```
        NAME    DEMOMAIN
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  WORK_1, WORK_2, WORK_3, FLAG, FLAG_2
        PUBLIC  WORK_4, WORK_5, WORK_6, WORK_7, AGDC_SEG
        ;
        EXTRN   ASYNC:NEAR, PUSH_EXE:NEAR, POP_EXE:NEAR, DEMO_INIT:NEAR
        EXTRN   DMCLO_C_WAIT:NEAR, DMCL1_C_WAIT:NEAR, DMCLO_C_FREE:NEAR
        EXTRN   MES_CL_WAIT:NEAR
        EXTRN   RECFILL_DEMO:NEAR, CRLFILL_DEMO:NEAR
        EXTRN   ELPSFILL_DEMO:NEAR, TRIFILL_DEMO:NEAR, TRAFILL_DEMO:NEAR
        EXTRN   LINE_DEMO:NEAR, CRL_DEMO:NEAR, ELPS_DEMO:NEAR
        EXTRN   CHR_DRAW_DEMO_L:NEAR, CHR_DRAW_DEMO_H:NEAR
        EXTRN   CHR_DRAW_DEMO_L_90:NEAR, CHR_DRAW_DEMO_H_90:NEAR
        EXTRN   MESSAGE:NEAR, LINE_DEMO_H:NEAR, ENLARGE_DEMO:NEAR
        EXTRN   WAIT:NEAR, MESSAGE_WAIT:NEAR, GET_PUT_DEMO:NEAR
        EXTRN   PUT_DATA_TRAN:NEAR, SHRINK_DEMO:NEAR, SHRINK_PUT:NEAR
        EXTRN   SHRINK_DATA:NEAR, SHRINK_EXE:NEAR, PAINT_DEMO:NEAR
        EXTRN   FRCOPY_DEMO:NEAR, RECT_DEMO:NEAR, SCROLL_DEMO:NEAR
        EXTRN   CLIP_DEMO:NEAR, GLIOMAIN:NEAR
        ;
DEMOMAIN        PROC    NEAR
;
        ;
        ORG     100H
        ;
;
;
;DATA BUFFER FIELD
;
WORK_1  DW      0                               ;0100H
WORK_2  DW      0                               ;0102H
WORK_3  DW      0                               ;0104H
FLAG    DB      0                               ;0106H
                                                ;BIT0 : REAL TIME/WAIT
                                                ;BIT1 : SLANT TEXT/
                                                ;BIT2 : PIXEL MODE/
FLAG_2  DB      0                               ;0107H
WORK_4  DW      0                               ;0108H
WORK_5  DW      0                               ;010AH
WORK_6  DW      0                               ;010CH
WORK_7  DW      0                               ;010EH
AGDC_SEG        DW      0                       ;0110H
        ;
;
;JUMP TABLE
;
;
        ;
```



```
E1:     JMP NEAR PTR ASYNC                      ;0112H
E2:     JMP NEAR PTR DEMO_EXE                   ;0115H
E3:     JMP NEAR PTR DEMO_INIT                  ;0118H
E4:     JMP NEAR PTR GLIOMAIN                   ;011BH
        ;
;
;
;       < DEMONSTRATION >
;
;
        ;
DEMO_EXE:
        PUSH    AX
        CALL    PUSH_EXE
        MOV     AL,1
        OUT     0D1H,AL                         ;ENABLE /CSIR/
        IN      AL,0D1H
        TEST    AL,1
        MOV     AX,0D000H
        JZ      DEMO_EXE_1                      ;PC-XA/XL -- 8000H
                                                ;PC9800   -- D000H
        MOV     AX,8000H
DEMO_EXE_1:
        MOV     DS,AX                           ;OPEN MEMORY WINDOW
        MOV     CS:WORD PTR AGDC_SEG,AX
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    DMCLO_C_WAIT
        ;
        CALL    DEMO_THRU
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,0
        CALL    MES_CL_WAIT                     ;MESSAGE(0)
        MOV     SI,32
        CALL    MES_CL_WAIT                     ;MESSAGE(32)
        ;
;LINE, RECTANGLE, CIRCLE, ELLIPSE
        ;
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    VECT_DEMO
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        CALL    VECT_DEMO
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    VECT_DEMO
        ;
;ENLARGE
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,16
        CALL    MES_CL_WAIT                     ;MESSAGE(16)
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
```



```
        CALL    LINE_DEMO_H
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        CALL    ENLARGE_DEMO
        MOV     SI,17
             CALL       MES_CL_WAIT                     ;MESSAGE(17)
        ;
;SHRINK
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,20
        CALL    MES_CL_WAIT                     ;MESSAGE(20)
        CALL    SHRINK_DEMO
        MOV     SI,22
        CALL    MES_CL_WAIT                     ;MESSAGE(22)
        ;
;GET, PUT
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,18
        CALL    MES_CL_WAIT                     ;MESSAGE(18)
        CALL    GET_PUT_DEMO
        MOV     SI,19
        CALL    MES_CL_WAIT                     ;MESSAGE(19)
        ;
;FR_COPY
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,26
        CALL    MES_CL_WAIT                     ;MESSAGE(26)
        CALL    FRCOPY_DEMO
        MOV     SI,27
        CALL    MES_CL_WAIT                     ;MESSAGE(27)
        ;
;PAINT
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,23
        CALL    MES_CL_WAIT                     ;MESSAGE(23)
        CALL    PAINT_DEMO
        MOV     SI,24
        CALL    MES_CL_WAIT                     ;MEESAGE(24)
        ;
;CHARACTER
        ;
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    CHR_DRAW_DEMO
        MOV     CS:BYTE PTR FLAG,2              ;REAL TIME MODE (SLANT)
        CALL    CHR_DRAW_DEMO
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    CHR_DRAW_DEMO_90
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
```



```
        MOV     SI,13
        CALL    MES_CL_WAIT                     ;MESSAGE(13)
        CALL    CHR_DRAW_DEMO
        MOV     CS:BYTE PTR FLAG,3              ;WAIT MODE (SLANT)
        MOV     SI,25
        CALL    MES_CL_WAIT                     ;MESSAGE(25)
        CALL    CHR_DRAW_DEMO
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,44
        CALL    MES_CL_WAIT                     ;MESSAGE(44)
        CALL    CHR_DRAW_DEMO_90
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    CHR_DRAW_DEMO
        MOV     CS:BYTE PTR FLAG,2              ;REAL TIME MODE (SLANT)
        CALL    CHR_DRAW_DEMO
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    CHR_DRAW_DEMO_90
        ;
;SCROLL
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,31
        CALL    MES_CL_WAIT                     ;MESSAGE(31)
        CALL    SCROLL_DEMO
        CALL    DMCLO_C_WAIT
        ;
;RECFILL, CRLFILL, ELPSFILL, TRIFILL, TRAFILL
        ;
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    FILL_DEMO
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        CALL    FILL_DEMO
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    FILL_DEMO
        ;
;CLIPPING
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,33
        CALL    MES_CL_WAIT                     ;MESSAGE(33)
        CALL    CLIP_DEMO
        CALL    DMCLO_C_WAIT
        ;
        CALL    DEMO_THRU
        ;
;DRAWING SPEED COMPARISON UNDER COMMON N88BASIC (PC-XL VS. AGDC)
        ;
        MOV     CS:BYTE PTR FLAG,1              ;WAIT MODE
        MOV     SI,42
        CALL    MES_CL_WAIT                     ;MESSAGE(42)
        MOV     SI,43
```

```
        CALL    MES_CL_WAIT                     ;MESSAGE(43)
        JMP     POP_EXE
        ;
;
VECT_DEMO:
        CALL    LINE_DEMO
        CALL    RECT_DEMO
        CALL    DMCLO_C_WAIT
        CALL    CRL_DEMO
        CALL    ELPS_DEMO
        RET
        ;
CHR_DRAW_DEMO:
        CALL    CHR_DRAW_DEMO_L
        MOV     SI,14
        CALL    MES_CL_WAIT                     ;MESSAGE(14)
        CALL    CHR_DRAW_DEMO_H
        JMP     CHR_DRAW_DEMO_90_1
        ;
CHR_DRAW_DEMO_90:
        CALL      CHR_DRAW_DEMO_L_90
        MOV     SI,14
        CALL    MES_CL_WAIT                     ;MESSAGE(14)
        CALL    CHR_DRAW_DEMO_H_90
CHR_DRAW_DEMO_90_1:
        MOV     AX,0
        MOV     BX,29
        MOV     CX,1119
        MOV     DX,-29
        CALL    DMCLO_C_FREE
        MOV     SI,15
        CALL    MES_CL_WAIT                     ;MESSAGE(15)
        RET
        ;
FILL_DEMO:
        CALL    RECFILL_DEMO
        CALL    CRLFILL_DEMO
        CALL    ELPSFILL_DEMO
        CALL    TRIFILL_DEMO
        CALL    TRAFILL_DEMO
        RET
        ;
DEMO_THRU:
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    VECT_DEMO
        CALL    LINE_DEMO_H
        CALL    ENLARGE_DEMO
        CALL    DMCLO_C_WAIT
        CALL    SHRINK_DEMO
        CALL    DMCLO_C_WAIT
        CALL    GET_PUT_DEMO
```

-5-

```
        CALL    DMCLO_C_WAIT
        CALL    FRCOPY_DEMO
        CALL    DMCLO_C_WAIT
        CALL    PAINT_DEMO
        CALL    DMCLO_C_WAIT
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    CHR_DRAW_DEMO
        MOV     CS:BYTE PTR FLAG,2              ;REAL TIME MODE (SLANT)
        CALL    CHR_DRAW_DEMO
        MOV     CS:BYTE PTR FLAG,0              ;REAL TIME MODE
        CALL    CHR_DRAW_DEMO_90
        CALL    SCROLL_DEMO
        CALL    DMCLO_C_WAIT
        CALL    FILL_DEMO
        CALL    CLIP_DEMO
        CALL    DMCLO_C_WAIT
        RET
;
DEMOMAIN        ENDP
PROG    ENDS
        END
```

-6-

```
        NAME    EQUATE_DEMO
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  EADORGL, EADORGH, dADORG, EAD1L, EAD1H, dAD1
        PUBLIC  EAD2L, EAD2H, dAD2, PDISPSL, PDISPSH, PDISPDL, PDISPDH
        PUBLIC  PMAX, MOD10, PTNPL, PTNPH, STACKL, STACKH, BANK, CTRL
        PUBLIC  STATUS, IRR, X, Y, DXX, DY, XS, YS, XE, YE, XC, YC
        PUBLIC  DHH, DV, PITCHS, PITCHD, STMAX, PLANES, PTNCNT
        PUBLIC  XCLMIN, YCLMIN, XCLMAX, YCLMAX, MAGETC, COM
        PUBLIC  DISP_FLAGS, DISP_PITCH, DADL, DADH_WC
        PUBLIC  GCSRX, GCSRYS, GCSRYE, SYNC
        PUBLIC  WINDOW_SEG_XL, WINDOW_SEG_98
        ;
EQUATE_DEMO     PROC    NEAR
        ;
;
;
;AGDC ADDRESS TABLE
;
        ;
EADORGL EQU      0H
EADORGH EQU      2H
dADORG  EQU      3H
EAD1L   EQU      4H
EAD1H   EQU      6H
dAD1    EQU      7H
EAD2L   EQU      8H
EAD2H   EQU      0AH
dAD2    EQU      0BH
PDISPSL EQU      0CH
PDISPSH EQU      0EH
PDISPDL EQU      10H
PDISPDH EQU      12H
PMAX    EQU      14H
MOD10   EQU      16H
PTNPL   EQU      18H
PTNPH   EQU      1AH
STACKL  EQU      1CH
STACKH  EQU      1EH
        ;
BANK    EQU      3CH
CTRL    EQU      3DH
        ;
STATUS  EQU      3CH
IRR     EQU      3EH
X       EQU      40H
Y       EQU      42H
DXX     EQU      44H
DY      EQU      46H
```

-1-

```
XS      EQU      48H
YS      EQU      4AH
XE      EQU      4CH
YE      EQU      4EH
XC      EQU      50H
YC      EQU      52H
DHH     EQU      54H
DV      EQU      56H
PITCHS  EQU      58H
PITCHD  EQU      5AH
STMAX   EQU      5CH
PLANES  EQU      5EH
PTNCNT  EQU      60H
XCLMIN  EQU      62H
YCLMIN  EQU      64H
XCLMAX  EQU      66H
YCLMAX  EQU      68H
        ;
MAGETC  EQU      6CH
COM     EQU      6EH
DISP_FLAGS      EQU      70H
DISP_PITCH      EQU      72H
DADL    EQU      74H
DADH_WC EQU      76H
GCSRX   EQU      78H
GCSRYS  EQU      7AH
GCSRYE  EQU      7CH
SYNC    EQU      7EH
        ;
;       MISCELLANEOUS EQUATE
        ;
WINDOW_SEG_XL   EQU      8000H
WINDOW_SEG_98   EQU      0D000H
        ;
;
EQUATE_DEMO     ENDP
PROG    ENDS
        END
```

-2-

```
        NAME    DEMO_INIT
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  DEMO_INIT
        ;
        EXTRN   PUSH_EXE:NEAR, POP_EXE:NEAR, DMCL1_C:NEAR
        ;
        EXTRN   CTRL:BYTE, AGDC_SEG:WORD
        EXTRN   BANK:BYTE, DISP_FLAGS:WORD, SYNC:WORD, DISP_PITCH:WORD
        EXTRN   DADL:WORD, GCSRX:WORD, GCSRYS:WORD, GCSRYE:WORD
        EXTRN   EADORGL:WORD, EADORGH:WORD, DADORG:WORD, DADH_WC:WORD
        EXTRN   PDISPSL:WORD, PDISPSH:WORD, PDISPDL:WORD, PDISPDH:WORD
        EXTRN   PMAX:WORD, STACKL:WORD, STACKH:WORD, STMAX:WORD
        EXTRN   PITCHS:WORD, PITCHD:WORD
        EXTRN   XCLMIN:WORD, YCLMIN:WORD, XCLMAX:WORD, YCLMAX:WORD
        EXTRN   MAGETC:WORD, STATUS:WORD
        ;
DEMO_INIT       PROC    NEAR
        ;
;
;       < INITIALIZE AGDC, DISPLAY MEMORY, AND WORKING AREA >
;
;
        ;
        PUSH    AX
        CALL    PUSH_EXE
        MOV     AL,1
        OUT     0D1H,AL                         ;ENABLE /CSIR/
        IN      AL,0D1H
        TEST    AL,1
        MOV     AX,0D000H
        JZ      MOD_9801                        ;CHECK IF 98XA/9801
        MOV     AX,8000H
MOD_9801:
        MOV     DS,AX                           ;OPEN MEMORY WINDOW
                                                ;98XA/XL -- 8000H
                                                ;9801XX  -- D000H
        MOV     CS:WORD PTR AGDC_SEG,AX
        ;
        CALL    INIT_AGDC
        CALL    DMCL1_C
        CALL    TILE_TRAN
        CALL    SAD_GEN
        JMP     POP_EXE
        ;
;
;       < MESSAGE PAGE ADDRESS TABLE GENERATION >
;
        ;
```



```
        MOV     DS:WORD PTR EADORGH,0           ;EADORGH=0
        MOV     DS:WORD PTR DADORG,0            ;DADORG=0
        MOV     DS:WORD PTR PDISPSL,0           ;PDISPSL=0
        MOV     DS:WORD PTR PDISPSH,1           ;PDISPSH=1
        MOV     DS:WORD PTR PDISPDL,0           ;PDISPDL=0
        MOV     DS:WORD PTR PDISPDH,1           ;PDISPDH=1
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR STACKL,0FFFFH       ;STACKL=0FFFFH
        MOV     DS:WORD PTR STACKH,2            ;STACKH=2
        MOV     DS:WORD PTR STMAX,7FFH          ;STMAX=7FFH
        MOV     DS:WORD PTR PITCHS,46H          ;PITCHS=46H
        MOV     DS:WORD PTR PITCHD,46H          ;PITCHD=46H
        MOV     DS:WORD PTR XCLMIN,0            ;XCLMIN=0
        MOV     DS:WORD PTR YCLMIN,0            ;YCLMIN=0
        MOV     DS:WORD PTR XCLMAX,1119         ;XCLMAX=1119
        MOV     DS:WORD PTR YCLMAX,749          ;YCLMAX=749
        MOV     DS:WORD PTR MAGETC,1FFH         ;MAGV=15,MAGH=15
                                                ;NON-CLIP
        RET
        ;
;
;       < TILING PATTERN TRANSFER >
;
;
        ;
TILE_TRAN:
        TEST    DS:WORD PTR STATUS,3
        JNZ     TILE_TRAN                       ;CHECK IF PPBUSY=1/0
                                                ;CHECK IF DPBUSY=1/0
        PUSH    DS
        MOV     AX,CS
        MOV     DS,AX
        MOV     SI,OFFSET TILING_DATA
        MOV     AX,WORD PTR AGDC_SEG
        MOV     ES,AX
        MOV     DI,0F000H
        MOV     CX,300H                         ;TILING PATTERN LENGTH
        MOV     AL,2
        OUT     0D1H,AL                         ;ENABLE *CSDM*
        REPZ    MOVSW
        MOV     AL,1
        OUT     0D1H,AL                         ;ENABLE *CSIR*
        POP     DS
        RET
        ;
;
;
;       < TILING DATA >
;
        ;
TILING_DATA:
```



```
SAD_GEN:
        PUSH    DS
        MOV     AX,54F0H                        ;PAGE ADDRESS SEGMENT
        MOV     DS,AX
        MOV     AX,5500H                        ;MESSAGE SEGMENT
        MOV     ES,AX
        MOV     SI,0
        MOV     DI,0
SAD_GEN_2:
        MOV     AL,0CH
SAD_GEN_1:
        SCASB
        JNZ     SAD_GEN_1                       ;CHECK IF "PAGE MARK"
        INC     DI
        MOV     [SI],DI                         ;STORE PAGE ADDRESS
        INC     SI
        INC     SI
        MOV     AL,25H
        SCASB
