

## COMPILE

```
0100 ! ** HP-75-Compiler **
0110 DIM C$[256],B$[96],R$[256],F$[30],H9$[16],P$[128]
0120 ON ERROR OFF ERROR
0130 PURGE 'ASSEM1'
0140 DELAY 0 @ H9$='0123456789ABCDEF'
0150 FOR I=1 TO 6 @ A(I)=0 @ NEXT I
0160 DISP ' * * A S S E M B L E R * * ' @ WAIT 2
0170 C$='' @ Z5,A,Y,B=0 @ A9,D9=-1 @ P,F=1
0180 ! - Verarbeitung des Textfiles -
0190 INPUT 'Filename: ';A$
0200 ASSIGN # 1 TO A$,TEXT @ ASSIGN # 2 TO 'ASSEM1',TEXT
0210 EDIT A$ @ RENUMBER 10,10,1
0220 EDIT 'COMPILE' @ RESTORE # 1 @ DISP 'Compilation . . . .' @ GOSUB 9000 @ P=1
0230 GOSUB 9970 @ Z5=Z5+1 @ READ # 1,Z5*10 ; B$ @ L=LEN(B$) @ E$='' @ Y=Y+10
0240 I=6 @ IF B$[1,2]='ID' OR B$[1,2]='ZD' THEN C$=C$&B$[8,9]&B$[6,7] @ A=A+2 @ GOTO 500
0250 IF B$[1,3]='DEF' THEN C$=C$&'0000' @ A=A+2 @ GOTO 500
0260 IF B$[1,3]='LAB' THEN A9,D9=0 @ GOTO 1900
0270 IF B$[1,3]='END' AND LEN(B$)#3 THEN 1800
0280 IF B$[1,2]='AS' THEN 1700
0290 IF B$[1,3]='END' THEN C$=C$&'FF' @ GOTO 600
0300 IF L=3 THEN D$=B$[1,3] @ GOTO 400 ELSE D$=B$[1,4]
0310 IF B$[1,3]='BYT' THEN C$=C$&B$[6,7] @ A=A+1 @ GOTO 500
0320 IF B$[1,1]='!' THEN 230
0330 IF B$[I,I]='"' THEN 1500
0340 IF B$[I,I]="R" THEN 1000
0350 IF B$[I,I]="X" THEN 1100
0360 IF B$[I,I]="$" THEN 1200
0370 IF B$[I,I]="=" THEN 1400
0380 IF I=10 AND POS('-',B$[I,I]) THEN 1300
0390 DISP 'ERR L'&STR$(Y) @ GOTO 230
0400 ! - Befehlskodierung -
0410 RESTORE @ Q=128
0420 READ R$ @ IF R$=D$ THEN 440
```

## COMPILE

```
0430 Q=Q+1 @ IF Q=256 THEN DISP 'ERR L'&STR$(Z) @ GOTO 230 ELSE 420
0440 A=A+1 @ C$=C$&FNY$(Q)&E$
0500 ! - Weitergabe des Hex-Codes -
0520 GOTO 230
0600 ! - Vervollstaendigen der Hex-Codes -
0610 PRINT # 2,P ; ' '&C$ @ F=3
0620 Z9=2 @ C$=''
0625 READ # 1,Z9*10 ; B$ @ IF B$[1,3]# 'DEF' THEN 655
0630 IF B$[6,8]='MAI' THEN C=VAL(B$[10,LEN(B$)]) @ C$=C$&FNY$(B(C)) @ GOTO 650
0635 IF B$[6,8]# 'PAR' THEN C$=C$&FNY$(A(LEN(B$)-7)) @ GOTO 650
0640 IF LEN(B$)=10 THEN C$=C$&FNY$(A(2)) @ GOTO 650
0645 C=VAL(B$[12,LEN(B$)]) @ C$=C$&FNY$(C(C))
0650 Z9=Z9+1 @ GOTO 625
0655 IF B$[1,2]# 'ZD' THEN 662
0660 C$=C$&B$[8,9]&B$[6,7] @ Z9=Z9+1 @ GOTO 625
0662 IF B$[1,5]='LABEL' THEN Z9=Z9+1 @ GOTO 625
0665 READ # 2,1 ; R$ @ R$=R$[2] @ IF P>2 THEN READ # 2,2 ; P$ @ P$=P$[2] ELSE P$=''
0670 R$=R$&P$
0675 R$[5,LEN(C$)+4]=C$
0680 PRINT # 2,1 ; ' '&R$[1,64]
0685 IF P>2 THEN PRINT # 2,2 ; ' '&R$[65,128]
0690 READ # 2,P ; C$
0700 ! - Umwandlung Hex-Codes in LEX-File -
0710 I=1354 @ P=P-1
0730 R$=''
0740 FOR J=I TO I+255 @ R$=R$&CHR$(PEKE(J)) @ NEXT J
0750 U=POS(R$,'ASSEM1') @ IF U=0 THEN I=I+256 @ GOTO 740
0760 P1=U-9+I @ U=NUM(R$[U-9,U-9])*256+NUM(R$[U-10,U-10])-32768
0770 W=P*32+LEN(C$)/2-1 @ V=U+14
0780 FOR J=1 TO P
0790 FOR I=V TO V+64 STEP 2
0800 R$=CHR$(PEKE(I))&CHR$(PEKE(I+1))
0810 T=POS(H9$,R$[1,1])*16+POS(H9$,R$[2,2])-17
```

