PRINT N$ E
LSE PRINT A
370 A=PEEK(&H700F):PRINT"RAMDSK(A) REPLACES DRIVE :";A
380 A=PEEK(&H7014):PRINT"RAMDSK(B) REPLACES DRIVE :";:IF A=255 THEN PRINT N$ ELS
E PRINT A
390 A=PEEK(&H7010):PRINT"RAMDSK(A) = ";A+1;" TRACKS"
400 A=PEEK(&H7015):PRINT"RAMDSK(B) = ";A+1;" TRACKS"
410 PRINT@390,"SAVE ??? (Y/N)":GOSUB480:IF A$="N" THEN 450 ELSE IF A$="Y" THEN 4
20 ELSE 410
420 CLS:LINEINPUT"SAVE AS :";NM$:IF LEN(NM$)>8 THEN PRINT:PRINT"    name to long
 !!!":FORX=1TO300:NEXT:GOTO420
430 SAVEM NM$,&H6FF0,&H7AFF,&H6FF0
440 GOTO450
450 CLS:PRINT:PRINT"  RESETTING DEFAULT PARAMETERS":FORX=1TO900:NEXT
460 POKE &H7008,3:POKE&H6FF3,0:POKE&H6FF4,0:POKE&H6FF5,0:POKE&H6FF6,255:POKE&H6F
F7,255:POKE&H700F,0:POKE&H7010,34:POKE&H7014,255:POKE&H7015,34
```

```
480 A$=INKEY$:IF A$="" THEN 480 ELSE A=VAL(A$):IF A>2 THEN 480
490 RETURN
500 A$=INKEY$:IF A$="" THEN 500 ELSE A=VAL(A$):IF A>3 THEN 480
510 RETURN
520 'To use this utility proper-  ly, you must use an unmodified  copy of bankrd
sk. Always use a  backup copy when possible.
530 'Anyone interested in swap-  notes or ideas on the bankerII  please write to
 me.
540 '
550 'Richard  Flathmann                12 Dorchester Road                Ronkonkoma,
 NY   11779
560 REM RDCUSTMZ/BAS V1.00 *** END
```

```
     NAM BANKCOPY
* BANKER MEMORY UPGRADE
* V1.01
* V2.00 2/17/86 UPDATE TO 512K
*
*J & R ELECTRONICS
* J.W.JACKSON
* COPYRIGHT 1985 &1986
* ALL RIGHTS RESERVED
*
*************************************************
* THIS PROGRAM COPIES THE CONTENTS OF THE UPPER
* 32K ($8000 - $FEFF) OF ROM/RAM IN BANK 0
* TO THE UPPER 32K OF RAM IN BANKS 0-MAXBNK
* IF THE SYSTEM IS IN ROM MODE WHEN THE PROGRAM
* IS EXEC'D, THEN THE ROM WILL BE COPIED.
* IF THE SYSTEM IS IN RAM MODE WHEN THE PROGRAM
* IS EXEC'D, THE THE RAM CONTENTS WILL BE COPIED.
* THE "OK" PROMPT WILL BE CHANGED TO REPRESENT THE
* ACTIVE UPPER BANK ( I.E. ")0" )

       ORG $0E00
       BRA START
MAXBNK FCB $07 512K ( 3 FOR 256K VERSIONS )
MAPFLG FCB $00 IFEQ, IS ROM

START  ORCC #$50
       CLR MAPFLG,PCR SET ROM MODE FLAG
       LDX #$8000
       LDY #$FFC8 BANK )0 & (0
       LDB MAXBNK,PCR GET MAXIMIUM BANK #
       LDA $8000 GET DATA
       COMA COMPLEMENT IT
       STA $8000 AND STORE IT BACK
       CMPA $8000 CAN IT BE CHANGED?
       BNE COPY IS IN MAP 0
       COMA RESTORE TO ORIGINAL VALUE
       STA $8000 IN RAM
       COM MAPFLG,PCR SET RAM

COPY   PSHS B,X SAVE FOR LOOP
CPY2   TST MAPFLG,PCR ROM?
       BNE CPY1 SKIP ROM SWITCH
       STA $FFDE ROM MODE
CPY1   BITA 0,Y SWITCH TO BANK #00
       LDA 0,X GET DATA
       BITA B,Y  BANK )B & (0
       STA $FFDF RAM MODE
       STA 0,X+ STORE DATA

       CMPX #$FF00 END OF UPPER BANK?
       BNE CPY2
```

```
*NOW, CHANGE THE "OK" PROMPT
       LDX #$ABEE 'OK' PROMPT ADDRESS
       BITA B,Y  BANK )B (@
       STA $FFDF
       LDA #') UPPER BANK SYMBOL
       STA @,X
       ADDB #'@ MAKE ASCII '@-'9
       STB 1,X

