*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

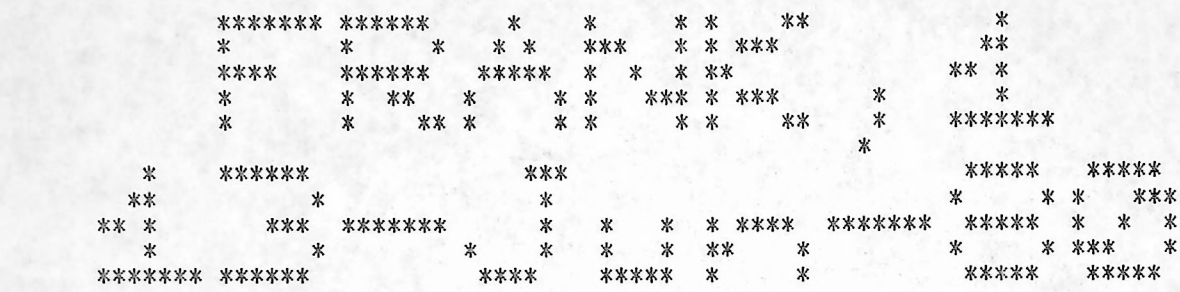
*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***



*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

*** START FRANK,1(511) - LENGTH IN BYTES START ***

```
10 CLEAR 2500
20 DIM S(255)
30 DIM PTR(128)
40 PRINT"DISK INFORMATION SCANNER"
50 INPUT"DRIVE NUMBER,PLATTER";DR,LU
60 LU=(LU*2)AND 3
70 FA=0
80 GOSUB 10000
90 PRINT
100 PRINT"                              VOLUME DESCRIPTOR"
110 PRINT
120 S$=""
130 FOR II=0 TO 19:S$=S$+CHR$(S(II)):NEXT II
140 PRINT"ONE GROUP IS EIGHT PAGES"
150 PRINT"VOLUME NAME IS                    ";S$
160 PRINT"CREATION DATE                     ";S(20);"/";S(21) +
 ;"/";S(22)
170 PRINT"LAST WRITTEN AT                   ";S(23);"/";S(24) +
 ;"/";S(25)
180 PRINT"CURRENT BACKUP                    ";(S(26)+(S(27)*2 +
 56))
190 PRINT"ALLOC POINTER ONE                 ";(S(28)+(S(29)*2 +
 56))
200 PRINT"ALLOC POINTER TWO                 ";(S(30)+(S(31)*2 +
 56))
210 PRINT"ALLOC POINTER THREE               ";(S(32)+(S(33)*2 +
 56))
220 PRINT"DIRECTORY POINTER ONE             ";(S(34)+(S(35)*2 +
 56))
230 PRINT"DIRECTORY POINTER TWO             ";(S(36)+(S(37)*2 +
 56))
240 PRINT"DIRECTORY POINTER THREE           ";(S(38)+(S(39)*2 +
 56))
250 PRINT"POINTER TO OPSYS IMAGE            ";(S(40)+(S(41)*2 +
 56))
260 PRINT"LENGTH OF OPSYS IMAGE IN PAGES";(S(42)+(S(43)*2 +
 56))
270 PRINT"CURRENT ALLOC TABLE POINTER       ";(S(64)+(S(65)*2 +
 56))
280 PRINT"LENGTH OF  ALLOCATION TABLE       ";(S(66)+(S(67)*2 +
 56))
290 PRINT"CURRENT DIRECTORY POINTER         ";(S(68)+(S(69)*2 +
 56))
300 PRINT"LENGTH OF CURRENT DIRECTORY       ";(S(70)+(S(71)*2 +
 56))
310 PRINT"READ WRITE OR READ ONLY           ";S(72)
320 PRINT"1=READ ONLY  3=READ/WRITE
330 PRINT"ADDRESS OF LAST PAGE              ";(S(76)+(S(77)*2 +
 56))
340 PRINT"NUMBER OF FREE GROUPS             ";(S(78)+(S(79)*2 +
 56))
350 PRINT"NUMBER OF RESERVED GROUPS         ";(S(80)+(S(81)*2 +
 56))
```

```
360 PRINT"NUMBER OF UNALLOCATABLE GROUPS";(S(82)+(S(83)*2 +
 56))
370 PRINT"POINTER TO SWAP AREA           ";(S(86)+(S(87)*2 +
 56))
380 PRINT"LENGTH OF SWAP AREA            ";(S(88)+(S(89)*2 +
 56))
390 AL=S(64)+(S(65)*256)
400 DI=S(68)+(S(69)*256)
410 LAL=S(66)+(S(67)*256)'LENGTH IN BYTES
420 LDI=S(70)+(S(71)*256)'LENGTH IN PAGES
430 DIM AL(LAL)
440 FOR II=0 TO LAL:AL(II)=0:NEXT II
450 FOR II=(S(40)+(S(41)*256))/8 TO ((S(42)+(S(43)*256)+S +
 (40)+(S(41)*256))/8)
460 AL(FIX(II/8))=AL(FIX(II/8)) OR (2^(II MOD 8))'ALLOCAT +
 E OPSYS IMAGE
470 NEXT II
480 AL(0)=AL(0)OR 6'ALLOCATE SECOND TWO GROUPS ANY WAY
490 AL(304)=AL(304) OR &HF8'ALLOCATE NONEXISTANT GROUPS A +
 ND VOLUME TABLE
500 PRINT"                                  DIRECTORY TABLE
510 FOR PA=DI TO (DI+LDI)
520 GOSUB 10000
530 FOR HL=0 TO 128 STEP 128
540 TY=S(HL+54)
550 IF TY=0 THEN GOTO 760
560 IF TY=255 THEN PA=DI+LDI:GOTO770
570 PRINT:PRINT
580 S$=""
590 FOR II=0TO23:S$=S$+CHR$(S(HL+II)):NEXTII
600 PRINT"FILE NAME IS                   ";S$
610 PRINT"CREATION DATE                  ";S(HL+24);"/";S(HL +
 +25);"/";S(HL+26)
620 PRINT"LAST WRRITTEN DATE             ";S(HL+27);"/";S(HL +
 +28);"/";S(HL+29)
630 PRINT"PROTECTION MODE                ";S(HL+30)
640 PRINT"FILE STORAGE TYPE              ";TY
650 PRINT"LOGICAL PAGE END OF FILE       ";(S(HL+56)+(S(HL+5 +
 7)*256))
660 PRINT"LOGICAL BYTE END OF FILE       ";(S(HL+58)+(S(HL+5 +
 9)*256))
670 PRINT"NUMBER OF GROUPS ALLOCATED     ";(S(HL+60)+(S(HL+6 +
 1)*256))
680 PRINT"LAST PHYSICALLY ALLOC GROUP    ";(S(HL+62)+(S(HL+6 +
 3)*256))
690 PRINT"                  GROUP POINTERS"
700 IF TY=3 THEN GOTO 1170
710 FOR II=HL+64 TO HL+126 STEP 2