## COMPILE

```
0820  POK U,ABS(T) @ U=U+1 @ NEXT I @ V=V+68 @ U=U-1 @ NEXT J
0830  IF LEN(C$)=0 THEN 870
0840  FOR I=2 TO LEN(C$) STEP 2
0850  T=POS(H9$,C$[I,I])*16+POS(H9$,C$[I+1,I+1])-17
0860  POK U,T @ U=U+1 @ NEXT I
0870  Q=W\256 @ Y=W-256*Q @ POK P1,Y @ POK P1+1,Q
0880  POK P1+2,141 @ POK P1+3,76
0890  DISP 'READY . . . .' @ BEEP @ BEEP @ BEEP
0900  STOP
1000  ! - Unterroutine fuer Registercodierung -
1010  IF B$[2,3]='RP' THEN 1150
1020  E=VAL(B$[I+1,I+1])*8+VAL(B$[I+2,I+2])+64*(I=6)
1030  IF I=6 THEN Z=D9 ELSE Z=A9
1040  IF E#Z THEN Z=E @ C$=C$&FNY$(E) @ A=A+1
1050  IF I=6 THEN D9=Z ELSE A9=Z
1060  D$=D$&'R'
1070  I=I+4 @ IF I>L THEN 400 ELSE 330
1100  ! - Unterroutine fuer ind. Variablen -
1110  E=VAL(B$[I+1,I+1])*8+VAL(B$[I+2,I+2])
1120  IF E#A9 THEN A9=E @ C$=C$&FNY$(E) @ A=A+1
1130  D$=D$&'X' @ I=I+4 @ IF I>L THEN 400 ELSE 330
1150  ! - Unterroutine fuer Registerpointer -
1160  E=VAL(B$[7,7])*8+VAL(B$[8,8])+64*(B$[1,1]='D')
1170  IF B$[1,1]='D' THEN D9=E ELSE A9=E
1180  A=A+1 @ C$=C$&FNY$(E) @ GOTO 500
1200  ! - Unterroutine fuer Adressencodierung -
1210  E$=E$&B$[I+3,I+4]&B$[I+1,I+2]
1220  A=A+2 @ I=I+6 @ D$=D$&'$' @ IF I>L THEN 400 ELSE 330
1300  ! - Unterroutine fuer Stacks -
1310  E=VAL(B$[12,12])*8+VAL(B$[13,13])
1320  IF E#A9 THEN A9=E @ C$=C$&FNY$(E) @ A=A+1
1330  D$=D$&B$[10,10] @ GOTO 400
1400  ! - Unterroutine fuer Literal -
```