        JNZ     SAD_GEN_2                       ;CHECK IF "END MARK '%'"
        POP     DS
        RET
        ;
;
;       < INITIALIZE AGDC >
;
;
        ;
INIT_AGDC:
        MOV     DS:BYTE PTR CTRL,3              ;CTRL=3  SOFTWARE RESET
                                                ;       PROCESSOR ABORT
        MOV     DS:BYTE PTR BANK,1              ;BANK=1
        MOV     DS:WORD PTR DISP_FLAGS,8630H    ;DISP_FLAGS=8630H
        MOV     DS:WORD PTR DISP_FLAGS,8632H    ;DISP_FLAGS=8632H
        MOV     DS:WORD PTR SYNC,4              ; HS=4
        MOV     DS:WORD PTR SYNC,8              ;HBP=8
        MOV     DS:WORD PTR SYNC,22H            ; HH=22H
        MOV     DS:WORD PTR SYNC,45H            ; HD=45H
        MOV     DS:WORD PTR SYNC,6              ;HFP=6
        MOV     DS:WORD PTR SYNC,5              ; VS=5
        MOV     DS:WORD PTR SYNC,16H            ;VBP=16H
        MOV     DS:WORD PTR SYNC,177H           ; LF=177H
        MOV     DS:WORD PTR SYNC,09H            ;VFP=9H
        MOV     DS:WORD PTR DISP_FLAGS,8630H    ;DISP_FLAGS=8630H
        MOV     DS:WORD PTR DISP_PITCH,46H      ;DISP_PITCH=46H
        MOV     DS:WORD PTR DADL,0              ;DADL=0
        MOV     DS:WORD PTR DADH_WC,4500H       ;WC=45H,DADH=0
        MOV     DS:WORD PTR GCSRX,0C020H        ;GCSRX=C020H
        MOV     DS:WORD PTR GCSRYS,100H         ;GCSRYS=100H
        MOV     DS:WORD PTR GCSRYE,110H         ;GCSRYE=110H
        MOV     DS:WORD PTR EADORGL,0CCCEH      ;EADORGL=0CCCEH
```



```
        DW       7EEFH, 34A5H, 1C63H, 0C21H     ;TILE(0)-R
        DW      0FFFFH, 6DEFH, 34A5H, 1C63H     ;
        DW       0C21H,0FFFFH, 6DEFH, 34A5H     ;
        DW       1C63H, 0C21H,0FFFFH,0EF7BH     ;
        DW       0888H, 0000H, 0000H, 0401H     ;TILE(0)-G
        DW       20A2H, 0000H, 1080H, 0400H     ;
        DW       0020H, 0445H, 0000H, 1400H     ;
        DW       0020H, 0001H, 2208H, 0410H     ;
        DW       28AAH, 0401H, 0020H, 0401H     ;TILE(0)-B
        DW      0AAAAH, 0401H, 10A0H, 0401H     ;
        DW       0020H, 5555H, 0020H, 1401H     ;
        DW       0020H, 0401H,0AAAAH, 0411H     ;
        ;
        DW       1084H, 0842H, 0000H, 0000H     ;TILE(1)-R
        DW       0000H, 294AH, 1004H, 0842H     ;
        DW       0000H, 0000H, 294AH, 1084H     ;
        DW       0842H, 0000H, 0000H, 294AH     ;
        DW       6D6BH, 30A5H, 1C63H, 0C21H     ;TILE(1)-G
        DW      0FBFFH, 54A5H, 20A1H, 1421H     ;
        DW       0C21H,0FBFFH, 54A5H, 2421H     ;
        DW       1021H, 0C21H,0FBFFH, 14A5H     ;
        DW       0000H, 0000H, 0000H, 0000H     ;TILE(1)-B
        DW       0000H, 0000H, 0000H, 0001H     ;
        DW       0000H, 0000H, 0000H, 0000H     ;
        DW       0000H, 0000H, 0000H, 0000H     ;
        ;
        DW      0BBBBH,0FFFFH,0EEEEH,0FFFFH     ;TILE(2)-R
        DW      0BBBBH,0FFFFH,0EEEEH,0FFFFH     ;
        DW      0BBBBH,0FFFFH,0EEEEH,0FFFFH     ;
        DW      0BBBBH,0FFFFH,0EEEEH,0FFFFH     ;
        DW       1111H, 0000H, 4444H, 0000H     ;TILE(2)-G
        DW       1111H, 0000H, 4444H, 0000H     ;
        DW       1111H, 0000H, 4444H, 0000H     ;
        DW       1111H, 0000H, 4444H, 0000H     ;
        DW       4444H, 0000H, 1111H, 0000H     ;TILE(2)-B
        DW       4444H, 0000H, 1111H, 0000H     ;
        DW       4444H, 0000H, 1111H, 0000H     ;
        DW       4444H, 0000H, 1111H, 0000H     ;
        ;
        DW      0DDDDH, 5555H, 7777H, 5555H     ;TILE(3)-R
        DW      0DDDDH, 57D5H, 77F7H, 57D5H     ;
        DW      0D7DDH, 53D5H, 7077H, 5555H     ;
        DW      0DDDDH, 5555H, 7777H, 5555H     ;
        DW       2222H, 8888H, 8888H, 2222H     ;TILE(3)-G
        DW       2222H, 8808H, 8008H, 2022H     ;
        DW       2022H, 8008H, 8088H, 2222H     ;
        DW        2222H, 8888H, 8888H, 2222H    ;
        DW      0AAAAH,0FFFFH,0AAAAH,0FFFFH     ;TILE(3)-B
        DW      0AAAAH,0FFFFH,0A7EAH,0F7FFH     ;
        DW      0A7EAH,0F3FFH,0A0AAH,0FFFFH     ;
        DW      0AAAAH,0FFFFH,0AAAAH,0FFFFH     ;
```

```
        ;
        DW      4149H, 8622H, 0C22H, 8400H      ;TILE(4)-R
        DW      0CA11H, 556EH, 90B1H, 0802H     ;
        DW      0800H,0A755H, 69E2H, 3C02H      ;
        DW      8442H, 0801H, 5BEBH, 4629H      ;
        DW      1C6EH,0AA96H, 5509H, 2C21H      ;TILE(4)-G
        DW      6565H, 00CBH,0B481H, 4528H      ;
        DW      20A5H, 74EAH, 2C86H, 14E3H      ;
        DW      5909H, 2429H, 061CH,0E510H      ;
        DW      20D8H, 9603H, 0420H, 0400H      ;TILE(4)-B
        DW      0D28BH, 2C4AH, 90A5H, 1041H     ;
        DW      0400H, 2D40H, 28ABH, 9606H      ;
        DW      0020H, 0081H, 0A2AH, 4C43H      ;
        ;
        DW      3118H, 1008H, 0700H, 0300H      ;TILE(5)-R
        DW      0C30DH,0C003H, 4001H, 4270H     ;
        DW      1718H, 0380H,0C100H, 4000H      ;
        DW      0061H, 84F1H,0C639H, 6310H      ;
        DW      7FFFH, 1008H,0FFFFH, 0300H      ;TILE(5)-G
        DW      0F7FFH,0C003H,0FFF7H, 4270H     ;
        DW      0FFFFH, 0380H,0FFDFH, 4000H     ;
        DW      0F7FFH, 84F1H,0FFFFH, 6310H     ;
        DW      3B8AH,0FFFFH,0AFAAH,0EFFFH      ;TILE(5)-B
        DW      0E3AFH,0FFEFH,0EAA3H,0FFFFH     ;
        DW      08FBAH,0FF8FH,0EB8AH,0EFFFH     ;
        DW      0A2EBH,0FFFFH,0EE8BH,0FFFEH     ;
        ;
        DW      0AAAAH,0FFFFH,0BBBBH,0FFFFH     ;TILE(6)-R
        DW      0AAAAH,0FFFFH,0BBBBH,0FFFFH     ;
        DW      0AAAAH,0FFFFH,0BBBBH,0FFFFH     ;
        DW      0AAAAH,0FFFFH,0BBBBH,0FFFFH     ;
        DW      5555H, 0000H, 4444H, 0000H      ;TILE(6)-G
        DW      5555H, 0000H, 4444H, 0000H      ;
        DW      5555H, 0000H, 4444H, 0000H      ;
        DW      5555H, 0000H, 4444H, 0000H      ;
        DW      5555H, 0000H, 5555H, 0000H      ;TILE(6)-B
        DW      5555H, 0000H, 5555H, 0000H      ;
        DW      5555H, 0000H, 5555H, 0000H      ;
        DW      5555H, 0000H, 5555H, 0000H      ;
        ;
        DW      0000H, 0000H, 0000H, 0200H      ;TILE(7)-R
        DW      0600H, 0700H, 0700H, 0784H      ;
        DW      06E4H, 07F8H, 03A0H, 01E0H      ;
        DW      0000H, 0000H, 0000H, 0000H      ;
        DW      112AH, 2240H, 48C8H, 9311H      ;TILE(7)-G
        DW      2222H, 4244H, 8088H, 1A19H      ;
        DW      2102H, 6084H, 8050H, 9411H      ;
        DW      2202H, 0644H, 8C0DH, 1012H      ;
        DW      0000H, 0000H, 0000H, 0000H      ;TILE(7)-B
        DW      0400H, 0100H, 0500H, 0084H      ;
        DW      0444H, 0228H, 0100H, 00A0H      ;
```



```
        DW      0000H, 0000H, 0000H, 0000H      ;
        ;
        DW      0000H, 0000H, 0000H, 0000H      ;TILE(8)-R
        DW      0000H, 0000H, 0000H, 0000H      ;
        DW      0000H, 0000H, 0000H, 0000H      ;
        DW      0000H, 0000H, 0000H, 0000H      ;
        DW      8220H, 559DH,0A228H, 1095H      ;TILE(8)-G
        DW      0000H, 8431H, 2220H,0D79DH      ;
        DW      2208H, 0000H, 8220H, 579DH      ;
        DW      0A228H, 1091H, 0000H, 4210H     ;
        DW      598CH, 2242H, 08C0H, 0842H      ;TILE(8)-B
        DW      0B5ADH, 5946H, 1088H, 2842H     ;
        DW      0842H,0B4ADH, 5942H, 2042H      ;
        DW      0800H, 0842H,0B5ADH, 294AH      ;
        ;
        DW      0FFFFH,0FFFFH,0FFFFH,0FFFFH     ;TILE(9)-R
        DW      0FFFFH,0FFFFH,0FFFFH,0FFFFH     ;
        DW      0FFFFH,0FFFFH,0FFFFH,0FFFFH     ;
        DW      0FFFFH,0FFFFH,0FFFFH,0FFFFH     ;
        DW      7777H,0AAAAH,0DDDDH,0AAAAH      ;TILE(9)-G
        DW      7777H,0AAAAH,0DDDDH,0AAAAH      ;
        DW      7777H,0AAAAH,0DDDDH,0AAAAH      ;
        DW      7777H,0AAAAH,0DDDDH,0AAAAH      ;
        DW      0DDDDH, 0000H, 5555H, 0000H             ;TILE(9)-B
        DW      0DDDDH, 0000H, 5555H, 0000H     ;
        DW      0DDDDH, 0000H, 5555H, 0000H     ;
        DW      0DDDDH, 0000H, 5555H, 0000H     ;
        ;
        DW      0F914H, 677CH,0E6F6H, 68EFH     ;TILE(A)-R
        DW      0E5A5H,0F6E3H,0C86EH, 3AADH             ;
        DW      0B472H, 0D2FH, 081EH,0FE29H     ;
        DW      0A9BBH,0F77EH, 3396H, 0D23H     ;
        DW      0FFF7H,0E75FH,0F6FEH,0E8E7H     ;TILE(A)-G
        DW      0E5F3H, 7DEBH,0FFEFH, 7FFFH     ;
        DW      0FB32H, 3D1FH, 0E8EH,0FF5BH     ;
        DW      0FDBFH, 75FFH,0FFFEH, 7DEBH     ;
        DW      0B800H, 4574H,0A6A2H, 40C5H     ;TILE(A)-B
        DW      0A0A0H,0F641H, 886AH, 308DH     ;
        DW      0B422H, 050DH, 081EH, 5621H     ;
        DW      0A8AAH,0D756H, 2296H, 0501H     ;
        ;
        DW      4000H, 8008H, 0000H, 0000H      ;TILE(B)-R
        DW      0000H, 0000H, 0000H, 2020H      ;
        DW      0000H, 0280H, 0000H, 0200H      ;
        DW      0000H, 0022H, 0000H, 0000H      ;
        DW      0EAAAH,0D559H,0AAA8H, 5575H     ;TILE(B)-G
        DW      2AAAH, 1575H,0AAA2H, 7535H      ;
        DW      0AAA2H, 57C5H,0BE8AH, 5615H     ;
        DW      0A0BAH, 557FH,0AAB3H, 5546H     ;
        DW      4000H,0AAA8H, 0000H, 2A88H      ;TILE(B)-B
        DW      0000H, 028AH, 0050H,0AAE2H      ;
```



```
TILING_DATA_SS:
        DW      7EEFH, 34A5H, 1C63H, 0C21H      ;TILE(0)-R (0)
        DW      0FFFFH, 6DEFH, 34A5H, 1C63H     ;
        DW      0C21H,0FFFFH, 6DEFH, 34A5H      ;
        DW      1C63H, 0C21H,0FFFFH,0EF7BH      ;
        ;
        DW      6D6BH, 30A5H, 1C63H, 0C21H      ;TILE(1)-G (1)
        DW      0FBFFH, 54A5H, 20A1H, 1421H     ;
        DW      0C21H,0FBFFH, 54A5H, 2421H      ;
        DW      1021H, 0C21H,0FBFFH, 14A5H      ;
        ;
        DW      4149H, 8622H, 0C22H, 8400H      ;TILE(4)-R (2)
        DW      0CA11H, 556EH, 90B1H, 0802H     ;
        DW      0800H,0A755H, 69E2H, 3C02H      ;
        DW      8442H, 0801H, 5BEBH, 4629H      ;
        ;
        DW      1C6EH,0AA96H, 5509H, 2C21H      ;TILE(4)-G (3)
        DW      6565H, 00CBH,0B481H, 4528H      ;
        DW      20A5H, 74EAH, 2C86H, 14E3H      ;
        DW      5909H, 2429H, 061CH,0E510H      ;
        ;
        DW      3118H, 1008H, 0700H, 0300H      ;TILE(5)-R (4)
        DW      0C30DH,0C003H, 4001H, 4270H     ;
        DW      1718H, 0380H,0C100H, 4000H      ;
        DW      0061H, 84F1H,0C639H, 6310H      ;
        ;
        DW      112AH, 2240H, 48C8H, 9311H      ;TILE(7)-G (5)
        DW      2222H, 4244H, 8088H, 1A19H      ;
        DW      2102H, 6084H, 8050H, 9411H      ;
        DW      2202H, 0644H, 8C0DH, 1012H      ;
        ;
        DW      8220H, 559DH,0A228H, 1095H      ;TILE(8)-G (6)
        DW      0000H, 8431H, 2220H,0D79DH      ;
        DW      2208H, 0000H, 8220H, 579DH      ;
        DW      0A228H, 1091H, 0000H, 4210H     ;
        ;
        DW      598CH, 2242H, 08C0H, 0842H      ;TILE(8)-B (7)
        DW      0B5ADH, 5946H, 1088H, 2842H     ;
        DW      0842H,0B4ADH, 5942H, 2042H      ;
        DW      0800H, 0842H,0B5ADH, 294AH      ;
        ;
        DW      0F914H, 677CH,0E6F6H, 68EFH     ;TILE(A)-R (8)
        DW      0E5A5H,0F6E3H,0C86EH, 3AADH     ;
        DW      0B472H, 0D2FH, 081EH,0FE29H     ;
        DW      0A9BBH,0F77EH, 3396H, 0D23H     ;
        ;
        DW      0B800H, 4574H,0A6A2H, 40C5H     ;TILE(A)-B (9)
        DW      0A0A0H,0F641H, 886AH, 308DH     ;
```



```
        DW      0B422H, 050DH, 081EH, 5621H     ;
        DW      0A8AAH,0D756H, 2296H, 0501H     ;
        ;
        DW      0EAAAH,0D559H,0AAA8H, 5575H     ;TILE(B)-G (A)
        DW      2AAAH, 1575H,0AAA2H, 7535H      ;
        DW      0AAA2H, 57C5H,0BE8AH, 5615H     ;
        DW      0A0BAH, 557FH,0AAB3H, 5546H     ;
        ;
        DW      4000H,0AAA8H, 0000H, 2A88H      ;TILE(B)-B (B)
        DW      0000H, 028AH, 0050H,0AAE2H      ;
        DW      0101H,0AAAAH, 0000H,0A20AH      ;
        DW      0000H,0A0A2H, 0000H,0AAA1H      ;
        ;
;
DEMO_INIT       ENDP
PROG    ENDS
        END
```

```
        NAME    DMCL
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  DMCLO_C, DMCLO_B, DMCL1_C, DMCL1_B, DMCLO_C_WAIT
        PUBLIC  DMCL1_C_WAIT, WAIT, DMCLO_C_FREE, MES_CL_WAIT
        PUBLIC  MESSAGE_WAIT, MES_CL_WAIT_FILL
        ;
        EXTRN   MESSAGE:NEAR
        ;
        EXTRN   FLAG:WORD, PTNCNT:WORD, PMAX:WORD, PLANES:WORD
        EXTRN   DY:WORD, Y:WORD, DXX:WORD, X:WORD, MOD10:WORD
        EXTRN   STATUS:WORD, COM:WORD, MAGETC:WORD, dAD1:BYTE
        EXTRN   EAD1H:WORD, EAD1L:WORD, DHH:WORD, DV:WORD
        EXTRN   WORK_7:WORD
        ;
DMCL    PROC    NEAR
        ;
;
;
;       < DISPLAY MEMORY CLEAR >
;
; [DMCLO_C] CLEARS VISIBLE 3 PLANES WITHOUT WORKING AREA TO "FFFFH"
; [DMCL1_C] CLEARS VISIBLE 3 PLANES AND NON-VISIBLE 4 PLANES TO "FFFFH"
;
        ;
DMCLO_C_WAIT:
        TEST    CS:BYTE PTR FLAG,1
        JZ      DMCLO_C                         ;CHECK IF 'WAIT'=1/0
        TEST    CS:BYTE PTR FLAG,4
        JNZ     DMCLO_C_END                     ;CHECK IF 'PIXEL'=1/0
        MOV     AX,4
        CALL    WAIT
        ;
DMCLO_C:
        MOV     AX,0                            ;X=0
        MOV     BX,749                          ;Y=749
        MOV     CX,1119                         ;DX=1119
        MOV     DX,-755                         ;DY=-755
        CALL    DMCLO_C_FREE
DMCLO_C_END:
        RET
        ;
DMCLO_B:
        TEST    DS:WORD PTR STATUS,1
        JNZ     DMCLO_B                         ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PTNCNT,0            ;PTNCNT=0
DMCLO:
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PLANES,7            ;PLANES=7
```



```
        MOV     DS:WORD PTR COM,903EH           ;<R_RECFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
                                                ;FAST=1
        RET
        ;
;
;       < WAIT >
;
        ;
WAIT:
        TEST    CS:BYTE PTR FLAG,1
        JZ      WAIT_END                        ;CHECK IF 'WAIT'=1/0
        MOV     CX,0FFFFH
WAIT_1:
        LOOPNZ  WAIT_1                          ;LOOP UNTIL CX=0
        DEC     AX
        JNZ     WAIT                            ;LOOP UNTIL AX=0
WAIT_END:
        RET
        ;
;
        ;
MES_CL_WAIT_FILL:
        MOV     CS:WORD PTR WORK_7,10
        JMP     MES_CL_WAIT_2
MES_CL_WAIT:
        MOV     CS:WORD PTR WORK_7,45
MES_CL_WAIT_2:
        CALL    MESSAGE_WAIT
        TEST    CS:WORD PTR FLAG,4
        JNZ     MES_CL_WAIT_1
        CALL    DMCLO_C_WAIT
        RET
MES_CL_WAIT_1:
        CALL    DMCL1_C_WAIT
        RET
        ;
MESSAGE_WAIT:
        TEST    CS:BYTE PTR FLAG,1
        JZ      MESSAGE_WAIT_1                  ;CHECK IF "WAIT"=1/0
        TEST    CS:BYTE PTR FLAG,4
        JNZ     MESSAGE_WAIT_1                  ;CHECK IF "PIXEL"=1/0
        CALL    MESSAGE                         ;MESSAGE(?)