       PULS B,X RESTORE BANK # & INDEX TO DATA
       DECB NEXT LOWER BANK #
       BGE COPY LAST IS #@
       RTS

       END
*** BANKCOPY/TXT *****
```

```
10 'PCOPYDEM.BAS V2.02
20 CLEAR 200,&H6FEF
30 GOTO60' ***********************************
40 TD=0' 0=DISK 1=TAPE
50 GOTO80' THIS STUFF DUE TO PCLEAR BUG ..
60 PCLEAR 8
70 GOTO40
80 CLS
90 PRINT " ***********************************"
100 PRINT " *    BANKER MEMORY EXPANSION   *"
110 PRINT " *      PCOPYMOR DEMO 2.02      *"
120 PRINT " *                              *"
130 PRINT " *      J & R ELECTRONICS       *"
140 PRINT " *      P.O.BOX 2572            *"
150 PRINT " *      COLUMBIA, MARYLAND      *"
160 PRINT " *                 21045        *"
170 PRINT " *                              *"
180 PRINT " *      J.W.JACKSON             *"
190 PRINT " *      (C) 1985                *"
200 PRINT " *      ALL RIGHTS RESERVED     *"
210 PRINT " ***********************************"
220 GOSUB 850 ' WAIT FOR KEY
230 CLS
240 PRINT" THIS DEMO LOADM'S PCOPYMOR
250 PRINT" AND EXECS IT,THEN LOADS
260 PRINT" 10 PMODE 4 PICTURES
270 PRINT" INTO BANKER MEMORY.
280 PRINT" WHEN THE (OK) PROMPT APPEARS
290 PRINT" PCOPYMOR WILL REMAIN PATCHED
300 PRINT" AND YOU MAY PCOPY 1 TO MORE
310 PRINT" THAN NORMAL EXTENDED BASIC.
320 PRINT" J&R HOPES YOU ENJOY IT!
330 GOSUB 850 ' WAIT FOR KEY
340 IF PEEK(&H9723)=&H7E THEN390
350 PRINT:IF TD=1 THEN PRINT"C";
360 PRINT"LOADM'ING PCOPYMOR "
370 IF TD=0 THEN LOADM"PCOPYMOR" ELSE CLOADM"PCOPYMOR"
380 EXEC '***********************************
390 CLS
400 PRINT" LOADING 10 PMODE 4 PICS "
410 PRINT"  INTO BANKER MEMORY      "
420 PRINT"  THIS USES 40 PAGES OF   "
430 PRINT"  THE MOR PROVIDED BY THE"
440 PRINT"  PCOPYMOR PATCH.         "
450 PMODE 4,1:SCREEN 1,0
460 PCLS'*******************************
470 P$="PIC"
480 IF PEEK(&HBC)=6 THEN OS=&HF800 ELSE OS=0
490 FOR I= 0 TO 9
500 N$=RIGHT$(STR$(I),1)
510 IF TD=0 THEN LOADM P$+N$ ELSE CLOADM P$+N$,OS
520 J=(I+1)*4+1
530 PCOPY 1 TO J
540 PCOPY 2 TO J+1
550 PCOPY 3 TO J+2
560 PCOPY 4 TO J+3
570 NEXT'*******************************
580 CLS
590 PRINT" 40 PAGES (61,444 BYTES) "
600 PRINT"  WILL BE TRANFERRED      "
```

```
620 PRINT"   BANKER USING THE PATCH "
630 PRINT"   PCOPYMOR PROGRAM.      "
640 PRINT"   SMOOTHER ANIMATION     "
650 PRINT"   COULD BE GAINED BY     "
660 PRINT"   PROGRAMMING THE BANKER "
670 PRINT"   VDG BANK AND THE BAM   "
680 PRINT"   OFFSET IN MACHINE CODE."
690 PRINT
700 PRINT"   PRESS F FOR (F)AST     "
710 PRINT"          S FOR (S)LOW     "
720 FOR ZZ=0TO5000'********************
730 ZZ$=INKEY$:IF ZZ$<>"" THEN 750
740 NEXT ZZ'***************************
750 PMODE 4,1:SCREEN 1,0
760 FOR I=5 TO 41  STEP 4
770 PCOPY I TO 1
780 PCOPY I+1 TO 2
790 PCOPY I+2 TO 3
800 PCOPY I+3 TO 4
810 IF ZZ$="F"THEN 830'***************
820 FORXX=0TO1000:NEXT XX
830 NEXT I
840 GOTO580
850 PRINT" PRESS A KEY ";
860 IF INKEY$=""THEN 860 ELSE RETURN
870 END' PCOPYDEM.BAS ****************
```