## COMPILE

```
1410 E$=E$&B$[I+1,I+2] @ A=A+1 @ D$=D$&'='
1420 IF L=8 OR L=12 AND I=10 THEN 400
1430 FOR I=14 TO L-1 STEP 3
1440 E$=E$&B$[I,I+1] @ A=A+1 @ NEXT I
1450 GOTO 400
1500 ! - Unterroutine fuer Labels -
1510 ON ERROR DISP 'ERR L'&STR$(Y) @ GOTO 230
1520 COPY ':CA' TO 'ASSEM2'
1530 ASSIGN # 3 TO 'ASSEM2'
1540 FOR I=1 TO 100
1550 READ # 3,I ; F$ @ IF F$[6,LEN(F$)]=B$[I+1,L-1] THEN 1570
1560 NEXT I @ DISP 'ERR L'&STR$(Y) @ GOTO 230
1570 E$=E$&B$[3,4]&B$[1,2] @ A=A+2 @ D$=D$&'$' @ GOTO 400
1700 ! - Unterroutine fuer Mnemonics -
1710 FOR X=7 TO L-2 @ C$=C$&FNY$(NUM(B$[X,X])) @ A=A+1 @ NEXT X
1720 C$=C$&FNY$(NUM(B$[L-1,L-1])+128) @ A=A+1 @ GOTO 500
1800 ! - Unterroutine fuer Ende Label -
1810 IF B$[6,6]='P' THEN A=A+2 @ C$=C$&'FFFF' @ GOTO 500
1820 C$=C$&'FF' @ A=A+1 @ GOTO 500
1900 ! - Unterroutine fuer Labels -
1910 IF B$[7,9]='MAI' THEN C=VAL(B$[11,L]) @ B(C)=A @ GOTO 230
1920 IF B$[7,9]# 'PAR' THEN A(L-8)=A @ GOTO 230
1930 IF LEN(B$)=11 THEN A(2)=A @ GOTO 230
1940 C=VAL(B$[13,L]) @ C(C)=A @ B=MAX(B,C) @ GOTO 230
2000 ! - Befehlstabelle -
2010 DATA ELB R,ELM R,ERB R,ERM R,LLB R,LLM R,LRB R,LRM R,ICB R,ICM R,DCB R,DCM R,TCB R,TCM R
2020 DATA NCB R,NCM R
2030 DATA TSB R,TSM R,CLB R,CLM R,ORB RR,ORM RR,XRB RR,XRM RR,BIN,BCD,SAD,DCE,ICE,CLE,RTN,PAD
2040 DATA LDB RR,LDM RR,STB RR,STM RR,LDBDRR,LDMDRR,STBDRR,STMDRR,LDB R=,LDM R=,STB R=,STM R=
2050 DATA LDBIRR,LDMIRR,STBIRR,STMIRR
2060 DATA LDBDR$,LDMDR$,STBDR$,STMDR$,LDBDRX$,LDMDRX$,STBDRX$,STMDRX$,LDBIR$,LDMIR$,STBIR$
2070 DATA STMIR$,LDBIRX$,LDMIRX$,STBIRX$,STMIRX$
2080 DATA CMB RR,CMM RR,ADB RR,ADM RR,SBB RR,SBM RR,JSB X$,ANM RR,CMB R=,CMM R=,ADB R=,ADM R=
```

## COMPILE

```
2090 DATA SBB R=,SBM R=,JSB $,ANM R=
2100 DATA CMBDR$,CMMDR$,ADBDR$,ADMDR$,SBBDR$,SBMDR$, ,ANMDR$,CMBDRR,CMMDRR,ADBDRR,ADMDDR
2110 DATA SBBDRR,SBMDRR, ,ANMDRR
2120 DATA POBDR+,POMDR+,POBDR-,POMDR-,PUBDR+,PUMDR+,PUBDR-,PUMDR-,POBIR+,POMIR+,POBIR-,POMIR-
2130 DATA PUBIR+,PUMIR+,PUBIR-,PUMIR-
2140 DATA JMP =,JNO =,JOD =,JEV =,JNG =,JPS =,JNZ =,JZR =,JEN =,JEZ =,JNC =,JCY =
2150 DATA JLZ =,JLN =,JRZ =,JRN =
9000 ! - Unterroutine zur Herstellung der Syntax -
9010 READ # 1,P*10 ; B$ @ J=1 @ FOR I=1 TO LEN(B$)
9020 IF B$[I,I]# ' ' THEN B$[J,J]=B$[I,I] @ J=J+1
9030 NEXT I @ B$=B$[1,J-1]
9040 IF B$[1,1]='!' THEN B$='! ' & B$[2,LEN(B$)] @ P=P+1 @ GOTO 9120
9050 IF LEN(B$)=3 THEN 9120
9060 IF POS(H9$[1,10],B$[3,3])#0 THEN B$=B$[1,2]&' ' & B$[3,LEN(B$)] @ GOTO 9120
9070 IF B$[1,5]='LABEL' THEN B$=B$[1,5]&' ' & B$[6,LEN(B$)] @ GOTO 9120
9080 IF B$[1,4]='ENDE' THEN B$=B$[1,4]&' ' & B$[5,LEN(B$)] @ GOTO 9120
9090 IF B$[1,3]='BYT' THEN B$=B$[1,3]&' ' & B$[4,LEN(B$)] @ GOTO 9120
9100 IF POS('DI',B$[4,4])#0 THEN B$=B$[1,4]&' ' & B$[5,LEN(B$)] @ GOTO 9120
9110 B$=B$[1,3]&' ' & B$[4,LEN(B$)]
9120 PRINT # 1,P*10 ; B$ @ P=P+1
9130 IF B$='END' THEN P=P-1 @ RETURN ELSE 9010
9900 DEF FNY$(X)
9910 H1$='' @ H2=ABS(X) @ H9$='0123456789ABCDEF'
9920 H1=1+MOD(H2,16) @ H1$=H9$[H1,H1]&H1$ @ H2=H2\16 @ IF H2 THEN 9920
9930 H1$='000'&H1$ @ H1$=H1$[LEN(H1$)-F,LEN(H1$)]
9940 IF F=3 THEN H1$=H1$[3,4]&H1$[1,2]
9950 FNY$=H1$
9960 END DEF
9970 IF LEN(C$)>64 THEN PRINT # 2,P ; ' ' & C$[1,64] @ C$=C$[65,LEN(C$)] @ P=P+1
9980 RETURN
```