        MOV     AX,CS:WORD PTR WORK_7
        CALL    WAIT
MESSAGE_WAIT_1:
        RET
        ;
;
DMCL    ENDP
PROG    ENDS
```



```
        MOV     DS:WORD PTR DY,-749             ;DY=-749
        JMP     DMCLO_1
        ;
DMCL1_C_WAIT:
        TEST    CS:BYTE PTR FLAG,1
        JZ      DMCL1_C                         ;CHECK IF 'WAIT'=1/0
        MOV     AX,4
        CALL    WAIT
        ;
DMCL1_C:
        MOV     DS:WORD PTR PTNCNT,0FFFFH       ;PTNCNT=0FFFFH
        JMP     DMCL1
        ;
DMCL1_B:
        MOV     DS:WORD PTR PTNCNT,0            ;PTNCNT=0
DMCL1:
        MOV     DS:WORD PTR EAD1H,4             ;EAD1H=4
        MOV     DS:WORD PTR EAD1L,0             ;EAD1L=0
        MOV     DS:BYTE PTR dAD1,0              ;dAD1=0
        MOV     DS:WORD PTR DHH,1119            ;DHH=1119
        MOV     DS:WORD PTR DV,2250             ;DV=2250
        MOV     DS:WORD PTR MAGETC,1FFH         ;MAGETC=1FFH NON-CLIP
        MOV     DS:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
        MOV     DS:WORD PTR COM,8E1EH           ;<A_RECFILL_A>
                                                ;TL=0,SS=1,WL=1,WR=1
                                                ;FAST=1
        RET
        ;
DMCLO_1:
        MOV     DS:WORD PTR Y,749               ;Y=749
        MOV     DS:WORD PTR DXX,1119            ;DX=1119
        MOV     DS:WORD PTR X,0                 ;X=0
        MOV     DS:WORD PTR MAGETC,1FFH         ;MAGETC=1FFH NON-CLIP
        MOV     DS:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
        MOV     DS:WORD PTR COM,903EH           ;<R_RECFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
                                                ;FAST=1
        RET
        ;
DMCLO_C_FREE:
        TEST    DS:WORD PTR STATUS,1
        JNZ     DMCLO_C_FREE                    ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PTNCNT,0FFFFH       ;PTNCNT=0FFFFH
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PLANES,7            ;PLANES=7
        MOV     DS:WORD PTR MAGETC,1FFH         ;MAGETC=1FFH NON-CLIP
        MOV     DS:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
        MOV     DS:WORD PTR X,AX                ; X=(?)
        MOV     DS:WORD PTR Y,BX                ; Y=(?)
        MOV     DS:WORD PTR DXX,CX                ;DX=(?)
        MOV     DS:WORD PTR DY,DX                               ;DY=(?)
```



```
        END
```

```
        NAME    GLIOMAIN
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  GLIOMAIN
        ;
        EXTRN   STATUS:WORD, PTNCNT:WORD, PMAX:WORD, PLANES:WORD
        EXTRN   MAGETC:WORD, MOD1O:WORD, X:WORD, Y:WORD, DXX:WORD
        EXTRN   DY:WORD, COM:WORD, XE:WORD, YE:WORD, XS:WORD, YS:WORD
        EXTRN   XC:WORD, YC:WORD, DHH:WORD, DV:WORD
        EXTRN   XCLMIN:WORD, YCLMIN:WORD, XCLMAX:WORD, YCLMAX:WORD
        EXTRN   EAD2L:WORD, EAD2H:WORD, dAD2:BYTE, PITCHS:WORD
        EXTRN   EADORGL:WORD, EADORGH:WORD, dADORG:BYTE
        EXTRN   WORK_1:WORD,
        ;
GLIOMAIN        PROC    NEAR
        ;
;
;       < GRAPH LIO/BIO >
;
;
        ;
        PUSH    BX
        CALL    G_PUSH_EXE
        CALL    INIT_SEG
        CMP     AH,5
        JZ      GCLS                            ;"GCLS"
        CMP     AH,7
        JZ      LINE                            ;"LINE","REC","REC_FILL"
        CMP     AH,8
        JZ      CRL                             ;"CRL","ELPS","CRL_FILL"
                                                ;"ELPS_FILL"
        CMP     AH,9
        JZ      PAINT                           ;"PAINT"
        CMP     AH,0DH
        JZ      COPY                            ;"COPY"
        JMP     G_POP_EXE
        ;
CRL:
        JMP     CRL_1
        ;
PAINT:
        JMP     PAINT_1
        ;
COPY:
        JMP     COPY_1
        ;
GCLS:
        MOV     AX,ES
        MOV     DS,AX
```



```
        JMP     REC_RECFILL_EXE
        ;
G_RECFILL_EXE:
        MOV     CS:WORD PTR WORK_1,8C3CH        ;<A_RECFILL_C>
                                                ;TL=0,SS=1,WL=1,WR=1
                                                ;FAST=1
REC_RECFILL_EXE:
        TEST    DS:WORD PTR STATUS,1
        JNZ     REC_RECFILL_EXE                 ;CHECK IF PPBUSY=1/0
        CALL    COLOR_CAL
        MOV     DS:WORD PTR X,BX                ; X=(?)
        MOV     DS:WORD PTR Y,CX                ; Y=(?)
        MOV     DS:WORD PTR XS,SI               ;XS=(?)
        MOV     DS:WORD PTR YS,DI               ;YS=(?)
GENERAL_CP:
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNCNT,0FFFFH       ;PTNCNT=0FFFFH
        MOV     DS:WORD PTR MOD10,1             ;MOD1=0,MODO=1
        MOV     AX,CS:WORD PTR WORK_1
        MOV     DS:WORD PTR COM,AX              ;<COM.FLAG>
        JMP     G_POP_EXE
        ;
CRL_1:
        MOV     BX,DS:[BP]                      ;XC=(?)
        MOV     CX,DS:[BP+2]
        XOR     CX,0FFFFH
        INC     CX                              ;YC=(?)
        MOV     SI,DS:[BP+4]                    ;DX=(?)
        MOV     DI,DS:[BP+6]                    ;DY=(?)
        MOV     AX,DS:[BP+8]                    ;
        MOV     DX,ES
        MOV     DS,DX
        CMP     SI,DI
        JZ      CRL_2
        MOV     CS:WORD PTR WORK_1,5C00H        ;<ELPS> CF=0,IP=0
                                                ;PXEN=0
        TEST    AH,20H
        JZ      ELPS_ELPSFILL_EXE
        MOV     CS:WORD PTR WORK_1,5C3CH        ;<ELPSFILL> TL=0,SS=1
ELPS_ELPSFILL_EXE:
        TEST    DS:WORD PTR STATUS,1
        JNZ     ELPS_ELPSFILL_EXE               ;CHECK IF PPBUSY=1/0
        CALL    COLOR_CAL
        MOV     DS:WORD PTR XC,BX               ;XC=(?)
        MOV     DS:WORD PTR YC,CX               ;YC=(?)
        CALL    DHDV_CAL
        MOV     DS:WORD PTR DHH,BX              ;DHH=(?)
        MOV     DS:WORD PTR DV,AX               ;DV=(?)
        MOV     DS:WORD PTR DY,DI               ;DY=(?)
        JMP     GENERAL_CP
```



```
GCLS_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     GCLS_1                          ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PTNCNT,0            ;PTNCNT=0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PLANES,7            ;PLANES=7
        MOV     DS:WORD PTR MAGETC,1FFH         ;MAGETC=1FFH NON-CLIP
        MOV     DS:WORD PTR MOD10,0             ;MOD1=0,MODO=0
        MOV     DS:WORD PTR X,0                 ; X=0
        MOV     DS:WORD PTR Y,-749              ; Y=-749
        MOV     DS:WORD PTR DXX,1119            ;DX=1119
        MOV     DS:WORD PTR DY,755              ;DY=755
        MOV     DS:WORD PTR COM,903EH           ;<R_RECFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
                                                ;FAST=1
        JMP     G_POP_EXE
        ;
LINE:
        MOV     BX,DS:[BP]                      ;XS=(?)
        MOV     CX,DS:[BP+2]
        XOR     CX,0FFFFH
        INC     CX                              ;YS=(?)
        MOV     SI,DS:[BP+4]                    ;XE=(?)
        MOV     DI,DS:[BP+6]
        XOR     DI,0FFFFH
        INC     DI                              ;YE=(?)
        MOV     AX,DS:[BP+8]
        MOV     DX,ES
        MOV     DS,DX
        CMP     AH,0
        JZ      G_LINE_EXE
        CMP     AH,1
        JZ      G_REC_EXE
        CMP     AH,2
        JZ      G_RECFILL_EXE
        JMP     G_POP_EXE
        ;
G_LINE_EXE:
        TEST    DS:WORD PTR STATUS,1
        JNZ     G_LINE_EXE                      ;CHECK IF PPBUSY=1/0
        MOV     CS:WORD PTR WORK_1,1800H        ;<A_LINE_M1> PL=0
                                                ;IP=0,PXEN=0
        CALL    COLOR_CAL
        MOV     DS:WORD PTR X,BX                ; X=(?)
        MOV     DS:WORD PTR Y,CX                ; Y=(?)
        MOV     DS:WORD PTR XE,SI               ;XE=(?)
        MOV     DS:WORD PTR YE,DI               ;YE=(?)
        JMP     GENERAL_CP
        ;
G_REC_EXE:
        MOV     CS:WORD PTR WORK_1,4800H        ;<A_REC> PXEN=0
```



```
CRL_2:
        MOV     CS:WORD PTR WORK_1,5000H        ;<CRL> CF=0,IP=0
                                                ;PXEN=0
        TEST    AH,20H
        JZ      CRL_CRLFILL_EXE
        MOV     CS:WORD PTR WORK_1,503CH        ;<CRLFILL> TL=0,SS=1
CRL_CRLFILL_EXE:
        TEST    DS:WORD PTR STATUS,1
        JNZ     CRL_CRLFILL_EXE                 ;CHECK IF PPBUSY=1/0
        CALL    COLOR_CAL
        MOV     DS:WORD PTR XC,BX               ;X=(?)
        MOV     DS:WORD PTR YC,CX               ;Y=(?)
        MOV     DS:WORD PTR DXX,SI              ;DXX=(?)
        JMP     GENERAL_CP
        ;
PAINT_1:
        MOV     BX,DS:[BP]                      ;X=(?)
        MOV     CX,DS:[BP+2]
        XOR     CX,0FFFFH
        INC     CX                              ;Y=(?)
        MOV     AX,DS:[BP+4]                    ;AH-->B.COL, AL-->D.COL
        MOV     DX,ES
        MOV     DS,DX
PAINT_2:
        TEST    DS:WORD PTR STATUS,1
        JNZ     PAINT_2                         ;CHECK IF PPBUSY=1/0
        MOV     CS:WORD PTR WORK_1,6830H        ;<PAINT>
                                                ;PMOD=0, TL=0, SS=1
        CALL    PAINT_COL
        MOV     DS:WORD PTR XCLMIN,0            ;XCLMIN=0
        MOV     DS:WORD PTR YCLMIN,-749         ;YCLMIN=-749
        MOV     DS:WORD PTR XCLMAX,1119         ;XCLMAX=1119
        MOV     DS:WORD PTR YCLMAX,0            ;YCLMAX=0
        MOV     DS:WORD PTR MAGETC,0FFH         ;MAGETC=0FFH
        MOV     DS:WORD PTR X,BX                ;X=(?)
        MOV     DS:WORD PTR Y,CX                ;Y=(?)
        JMP     GENERAL_CP
        ;
COPY_1:
        MOV     BX,DS:[BP]                      ;X=(?)
        ADD     BX,2                            ;BODY-->CHARACTER FACE
        MOV     CX,DS:[BP+2]
        ADD     CX,3                            ;BODY-->CHARACTRE FACE
        XOR     CX,0FFFFH
        INC     CX                              ;Y=(?)
        MOV     AX,DS:[BP+4]                    ;KANJI CODE
        MOV     DI,DS:[BP+8]                    ;
        MOV     DX,ES
        MOV     DS,DX
        ;
        MOV     DX,1
```

```
        SHL     AL,1
        SHL     AX,1
        SHL     AX,1
        RCL     DX,1
        SHL     AX,1
        RCL     DX,1
        SHL     AX,1
        RCL     DX,1
        SHL     AX,1
        RCL     DX,1
        ;
COPY_2:
        TEST    DS:WORD PTR STATUS,1
        JNZ     COPY_2                  ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR EAD2L,AX    ;EAD2L=(?)
        MOV     DS:WORD PTR EAD2H,DX    ;EAD2H=(?)
        MOV     DS:BYTE PTR dAD2,0      ;dAD2=0
        MOV     AX,DI
        CALL    COLOR_CAL
        MOV     DS:WORD PTR PITCHS,2    ;PITCHS=2
        MOV     DS:WORD PTR DHH,23      ;DH=23
        MOV     DS:WORD PTR DV,23       ;DV=23
        MOV     DS:WORD PTR X,BX        ;X=(?)
        MOV     DS:WORD PTR Y,CX        ;Y=(?)
        MOV     DS:WORD PTR PMAX,4      ;PMAX=4
        MOV     DS:WORD PTR MOD10,2     ;MOD1=0,MODO=2
        MOV     DS:WORD PTR COM,8008H   ;<A_COPY_AC> COPY
                                        ;ESE=0,REV=0,ROT=0,{MD}
                                        ;FAST=0 <<!!BUG!!>>
        JMP     G_POP_EXE
        ;
;
        ;
INIT_SEG:
        PUSH    AX
        MOV     AL,9
        OUT     0D1H,AL                 ;ENABLE /CSIR/
        IN      AL,0D1H
        TEST    AL,1
        MOV     AX,0D000H
        JZ      MOD_9801                ;CHECK IF 98XA/9801
        MOV     AX,8000H
MOD_9801:
        MOV     ES,AX
        MOV     AL,0F0H
        OUT     91H,AL
        OUT     93H,AL
INIT_SEG_1:
        TEST    ES:WORD PTR STATUS,1
        JNZ     INIT_SEG_1              ;CHECK IF PPBUSY=1/0
        MOV     ES:WORD PTR EADORGL,0   ;EADORGL=0
```



```
        MOV     ES:WORD PTR EADORGH,0   ;EADORGH=0
        MOV     ES:BYTE PTR dADORG,0    ;dADORG=0
        MOV     BP,100H
        POP     AX
        RET
        ;
COLOR_CAL:
        AND     AX,7
        MOV     BP,AX
        MOV     AL,CS:BYTE PTR [COL+BP]
        MOV     DS:WORD PTR PLANES,AX   ;PLANES=(?)
        RET
        ;
COL:
        DB      0, 4, 1, 5, 2, 6, 3, 7  ;FIT COLOR BITS
        ;
PAINT_COL:
        MOV     DX,AX
        CALL    COLOR_CAL
        MOV     AX,DX
        MOV     AL,AH
        AND     AX,7
        MOV     BP,AX
        MOV     AL,CS:BYTE PTR [COL+BP]
        MOV     DS:WORD PTR DXX,AX      ;DXX=(?)
        RET
        ;
DHDV_CAL:
        MOV     AX,SI
        MUL     AX                      ;DH=DX**2
        MOV     BX,AX
        MOV     CX,DX
        ;
        MOV     AX,DI
        MUL     AX                      ;DV=DY**2
DHDV_CAL_1:
        OR      DX,DX
        JZ      EXIT_ROT
        SHR     DX,1
        RCR     AX,1
        SHR     CX,1
        RCR     BX,1
        JMP     DHDV_CAL_1
        ;
EXIT_ROT_1:
        SHR     AX,1
        SHR     CX,1
        RCR     BX,1
EXIT_ROT:
        OR      CX,CX
        JNZ     EXIT_ROT_1
```



```
        ;
        MOV     DX,0FC00H
DHDV_CAL_3:
        TEST    AX,DX
        JZ      DHDV_CAL_2
        SHR     AX,1
        SHR     BX,1
        JMP     DHDV_CAL_3
        ;
DHDV_CAL_4:
        SHR     AX,1
        SHR     BX,1
DHDV_CAL_2:
        TEST    BX,DX
        JNZ     DHDV_CAL_4
        RET
        ;
;
;       < PUSH POP >
;
;
        ;
G_POP_EXE:
        POP     ES
        POP     DS
        POP     DI
        POP     SI
        POP     BP
        POP     DX
        POP     CX
        POP     AX
        POP     BX
        MOV     AX,0                    ;NORMAL END
        IRET
        ;
G_PUSH_EXE:
        POP     BX
        PUSH    AX
        PUSH    CX
        PUSH    DX
        PUSH    BP
        PUSH    SI
        PUSH    DI
        PUSH    DS
        PUSH    ES
        PUSH    BX
        RET
        ;
;
GLIOMAIN        ENDP
PROG    ENDS
```



```
        END
```

```
        NAME    LINE
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  LINE_DEMO, LINE_DEMO_H
        ;
        EXTRN   DMCLO_C_WAIT:NEAR, MES_CL_WAIT:NEAR
        ;
        EXTRN   WORK_1:WORD, WORK_2:WORD, WORK_3:WORD, WORK_4:WORD
        EXTRN   WORK_5:WORD, WORK_6:WORD, STATUS:WORD, PMAX:WORD
        EXTRN   PTNCNT:WORD, X:WORD, Y:WORD, MOD10:WORD, PLANES:WORD
        EXTRN   XE:WORD, YE:WORD, COM:WORD, PTNCNT:WORD, FLAG:BYTE
        ;
LINE    PROC    NEAR
        ;
;
;       < LINE DEMO >
;
;
        ;
LINE_DEMO:
        MOV     SI,10
        CALL    MES_CL_WAIT                     ;MESSAGE(10)
        ;
        MOV     CS:WORD PTR WORK_1,64           ;REPETITION COUNTS (H)
        MOV     CS:WORD PTR WORK_2,48           ;REPETITION COUNTS (V)
        MOV     CS:WORD PTR WORK_3,10           ;DISTANCE (X)
        MOV     CS:WORD PTR WORK_4,10           ;DISTANCE (Y)
        MOV     CS:WORD PTR WORK_5,319          ;X=319
        MOV     CS:WORD PTR WORK_6,239          ;Y=239
        CALL    LINE_EXE_ALL
        MOV     SI,11
        CALL    MES_CL_WAIT                     ;MESSAGE(11)
        CALL    LINE_DEMO_H
        MOV     SI,12
        CALL    MES_CL_WAIT                     ;MESSAGE(12)
        RET
        ;
LINE_DEMO_H:
        MOV     CS:WORD PTR WORK_1,56           ;REPETITION COUNTS (H)
        MOV     CS:WORD PTR WORK_2,50           ;REPETITION COUNTS (V)
        MOV     CS:WORD PTR WORK_3,20           ;DISTANCE (X)
        MOV     CS:WORD PTR WORK_4,15           ;DISTANCE (Y)
        MOV     CS:WORD PTR WORK_5,559          ;X=559
        MOV     CS:WORD PTR WORK_6,374          ;Y=374
LINE_EXE_ALL:
        TEST    DS:WORD PTR STATUS,1
        JNZ     LINE_EXE_ALL                    ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNCNT,0FFFFH       ;PTNCNT=0FFFFH
```



```
LINE_EXE_1:
        MOV     DS:WORD PTR PTNCNT,SI           ;PTNCNT=(?)