720 GP=S(II)+(S(II+1)*256)
730 IF GP<>65535! THEN PRINT GP;:IF AL(FIX(GP/8))AND(2^(G +
 P MOD 8)) THEN PRINT"<ALLOCATED";ELSE AL(FIX(GP/8))=AL(F +
 IX(GP/8)) OR (2^(GP MOD 8))
740 NEXT II
750 PRINT
```

```
760 NEXT HL
770 NEXT PA
780 PRINT
790 PRINT"          SEARCHING ALLOCATION TABLE FOR DIFFER +
 ENCES"
800 PRINT
810 PA=AL-1
820 FOR K=0 TO (LAL-1)
830 BC=K MOD 256
840 IF BC=0 THEN PA=PA+1:GOSUB 10000'GET PAGE OF ALLOC TABLE
850 IF AL(K)=S(BC)THEN910
860 FOR BIT=0TO7
870 IF (AL(K)AND (2^BIT))=(S(BC)AND(2^BIT)) THEN 900
880 IF (AL(K)AND(2^BIT))<(S(BC)AND(2^BIT))THENPRINT"GROUP +
 ";((K*8)+BIT);"IS ALLOCATED IN THE TABLE BUT NOT USED":G +
 OTO900
890 PRINT"GROUP";((K*8)+BIT);"IS IN USE BUT NOT IN ALLOCA +
 TION TABLE"
900 NEXT BIT
910 NEXT K
920 INPUT"WANT TO PUT NEW ALLOCATION TABLE ON THE DISK";S$
930 IF S$<>"Y" THEN GOTO 1010
940 PA=AL-1
950 FORK=0 TO (LAL-1)
960 BC=K MOD 256
970 S(BC)=AL(K)
980 IF BC=255 THEN PA =PA+1:GOSUB 20000
990 NEXT K
1000 IF (LAL-1)MOD256<>0 THEN PA=PA+1:GOSUB 20000
1010 FS=0
1020 PRINT"CALCULATING NUMBERR OF FRREE GROUPS"
1030 FOR K=0 TO (LAL-1)
1040 FOR BIT =0 TO 7
1050 IF (AL(K)AND(2^BIT))=0 THEN FS=FS+1
1060 NEXT BIT
1070 NEXT K
1080 PRINT"ACTUAL NUMBER OF FREE GROUPS";FS
1090 IF S$<>"Y" THEN GOTO 1150
1100 PA=0
1110 GOSUB 10000
1120 S(78)=FSMOD256
1130 S(79)=FIX(FS/256)
1140 GOSUB 20000
1150 '
1160 GOTO1370
1170 'TYPE 3 RETURN TO 660
1180 TEM=PA
1190 FOR II =HL+64 TO HL+126 STEP 2
1200 PTR(((II-HL)-64)/2)=(S(II)+(S(II+1)*256))
1210 NEXT II
1220 FOR UI=0 TO 31
1230 IF PTR(UI)=65535! THEN GOTO 1340
1240 PRINT
1250 PRINT"*";PTR(UI);"*"
```

```
1260 IF AL(FIX(PTR(UI)/8))AND(2^(PTR(UI)MOD8))THEN PRINT" +
 <**ALLOCATED";ELSE AL(FIX(PTR(UI)/8))=AL(FIX(PTR(UI)/8)) +
 OR(2^(PTR(UI)MOD8))
1270 FOR GX=0 TO 7
1280 PA=(PTR(UI)*8)+GX:GOSUB10000
1290 FORJJ=0TO254 STEP2
1300 GP=S(JJ)+(S(JJ+1)*256)
1310 IFGP<>65535! THEN PRINTGP;:IFAL(FIX(GP/8))AND(2^(GPM +
 OD8))THEN PRINT"<ALLOCATED";ELSEAL(FIX(GP/8))=AL(FIX(GP/ +
 8))OR(2^(GPMOD8))
1320 NEXT JJ
1330 NEXT GX
1340 NEXT UI
1350 PA=TEM:GOSUB 10000
1360 GOTO750
1370 '
1380 END
10000 'SUBROUTINE GET PAGE FROM DISK"
10010 SC=PA MOD 24
10020 TK=FIX((PA MOD 48)/24)+LU
10030 CY=FIX(PA/48)
10040 BF=0
10050 A=INP(160)
10060 GOSUB 30290
10070 GOSUB 30000
10080 GOSUB 30260
10090 GOSUB 30000
10100 GOSUB 30180
10110 GOSUB 30000
10120 RETURN
20000 'SUBROUTINE PUT PAGE
20010 SC=PA MOD 24
20020 TK=FIX((PAMOD48)/24)+LU
20030 CY=FIX(PA/48)
20040 BF=0
20050 A=INP(160)
20060 GOSUB 30290
20070 GOSUB 30000
20080 GOSUB 30130
20090 GOSUB 30000
20100 GOSUB 30230
20110 GOSUB 30000
20120 IF E<>0 THEN STOP
20130 RETURN
30000 'CHECK ERROR STATUS
30010 WAIT160,128
30020 E=INP(161)
30030 IF EAND1 THEN PRINT"DRIVE NOT READY"
30040 IF EAND2 THEN PRINT"ILLEGAL SECTOR"
30050 IF EAND4 THEN PRINT"CRC ERROR IN SECTOR DATA"
30060 IF EAND8 THEN PRINT"CRC ERROR IN HEADER"
30070 IF EAND16 THEN PRINT"HEADER HAS WRONG SECTOR"
30080 IF EAND32 THEN PRINT "HEADEER HAS WRONG TRACK"
30090 IF EAND64 THEN PRINT"HEADER HAS WRONGJ HEAD NUMBER"
```

```
30100 IF EAND128 THEN PRINT"PLATTER IS WRITE PROTECTED"
30110 IF (E AND 127) <>0 THEN STOP
30120 RETURN
30130 OUT 167,0: A=INP(163):A=INP(167)
30140 OUT 163,64+BF
30150 WAIT 166,128
30160 FOR II=0 TO 255:OUT167,S(II):NEXT II
30170 RETURN
30180 OUT 167,0
30190 A=INP(161): A=INP(163): A=INP(165)
30200 OUT 163,64+16+BF: WAIT 164,128
30210 FOR II=0 TO 255: S(II) = INP(165):NEXT II
30220 RETURN
30230 OUT 167,SC+TK*32
30240 A=INP(163): OUT 163,32+DR*4+BF
30250 RETURN
30260 OUT 167,SC+TK*32
30270 A=INP (163): OUT 163,32+16+DR*4+BF
30280 RETURN
30290 OUT 167,CY AND 255
30300 A = INP(163): OUT 163,(CY/256)+DR*4
30310 RETURN
```