## ASSEM2

0001	018ACONBIN
0002	1D37CONINT
0003	4C91ERROR
0004	4C99ERROR+
0005	0E70NUMVAL
0006	3E8BONEB
0007	3EACONEI
0008	3ECDONER
0009	4694ROMJSB
0010	466DSYSJSB

## PRDIR

```
0100 ! ** Directory ausdrucken **
0110 INTEGER F,L,P,R,R1,R2,T,S,N,Y
0120 DIM D$(8),E$(1),N$(8),R$(256),T$(1)
0130 E$=CHR$(27)
0140 ! - Rec 0 lesen -
0150 SENDIO ':ca','unl,unt,lad#,ddl4',CHR$(0)&CHR$(0) @ GOSUB 510
0160 F=NUM(R$(20,20)) @ R=510-F @ F=F*8
0170 INPUT 'Bandkennzeichnung: '; B$ @ B$=UPRC$(B$)
0180 IF LEN(B$)>15 THEN B$=B$(1,15)
0190 ! - Drucker einstellen -
0200 PWIDTH 75 @ PRINT E$&'!'&CHR$(8)
0210 IMAGE 3x,17a,3d,' Files',3x,8a,/
0220 ! - Ueberschrift drucken -
0230 PRINT USING 210 ; B$,F,FND$(R$(37,39))
0240 IMAGE 3x,6a,6a,9a,4a,5a,5a
0250 PRINT USING 240 ; 'Nr.', 'Rec', 'Name', 'Len', 'Typ', 'Datum'
0260 ! - Band auf Rec 2 positionieren -
0270 SENDIO ':ca','lad#,ddl4',CHR$(0)&CHR$(2)
0280 N,L=0
0290 ! - Rec lesen -
0300 IF MOD(L,8)=0 THEN GOSUB 510
0310 ! - Rec in 8 Files aufteilen -
0320 P=MOD(L,8)*32+1 @ F$=R$(P,P+31) @ L=L+1
0330 ! - File zerlegen -
0340 N$=F$(1,8) @ IF N$(1,1)=CHR$(255) THEN 460
0350 T=NUM(F$(12,12)) @ IF T=0 THEN 300
0360 S=NUM(F$(15,15))
0370 R1=NUM(F$(16,16))
0380 R2=NUM(F$(20,20))
0390 D$=FND$(F$(21,23))
0400 N=N+1 @ R=R-R2
0410 ! - Fileausdruck -
0420 PRINT E$&'!'&CHR$(2);
```

## PRDIR

```
0430 IMAGE 3x, 3z,2x,d,'-',3z,2x,9a,3z,2x,3a,8a
0440 PRINT USING 430 ; N,S,R1,N$,R2,FNT$(T),D$
0450 IF F#L THEN 300
0460 ! - Schlusszeile -
0470 PRINT E$&'!'&CHR$(8)
0480 IMAGE 3x,3d,x,17a,3d,x,12a
0490 PRINT USING 480 ; F-N,'File(s) frei',R,'Records frei'
0500 END
0510 ! - Record holen -
0520 R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0530 RETURN
0540 ! - Datum/Zeichenumwandlung(dez,hex)
0550 DEF FND$(A$) = FNH$(A$[3,3])&'.'&FNH$(A$[2,2])&'.'&FNH$(A$[1,1])
0560 DEF FNH$(Z$) = STR$(NUM(Z$)\16)&STR$(MOD(NUM(Z$),16))
0570 ! - Filetyp -
0580 DEF FNT$(Y)
0590 FNT$='?'
0600 IF Y=1 THEN FNT$='I'
0610 IF Y=82 THEN FNT$='T'
0620 IF Y=83 THEN FNT$='A'
0630 IF Y=136 THEN FNT$='B'
0640 IF Y=137 THEN FNT$='L'
0650 END DEF
```