        MOV     DS:WORD PTR MOD10,44H           ;MOD1=4,MOD0=4
        MOV     DS:WORD PTR COM,1814H           ;<A_LINE_M1> PL=0
                                                ;IP=0,PXEN=1,BPPX=1
        ADD     SI,1111H                        ;PTNCNT+1111H
        RET
        ;
;
LINE    ENDP
PROG    ENDS
        END
```



```
        MOV     AX,CS:WORD PTR WORK_5
        MOV     DS:WORD PTR X,AX                ;X=(?)
        MOV     AX,CS:WORD PTR WORK_6
        MOV     DS:WORD PTR Y,AX                ;Y=(?)
        MOV     DS:WORD PTR MOD10,10H           ;MOD1=1,MOD0=0
        MOV     BX,0                            ;XE=0
        MOV     DX,0                            ;YE=0
        MOV     SI,1                            ;PLANES=1
        TEST    CS:BYTE PTR FLAG,4
        JNZ     LINE_EXE_ALL_1                  ;CHECK IF PIXEL=1/0
        MOV     SI,1111H                        ;PTNCNT=1111H
LINE_EXE_ALL_1:
        MOV     CX,CS:WORD PTR WORK_1           ;REPETITION COUNTS (H)
LINE_LOW:
        CALL    LINE_EXE
        ADD     BX,CS:WORD PTR WORK_3           ;XE+(?) --> XE
        LOOP    LINE_LOW
        ADD     DX,CS:WORD PTR WORK_4           ;YE=(?)
        MOV     CX,CS:WORD PTR WORK_2           ;REPETITION COUNTS (V)
LINE_RIGHT:
        CALL    LINE_EXE
        ADD     DX,CS:WORD PTR WORK_4           ;YE+(?) --> YE
        LOOP    LINE_RIGHT
        SUB     BX,CS:WORD PTR WORK_3           ;XE-(?) --> XE
        MOV     CX,CS:WORD PTR WORK_1           ;REPETITION COUNTS (H)
        DEC     CX
LINE_HIGH:
        CALL    LINE_EXE
        SUB     BX,CS:WORD PTR WORK_3           ;XE-(?) --> XE
        LOOP    LINE_HIGH
        SUB     DX,CS:WORD PTR WORK_4           ;YE-(?) --> YE
        MOV     CX,CS:WORD PTR WORK_2           ;REPETITION COUNTS (V)
LINE_LEFT:
        CALL    LINE_EXE
        SUB     DX,CS:WORD PTR WORK_4           ;YE-(?) --> YE
        LOOP    LINE_LEFT
        RET
        ;
LINE_EXE:
        TEST    DS:WORD PTR STATUS,1
        JNZ     LINE_EXE                        ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XE,BX               ;XE=(?)
        MOV     DS:WORD PTR YE,DX               ;YE=(?)
        TEST    CS:BYTE PTR FLAG,4
        JNZ     LINE_EXE_1                      ;CHECK IF PIXEL=1/0
        MOV     DS:WORD PTR PLANES,SI                ;PLANES=(?)
        MOV     DS:WORD PTR COM,1800H           ;<A_LINE_M1> PL=0
                                                ;IP=0,PXEN=0
        INC     SI                              ;PLANES+1
        RET
        ;
```



```
        NAME    RECT
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  RECT_DEMO
        ;
        EXTRN   MES_CL_WAIT:NEAR
        ;
        EXTRN   STATUS:WORD, PMAX:WORD, PTNCNT:WORD, X:WORD, Y:WORD
        EXTRN   MOD10:WORD, PLANES:WORD, XS:WORD, YS:WORD, COM:WORD
        ;
RECT    PROC    NEAR
        ;
;
;       < RECTANGLE DEMO >
;
;
        ;
RECT_DEMO:
        MOV     SI,28                           ;MESSAGE(28)
        CALL    MES_CL_WAIT
        CALL    RECT_DEMO_L
        MOV     SI,29                           ;MESSAGE(29)
        CALL    MES_CL_WAIT
        CALL    RECT_DEMO_H
        MOV     SI,30                           ;MESSAGE(30)
        CALL    MES_CL_WAIT
        RET
        ;
RECT_DEMO_L:
        MOV     SI,639                          ;XS=639
        MOV     DI,479                          ;YS=479
        MOV     CX,60                           ;REPETITION COUNTS
        JMP     RECT_EXE_ALL
        ;
RECT_DEMO_H:
        MOV     SI,1119                         ;XS=1119
        MOV     DI,749                          ;YS=749
        MOV     CX,93                           ;REPETITION COUNTS
RECT_EXE_ALL:
        MOV     AX,0                            ;X=0
        MOV     BX,0                            ;Y=0
RECT_EXE_ALL_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     RECT_EXE_ALL_1                  ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNCNT,0FFFFH       ;PTNCNT=0FFFFH
        MOV     DS:WORD PTR X,AX                ;X=(?)
        MOV     DS:WORD PTR Y,BX                ;Y=(?)
        MOV     DS:WORD PTR MOD10,10H           ;MOD1=1,MOD0=0
```

```
        MOV     DS:WORD PTR PLANES,CX           ;PLANES=(?)
        MOV     DS:WORD PTR XS,SI               ;XS=(?)
        MOV     DS:WORD PTR YS,DI               ;YS=(?)
        MOV     DS:WORD PTR COM,4800H           ;<A_REC>
                                                ;PXEN=0
        ADD     AX,5                            ;X+5  --> X
        ADD     BX,4                            ;Y+4  --> Y
        SUB     SI,5                            ;XS-5 --> XS
        SUB     DI,4                            ;YS-4 --> YS
        LOOP    RECT_EXE_ALL_1
        RET
        ;
;
RECT    ENDP
PROG    ENDS
        END
```



```
        MOV     DS:WORD PTR PLANES,AX           ;PLANES=(?)
        MOV     DS:WORD PTR DXX,BX              ;DXX=(?)
        MOV     DS:WORD PTR COM,5000H           ;<CRL> CF=0,IP=0
                                                ;PXEN=0
        INC     AX                              ;PLANES+1
        SUB     BX,5                            ;DXX-5
        JNC     CRL_EXE_ALL_1
        RET
        ;
;
;       < ELLIPSE DEMO >
;
;
        ;
ELPS_DEMO:
        MOV     SI,39                           ;MESSAGE(39)
        CALL    MES_CL_WAIT
        MOV     CS:WORD PTR WORK_1,16           ;DH=16
        MOV     CS:WORD PTR WORK_2,9            ;DV=9
        CALL    ELPS_DEMO_1
        MOV     CS:WORD PTR WORK_1,9            ;DH=9
        MOV     CS:WORD PTR WORK_2,16           ;DV=16
        CALL    ELPS_DEMO_1
        RET
        ;
ELPS_DEMO_1:
        CALL    ELPS_DEMO_L
        MOV     SI,40                           ;MESSAGE(40)
        CALL    MES_CL_WAIT
        CALL    ELPS_DEMO_H
        MOV     SI,41                           ;MESSAGE(41)
        CALL    MES_CL_WAIT
        RET
        ;
ELPS_DEMO_L:
        MOV     SI,319                          ;XC=319
        MOV     DI,239                          ;YC=239
        MOV     BX,238                          ;DV=238
        JMP     ELPS_EXE_ALL
        ;
ELPS_DEMO_H:
        MOV     SI,559                          ;XC=559
        MOV     DI,374                          ;YC=374
        MOV     BX,373                          ;DV=373
ELPS_EXE_ALL:
        TEST    DS:WORD PTR STATUS,1
        JNZ     ELPS_EXE_ALL                    ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNCNT,0FFFFH       ;PTNCNT=0FFFFH
        MOV     CX,1
ELPS_EXE_ALL_1:
```



```
        NAME    CRL_ELPS
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  CRL_DEMO, ELPS_DEMO
        ;
        EXTRN   MES_CL_WAIT:NEAR
        ;
        EXTRN   STATUS:WORD, PMAX:WORD, PTNCNT:WORD, XC:WORD, YC:WORD
        EXTRN   MOD10:WORD, PLANES:WORD, DXX:WORD, COM:WORD
        EXTRN   WORK_1:WORD, WORK_2:WORD, DHH:WORD, DV:WORD, DY:WORD
        ;
CRL_ELPS        PROC    NEAR
        ;
;
;       < CIRCLE DEMO >
;
;
        ;
CRL_DEMO:
        MOV     SI,36                           ;MESSAGE(36)
        CALL    MES_CL_WAIT
        CALL    CRL_DEMO_L
        MOV     SI,37                           ;MESSAGE(37)
        CALL    MES_CL_WAIT
        CALL    CRL_DEMO_H
        MOV     SI,38                           ;MESSAGE(38)
        CALL    MES_CL_WAIT
        RET
        ;
CRL_DEMO_L:
        MOV     SI,319                          ;XC=319
        MOV     DI,239                          ;YC=239
        MOV     BX,238                          ;DXX=238
        JMP     CRL_EXE_ALL
        ;
CRL_DEMO_H:
        MOV     SI,559                          ;XC=559
        MOV     DI,374                          ;YC=374
        MOV     BX,373                          ;DXX=373
CRL_EXE_ALL:
        MOV     AX,1
CRL_EXE_ALL_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     CRL_EXE_ALL_1                   ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNCNT,0FFFFH       ;PTNCNT=0FFFFH
        MOV     DS:WORD PTR XC,SI               ;X=(?)
        MOV     DS:WORD PTR YC,DI               ;Y=(?)
        MOV     DS:WORD PTR MOD10,10H           ;MOD1=1,MOD0=0
```



```
        TEST    DS:WORD PTR STATUS,1
        JNZ     ELPS_EXE_ALL_1                  ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XC,SI               ;XC=(?)
        MOV     DS:WORD PTR YC,DI               ;YC=(?)
        MOV     AX,CS:WORD PTR WORK_1
        MOV     DS:WORD PTR DHH,AX              ;DHH=(?)
        MOV     AX,CS:WORD PTR WORK_2
        MOV     DS:WORD PTR DV,AX               ;DV=(?)
        MOV     DS:WORD PTR MOD10,10H           ;MOD1=1,MOD0=0
        MOV     DS:WORD PTR PLANES,CX           ;PLANES=(?)
        MOV     DS:WORD PTR DY,BX               ;DY=(?)
        MOV     DS:WORD PTR COM,5C00H           ;<ELPS> CF=0,IP=0
                                                ;PXEN=0
        INC     CX                              ;PLANES+1
        SUB     BX,5                            ;DXX-5
        JNC     ELPS_EXE_ALL_1
        RET
        ;
;
CRL_ELPS        ENDP
PROG    ENDS
        END
```

```
        NAME    RECFILL
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  RECFILL_DEMO
        ;
        EXTRN   DMCLO_C_WAIT:NEAR, MES_CL_WAIT:NEAR
        EXTRN   MES_CL_WAIT_FILL:NEAR
        ;
        EXTRN   STATUS:WORD, PDISPSL:WORD, PDISPSH:WORD, PMAX:WORD
        EXTRN   PTNPH:WORD, PTNPL:WORD, X:WORD, Y:WORD, DXX:WORD
        EXTRN   DY:WORD, PTNCNT:WORD, PLANES:WORD, MOD10:WORD
        EXTRN   COM:WORD, AGDC_SEG:WORD
        ;
RECFILL_DEMO    PROC    NEAR
        ;
;
;       < RECTANGLE FILLING DEMO >
;
;
        ;
        MOV     SI,1
        CALL    MES_CL_WAIT                     ;MESSAGE(1)
        ;
RECFILL_DEMO_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     RECFILL_DEMO_1                  ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PDISPSL,10H         ;PDISPSL=10H
        MOV     DS:WORD PTR PDISPSH,0           ;PDISPSH=0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNPH,0             ;PTNPH=0
        MOV     BX,903EH                        ;<R_RECFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
                                                ;FAST=1
        CALL    RECFILL_EXE
        MOV     SI,2
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(2)
        ;
        MOV     BX,90BCH                        ;<R_RECFILL>
                                                ;TL=1,SS=1,WL=1,WR=1
                                                ;FAST=0
        CALL    RECFILL_EXE
        MOV     SI,3
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(3)
        ;
        MOV     BX,90ACH                        ;<R_RECFILL>
                                                ;TL=1,SS=0,WL=1,WR=1
                                                ;FAST=0
        CALL    RECFILL_EXE
        MOV     SI,4
```



```
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(4)
        ;
        MOV     BX,903EH                        ;<R_RECFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
                                                ;FAST=1
        CALL    RECFILL_EXE_1
        MOV     SI,5
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(5)
        ;
        MOV     BX,90BCH                        ;<R_RECFILL>
                                                ;TL=1,SS=1,WL=1,WR=1
                                                ;FAST=0
        CALL    RECFILL_EXE_1
        MOV     SI,6
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(6)
        ;
        MOV     BX,90ACH                        ;<R_RECFILL>
                                                ;TL=1,SS=0,WL=1,WR=1
                                                ;FAST=0
        CALL    RECFILL_EXE_1
        MOV     SI,7
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(7)
        ;
        RET
        ;
RECFILL_EXE:
        MOV     DI,1                            ;DRAW COUNTS
RECFILL_EXE1:
        CALL    RECFILL_EXE_L
        DEC     DI
        JNZ     RECFILL_EXE1
        RET
        ;
RECFILL_EXE_1:
        MOV     DI,1                            ;DRAW COUNTS
RECFILL_EXE_11:
        CALL    RECFILL_EXE_H
        DEC     DI
        JNZ     RECFILL_EXE_11
        RET
        ;
RECFILL_EXE_L:
        MOV     SI,OFFSET RECFILL_DATA_L
        JMP     RECFILL_1
RECFILL_EXE_H:
        MOV     SI,OFFSET RECFILL_DATA_H
RECFILL_1:
        PUSH    DS
        MOV     AX,CS
        MOV     DS,AX
        MOV     AX,WORD PTR AGDC_SEG
```



```
        MOV     ES,AX
        MOV     CX,12                           ;REPETITION COUNTS
        MOV     DX,0F7D0H                       ;TILE    .. 0F800H
        TEST    BL,10H
        JZ      RECFILL_2                       ;CHECK IF "SS"=1/0
        MOV     DX,0FA30H                       ;TILE_SS .. 0FA40H
RECFILL_2:
        TEST    ES:WORD PTR STATUS,1
        JNZ     RECFILL_2                       ;CHECK IF PPBUSY=1/0
        TEST    BL,10H
        JNZ     RECFILL_2_1                     ;CHECK IF "SS"=1/0
        ADD     DX,20H
RECFILL_2_1:
        ADD     DX,10H
        MOV     ES:WORD PTR PTNPL,DX            ;PTNPL=(?)
        LODSW
        MOV     ES:WORD PTR X,AX                ;X=(?)
        LODSW
        MOV     ES:WORD PTR Y,AX                ;Y=(?)
        LODSW
        MOV     ES:WORD PTR DXX,AX              ;DX=(?)
        LODSW
        MOV     ES:WORD PTR DY,AX               ;DY=(?)
        TEST    BL,80H
        JNZ     RECFILL_TL1                     ;CHECK IF "TL"=1/0
        MOV     ES:WORD PTR PTNCNT,0            ;PTNCNT=0
        MOV     AX,CX
        AND     AX,7
        JNZ     RECFILL_3                       ;CHECK IF "WHITE"
        MOV     AX,3
RECFILL_3:
        MOV     ES:WORD PTR PLANES,AX           ;PLANES=(?)