```
10 'BANKTEST.BAS V2.01
20 CLEAR 200,&H6FFF
30 LA=&H7800' LOAD ADDRESS
40 BE=LA + &H0C ' BANK ERROR ADDRESS
50 MX=07 ' MAX BANK 3 FOR 256K, 7 FOR 512K
60 TD=0 ' 0=DISK 1=TAPE
70 '*****************************************
80 CLS
90 PRINT " ***********************************"
100 PRINT " *      BANKER TEST V2.01        *"
110 PRINT " *      MEMORY BANK TEST         *"
120 PRINT " *                               *"
130 PRINT " *      J & R ELECTRONICS        *"
140 PRINT " *      P.O.BOX 2572             *"
150 PRINT " *      COLUMBIA, MARYLAND       *"
160 PRINT " *                  21045        *"
170 PRINT " *                               *"
180 PRINT " *       J.W.JACKSON             *"
190 PRINT " *          (C) 1985             *"
200 PRINT " *      ALL RIGHTS RESERVED      *"
210 PRINT " ***********************************"
220 ' *****************************************
230 IF TD > 0 THEN PRINT"C";
240 PRINT"LOADING MACHINE LANGUAGE SEGMENT"
250 IF TD=0 THEN LOADM"BANKTEST" ELSE CLOADM"BANKTEST"
260 GOSUB 760' WAIT FOR KEY
270 CLS
280 PRINT"      MENU SELECTIONS "
290 PRINT"1. SINGLE BANK
300 PRINT"2. AUTOMATIC
310 PRINT"3. QUIT
320 PRINT" YOUR CHOICE (1,2,3) ";
330 YC$=INKEY$:IF YC$=""THEN330
340 YC=VAL(YC$)
350 PRINT YC
360 IF YC<1 OR YC>3 THEN 270
370 ON YC GOSUB 400,510,390
380 GOTO260
390 END'*****************************************
400 CLS
410 PRINT"      BANK SELECTIONS "
420 PRINT"1. TEST UPPER BANK
430 PRINT"2. TEST LOWER BANK
440 PRINT" YOUR CHOICE (1,2) ";
450 YC$=INKEY$:IF YC$=""THEN450
460 YC=VAL(YC$)
470 PRINT YC
480 IF YC<1 OR YC>2 THEN 400
490 ON YC GOSUB 610,640
500 RETURN'*************************************
510 FOR BK = 0 TO MX' ALL BANKS
520 GOSUB 740' SET BANK
530 GOSUB 650' LOWER
540 FORZZ=0TO500:NEXT
550 IF PEEK(BE)=0 THEN GOSUB 760'ERROR?
560 GOSUB 620' UPPER
570 FORZZ=0TO500:NEXT
580 IF PEEK(BE)=0 THEN GOSUB 760'ERROR?
590 NEXT BK
600 RETURN'*************************************
```

```
620 POKE LA+&H13,255:EXEC LA'UP/LO FLAG
630 RETURN'**********************************
640 GOSUB670'GET BANK #
650 POKE LA+&H13,0:EXEC LA' UP/LO FLAG
660 RETURN'**********************************
670 PRINT: PRINT" SELECT BANK :":PRINT"    (";
680 FOR ZZ=0 TO MX-1:PRINTSTR$(ZZ)",";:NEXT ZZ
690 PRINTSTR$(MX)")";
700 BK$=INKEY$:IF BK$=""THEN700
710 BK=VAL(BK$)
720 PRINTBK
730 IF BK<0 OR BK>MX THEN 670
740 POKE LA+&H0B,BK
750 RETURN'**********************************
760 PRINT"PRESS A KEY";
770 IF INKEY$="" THEN 770 ELSE PRINT
780 RETURN'**********************************
790 END 'BANKTEST.BAS V2.01 ***************
```