## PRREC

```
0100 ! ** Records vom Band ausdrucken **
0110 INTEGER R,A,D,I,J,K,L,S,Z,M,N,P,Q
0120 DIM R$(256),E$(1),H$(128),P$(3),S$(4),D$(128),A$(64),R1$(16),F$(32)
0130 E$=CHR$(27) @ S$=' '
0140 ! - Eingaben -
0150 INPUT 'Start bei Spur : ',S$; S$ @ S=VAL(S$) @ S$=' '
0155 INPUT 'Start bei Record : '; R1$ @ R1$=UPRC$(R1$) @ FOR M=1 TO LEN(R1$)-1
0156 IF NUM(R1$(M,M+1))<48 OR NUM(R1$(M,M+1))>57 THEN 570
0157 NEXT M @ R=VAL(R1$)
0160 INPUT 'Anzahl der Rec.: ';A
0170 ! - Drucker einstellen -
0180 PWIDTH 130 @ PRINT E$&'!'&CHR$(22);
0190 ! - Band positionieren -
0200 SENDIO ':ca','unl,lad#,ddl4',CHR$(S)&CHR$(R)
0210 ! - Beginn 1.Rec -
0215 DISP 'Wahl: '&CHR$(196)&'ez','&CHR$(200)&'ex','&CHR$(193)&'SCII: ';
0216 INPUT ''; Q$ @ Q$=UPRC$(Q$) @ D1,D2,D3=0
0217 IF POS(Q$,'D')#0 THEN D1=1
0218 IF POS(Q$,'H')#0 THEN D2=1
0219 IF POS(Q$,'A')#0 THEN D3=1
0220 FOR K=1 TO A
0230 ! - Spur- und Rec.Nr. lesen/drucken -
0240 P$=ENTIO$(':ca','unl,tad#,ddt3,sda')
0250 S=NUM(P$(1,1)) @ R=NUM(P$(2,2))
0260 PRINT 'Spur';S;' Record';R @ PRINT
0270 ! - Rec lesen -
0280 R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0290 ! - Rec aufspalten -
0300 FOR J=1 TO 256 STEP 32
0310 ! - Druckzeilen (dez,hex,ASCII) aufbauen -
0320 D$,H$,A$=''
0330 FOR I=J TO J+31
0340 D=NUM(R$(I,I))
```

## PRREC

```
0350 D$=D$&FNF$(STR$(D))
0360 H$=H$&FNF$(FNH1$(D))
0370 A$=A$&FNA$(D) & ' '
0380 NEXT I
0390 ! - 3 Zeilen drucken -
0400 IF D1=1 THEN PRINT D$
0401 IF D2=1 THEN PRINT H$
0402 IF D3=1 THEN PRINT E$&'!'&CHR$(36);' ';A$
0410 PRINT E$&'!'&CHR$(22)
0420 NEXT J
0430 PRINT
0440 NEXT K
0450 END
0460 ! - Zeichenumwandlung dez/hex -
0470 DEF FNH1$(D) = FNH0$(D\16) & FNH0$(MOD(D,16))
0480 DEF FNH0$(Z) = CHR$(Z+48+7*(Z>9))
0490 ! - Ausdruck formatieren -
0500 DEF FNF$(Z$)
0510 L=4-LEN(Z$) @ FNF$=S$[1,L]&Z$
0520 END DEF
0530 ! - ASCII-Zeichen -
0540 DEF FNA$(D)
0550 IF D>31 AND D<127 THEN FNA$=CHR$(D) ELSE FNA$=' '
0560 END DEF
0570 R1$=UPRC$(R1$) & ' '
0580 R1$=R1$[1,8] @ SENDIO ':ca','unl,lad#,ddl4',CHR$(0)&CHR$(2)
0590 N,Q=0
0600 IF MOD(Q,8)=0 THEN R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0610 P=MOD(Q,8)*32+1 @ F$=R$[P,P+31] @ Q=Q+1
0620 IF F$[1,1]=CHR$(255) THEN 680
0630 N$=F$[1,8] @ A=NUM(F$[20,20])
0640 IF NUM(F$[12,12])=0 THEN 600
0650 S=NUM(F$[15,15]) @ R=NUM(F$[16,16])
```

# PRREC

```
0660  IF N$=R1$ THEN 180
0670  N=N+1 @ IF A#Q THEN 600 ELSE Q=Q+1 @ GOTO 600
0680  DISP 'File nicht gefunden' @ END
```