        MOV     ES:WORD PTR MOD10,1             ;MOD1=0,MOD0=1
        JMP     RECFILL_TL
        ;
RECFILL_TL1:
        MOV     ES:WORD PTR PTNCNT,16           ;PTNCNT=16
        MOV     ES:WORD PTR PLANES,7            ;PLANES=7
        MOV     ES:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
RECFILL_TL:
        MOV     ES:WORD PTR COM,BX              ;<COM.FLAGS>
        LOOP    RECFILL_2
        POP     DS
        RET
        ;
RECFILL_DATA_L:
        DW      0032H,01AEH,0186H,0FEDEH        ; 1. X,Y,DX,DY
        DW      003CH,0104H,017CH,0FEE8H        ; 2. X,Y,DX,DY
        DW      010EH,019AH,0172H,0FEF2H        ; 3. X,Y,DX,DY
        DW      0104H,0118H,0168H,0FEFCH        ; 4. X,Y,DX,DY
        DW      005AH,0186H,015EH,0FF06H        ; 5. X,Y,DX,DY
```



```
        DW      0064H,012CH,0154H,0FF10H        ; 6. X,Y,DX,DY
        DW      00E6H,0172H,014AH,0FF1AH        ; 7. X,Y,DX,DY
        DW      00DCH,0140H,0140H,0FF24H        ; 8. X,Y,DX,DY
        DW      0082H,015EH,0136H,0FF2EH        ; 9. X,Y,DX,DY
        DW      008CH,0154H,012CH,0FF38H        ;10. X,Y,DX,DY
        DW      00BEH,014AH,0122H,0FF42H        ;11. X,Y,DX,DY
        DW      0084H,0168H,0118H,0FF4CH        ;12. X,Y,DX,DY
;       DW      00AAH,0136H,010EH,0FF56H        ;13. X,Y,DX,DY
        ;
RECFILL_DATA_H:
        DW      001EH,02BCH,02B0H,0FE36H        ; 1. X,Y,DX,DY
        DW      002DH,0195H,02A4H,0FE42H        ; 2. X,Y,DX,DY
        DW      01B8H,029EH,0298H,0FE4EH        ; 3. X,Y,DX,DY
        DW      01A9H,01B3H,028CH,0FE5AH        ; 4. X,Y,DX,DY
        DW      005AH,0280H,0280H,0FE66H        ; 5. X,Y,DX,DY
        DW      0069H,01D1H,0274H,0FE72H        ; 6. X,Y,DX,DY
        DW      017CH,0262H,0268H,0FE7EH        ; 7. X,Y,DX,DY
        DW      016DH,01EFH,025CH,0FE8AH        ; 8. X,Y,DX,DY
        DW      0096H,0244H,0250H,0FE96H        ; 9. X,Y,DX,DY
        DW      00A5H,020DH,0244H,0FEA2H        ;10. X,Y,DX,DY
        DW      0140H,0226H,0238H,0FEAEH        ;11. X,Y,DX,DY
        DW      0131H,022BH,022CH,0FEBAH        ;12. X,Y,DX,DY
;       DW      00D2H,0208H,0220H,0FEC6H        ;13. X,Y,DX,DY
        ;
;
RECFILL_DEMO    ENDP
PROG    ENDS
        END
```

```
        NAME    CRLFILL
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  CRLFILL_DEMO, ELPSFILL_DEMO
        ;
        EXTRN   MES_CL_WAIT:NEAR, MES_CL_WAIT_FILL:NEAR
        ;
        EXTRN   STATUS:WORD, PDISPSL:WORD, PDISPSH:WORD, PMAX:WORD
        EXTRN   PTNPH:WORD, AGDC_SEG:WORD, PTNPL:WORD, DXX:WORD
        EXTRN   XC:WORD, YC:WORD, PTNCNT:WORD, PLANES:WORD, MOD10:WORD
        EXTRN   COM:WORD, WORK_1:WORD, WORK_2:WORD, DY:WORD
        EXTRN   DHH:WORD, DV:WORD
        ;
CRLFILL PROC    NEAR
        ;
;
;       < CIRCLE FILLING DEMO >
;
;
        ;
CRLFILL_DEMO:
        MOV     SI,34
        CALL    MES_CL_WAIT                     ;MESSAGE(34)
        ;
CRLFILL_DEMO_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     CRLFILL_DEMO_1                  ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PDISPSL,10H         ;PDISPSL=10H
        MOV     DS:WORD PTR PDISPSH,0           ;PDISPSH=0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNPH,0             ;PTNPH=0
        MOV     BX,503CH                        ;<CRLFILL> TL=0,SS=1
        CALL    CRLFILL_EXE
        MOV     SI,2
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(2)
        ;
        MOV     BX,50BCH                        ;<CRLFILL> TL=1,SS=1
        CALL    CRLFILL_EXE
        MOV     SI,3
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(3)
        ;
        MOV     BX,50ACH                        ;<CRLFILL> TL=1,SS=0
        CALL    CRLFILL_EXE
        MOV     SI,4
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(4)
        ;
        MOV     BX,503CH                        ;<R_RECFILL> TL=0,SS=1
        CALL    CRLFILL_EXE_1
        MOV     SI,5
```



```
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(5)
        ;
        MOV     BX,50BCH                        ;<CRLFILL> TL=1,SS=1
        CALL    CRLFILL_EXE_1
        MOV     SI,6
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(6)
        ;
        MOV     BX,50ACH                        ;<CRLFILL> TL=1,SS=0
        CALL    CRLFILL_EXE_1
        MOV     SI,7
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(7)
        ;
        RET
        ;
CRLFILL_EXE:
        MOV     DI,1                            ;DRAW COUNTS
CRLFILL_EXE1:
        CALL    CRLFILL_EXE_L
        DEC     DI
        JNZ     CRLFILL_EXE1
        RET
        ;
CRLFILL_EXE_1:
        MOV     DI,1                            ;DRAW COUNTS
CRLFILL_EXE_11:
        CALL    CRLFILL_EXE_H
        DEC     DI
        JNZ     CRLFILL_EXE_11
        RET
        ;
CRLFILL_EXE_L:
        MOV     SI,OFFSET CRLFILL_DATA_L
        JMP     CRLFILL_1
        ;
CRLFILL_EXE_H:
        MOV     SI,OFFSET CRLFILL_DATA_H
CRLFILL_1:
        PUSH    DS
        MOV     AX,CS
        MOV     DS,AX
        MOV     AX,WORD PTR AGDC_SEG
        MOV     ES,AX
        MOV     CX,12                           ;REPETITION COUNTS
        MOV     DX,0F7D0H                       ;TILE    .. 0F800H
        TEST    BL,10H
        JZ      CRLFILL_2                       ;CHECK IF "SS"=1/0
        MOV     DX,0FA30H                       ;TILE_SS .. 0FA40H
CRLFILL_2:
        TEST    ES:WORD PTR STATUS,1
        JNZ     CRLFILL_2                       ;CHECK IF PPBUSY=1/0
        TEST    BL,10H
```



```
        JNZ     CRLFILL_2_1                     ;CHECK IF "SS"=1/0
        ADD     DX,20H
CRLFILL_2_1:
        ADD     DX,10H
        MOV     ES:WORD PTR PTNPL,DX            ;PTNPL=(?)
        LODSW
        MOV     ES:WORD PTR DXX,AX              ;DXX=(?)
        LODSW
        MOV     ES:WORD PTR XC,AX               ;XC=(?)
        LODSW
        MOV     ES:WORD PTR YC,AX               ;YC=(?)
        TEST    BL,80H
        JNZ     CRLFILL_TL1                     ;CHECK IF "TL"=1/0
        MOV     ES:WORD PTR PTNCNT,0            ;PTNCNT=0
            MOV AX,CX
        AND     AX,7
        JNZ     CRLFILL_3                       ;CHECK IF "WHITE"
        MOV     AX,3
CRLFILL_3:
        MOV     ES:WORD PTR PLANES,AX           ;PLANES=(?)
        MOV     ES:WORD PTR MOD10,1             ;MOD1=0,MOD0=1
        JMP     CRLFILL_TL
        ;
CRLFILL_TL1:
        MOV     ES:WORD PTR PTNCNT,16           ;PTNCNT=16
        MOV     ES:WORD PTR PLANES,7            ;PLANES=7
        MOV     ES:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
CRLFILL_TL:
        MOV     ES:WORD PTR COM,BX              ;<COM.FLAGS>
        LOOP    CRLFILL_2
        POP     DS
        RET
        ;
;
;       < ELLIPSE FILLING DEMO >
;
;
        ;
ELPSFILL_DEMO:
        MOV     SI,35
        CALL    MES_CL_WAIT                     ;MESSAGE(35)
        ;
        MOV     CS:WORD PTR WORK_1,16           ;DH=16
        MOV     CS:WORD PTR WORK_2,9            ;DV=9
        CALL    ELPSFILL_DEMO_1
        MOV     CS:WORD PTR WORK_1,9            ;DH=9
        MOV     CS:WORD PTR WORK_2,16           ;DV=16
        CALL    ELPSFILL_DEMO_1
        RET
        ;
ELPSFILL_DEMO_1:
```



```
        TEST    DS:WORD PTR STATUS,1
        JNZ     ELPSFILL_DEMO_1                 ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PDISPSL,10H         ;PDISPSL=10H
        MOV     DS:WORD PTR PDISPSH,0           ;PDISPSH=0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNPH,0             ;PTNPH=0
        MOV     BX,5C3CH                        ;<ELPSFILL> TL=0,SS=1
        CALL    ELPSFILL_EXE
        MOV     SI,2
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(2)
        ;
        MOV     BX,5CBCH                        ;<ELPSFILL> TL=1,SS=1
        CALL    ELPSFILL_EXE
        MOV     SI,3
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(3)
        ;
        MOV     BX,5CACH                        ;<ELPSFILL> TL=1,SS=0
        CALL    ELPSFILL_EXE
        MOV     SI,4
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(4)
        ;
        MOV     BX,5C3CH                        ;<R_RECFILL> TL=0,SS=1
        CALL    ELPSFILL_EXE_1
        MOV     SI,5
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(5)
        ;
        MOV     BX,5CBCH                        ;<ELPSFILL> TL=1,SS=1
        CALL    ELPSFILL_EXE_1
        MOV     SI,6
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(6)
        ;
        MOV     BX,5CACH                        ;<ELPSFILL> TL=1,SS=0
        CALL    ELPSFILL_EXE_1
        MOV     SI,7
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(7)
        ;
        RET
        ;
ELPSFILL_EXE:
        MOV     DI,1                            ;DRAW COUNTS
ELPSFILL_EXE1:
        CALL    ELPSFILL_EXE_L
        DEC     DI
        JNZ     ELPSFILL_EXE1
        RET
        ;
ELPSFILL_EXE_1:
        MOV     DI,1                            ;DRAW COUNTS
ELPSFILL_EXE_11:
        CALL    ELPSFILL_EXE_H
        DEC     DI
```

```
        JNZ     ELPSFILL_EXE_11
        RET
        ;
ELPSFILL_EXE_L:
        MOV     SI,OFFSET CRLFILL_DATA_L
        JMP     ELPSFILL_1
        ;
ELPSFILL_EXE_H:
        MOV     SI,OFFSET CRLFILL_DATA_H
ELPSFILL_1:
        PUSH    DS
        MOV     AX,CS
        MOV     DS,AX
        MOV     AX,WORD PTR AGDC_SEG
        MOV     ES,AX
        MOV     CX,12                           ;REPETITION COUNTS
        MOV     DX,0F7D0H                       ;TILE    .. 0F800H
        TEST    BL,10H
        JZ      ELPSFILL_2                      ;CHECK IF "SS"=1/0
        MOV     DX,0FA30H                       ;TILE_SS .. 0FA40H
ELPSFILL_2:
        TEST    ES:WORD PTR STATUS,1
        JNZ     ELPSFILL_2                      ;CHECK IF PPBUSY=1/0
        TEST    BL,10H
        JNZ     ELPSFILL_2_1                    ;CHECK IF "SS"=1/0
        ADD     DX,20H
ELPSFILL_2_1:
        ADD      DX,10H
        MOV       ES:WORD PTR PTNPL,DX          ;PTNPL=(?)
        LODSW
        MOV     ES:WORD PTR DY,AX               ;DY=(?)
        LODSW
        MOV     ES:WORD PTR XC,AX               ;XC=(?)
        LODSW
        MOV     ES:WORD PTR YC,AX               ;YC=(?)
        MOV     AX,CS:WORD PTR WORK_1
        MOV     ES:WORD PTR DHH,AX              ;DHH=(?)
        MOV     AX,CS:WORD PTR WORK_2
        MOV     ES:WORD PTR DV,AX               ; DV=(?)
        TEST    BL,80H
        JNZ     ELPSFILL_TL1                    ;CHECK IF "TL"=1/0
        MOV     ES:WORD PTR PTNCNT,0            ;PTNCNT=0
        MOV     AX,CX
        AND     AX,7
        JNZ     ELPSFILL_3                      ;CHECK IF "WHITE"
        MOV     AX,3
ELPSFILL_3:
        MOV     ES:WORD PTR PLANES,AX           ;PLANES=(?)
        MOV     ES:WORD PTR MOD10,1             ;MOD1=0,MOD0=1
        JMP     ELPSFILL_TL
        ;
```



```
ELPSFILL_TL1:
        MOV     ES:WORD PTR PTNCNT,16           ;PTNCNT=16
        MOV     ES:WORD PTR PLANES,7            ;PLANES=7
        MOV     ES:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
ELPSFILL_TL:
        MOV     ES:WORD PTR COM,BX              ;<COM.FLAGS>
        LOOP    ELPSFILL_2
        POP     DS
        RET
        ;
CRLFILL DATA_L:
        DW      150, 200, 160                   ; 1. DX, XC, YC
        DW      145, 490, 310                   ; 2. DX, XC, YC
        DW      140, 480, 180                   ; 3. DX, XC, YC
        DW      135, 230, 290                   ; 4. DX, XC, YC
        DW      130, 240, 200                   ; 5. DX, XC, YC
        DW      125, 450, 270                   ; 6. DX, XC, YC
        DW      120, 440, 220                   ; 7. DX, XC, YC
        DW      115, 270, 250                   ; 8. DX, XC, YC
        DW      110, 280, 240                   ; 9. DX, XC, YC
        DW      105, 410, 230                   ;10. DX, XC, YC
        DW      100, 400, 260                   ;11. DX, XC, YC
        DW       95, 310, 210                   ;12. DX, XC, YC
        ;
CRLFILL DATA_H:
        DW      240, 326, 256                   ; 1. DX, XC, YC
        DW      232, 784, 496                   ; 2. DX, XC, YC
        DW      224, 768, 288                   ; 3. DX, XC, YC
        DW      216, 364, 464                   ; 4. DX, XC, YC
        DW      208, 380, 320                   ; 5. DX, XC, YC
        DW      200, 720, 432                   ; 6. DX, XC, YC
        DW      192, 704, 352                   ; 7. DX, XC, YC
        DW      184, 428, 400                   ; 8. DX, XC, YC
        DW      176, 444, 384                   ; 9. DX, XC, YC
        DW      168, 656, 368                   ;10. DX, XC, YC
        DW      160, 640, 416                   ;11. DX, XC, YC
        DW      152, 492, 336                   ;12. DX, XC, YC
        ;
;
CRLFILL ENDP
PROG    ENDS
        END
```



```
        NAME    TRAFILL
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  TRAFILL_DEMO, TRAFILL_EXE_H
        ;
        EXTRN   MES_CL_WAIT:NEAR, MES_CL_WAIT_FILL:NEAR
        ;
        EXTRN   STATUS:WORD, PDISPSL:WORD, PDISPSH:WORD, PMAX:WORD
        EXTRN   PTNPH:WORD, PTNPL:WORD, X:WORD, Y:WORD, XS:WORD
        EXTRN   YS:WORD, YE:WORD, XE:WORD, PTNCNT:WORD, PLANES:WORD
        EXTRN   MOD10:WORD, COM:WORD, AGDC_SEG:WORD
        ;
TRAFILL PROC    NEAR
        ;
;
;       < TRAPEZOID FILLING DEMO >
;
;
        ;
TRAFILL_DEMO:
        MOV     SI,9                            ;MESSAGE(9)
        CALL    MES_CL_WAIT
        MOV     BX,703CH                        ;<A_TRAFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
        CALL    TRAFILL_EXE_L
        MOV     SI,2
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(2)
        MOV     BX,70BCH                        ;<A_TRAFILL>
                                                ;TL=1,SS=1,WL=1,WR=1
        CALL    TRAFILL_EXE_L
        MOV     SI,3
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(3)
        MOV     BX,70ACH                         ;<A_TRAFILL>
                                                ;TL=1,SS=0,WL=1,WR=1
        CALL    TRAFILL_EXE_L
        MOV     SI,4
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(4)
        MOV     BX,703CH                        ;<A_TRAFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
        CALL    TRAFILL_EXE_H
        MOV     SI,5
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(5)
        MOV     BX,70BCH                        ;<A_TRAFILL>
                                                ;TL=1,SS=1,WL=1,WR=1
        CALL    TRAFILL_EXE_H
        MOV     SI,6
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(6)
        MOV     BX,70ACH                        ;<A_TRAFILL>
                                                ;TL=1,SS=0,WL=1,WR=1
```



```
        CALL    TRAFILL_EXE_H
        MOV     SI,7
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(7)
        RET
        ;
TRAFILL_EXE_L:
        MOV     SI,OFFSET TRAFILL_DATA_L
        JMP     TRAFILL_1
TRAFILL_EXE_H:
        MOV     SI,OFFSET TRAFILL_DATA_H
TRAFILL_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     TRAFILL_1                       ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PDISPSL,10H         ;PDISPSL=10H
        MOV     DS:WORD PTR PDISPSH,0           ;PDISPSH=0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNPH,0             ;PTNPH=0
        PUSH    DS
        MOV     AX,CS
        MOV     DS,AX
        MOV     AX,WORD PTR AGDC_SEG
        MOV     ES,AX
        MOV     CX,10
        MOV     DX,0F7D0H                       ;TILE    .. 0F800H
        TEST    BL,10H
        JZ      TRAFILL_2                       ;CHECK IF "SS"=1/0
        MOV     DX,0FA30H                       ;TILE_SS .. 0FA40H
TRAFILL_2:
        TEST    ES:WORD PTR STATUS,1
        JNZ     TRAFILL_2                       ;CHECK IF PPBUSY=1/0
        TEST    BL,10H
        JNZ     TRAFILL_2_1                     ;CHECK IF "SS"=1/0
        ADD     DX,20H
TRAFILL_2_1:
        ADD     DX,10H
        MOV     ES:WORD PTR PTNPL,DX            ;PTNPL=(?)
        LODSW
        MOV     ES:WORD PTR X,AX                ; X=(?)
        LODSW
         MOV    ES:WORD PTR Y,AX                ; Y=(?)
        LODSW
        MOV     ES:WORD PTR XS,AX               ;XS=(?)
        LODSW
        MOV     ES:WORD PTR YS,AX               ;YS=(?)
        LODSW
        MOV     ES:WORD PTR YE,AX               ;YE=(?)
        LODSW
        MOV     ES:WORD PTR XE,AX               ;XE=(?)
        TEST    BL,80H
        JNZ     TRAFILL_TL1                     ;CHECK IF "TL"=1/0
        MOV      AX,0
```

```
        MOV     ES:WORD PTR PTNCNT,AX           ;PTNCNT=0
        MOV     AX,CX
        AND     AX,7
        JNZ     TRAFILL_3                       ;CHECK IF "WHITE"
        MOV     AX,3
TRAFILL_3:
        MOV     ES:WORD PTR PLANES,AX           ;PLANES=(?)