Total of 5 pages + 2 banner

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

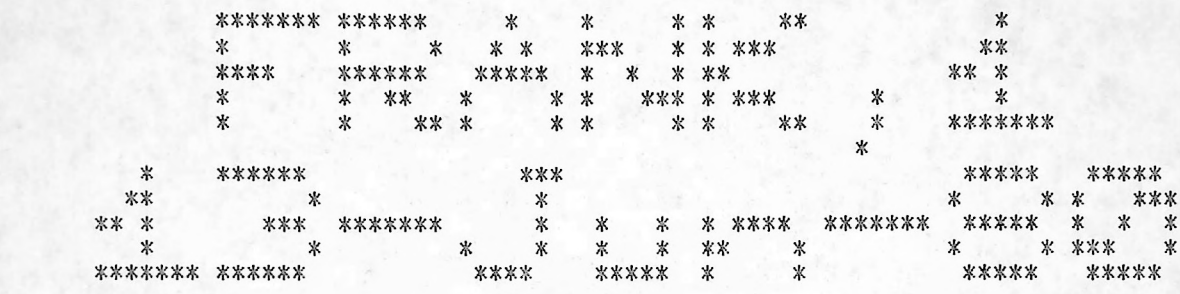
*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***



*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***

*** END FRANK,1(511) - LENGTH IN BYTES END ***