## LEXAN

```
0100 ! ** Analyse von LEX-Files **
0110 INTEGER N,L,P,F,T,S,R,K,A,J,I
0120 DIM E$(1),A1$(200),A$(8),R$(256),F$(32),N$(8),H$(200),S$(4)
0130 E$=CHR$(27)
0135 SENDIO ':ca','unl,lad#,ddl7',' ' @ R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0140 ! - Positionieren des Bandes auf Directory -
0150 INPUT 'Programmname: '; A1$ @ A1$=UPRC$(A1$) & '
0160 A$=A1$(1,8) @ SENDIO ':ca','lad#,ddl4',CHR$(0) & CHR$(2)
0170 N,L=0
0180 ! - Rec lesen -
0190 IF MOD(L,8)=0 THEN R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0200 P=MOD(L,8)*32+1 @ F$=R$(P,P+31) @ L=L+1
0210 IF F$(1,1)=CHR$(255) THEN 270
0220 N$=F$(1,8) @ F=NUM(F$(20,20))
0230 T=NUM(F$(12,12)) @ IF T=0 THEN 190
0240 S=NUM(F$(15,15)) @ R=NUM(F$(16,16)) @ A=F
0250 IF N$=A$ THEN 280
0260 N=N+1 @ IF F#L THEN 190 ELSE L=L+1 @ GOTO 190
0270 DISP 'File nicht gefunden' @ END
0280 A1$=A$ @ A$=FNH1$(NUM(F$(23,23))) & '.' & FNH1$(NUM(F$(22,22))) & '.'
0290 A$=A$ & FNH1$(NUM(F$(21,21)))
0300 ! - Band auf Programm positionieren -
0310 SENDIO ':ca','unl,lad#,ddl4',CHR$(S) & CHR$(R)
0320 ! - Rec lesen -
0330 R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0340 F=NUM(R$(4,4))*256+NUM(R$(3,3))
0350 PWIDTH 75 @ PRINT E$&'!' & CHR$(8)
0360 IMAGE 3x,17a,5d,'Bytes ',3x,8a,/
0370 PRINT USING 360 ; A1$,F,A$
0380 PRINT @ PRINT
0390 PWIDTH 130 @ PRINT E$&'!' & CHR$(22);
0400 PRINT 'Befehle:'
0410 M=NUM(R$(30,30))*256+NUM(R$(29,29))+19
```

## LEXAN

```
0420 A2=NUM(R$[20,20]) @ A1=NUM(R$[19,19])
0430 K=NUM(R$[24,24])*256+NUM(R$[23,23])+19
0440 J=NUM(R$[28,28])*256+NUM(R$[27,27])+19
0450 IF R$[K,K]=CHR$(255) THEN PRINT 'keine' @ GOTO 580
0460 H$=R$[K,J] @ D=0
0470 FOR I=1 TO POS(H$,CHR$(255))
0480 IF NUM(H$[I,I])>128 THEN D=D+1
0490 NEXT I
0500 FOR I=1 TO D-1
0510 PRINT 180;A1;A2;I;' ';
0520 FOR B=1 TO POS(H$,CHR$(255))
0530 IF NUM(H$[B,B])>127 THEN 550
0540 NEXT B @ DISP 'Falsche Codierung !!' @ STOP
0550 A1$=H$[1,B] @ A1$[B,B]=CHR$(NUM(A1$[B,B])-128)
0560 PRINT A1$ @ H$=H$[B+1,LEN(H$)]
0570 NEXT I
0580 PRINT 'Fehlermeldungen:'
0590 IF R$[J,J]=CHR$(255) THEN PRINT 'keine' @ STOP
0600 H$=R$[J+1,M] @ D=0
0605 IF NUM(H$[1,1])>127 THEN H$=H$[2,LEN(H$)]
0607 H$=H$[1,POS(H$,CHR$(255))]
0610 FOR I=1 TO LEN(H$)
0620 IF NUM(H$[I,I])>128 THEN D=D+1
0630 NEXT I
0640 FOR I=1 TO D-1
0650 FOR B=1 TO LEN(H$)
0660 IF NUM(H$[B,B])>127 THEN 680
0670 NEXT B @ DISP 'Falsche Codierung !!' @ STOP
0680 A1$=H$[1,B] @ A1$[B,B]=CHR$(NUM(A1$[B,B])-128)
0690 PRINT A1$ @ H$=H$[B,LEN(H$)]
0700 NEXT I
0710 PRINT @ PRINT
0720 STOP
```