        MOV     ES:WORD PTR MOD10,1             ;MOD1=0,MOD0=1
        JMP     TRAFILL_TL
TRAFILL_TL1:
        MOV     ES:WORD PTR PTNCNT,16           ;PTNCNT=16
        MOV     ES:WORD PTR PLANES,7            ;PLANES=7
        MOV     ES:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
TRAFILL_TL:
        MOV     ES:WORD PTR COM,BX              ;<COM.FLAGS>
        DEC     CX
        JNZ     TRAFILL_2
        POP     DS
        RET
        ;
TRAFILL_DATA_L:
        DW      0000H,0096H,012CH,001EH,0000H,010EH; 1. X,Y,XS,YS,YE,XE
        DW      0091H,00AFH,0172H,0064H,0019H,0154H; 2. X,Y,XS,YS,YE,XE
        DW      0122H,00C8H,01B8H,00AAH,0032H,019AH; 3. X,Y,XS,YS,YE,XE
        DW      01B3H,00E1H,01FEH,00F0H,004BH,01E0H; 4. X,Y,XS,YS,YE,XE
        DW      0244H,00FAH,0244H,0136H,0064H,0226H; 5. X,Y,XS,YS,YE,XE
        DW      0153H,014AH,027FH,0171H,01DFH,0261H; 6. X,Y,XS,YS,YE,XE
        DW      0158H,0131H,01EEH,012BH,01C6H,021BH; 7. X,Y,XS,YS,YE,XE
        DW      015DH,0118H,015DH,00E5H,01ADH,01D5H; 8. X,Y,XS,YS,YE,XE
        DW      0162H,00FFH,00CCH,009FH,0194H,018FH; 9. X,Y,XS,YS,YE,XE
        DW      0167H,00E6H,003BH,0059H,017BH,0149H;10. X,Y,XS,YS,YE,XE
        ;
TRAFILL_DATA_H:
        DW      0000H,00EAH,020DH,0035H,0000H,01D9H; 1. X,Y,XS,YS,YE,XE
        DW      00FEH,0111H,0288H,00B0H,0027H,0254H; 2. X,Y,XS,YS,YE,XE
        DW      01FCH,0138H,0303H,012BH,004EH,02CFH; 3. X,Y,XS,YS,YE,XE
        DW      02FAH,015FH,037EH,01A6H,0075H,034AH; 4. X,Y,XS,YS,YE,XE
        DW      03F8H,0186H,03F9H,0221H,009CH,03C5H; 5. X,Y,XS,YS,YE,XE
        DW      0251H,01E4H,045EH,0286H,02ECH,042AH; 6. X,Y,XS,YS,YE,XE
        DW      025AH,01BDH,0360H,020CH,02C5H,03B0H; 7. X,Y,XS,YS,YE,XE
        DW      0263H,0196H,0262H,0192H,029EH,0336H; 8. X,Y,XS,YS,YE,XE
        DW      026CH,016FH,0164H,0118H,0277H,02BCH; 9. X,Y,XS,YS,YE,XE
        DW      0275H,0148H,0066H,009EH,0250H,0242H;10. X,Y,XS,YS,YE,XE
        ;
;
TRAFILL ENDP
PROG    ENDS
        END
```



```
        CALL    TRIFILL_EXE
        MOV     SI,4
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(4)
        ;
        MOV     CS:WORD PTR WORK_1,560          ;X=560
        MOV     CS:WORD PTR WORK_2,374          ;Y=374
        MOV     BX,6C3CH                        ;<A_TRIFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
        CALL    TRIFILL_EXE_1
        MOV     SI,5
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(5)
        ;
        MOV     BX,6CBCH                        ;<A_TRIFILL>
                                                ;TL=1,SS=1,WL=1,WR=1
        CALL    TRIFILL_EXE_1
        MOV     SI,6
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(6)
        ;
        MOV     BX,6CACH                        ;<A_TRIFILL>
                                                ;TL=1,SS=0,WL=1,WR=1
        CALL    TRIFILL_EXE_1
        MOV     SI,7
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(7)
        ;
        RET
        ;
TRIFILL_EXE:
        MOV     DI,1                            ;DRAW COUNTS
TRIFILL_EXE1:
        CALL    TRIFILL_EXE_L
        DEC     DI
        JNZ     TRIFILL_EXE1
        RET
        ;
TRIFILL_EXE_1:
        MOV     DI,1                            ;DRAW COUNTS
TRIFILL_EXE_11:
        CALL    TRIFILL_EXE_H
        DEC     DI
        JNZ     TRIFILL_EXE_11
        RET
        ;
TRIFILL_EXE_L:
        MOV     SI,OFFSET TRIFILL_DATA_L
        JMP     TRIFILL_1
TRIFILL_EXE_H:
        MOV     SI,OFFSET TRIFILL_DATA_H
TRIFILL_1:
        PUSH    DS
        MOV     AX,CS
        MOV     DS,AX
```



```
        NAME    TRIFILL
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  TRIFILL_DEMO
        ;
        EXTRN   DMCLO_C_WAIT:NEAR, MES_CL_WAIT:NEAR
        EXTRN   MES_CL_WAIT_FILL:NEAR
        ;
        EXTRN   STATUS:WORD, PDISPSL:WORD, PDISPSH:WORD,PMAX:WORD
        EXTRN   PTNPH:WORD, WORK_1:WORD, WORK_2:WORD, PTNPL:WORD
        EXTRN   X:WORD, Y:WORD, XC:WORD, YC:WORD, XS:WORD, YS:WORD
        EXTRN   PTNCNT:WORD, PLANES:WORD, MOD10:WORD, COM:WORD
        EXTRN   AGDC_SEG:WORD
        ;
TRIFILL PROC    NEAR
        ;
;
;       < TRIANGLE FILLING DEMO >
;
; [TRIFILL_DEMO]
;
        ;
TRIFILL_DEMO:
        MOV     SI,8
        CALL    MES_CL_WAIT                     ;MESSAGE(8)
        ;
TRIFILL_DEMO_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     TRIFILL_DEMO_1                  ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PDISPSL,10H         ;PDISPSL=10H
        MOV     DS:WORD PTR PDISPSH,0           ;PDISPSH=0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PTNPH,0             ;PTNPH=0
        MOV     CS:WORD PTR WORK_1,320          ;X=320
        MOV     CS:WORD PTR WORK_2,240          ;Y=240
        MOV     BX,6C3CH                        ;<A_TRIFILL>
                                                ;TL=0,SS=1,WL=1,WR=1
        CALL    TRIFILL_EXE
        MOV     SI,2
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(2)
        ;
        MOV     BX,6CBCH                        ;<A_TRIFILL>
                                                ;TL=1,SS=1,WL=1,WR=1
        CALL    TRIFILL_EXE
        MOV     SI,3
        CALL    MES_CL_WAIT_FILL                ;MESSAGE(3)
        ;
        MOV     BX,6CACH                        ;<A_TRIFILL>
                                                ;TL=1,SS=0,WL=1,WR=1
```



```
        MOV     AX,WORD PTR AGDC_SEG
        MOV     ES,AX
        MOV     CX,12                           ;REPETITON COUNTS
        MOV     DX,0F7D0H                       ;TILE    .. 0F800H
        TEST    BL,10H
        JZ      TRIFILL_2                       ;CHECK IF "SS"=1/0
          MOV   DX,0FA30H                       ;TILE_SS .. 0FA40H
TRIFILL_2:
        TEST    ES:WORD PTR STATUS,1
        JNZ     TRIFILL_2                       ;CHECK IF PPBUSY=1/0
        TEST    BL,10H
        JNZ     TRIFILL_2_1                     ;CHECK IF "SS"=1/0
        ADD     DX,20H
TRIFILL_2_1:
        ADD     DX,10H
        MOV     ES:WORD PTR PTNPL,DX            ;PTNPL=(?)
        MOV     AX,CS:WORD PTR WORK_1
        MOV     ES:WORD PTR X,AX                ;X=(?)
        MOV     AX,CS:WORD PTR WORK_2
        MOV     ES:WORD PTR Y,AX                ;Y=(?)
        LODSW
        MOV     ES:WORD PTR XC,AX               ;XC=(?)
        LODSW
        MOV     ES:WORD PTR YC,AX               ;YC=(?)
        LODSW
        MOV     ES:WORD PTR XS,AX               ;XS=(?)
        LODSW
        MOV     ES:WORD PTR YS,AX               ;YS=(?)
        TEST    BL,80H
        JNZ     TRIFILL_TL1                     ;CHECK IF "TL"=1/0
        MOV     AX,0
        MOV     ES:WORD PTR PTNCNT,AX           ;PTNCNT=0
        MOV     AX,CX
        AND     AX,7
        JNZ     TRIFILL_3                       ;CHECK IF "WHITE"
        MOV     AX,3
TRIFILL_3:
        MOV     ES:WORD PTR PLANES,AX           ;PLANES=(?)
        MOV     ES:WORD PTR MOD10,1             ;MOD1=0,MOD0=1
        JMP     TRIFILL_TL
        ;
TRIFILL_TL1:
        MOV     ES:WORD PTR PTNCNT,16           ;PTNCNT=16
        MOV     ES:WORD PTR PLANES,7            ;PLANES=7
        MOV     ES:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
TRIFILL_TL:
        MOV     ES:WORD PTR COM,BX              ;<COM.FLAGS>
        DEC     CX
        JNZ     TRIFILL_2
        POP     DS
        RET
```

```
        ;
TRIFILL_DATA_L:
        DW      01CBH,0140H,0098H,01A3H         ; 1. X,Y,DX,DY
        DW      0190H,0178H,0056H,0137H         ; 2. X,Y,DX,DY
        DW      0140H,0190H,0051H,00B8H         ; 3. X,Y,DX,DY
        DW      00F0H,0178H,008DH,0048H         ; 4. X,Y,DX,DY
        DW      00B5H,0140H,00F9H,0006H         ; 5. X,Y,DX,DY
        DW      00A0H,00F0H,0178H,0001H         ; 6. X,Y,DX,DY
        DW      00B5H,00A0H,01E8H,003DH         ; 7. X,Y,DX,DY
        DW      00F0H,0065H,022AH,00A9H         ; 8. X,Y,DX,DY
        DW      0140H,0050H,022FH,0128H         ; 9. X,Y,DX,DY
        DW      0190H,0065H,01F3H,0198H         ;10. X,Y,DX,DY
        DW      01CBH,00A0H,0187H,01DAH         ;11. X,Y,DX,DY
        DW      01E0H,00F0H,0108H,01DFH         ;12. X,Y,DX,DY
        ;
TRIFILL_DATA_H:
        DW      0315H,01FAH,011BH,029EH         ; 1. X,Y,DX,DY
        DW      02B5H,025BH,00ACH,01ECH         ; 2. X,Y,DX,DY
        DW      0230H,027FH,00A6H,011AH         ; 3. X,Y,DX,DY
        DW      01ACH,025BH,0108H,0061H         ; 4. X,Y,DX,DY
        DW      014BH,01FBH,01BAH,0FFF2H        ; 5. X,Y,DX,DY
        DW      0127H,0176H,028CH,0FFECH        ; 6. X,Y,DX,DY
        DW      014BH,00F2H,0345H,004EH         ; 7. X,Y,DX,DY
        DW      01ABH,0091H,03B4H,0100H         ; 8. X,Y,DX,DY
        DW      0230H,006DH,038AH,01D2H         ; 9. X,Y,DX,DY
        DW      02B4H,0091H,0358H,028BH         ;10. X,Y,DX,DY
        DW      0315H,00F1H,02A6H,02FAH         ;11. X,Y,DX,DY
        DW      0339H,0176H,01D4H,0300H         ;12. X,Y,DX,DY
        ;
;
TRIFILL ENDP
PROG    ENDS
        END
```



```
        NAME    PAINT
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  PAINT_DEMO
        ;
        EXTRN   SHRINK_PUT:NEAR, SHRINK_EXE:NEAR, SHRINK_DATA:NEAR
        ;
        EXTRN   STATUS:WORD, XCLMIN:WORD, YCLMIN:WORD, XCLMAX:WORD
        EXTRN   YCLMAX:WORD, MAGETC:WORD, PMAX:WORD, PITCHS:WORD
        EXTRN   PDISPSH:WORD, PDISPSL:WORD, PTNCNT:WORD, MOD10:WORD
        EXTRN   PLANES:WORD, X:WORD, Y:WORD, COM:WORD, PTNPH:WORD
        EXTRN   PTNPL:WORD
        ;
PAINT   PROC    NEAR
        ;
;
;       < PAINT DEMO >
;
;
        ;
PAINT_DEMO:
        CALL    SHRINK_PUT
        MOV     SI,OFFSET SHRINK_DATA
        MOV     CX,6                            ;15/16 --> 10/16
        CALL    SHRINK_EXE
        ;
PAINT_DEMO_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     PAINT_DEMO_1                    ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XCLMIN,5            ;XCLMIN=5
        MOV     DS:WORD PTR YCLMIN,80           ;YCLMIN=80
        MOV     DS:WORD PTR XCLMAX,591          ;XCLMAX=591
        MOV     DS:WORD PTR YCLMAX,744          ;YCLMAX=744
        MOV     DS:WORD PTR MAGETC,0FFH         ;MAGETC=0FFH
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PITCHS,46H          ;PITCHS=46H
        MOV     DS:WORD PTR PDISPSH,1           ;PDISPSH=1
        MOV     DS:WORD PTR PDISPSL,0           ;PDISPSL=0
        MOV     DS:WORD PTR PTNCNT,0FFFFH       ;PTNCNT=0FFFFH
        MOV     DS:WORD PTR MOD10,1             ;MOD1=0,MODO=1
        MOV     DS:WORD PTR PLANES,6            ;PLANES=6
        MOV     DS:WORD PTR X,300               ;X=300
        MOV     DS:WORD PTR Y,560               ;Y=560
        MOV     DS:WORD PTR COM,6834H           ;<PAINT>
                                                ;PMOD=1, TL=0, SS=1
        ;
PAINT_DEMO_2:
        TEST    DS:WORD PTR STATUS,1
        JNZ     PAINT_DEMO_2                    ;CHECK IF PPBUSY=1/0
```



```
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR X,AX                ;X=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR Y,AX                ;Y=(?)
        MOV     DS:WORD PTR PTNCNT,16           ;PTNCNT=16
        ADD     DX,30H                          ;PTNPL+30H --> PTNPL
        MOV     DS:WORD PTR PTNPL,DX            ;PTNPL=(?)
        MOV     DS:WORD PTR COM,BX              ;<COM.FLAGS>
        LOOP    PAINT_NEC_1
        RET
        ;
PAINT_DATA:
        DW       101,642,  269,636,  479,565    ;16/16
        DW        95,463,  253,457,  450,361    ;15/16
        DW        90,217,  237,212,  420,150    ;14/16
        DW       704,660,  840,655, 1011,598    ;13/16
        DW       698,481,  824,477,  982,424    ;12/16
        DW       693,316,  808,312,  952,263    ;11/16
        DW       687,164,  792,161,  923,116    ;10/16
        ;
;
PAINT   ENDP
PROG    ENDS
        END
```



```
        MOV     DS:WORD PTR XCLMIN,596          ;XCLMIN=596
        MOV     DS:WORD PTR YCLMIN,80           ;YCLMIN=80
        MOV     DS:WORD PTR XCLMAX,1115         ;XCLMAX=1115
        MOV     DS:WORD PTR YCLMAX,744          ;YCLMAX=744
        MOV     DS:WORD PTR PLANES,3            ;PLANES=3
        MOV     DS:WORD PTR X,800               ;X=800
        MOV     DS:WORD PTR Y,560               ;Y=560
        MOV     DS:WORD PTR COM,6834H           ;<PAINT>
                                                ;PMOD=1, TL=0, SS=1
        ;
PAINT_DEMO_3:
        TEST    DS:WORD PTR STATUS,1
        JNZ     PAINT_DEMO_3                    ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XCLMIN,0            ;XCLMIN=0
        MOV     DS:WORD PTR YCLMIN,30           ;YCLMIN=30
        MOV     DS:WORD PTR XCLMAX,1119         ;XCLMAX=1119
        MOV     DS:WORD PTR YCLMAX,749          ;YCLMAX=749
        MOV     DS:WORD PTR PDISPSH,0           ;PDISPSH=0
        MOV     DS:WORD PTR PDISPSL,16          ;PDISPSL=16
        MOV     DS:WORD PTR PTNPH,0             ;PTNPH=0
        MOV     DS:WORD PTR PTNCNT,16           ;PTNCNT=16
        MOV     DS:WORD PTR PLANES,7            ;PLANES=7
        MOV     DS:WORD PTR MOD10,1             ;MOD1=0,MODO=1
        MOV     DS:WORD PTR PTNPL,0F800H        ;PTNPL=0F800H  TILE(0)
        MOV     DS:WORD PTR X,600               ;X=600
        MOV     DS:WORD PTR Y,40                ;Y=40
        MOV     DS:WORD PTR COM,68A4H           ;<PAINT>
                                                ;PMOD=1, TL=1, SS=0
        ;
        MOV     SI,OFFSET PAINT_DATA
        MOV     BX,68A4H                        ;<PAINT>
                                                ;PMOD=1, TL=1, SS=0
        MOV     CX,12                           ;REPETITION COUNTS
        CALL    PAINT_NEC
        MOV     CX,9                            ;REPETITION COUNTS
        CALL     PAINT_NEC
PAINT_DEMO_4:
        TEST    DS:WORD PTR STATUS,1
        JNZ     PAINT_DEMO_4                    ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XCLMIN,0            ;XCLMIN=0
        MOV     DS:WORD PTR YCLMIN,0            ;YCLMIN=0
        MOV     DS:WORD PTR XCLMAX,1119         ;XCLMAX=1119
        MOV     DS:WORD PTR YCLMAX,749          ;YCLMAX=749
        MOV     DS:WORD PTR MAGETC,1FFH         ;MAGETC=1FFH
        RET
        ;
PAINT_NEC:
        MOV     DX,0F7D0H
PAINT_NEC_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     PAINT_NEC_1                     ;CHECK IF PPBUSY=1/0
```

```
        NAME    SCROLL
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  SCROLL_DEMO
        ;
        EXTRN   CHR_DRAW_DEMO_H:NEAR
        ;
        EXTRN   WORK_1:WORD, WORK_2:WORD, WORK_3:WORD, WORK_4:WORD
        EXTRN   STATUS:WORD, PLANES:WORD, PMAX:WORD, PITCHS:WORD
        EXTRN   MOD10:WORD, XS:WORD, YS:WORD, X:WORD, Y:WORD
        EXTRN   DHH:WORD, DV:WORD, COM:WORD, PDISPSH:WORD
        EXTRN   PDISPSL:WORD
        ;
SCROLL  PROC    NEAR
        ;
;
;       < SCROLL DEMO >
;
;
        ;
SCROLL_DEMO:
        CALL    CHR_DRAW_DEMO_H
                                        ;END OF BACK-GROUND
        MOV     AX,0                    ;X1=0
        MOV     BX,260                  ;Y1=260
        MOV     SI,560                  ;X2=560
        MOV     DI,477                  ;Y2=477
        MOV     CS:WORD PTR WORK_1,0    ;X3=0
        MOV     CS:WORD PTR WORK_2,256  ;Y3=256
        MOV     CS:WORD PTR WORK_3,560  ;X4=560
        MOV     CS:WORD PTR WORK_4,477  ;Y4=477
        MOV     CX,220                  ;REPETITION COUNTS
SCROLL_DEMO_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     SCROLL_DEMO_1           ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PDISPSH,1   ;PDISPSH=1
        MOV     DS:WORD PTR PDISPSL,0   ;PDISPSL=0
        MOV     DS:WORD PTR PLANES,7    ;PLANES=7
        MOV     DS:WORD PTR PMAX,4      ;PMAX=4
        MOV     DS:WORD PTR PITCHS,46H  ;PITCHS=46H
        MOV     DS:WORD PTR MOD10,11H   ;MOD10=11H
        ;
SCROLL_DEMO_2:
        CALL    SCROLL_P1
        CMP     CX,110
        JB      SCROLL_SKIP
        CALL    SCROLL_P2
        CMP     CX,147
        JB      SCROLL_SKIP
```



```
        CALL    SCROLL_P3
        CMP     CX,166
        JB      SCROLL_SKIP
        CALL    SCROLL_P4
SCROLL_SKIP:
        LOOP    SCROLL_DEMO_2
        RET
        ;
SCROLL_P1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     SCROLL_P1               ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XS,AX       ;XS=(?)
        MOV     DS:WORD PTR YS,BX       ;YS=(?)
        MOV     DS:WORD PTR X,16        ; X=16
        MOV     DS:WORD PTR Y,734       ; Y=734
        MOV     DS:WORD PTR DHH,255     ;DH=255
        MOV     DS:WORD PTR DV,255      ;DV=255
        MOV     DS:WORD PTR COM,840EH   ;<A_COPY_CC> COPY
                                        ;ESE=0,REV=0,ROT=0
                                        ;{MSD},FAST=1
        ADD     AX,1                    ;XS+1
        RET
        ;
SCROLL_P2:
        TEST    DS:WORD PTR STATUS,1
        JNZ     SCROLL_P2               ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XS,SI       ;XS=(?)
        MOV     DS:WORD PTR YS,DI       ;YS=(?)