```
^^^^^^^_^^!^ic^|^@^b^
```

```
10 'BANKRPAG/BAS V2.00
20 CLS
100 PRINT " ************************************"
110 PRINT " *          BANKER PAGE             *"
120 PRINT " *             V2.00                *"
130 PRINT " *     BANKER DISPLAY PAGER         *"
140 PRINT " *                                  *"
150 PRINT " *      J & R ELECTRONICS           *"
160 PRINT " *      P.O.BOX 2572                *"
170 PRINT " *      COLUMBIA, MARYLAND          *"
180 PRINT " *                 21045            *"
190 PRINT " *                                  *"
200 PRINT " *      J.W.JACKSON                 *"
210 PRINT " *      (C) 1985                    *"
220 PRINT " *      ALL RIGHTS RESERVED         *"
230 PRINT " ************************************"
240 PRINT" PRESS A KEY "
250 XX$=INKEY$:IF XX$=""THEN250
260 CLS:PCLEAR 4
270 MX=3' MAX BANK 256K=3 512K=7
280 SP=10'SPEED OF REPEAT KEYS
290 OD=2' OFFSET DATA FOR SAM
300 OA=&HFFC6' OFFSET ADDRESS
310 VL=&HFFD0' BANKER VIDEO LATCH FFD0-FFDF
320 S0=OD:S1=VL'SAVE OLD
330 REM MAIN PROGRAM LOOP ****************
340 CLS
350 PRINT"        BANKRPAG
360 PRINT"  UP ARROW MOVES PAGE UP ONE
370 PRINT"  DN ARROW MOVES PAGE DOWN ONE
380 PRINT"  RT ARROW MOVES BANK UP ONE
390 PRINT"  LT ARROW MOVES BANK DOWN ONE
400 PRINT"  CLR KEY FOR THIS TEXT SCREEN
410 PRINT"  SPACE BAR RETURNS AFTER CLR
420 PRINT"  0 KEY FOR PMODE 0 DISPLAY
430 PRINT"  1 KEY FOR PMODE 1 DISPLAY
440 PRINT"  2 KEY FOR PMODE 2 DISPLAY
450 PRINT"  3 KEY FOR PMODE 2 DISPLAY
460 PRINT"  4 KEY FOR PMODE 4 DISPLAY
470 PRINT"  5 KEY FOR TEXT DISPLAY
480 PRINT"* BANK ";S1-&HFFD0" *":PRINT"* PAGE OFFSET ";S0*512;"/ &H"HEX$(S0*512)
" *"
490 REM PAGING MODE ****************************
500 GOSUB820' REPEAT KEY ROUTINE
510 KB$=INKEY$:IF KB$=""THEN510
520 REM GET DISPLAY OFFSET ********************
530 IF KB$=CHR$(12) THEN S0=OD:S1=VL:OD=2:VL=&HFFD0:SCREEN 0,0' CLEAR KEY
540 IF KB$<>CHR$(32) THEN 570 ' SPACE BAR ?
550 OD=S0:VL=S1 ' SPACE BAR
560 IF PM+1 >0 THEN PMODE PM,1:SCREEN 1,0
570 IF KB$=CHR$(94) THEN OD=OD +1'UP ARROW
580 IF KB$=CHR$( 9) THEN VL=VL +1'RT ARROW
590 IF KB$=CHR$(10) THEN OD=OD -1'DN ARROW
600 IF KB$=CHR$( 8) THEN VL=VL -1'LT ARROW
610 IF (KB$)"4" AND KB$<="9") THEN SCREEN 0,0:PM=-1
620 IF KB$= "4" THEN PMODE 4,1:SCREEN 1,0:PM=4
630 IF KB$= "3" THEN PMODE 3,1:SCREEN 1,0:PM=3
```

```
650 IF KB$= "1" THEN PMODE 1,1:SCREEN 1,0:PM=1
660 IF KB$= "0" THEN PMODE 0,1:SCREEN 1,0:PM=0
670 IF OD>127 THEN OD=0:VL=VL+1
680 IF OD < 0 THEN OD=127:VL=VL-1
690 IF VL>(&HFFD0+MX) THEN VL=&HFFD0
700 IF VL<&HFFD0 THEN VL=&HFFD0+MX
710 ZZ=PEEK(VL)' SET VID LATCH FOR BANK
720 GOSUB 760' SET SAM FOR PAGE
730 IF KB$=CHR$(12)THEN 330
740 GOTO500
750 END' MAIN PROGRAM LOOP ****************************************
760 REM' SET SAM DISPLAY OFFSET
770 D0=1' FIRST BIT
780 FOR SA= OA TO OA+12 STEP 2
790 IF (OD AND D0) =0 THEN POKE SA,0 ELSE POKE SA+1,0
800 D0=2*D0' NEXT BIT
810 NEXT SA
820 REM REPEAT KEY ROUTINE
830 FOR I=0TO SP :NEXT I'DELAY
840 FOR I=&H0152 TO &H0159:POKE I,&HFF:NEXT
850 RETURN' REPEAT KEY
860 END' BANKRPAG/BAS
```