## LEXAN

```
0730 DEF FNH1$(D) = FNH0$(D\16) & FNH0$(MOD(D,16))
0740 DEF FNH0$(Z) = CHR$(Z+48+7*(Z>9))
```

## LEXIN

```
0100 ! ** LEX-Files vom Listing eingeben **
0110 INTEGER Z9,N,L,P,P1,P2,P3,P5,S,P4,R9,I,J
0120 DIM A$(32),R$(256),H$(8),C$(10),D$(70)
0130 ! - Band positionieren -
0140 Z9=0 @ R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0150 SENDIO ':ca','unl,lad#,ddl4',CHR$(0)&CHR$(Z9)
0160 N,L=0 @ A$=CHR$(255)&CHR$(255)
0170 IF MOD(L,8)=0 THEN R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0180 P=POS(R$,A$)
0190 ! - Ermitteln des Programmrecords -
0200 IF P=0 THEN Z9=Z9+1 @ GOTO 150
0210 IF MOD(P,32)#0 THEN P=IP(P/32)+1 @ P=P*32
0211 P1=P-48 @ IF P1>0 THEN 220 ELSE SENDIO ':ca','unl,lad#,ddl4',CHR$(0)&CHR$(Z9-1)
0212 R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda') @ P1=P1+256
0220 P2=NUM(R$(P1,P1))+NUM(R$(P1+4,P1+4))
0230 S=NUM(R$(P1-1,P1-1)) @ R$='' @ P5=0 @ P3,J=1
0240 SENDIO ':ca','lad#,ddl4',CHR$(S)&CHR$(P2)
0250 ! - Eingabe des Listings -
0260 DISP STR$(J)&': '; @ INPUT ''; D$ @ D$=UPRC$(D$)
0270 IF LEN(D$)>64 THEN 260
0275 FOR I=1 TO 63 STEP 2 @ H$=D$(I,I+1)
0277 IF NUM(H$(1,1))>70 OR NUM(H$(2,2))>70 THEN P3=P3+1 @ GOTO 330
0280 P5=P5+1 @ P4=(POS('0123456789ABCDEF',H$(1,1))-1)*16+POS('0123456789ABCDEF',H$(2,2))-1
0290 R$=R$&CHR$(P4)
0300 IF P5>=256 THEN P3=P3+1 @ GOTO 305
0301 NEXT I @ J=J+1 @ GOTO 260
0305 SENDIO ':ca','lad#,ddl4',CHR$(S)&CHR$(P2+P3-2)
0310 SENDIO ':ca','lad#,ddl2',R$
0320 P5=0 @ R$='' @ J=J+1 @ GOTO 260
0330 IF LEN(R$)=256 THEN 350
0340 FOR R9=LEN(R$) TO 255 @ R$=R$&CHR$(255) @ NEXT R9
0345 SENDIO ':ca','lad#,ddl4',CHR$(S)&CHR$(P2+P3-2)
0350 SENDIO ':ca','lad#,ddl2',R$
```

## LEXIN

```
0360 SENDIO ':ca','lad#,ddl4',CHR$(S)&CHR$(P2)
0370 R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0380 C$=R$[11,18]&' '
0390 ! - Eintrag in Directory -
0400 SENDIO ':ca','unl,lad#,ddl4',CHR$(0)&CHR$(Z9)
0410 R$=ENTIO$(':ca','unl,tad#,ddt2,ddt4,ddt1,sda')
0420 A$=C$&CHR$(224)&CHR$(137)&CHR$(0)&CHR$(0)
0430 A$=A$&CHR$(S)&CHR$(P2)&CHR$(0)&CHR$(0)&CHR$(0)&CHR$(P3)
0440 H$=DATE$ @ P1=VAL(H$[1,1])*16+VAL(H$[2,2])
0450 P2=VAL(H$[4,4])*16+VAL(H$[5,5])
0460 P3=VAL(H$[7,7])*16+VAL(H$[8,8])
0470 A$=A$&CHR$(P1)&CHR$(P2)&CHR$(P3)
0480 H$=TIME$ @ P1=VAL(H$[1,1])*16+VAL(H$[2,2])
0490 P2=VAL(H$[4,4])*16+VAL(H$[5,5])
0500 P3=VAL(H$[7,7])*16+VAL(H$[8,8])
0510 A$=A$&CHR$(P1)&CHR$(P2)&CHR$(P3)
0520 A$=A$&CHR$(128)&CHR$(1)&' '
0530 R$[P-31,P]=A$
0540 SENDIO ':ca','unl,lad#,ddl4',CHR$(0)&CHR$(Z9)
0550 SENDIO ':ca','unl,lad#,ddl2',R$
0560 STOP
```