        MOV     DS:WORD PTR X,288       ; X=288
        MOV     DS:WORD PTR Y,734       ; Y=734
        MOV     DS:WORD PTR DHH,255     ;DH=255
        MOV     DS:WORD PTR DV,255      ;DV=255
        MOV     DS:WORD PTR COM,840EH   ;<A_COPY_CC> COPY
                                        ;ESE=0,REV=0,ROT=0
                                        ;{MSD},FAST=1
        SUB     DI,2                    ;YS-2
        RET
        ;
SCROLL_P3:
        TEST    DS:WORD PTR STATUS,1
        JNZ     SCROLL_P3               ;CHECK IF PPBUSY=1/0
        MOV     DX,CS:WORD PTR WORK_1
        MOV     DS:WORD PTR XS,DX       ;XS=(?)
        MOV     DX,CS:WORD PTR WORK_2
        MOV     DS:WORD PTR YS,DX       ;YS=(?)
        MOV     DS:WORD PTR X,560       ; X=560
        MOV     DS:WORD PTR Y,734       ; Y=734
        MOV     DS:WORD PTR DHH,255     ;DH=255
        MOV     DS:WORD PTR DV,255      ;DV=255
        MOV     DS:WORD PTR COM,840CH   ;<A_COPY_CC> COPY
                                        ;ESE=0,REV=0,ROT=0
```



```
                                        ;{MSD},FAST=0 <!!BUG!!>
        ADD     CS:WORD PTR WORK_1,3    ;XS+3
        ADD     CS:WORD PTR WORK_2,3    ;YS+3
        RET
        ;
SCROLL_P4:
        TEST    DS:WORD PTR STATUS,1
        JNZ     SCROLL_P4               ;CHECK IF PPBUSY=1/0
        MOV     DX,CS:WORD PTR WORK_3
        MOV     DS:WORD PTR XS,DX       ;XS=(?)
        MOV     DX,CS:WORD PTR WORK_4
        MOV     DS:WORD PTR YS,DX       ;YS=(?)
        MOV     DS:WORD PTR X,832       ; X=832
        MOV     DS:WORD PTR Y,734       ; Y=734
        MOV     DS:WORD PTR DHH,255     ;DH=255
        MOV     DS:WORD PTR DV,255      ;DV=255
        MOV     DS:WORD PTR COM,840CH   ;<A_COPY_CC> COPY
                                        ;ESE=0,REV=0,ROT=0
                                        ;{MSD},FAST=0 <!!BUG!!>
        SUB     CS:WORD PTR WORK_3,4    ;XS-4
        SUB     CS:WORD PTR WORK_4,4    ;YS-4
        RET
        ;
;
SCROLL  ENDP
PROG    ENDS
        END
```



```
        NAME    FRCOPY
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  FRCOPY_DEMO
        ;
        EXTRN   PUT_DATA_TRAN:NEAR
        ;
        EXTRN   STATUS:WORD, DHH:WORD, DV:WORD, X:WORD, Y:WORD
        EXTRN   PLANES:WORD, PMAX:WORD, MOD10:WORD, COM:WORD
        EXTRN   PITCHS:WORD, XS:WORD, YS:WORD, DXX:WORD, DY:WORD
        EXTRN   XE:WORD, YE:WORD, MAGETC:WORD
        ;
FRCOPY  PROC    NEAR
        ;
;
;       < FR_COPY DEMO >
;
;
        ;
FRCOPY_DEMO:
        TEST    DS:WORD PTR STATUS,1
        JNZ     FRCOPY_DEMO             ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR DHH,255     ;DH=255
        MOV     DS:WORD PTR DV,255      ;DV=255
        MOV     DS:WORD PTR X,30        ;X=30
        MOV     DS:WORD PTR Y,720       ;Y=720
        MOV     DS:WORD PTR PLANES,7    ;PLANES=7
        MOV     DS:WORD PTR PMAX,4      ;PMAX=4
        MOV     DS:WORD PTR MOD10,0     ;MOD1=0,MODO=0
        MOV     DS:WORD PTR COM,980FH   ;<PUT_C>
                                        ;ROT=0,{MSD}
        PUSH    DS
        MOV     AX,6C00H                ;SEGMENT "MAAJAN"
        MOV     DS,AX
        MOV     CX,3000H                ;256 X 256 X 3 PLANES
        CALL    PUT_DATA_TRAN
        POP     DS
                                        ;END OF "PUT"
        MOV     SI,OFFSET FRCOPY_DATA
        MOV     CX,16                   ;REPETITION COUNTS
FRCOPY_DEMO_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     FRCOPY_DEMO_1           ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PITCHS,46H  ;PITCHS=46H
        MOV     DS:WORD PTR XS,30       ;XS=30
        MOV     DS:WORD PTR YS,720      ;YS=720
        MOV     DS:WORD PTR X,704       ; X=704
        MOV     DS:WORD PTR Y,375       ; Y=375
        MOV     DS:WORD PTR DHH,255     ;DH=255
```

```
        MOV     DS:WORD PTR DV,255              ;DV=255
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR DXX,AX              ;DXX=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR DY,AX               ;DY=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR XE,AX               ;XE=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR YE,AX               ;YE=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR MAGETC,AX           ;MAGETC=(?)
        MOV     DS:WORD PTR COM,843EH           ;<A_COPY_CC> FR_COPY
                                                ;ROT=0,{MSD}
        LOOP    FRCOPY_DEMO_1
        RET
        ;
FRCOPY_DATA:
        DW       2,  1,  1, -2, 1DDH            ;FR(0)
        DW       2,  2,  2, -2, 1AAH            ;FR(1)
        DW       1,  2,  2, -1, 1DDH            ;FR(2)
        DW       0,  2,  2,  0, 1FFH            ;FR(3)
        DW      -1,  2,  2,  1, 1DDH            ;FR(4)
        DW      -2,  2,  2,  2, 1AAH            ;FR(5)
        DW      -2,  1,  1,  2, 1DDH            ;FR(6)
        DW      -2,  0,  0,  2, 1FFH            ;FR(7)
        DW      -2, -1, -1,  2, 1DDH            ;FR(8)
        DW      -2, -2, -2,  2, 1AAH            ;FR(9)
        DW      -1, -2, -2,  1, 1DDH            ;FR(A)
        DW       0, -2, -2,  0, 1FFH            ;FR(B)
        DW       1, -2, -2, -1, 1DDH            ;FR(C)
        DW       2, -2, -2, -2, 1AAH            ;FR(D)
        DW       2, -1, -1, -2, 1DDH            ;FR(E)
        DW       2,  0,  0, -2, 1FFH            ;FR(F)
        ;
;
FRCOPY  ENDP
PROG    ENDS
        END
```

-2-

```
        NAME    MESSAGE
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  MESSAGE
        ;
        EXTRN   WAIT:NEAR
        ;
        EXTRN   STATUS:WORD, PMAX:WORD, PITCHS:WORD, DHH:WORD
        EXTRN   DV:WORD, MOD10:WORD, PLANES:WORD, EAD2L:WORD
        EXTRN   EAD2H:WORD, dAD2:BYTE, X:WORD, Y:WORD, COM:WORD
        EXTRN   FLAG:BYTE, DXX:WORD
        ;
MESSAGE PROC    NEAR
        ;
;
;       < DRAW PREVIOUSLY ASSIGNED MESSAGE >
;
;PRINT MESSAGE ON PREDETERMINED PAGE (SI;  0 TO (?) )
        ;
        TEST    DS:WORD PTR STATUS,1
        JNZ     MESSAGE                         ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PITCHS,2            ;PITCHS=2
        MOV     DS:WORD PTR DHH,23              ;DH=23
        MOV     DS:WORD PTR DV,23               ;DV=23
        MOV     DS:WORD PTR MOD10,97H           ;MOD1=9,MOD0=7
        MOV     DS:WORD PTR PLANES,6            ;PLANES=6
        PUSH    DS
        MOV     AX,DS
        MOV     ES,AX
        MOV     BX,2                            ;X=2
        MOV     CX,746                          ;Y=746
        MOV     AX,54F0H                        ;PAGE ADDRESS SEGMENT
        MOV     DS,AX
        ADD     SI,SI
        MOV     SI,[SI]
        MOV     AX,5500H                        ;MESSAGE SEGMENT
        MOV     DS,AX
MES_2:
        MOV     AL,[SI]
        CMP     AL,80H
        JNC     MES_1                           ;CHECK IF 2 BYTES CODE
        CMP     AL,0DH
        JZ      MES_SKIP2                       ;CHECK IF "CR/LF"
        ;
;COLOR ATTRIBUTE
        ;
        MOV     DX,1
        CMP     AL,4DH
```

-1-

```
        JZ      CHG_COLOR                       ;"M"=MAGENTA
        INC     DX
        CMP     AL,43H
        JZ      CHG_COLOR                       ;"C"=CYAN
        INC     DX
        CMP     AL,42H
        JZ      CHG_COLOR                       ;"B"=BLUE
        INC     DX
        CMP     AL,59H
        JZ      CHG_COLOR                       ;"Y"=YELLOW
        INC     DX
        CMP     AL,47H
        JZ      CHG_COLOR                       ;"G"=GREEN
        INC     DX
        CMP     AL,52H
        JZ      CHG_COLOR                       ;"R"=RED
        INC     DX
        CMP     AL,4CH
        JZ      CHG_COLOR                       ;"L"=BLACK
        POP     DS
        RET;
        ;
CHG_COLOR:
        MOV     ES:WORD PTR PLANES,DX           ;PLANES=(?)
        INC     SI
        JMP     MES_2
        ;
MES_SKIP2:
        MOV     BX,2                            ;X=2
        SUB     CX,30                           ;Y=Y-30
        INC     SI
        INC     SI
        JMP     MES_2                           ;NEXT CHARACTER
        ;
;TRANSLATE SHIFT-JIS CODE TO JIS CODE
        ;
MES_1:
        MOV     AX,[SI]
        XCHG    AL,AH
        SUB     AH,70H
        OR      AL,AL
        JNS     SJIS_JIS_1
        DEC     AL
SJIS_JIS_1:
        ADD     AH,AH
        CMP     AL,9EH
        JC      SJIS_JIS_2
        SUB     AL,5EH
        JMP     SJIS_JIS_3
SJIS_JIS_2:
        DEC     AH
```

-2-

```
SJIS_JIS_3:
        SUB     AL,1FH
        ;
;ADJUST JIS CODE TO BOARD HARDWARE
        ;
        MOV     DX,1
        SHL     AL,1
        SHL     AX,1
        SHL     AX,1
        RCL     DX,1
        SHL     AX,1
        RCL     DX,1
        SHL     AX,1
        RCL     DX,1
        SHL     AX,1
        RCL     DX,1
        ;
        POP     DS
        MOV     DS:WORD PTR EAD2L,AX            ;EAD2L=(?)
        MOV     DS:WORD PTR EAD2H,DX            ;EAD2H=(?)
        MOV     DS:BYTE PTR dAD2,0              ;dAD2=0
        MOV     DS:WORD PTR X,BX                ;X=(?)
        MOV     DS:WORD PTR Y,CX                ;Y=(?)
        ADD     BX,28                           ;X=X+28
        MOV     AX,8008H                        ;<A_COPY_AC> COPY
                                                ;ESE=0,REV=0,ROT=0,{MD}
                                                ;FAST=0 <<!!BUG!!>>
        TEST    CS:FLAG,2
        JZ      MES_3                           ;CHECK IF "SLANT"=1/0
        MOV     AX,8009H                        ;<A_COPY_AC> COPY
                                                ;ESE=0,REV=0,ROT=0,{MD}
                                                ;SLANT
        MOV     DS:WORD PTR DXX,-4              ;DXX=-4
MES_3:
        MOV     DS:WORD PTR COM,AX              ;<COM_FLAGS>
        PUSH    DS
        MOV     AX,5500H                        ;MESSAGE SEGMENT
        MOV     DS,AX
        INC     SI
        INC     SI
        JMP     MES_2                           ;NEXT CHARACTER
        ;
;
MESSAGE ENDP
PROG    ENDS
        END
```

-3-

```
        NAME    CHRDRAW
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  CHR_DRAW_DEMO_L, CHR_DRAW_DEMO_H, CLIP_DEMO
        PUBLIC  CHR_DRAW_DEMO_L_90, CHR_DRAW_DEMO_H_90
        ;
        EXTRN   TRAFILL_EXE_H:NEAR, WAIT:NEAR, DMCLO_C_WAIT:NEAR
        ;
        EXTRN   WORK_2:WORD, WORK_3:WORD, WORK_4:WORD, WORK_5:WORD
        EXTRN   WORK_6:WORD, STATUS:WORD, PMAX:WORD, PITCHS:WORD
        EXTRN   EAD2H:WORD, WORK_1:WORD, EAD2L:WORD, X:WORD, Y:WORD
        EXTRN   DHH:WORD, DV:WORD, MOD10:WORD, PLANES:WORD, COM:WORD
        EXTRN   DXX:WORD, FLAG:WORD, MAGETC:WORD
        EXTRN   XCLMIN:WORD, YCLMIN:WORD, XCLMAX:WORD, YCLMAX:WORD
        ;
CHRDRAW PROC    NEAR
        ;
;
;       < CHARACTER FONT EXPANSION DEMO >
;
;
        ;
CHR_DRAW_DEMO_L:
        MOV     CS:WORD PTR WORK_2,4            ;X=4
        MOV     CS:WORD PTR WORK_3,476          ;Y=476
        MOV     CS:WORD PTR WORK_4,32           ;BODY (H)=32
        MOV     CS:WORD PTR WORK_5,30           ;BODY (V)=30
        MOV     CS:WORD PTR WORK_6,1014H        ;20 CHARS X 16 ROWS
        CALL    CHR_DRAW_EXE_ALL
        RET
        ;
CHR_DRAW_DEMO_H:
        MOV     CS:WORD PTR WORK_2,2            ;X=2
        MOV     CS:WORD PTR WORK_3,746          ;Y=746
        MOV     CS:WORD PTR WORK_4,28           ;BODY (H)=28
        MOV     CS:WORD PTR WORK_5,30           ;BODY (V)=30
        MOV     CS:WORD PTR WORK_6,1928H        ;40 CHARS X 25 ROWS
        CALL    CHR_DRAW_EXE_ALL
        RET
        ;
CHR_DRAW_EXE_ALL:
        TEST    DS:WORD PTR STATUS,1
        JNZ     CHR_DRAW_EXE_ALL                ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PITCHS,2            ;PITCHS=2
        MOV     DS:WORD PTR EAD2H,16H           ;EAD2H=16H
        MOV     CS:WORD PTR WORK_1,0            ;PLANES=0
        TEST    CS:BYTE PTR FLAG,2
        MOV     BX,8008H                        ;<A_COPY_AC> COPY
```



```
        MOV     CS:WORD PTR WORK_4,28           ;BODY (H)=28
        MOV     CS:WORD PTR WORK_5,30           ;BODY (V)=30
        MOV     CS:WORD PTR WORK_6,2819H        ;25 CHARS X 40 ROWS
        CALL    CHR_DRAW_EXE_ALL_90
        RET
        ;
CHR_DRAW_EXE_ALL_90:
        TEST    DS:WORD PTR STATUS,1
        JNZ     CHR_DRAW_EXE_ALL_90             ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PITCHS,2            ;PITCHS=2
        MOV     DS:WORD PTR EAD2H,16H           ;EAD2H=16H
        MOV     CS:WORD PTR WORK_1,0            ;PLANES=0
        MOV     BX,8018H                        ;<A_COPY_AC> 90_COPY
                                                ;ESE=0,REV=0,ROT=0,{MD}
        MOV     DX,840H                         ;EAD2L=840H (JIS-3021H)
        MOV     CX,CS:WORD PTR WORK_6           ;CHARACTERS X ROWS
        MOV     DI,CS:WORD PTR WORK_3           ;Y=(?)
        MOV     SI,CS:WORD PTR WORK_2           ;X=(?)
CHR_DRAW_1_90:
        MOV     DS:WORD PTR EAD2L,DX            ;EAD2L=(?)
        MOV     DS:WORD PTR X,SI                ;X=(?)
        MOV     DS:WORD PTR Y,DI                ;Y=(?)
        MOV     DS:WORD PTR DHH,23              ;DH=23
        MOV     DS:WORD PTR DV,23               ;DV=23
        MOV     DS:WORD PTR MOD10,97H           ;MOD1=9,MODO=7
        INC     CS:WORD PTR WORK_1              ;PLANES+1 --> PLANES
        MOV     AX,CS:WORD PTR WORK_1
        MOV     DS:WORD PTR PLANES,AX           ;PLANES=(?)
        MOV     DS:WORD PTR COM,BX              ;<COM.FLAGS>
        ADD     DX,40H                          ;CHARACTER CODE + 40H
        ADD     DI,CS:WORD PTR WORK_5           ;Y+(?)
        DEC     CL
        JNZ     CHR_DRAW_1_90
        MOV     CL,CS:BYTE PTR WORK_6
        ADD     SI,CS:WORD PTR WORK_4           ;X+(?)
        MOV     DI,CS:WORD PTR WORK_3           ;Y=(?)
        DEC     CH
        JNZ     CHR_DRAW_1_90
        RET
        ;
;
;       < CLIPPING DEMO >
;
;
        ;
CLIP_DEMO:
        CALL    CHR_DRAW_DEMO_H
                                                ;END OF BACK-GROUND
CLIP_DEMO_1:
        TEST    DS:WORD PTR STATUS,1
```



```
                                                ;ESE=0,REV=0,ROT=0,{MD}
                                                ;FAST=0 <<!!BUG!!>>
        JZ      CHR_DRAW_2                      ;CHECK IF "SLANT"=1/0
        MOV     BX,8009H                        ;<A_COPY_AC> COPY
                                                ;ESE=0,REV=0,ROT=0,{MD}
                                                ;SLANT
        MOV     DS:WORD PTR DXX,-4              ;DXX=-4
CHR_DRAW_2:
        MOV     DX,840H                         ;EAD2L=840H (JIS-3021H)
        MOV     CX,CS:WORD PTR WORK_6           ;CHARACTERS X ROWS
        MOV     DI,CS:WORD PTR WORK_3           ;Y=(?)
        MOV     SI,CS:WORD PTR WORK_2           ;X=(?)
CHR_DRAW_1:
        INC     CS:WORD PTR WORK_1              ;PLANES+1 --> PLANES
        MOV     AX,CS:WORD PTR WORK_1
        MOV     DS:WORD PTR PLANES,AX           ;PLANES=(?)
        MOV     DS:WORD PTR EAD2L,DX            ;EAD2L=(?)
        MOV     DS:WORD PTR X,SI                ;X=(?)
        MOV     DS:WORD PTR Y,DI                ;Y=(?)
        MOV     DS:WORD PTR DHH,23              ;DH=23
        MOV     DS:WORD PTR DV,23               ;DV=23
        MOV     DS:WORD PTR MOD10,97H           ;MOD1=9,MODO=7
        MOV     DS:WORD PTR COM,BX              ;<COM.FLAGS>
        ADD     DX,40H                          ;CHARACTER CODE + 40H
        ADD     SI,CS:WORD PTR WORK_4           ;X+(?)
        DEC     CL
        JNZ     CHR_DRAW_1
        MOV     CL,CS:BYTE PTR WORK_6
        SUB     DI,CS:WORD PTR WORK_5           ;Y-(?)
        MOV     SI,CS:WORD PTR WORK_2           ;X=(?)
        DEC     CH
        JNZ     CHR_DRAW_1
        RET
        ;
;
;       < 90 DEGREES ROTATED CHARACTER FONT EXPANSION DEMO >
;
;
        ;
CHR_DRAW_DEMO_L_90:
        MOV     CS:WORD PTR WORK_2,2            ;X=2
        MOV     CS:WORD PTR WORK_3,3              ;Y=3
        MOV     CS:WORD PTR WORK_4,28           ;BODY (H)=28
        MOV     CS:WORD PTR WORK_5,30           ;BODY (V)=30
        MOV     CS:WORD PTR WORK_6,1510H        ;16 CHARS X 21 ROWS
        CALL    CHR_DRAW_EXE_ALL_90
        RET
        ;
CHR_DRAW_DEMO_H_90:
        MOV     CS:WORD PTR WORK_2,2            ;X=2
        MOV     CS:WORD PTR WORK_3,3            ;Y=3
```



```
        JNZ     CLIP_DEMO_1
        MOV     DS:WORD PTR MAGETC,1FFH         ;MAGETC=0FFH NON-CLIP
        MOV     DS:WORD PTR XCLMIN,200          ;XCLMIN=200
        MOV     DS:WORD PTR YCLMIN,180          ;YCLMIN=180
        MOV     DS:WORD PTR XCLMAX,900          ;XCLMAX=900
        MOV     DS:WORD PTR YCLMAX,570          ;YCLMAX=570
        CALL    CLIP_DEMO_EXE
        ;
        CALL    CHR_DRAW_DEMO_H
                                                ;END OF BACK-GROUND
CLIP_DEMO_2:
        TEST    DS:WORD PTR STATUS,1
        JNZ     CLIP_DEMO_2
        MOV     DS:WORD PTR MAGETC,0FFH         ;MAGETC=0FFH CLIP-OUT
        CALL    CLIP_DEMO_EXE
        ;
        CALL    CHR_DRAW_DEMO_H
                                                ;END OF BACK-GROUND
CLIP_DEMO_3:
        TEST    DS:WORD PTR STATUS,1
        JNZ     CLIP_DEMO_3
        MOV     DS:WORD PTR MAGETC,2FFH         ;MAGETC=0FFH CLIP-IN
        ;
CLIP_DEMO_EXE:
        MOV     BX,70ACH                        ;<A_TRAFILL>
                                                ;TL=1,SS=0,WL=1,WR=1
        CALL    TRAFILL_EXE_H
        MOV     AX,40
        CALL    WAIT
        CALL    DMCLO_C_WAIT
        RET
        ;
;
CHRDRAW ENDP
PROG    ENDS
        END
```

```
        NAME    GETPUT
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  GET_PUT_DEMO, PUT_DATA_TRAN
        ;
        EXTRN   WAIT:NEAR
        ;
        EXTRN   STATUS:WORD, DHH:WORD, DV:WORD, X:WORD, Y:WORD
        EXTRN   PLANES:WORD, PMAX:WORD, MOD10:WORD, COM:WORD
        EXTRN   PITCHS:WORD, PDISPSH:WORD, PDISPSL:WORD, AGDC_SEG:WORD
;
GETPUT  PROC    NEAR
        ;
;
;       < GET/PUT DEMO >
;
;
        ;
GET_PUT_DEMO:
        CALL    INIT_PUT_DATA
        ;
        CALL    PUT_GET_DEMO_1
        CALL    PUT_GET_DEMO_1
        CALL    PUT_GET_DEMO_1
        CALL    PUT_DEMO_EXE
        RET
        ;
INIT_PUT_DATA:
        PUSH    DS
        MOV     AX,6600H                        ;SEGMENT "MAAJAN_BUF"
        MOV     DS,AX
        MOV     AX,6C00H                        ;SEGMENT "MAAJAN"
        MOV     ES,AX
        MOV     SI,0
        MOV     DI,0
        MOV     CX,3000H                        ;256 X 256 X 3 PLANES
        REPZ    MOVSW
        POP     DS
        RET
        ;
PUT_GET_DEMO_1:
        CALL    PUT_DEMO_EXE
        CALL    GET_DEMO_EXE
        MOV     AX,10
        CALL    WAIT
        RET
        ;
PUT_DEMO_EXE:
        MOV     SI,OFFSET GET_PUT_DATA
```



```
        MOV     DS:WORD PTR X,705               ; X=705
        MOV     DS:WORD PTR Y,700               ; Y=700
        MOV     DS:WORD PTR PITCHS,46H          ;PITCHS=46H
        MOV     DS:WORD PTR PDISPSH,1           ;PDISPSH=1
        MOV     DS:WORD PTR PDISPSL,0           ;PDISPSL=0
        MOV     DS:WORD PTR COM,9A0EH           ;<GET_C>
                                                ;{MSD}
        MOV     CX,3000H                        ;256 X 256 X 3 PLANES
        MOV     AX,6C00H                        ;SEGMENT "MAAJAN"
        MOV     ES,AX
        MOV     CX,3000H                        ;256 X 256 X 3 PLANES
GET_DATA_TRAN:
        MOV     SI,3EH                          ;ASSIGN GET/PUT PORT
        MOV     DI,0
GET_DATA_TRAN_1:
                MOV     AX,[SI]
        MOV     ES:[DI],AX
        INC     DI
        INC     DI
        LOOP    GET_DATA_TRAN_1
        RET
        ;
GET_PUT_DATA:
        DW      150,700, 960,445                ;1.--> 2. X,Y
        DW      960,315, 150, 60                ;3.--> 4. X,Y
        ;
;
GETPUT  ENDP
PROG    ENDS
        END
```



```
        MOV     DX,4                            ;REPETITION COUNTS
        MOV     BX,980FH                        ;<PUT_C>
                                                ;ROT=0,{MSD}
PUT_DEMO_EXE_1:
        TEST    DS:WORD PTR STATUS,1
        JNZ     PUT_DEMO_EXE_1                  ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR DHH,255             ;DH=255
        MOV     DS:WORD PTR DV,255              ;DV=255
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR X,AX                ;X=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR Y,AX                ;Y=(?)
        MOV     DS:WORD PTR PLANES,7            ;PLANES=7
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
        MOV     DS:WORD PTR COM,BX              ;<COM.FLAG>
        PUSH    DS
        MOV     AX,6C00H                        ;SEGMENT "MAAJAN"
        MOV     DS,AX
        MOV     CX,3000H                        ;256 X 256 X 3 PLANES
        CALL    PUT_DATA_TRAN
        POP     DS
        ADD     BX,20H                          ;CHANGE ROTATION
        DEC     DX
        JNZ     PUT_DEMO_EXE_1
        RET
        ;
PUT_DATA_TRAN:
        PUSH    SI
        MOV     AX,WORD PTR AGDC_SEG
        MOV     ES,AX
        MOV     SI,0
        MOV     DI,3EH                          ;ASSIGN GET/PUT PORT
PUT_DATA_TRAN_1:
        MOV     AX,[SI]
        MOV     ES:[DI],AX
        INC     SI
        INC     SI
        LOOP    PUT_DATA_TRAN_1
        POP     SI
        RET
        ;
GET_DEMO_EXE:
        TEST    DS:WORD PTR STATUS,1
        JNZ     GET_DEMO_EXE                    ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR DHH,255             ;DH=255
        MOV     DS:WORD PTR DV,255              ;DV=255
```



```
        NAME    ENLARGE
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  ENLARGE_DEMO
        ;
        EXTRN   STATUS:WORD, PITCHS:WORD, PMAX:WORD, MOD10:WORD
        EXTRN   X:WORD, Y:WORD, XS:WORD, YS:WORD, DHH:WORD, DV:WORD
        EXTRN   MAGETC:WORD, COM:WORD, PDISPSH:WORD, PDISPSL:WORD
        ;
ENLARGE PROC    NEAR
        ;
;
;       < ENLARGEMENT COPY DEMO >
;
;
        ;
ENLARGE_DEMO:
        TEST    DS:WORD PTR STATUS,1
        JNZ     ENLARGE_DEMO                    ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PDISPSH,1           ;PDISPSH=1
        MOV     DS:WORD PTR PDISPSL,0           ;PDISPSL=0
        MOV     DS:WORD PTR PITCHS,46H          ;PITCHS=46H
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
        MOV     SI,OFFSET ENLARGE_DATA
        MOV     CX,16                           ;REPETITION COUNTS
        MOV     DX,1FFH                         ;MAGH=15,MAGV=15
        MOV     BX,849FH                        ;<A_COPY_CC> ES_COPY
                                                ;ROT=0,{MSD}
ENLARGE_EXE:
        TEST    DS:WORD PTR STATUS,1
        JNZ     ENLARGE_EXE                     ;CHECK IF PPBUSY=1/0
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR X,AX                ;X=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR Y,AX                ;Y=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR XS,AX               ;XS=(?)
        MOV     DS:WORD PTR YS,374              ;YS=374
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR DHH,AX              ;DHH=(?)
```

```
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR DV,AX               ;DV=(?)
        MOV     DS:WORD PTR MAGETC,DX           ;MAGETC=(?)
        MOV     DS:WORD PTR COM,BX              ;<COM.FLAG>
        SUB     DX,11H                          ;MAGH-1 -->MAGH
                                                ;MAGV-1 -->MAGV
        LOOP    ENLARGE_EXE
        RET
;
ENLARGE_DATA:
        DW       30,749,389,339,109             ;X,Y,XS,DHH,DV  16/16
        DW       30,629,400,318,102             ;X,Y,XS,DHH,DV  16/15
        DW       30,509,410,297, 95             ;X,Y,XS,DHH,DV  16/14
        DW       30,389,421,275, 88             ;X,Y,XS,DHH,DV  16/13
        DW       30,269,432,254, 82             ;X,Y,XS,DHH,DV  16/12
        DW       30,149,442,233, 75             ;X,Y,XS,DHH,DV  16/11
        DW      390,749,453,212, 68             ;X,Y,XS,DHH,DV  16/10
        DW      390,629,464,190, 61             ;X,Y,XS,DHH,DV  16/ 9
        DW      390,509,474,169, 54             ;X,Y,XS,DHH,DV  16/ 8
        DW      390,149,485,148, 47             ;X,Y,XS,DHH,DV  16/ 7
        DW      750,749,495,127, 40             ;X,Y,XS,DHH,DV  16/ 6
        DW      750,629,506,105, 33             ;X,Y,XS,DHH,DV  16/ 5
        DW      750,509,517, 84, 27             ;X,Y,XS,DHH,DV  16/ 4
        DW      750,389,527, 63, 20             ;X,Y,XS,DHH,DV  16/ 3
        DW      750,269,538, 42, 13             ;X,Y,XS,DHH,DV  16/ 2
        DW      750,149,549, 20,  6             ;X,Y,XS,DHH,DV  16/ 1
        ;
;
ENLARGE ENDP
PROG    ENDS
        END
```



```
        NAME    SHRINK
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  SHRINK_DEMO, SHRINK_PUT, SHRINK_DATA, SHRINK_EXE
        ;
        EXTRN   MESSAGE_WAIT:NEAR, DMCLO_C_WAIT:NEAR, PUT_DATA_TRAN:NEAR
        ;
        EXTRN   STATUS:WORD, XS:WORD, YS:WORD, X:WORD, Y:WORD
        EXTRN   DHH:WORD, DV:WORD, MAGETC:WORD, MOD10:WORD, COM:WORD
        EXTRN   PMAX:WORD, PLANES:WORD, PITCHS:WORD, PDISPSL:WORD
        EXTRN   PDISPSH:WORD, EAD2H:WORD, dAD2:BYTE, EAD2L:WORD
        ;
SHRINK  PROC    NEAR
        ;
;
;       < SHRINK COPY DEMO >
;
;
        ;
SHRINK_DEMO:
        CALL    SHRINK_PUT                      ;16/16
        MOV     SI,OFFSET SHRINK_DATA
        MOV     CX,6                            ;15/16 --> 10/16
        CALL    SHRINK_EXE
        PUSH    SI
        MOV     SI,21
        CALL    MESSAGE_WAIT                    ;MESSAGE(21)
        POP     SI
        CALL    DMCLO_C_WAIT
        CALL    SHRINK_PUT                      ;16/16
        MOV     CX,9                            ; 9/16 -->  1/16
        CALL    SHRINK_EXE
        ;
        MOV     DX,10FH                         ;MAGETC=10FH
        MOV     CX,16                           ;MAGH=0 --> 15
SHRINK_MAGH:
        TEST    DS:WORD PTR STATUS,1
        JNZ     SHRINK_MAGH                     ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XS,10               ;XS=10
        MOV     DS:WORD PTR YS,739              ; Y=739
        MOV     DS:WORD PTR X,10                ; X=10
        MOV     DS:WORD PTR Y,499               ; Y=499
        MOV     DS:WORD PTR DHH,575             ;DH=575
        MOV     DS:WORD PTR DV,214              ;DV=214
        MOV     DS:WORD PTR MAGETC,DX           ;MAGH=(?),MAGV=0FH
        MOV     DS:WORD PTR MOD10,0             ;MOD1=0,MOD0=0
        MOV     DS:WORD PTR COM,840FH           ;<A_COPY_CC> ES_COPY
                                                ;REV=0,ROT=0,{MSD}
        ADD     DX,10H                          ;MAGH+1 --> MAGH
```



```
        LOOP    SHRINK_MAGH
        ;
        MOV     DX,1F0H                         ;MAGH=15,MAGV=0
        MOV     CX,16                           ;MAGV=0 --> 15
SHRINK_MAGV:
        TEST    DS:WORD PTR STATUS,1
        JNZ     SHRINK_MAGV                     ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR XS,10               ;XS=10
        MOV     DS:WORD PTR YS,739              ;YS=739
        MOV     DS:WORD PTR X,10                ; X=10
        MOV     DS:WORD PTR Y,259               ; Y=259
        MOV     DS:WORD PTR DHH,575             ;DH=575
        MOV     DS:WORD PTR DV,214              ;DV=214
        MOV     DS:WORD PTR MAGETC,DX           ;MAGH=(?),MAGV=0FH
        MOV     DS:WORD PTR COM,840FH           ;<A_COPY_CC> ES_COPY
                                                ;REV=0,ROT=0,{MSD}
        INC     DX                              ;MAGV+1 --> MAGV
        LOOP    SHRINK_MAGV
        RET
        ;
SHRINK_PUT:
        TEST    DS:WORD PTR STATUS,1
        JNZ     SHRINK_PUT                      ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR DHH,575             ;DH=575
        MOV     DS:WORD PTR DV,214              ;DV=214
        MOV     DS:WORD PTR X,10                ; X=10
        MOV     DS:WORD PTR Y,739               ; Y=739
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PLANES,6            ;PLANES=6
        MOV     DS:WORD PTR MOD10,097H          ;MOD1=9,MOD0=7
        MOV     DS:WORD PTR COM,980BH           ;<PUT_C>
                                                ;REV=0,ROT=0,{MD}
        PUSH    DS
        MOV     AX,5C00H                        ;SEGMENT "NECDEMO.PAT"
        MOV     DS,AX
        MOV     CX,7740                         ;576 X 215 X 1 PLANE
        CALL    PUT_DATA_TRAN
        POP     DS
        RET
        ;
SHRINK_EXE:
        TEST    DS:WORD PTR STATUS,3
        JNZ     SHRINK_EXE                      ;CHECK IF PPBUSY=1/0
        MOV     DS:WORD PTR PMAX,4              ;PMAX=4
        MOV     DS:WORD PTR PLANES,6            ;PLANES=6
        MOV     DS:WORD PTR PITCHS,46H          ;PITCHS=46H
        MOV     DS:WORD PTR PDISPSH,1           ;PDISPSH=1
        MOV     DS:WORD PTR PDISPSL,0           ;PDISPSL=0
        MOV     DS:WORD PTR MOD10,87H           ;MOD1=8,MOD0=7
SHRINK_EXE_1:
        TEST    DS:WORD PTR STATUS,1
```



```
        JNZ     SHRINK_EXE_1                    ;CHECK IF PPBUSY=1/0
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR X,AX                ;X=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR Y,AX                ;Y=(?)
        MOV     AX,CS:[SI]
        INC     SI
        INC     SI
        MOV     DS:WORD PTR MAGETC,AX           ;MAGETC=(?)
        MOV     DS:WORD PTR EAD2H,1             ;EAD2H=1 (GO)
        MOV     DS:BYTE PTR dAD2,10             ;dAD2=10 (GO)
        MOV     DS:WORD PTR EAD2L,700           ;EAD2L=700 (GO)
        MOV     DS:WORD PTR DHH,575             ;DH=575
        MOV     DS:WORD PTR DV,214              ;DV=214
        MOV     DS:WORD PTR COM,800BH           ;<A_COPY_AC> ES_COPY
                                                ;REV=0,ROT=0,{MD}
        LOOP    SHRINK_EXE_1
        RET
        ;
;
SHRINK_DATA:
        DW       10,514,1EEH                    ;   15/16  X,Y,MAGETC
        DW       10,302,1DDH, 630,739,1CCH      ;14,13/16  X,Y,MAGETC
        DW      630,554,1BBH, 630,383,1AAH      ;12,11/16  X,Y,MAGETC
        DW      630,225,199H                    ;10   /16  X,Y,MAGETC
        DW      630,739,188H, 630,608,177H      ; 9, 8/16  X,Y,MAGETC
        DW      630,490,166H, 630,386,155H      ; 7, 6/16  X,Y,MAGETC
        DW      630,295,144H, 630,218,133H      ; 5, 4/16  X,Y,MAGETC
        DW      630,154,122H, 630,104,111H      ; 3, 2/16  X,Y,MAGETC
        DW      630, 67,100H                    ;    1/16  X,Y,MAGETC
        ;
;
SHRINK  ENDP
PROG    ENDS
        END
```

```
        NAME    PUSH_POP
PGROUP  GROUP   PROG
PROG    SEGMENT BYTE PUBLIC     'PROG'
        ASSUME  CS:PGROUP
        ;
        PUBLIC  PUSH_EXE, POP_EXE
        ;
PUSH_POP        PROC    NEAR
        ;
POP_EXE:
        POP     ES
        POP     DS
        POP     DI
        POP     SI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        IRET
        ;
PUSH_EXE:
        POP     AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    SI
        PUSH    DI
        PUSH    DS
        PUSH    ES
        PUSH    AX
        RET
        ;
;
PUSH_POP        ENDP
PROG    ENDS
        END
```


I made entire 80286 assembly code above for the purpose of both functional verification and demonstration for customers as well as emulating Graphics BIOS on NEC PC-9800 series replacing the entry points.
The PC-9800 had been most popular 16 bit personal computer in Japan due to implementation of full graphics capability as standard feature but eventually defeated by IBM PC clones that assembled Chips & Technologies' chip sets.
This source code demonstrates how to drive μPD72120 Advanced Graphics Display Controller LSI from system side exhibiting concrete examples.

To see the details, zoom up.