## DEVICES

```
0100 ! ** Bestimmung der Zahl von IL-Einheiten **
0110 DIM R$(1),A$(30)
0120 INTEGER R,A,I
0190 ! - Eingabe der Einheitsart -
0200 INPUT 'Art: '; A$ @ P=POS('CMPDIEGA',UPRC$(A$))
0210 ! - Kontrolle der Zubehoerskennung -
0220 I=1 @ A=0 @ IF P=0 THEN DISP 'falsche Art' @ BEEP @ GOTO 200
0230 R$=ENTIO$('','UNL,UNT,TAD'&STR$(I)&','SAI')
0240 R=NUM(R$) @ IF R=0 THEN 300
0250 IF P=8 THEN A=A+1
0260 IF R>=16*(P-1) AND R<=16*P-1 THEN A=A+1
0270 I=I+1 @ GOTO 230
0300 ! - Herstellung der Zustandsmeldung -
0310 RESTORE @ FOR I=1 TO P @ READ A$ @ NEXT I
0320 DISP 'IL-Schleife: '&STR$(A)&' '&A$
0330 STOP
0400 DATA Controller,Massenspeicher,Drucker,Display,Interface,E-Instrumente
0410 DATA Graphikeinheiten,Einheiten
```

## PRMEM

```
0100 ! ** Memory des Rechners ausdrucken **
0110 INTEGER R,A,D,I,J,K,L,S,Z,M,N,P,Q
0120 DIM R$(256),E$(1),H$(128),P$(16),S$(4),D$(128),A$(64),R1$(16),F$(32)
0130 E$=CHR$(27) @ S$=' '
0150 S$=' '
0170 ! - Drucker einstellen -
0180 PWIDTH 130 @ PRINT E$&'!'&CHR$(6);
0210 ! - Beginn 1.Rec -
0215 DISP 'Wahl: '&CHR$(196)&'ez,'&CHR$(200)&'ex,'&CHR$(193)&'SCII: ' ;
0216 INPUT ' '; Q$ @ Q$=UPRC$(Q$) @ D1,D2,D3=0
0217 IF POS(Q$,'D')#0 THEN D1=1
0218 IF POS(Q$,'H')#0 THEN D2=1
0219 IF POS(Q$,'A')#0 THEN D3=1
0220 FOR K=0 TO 65535 STEP 256
0230 R$=' '
0240 FOR J=K TO K+255
0250 IF J>32767 THEN R$=R$&CHR$(PEKE(J-32768)) ELSE R$=R$&CHR$(PEEK(J))
0260 NEXT J
0270 FOR J=1 TO 256 STEP 32
0280 S=K+J-1 @ P$='0123456789ABCDEF' @ B$=' '
0290 R=1+MOD(S,16) @ B$=P$(R,R)&B$ @ S=S\16 @ IF S THEN 290
0300 FOR S=LEN(B$) TO 5 @ B$='0'&B$ @ NEXT S @ B$=B$[3,6] @ PRINT '$'&B$
0310 ! - Druckzeilen (dez,hex,ASCII) aufbauen -
0320 D$,H$,A$=' '
0330 FOR I=J TO J+31
0340 D=NUM(R$[I,I])
0350 D$=D$&FNF$(STR$(D))
0360 H$=H$&FNF$(FNH1$(D))
0370 A$=A$&FNA$(D) &' '
0380 NEXT I
0390 ! - 3 Zeilen drucken -
0400 IF D1=1 THEN PRINT D$
0401 IF D2=1 THEN PRINT H$
```

## PRMEM

```
0402 IF D3=1 THEN PRINT E$&'!'&CHR$(36);' ';A$
0410 PRINT E$&'!'&CHR$(6);
0420 NEXT J
0430 PRINT
0440 NEXT K
0450 END
0460 ! - Zeichenumwandlung dez/hex -
0470 DEF FNH1$(D) = FNH0$(D\16)&FNH0$(MOD(D,16))
0480 DEF FNH0$(Z) = CHR$(Z+48+7*(Z>9))
0490 ! - Ausdruck formatieren -
0500 DEF FNF$(Z$)
0510 L=4-LEN(Z$) @ FNF$=S$[1,L]&Z$
0520 END DEF
0530 ! - ASCII-Zeichen -
0540 DEF FNA$(D)
0550 IF D>31 AND D<127 THEN FNA$=CHR$(D) ELSE FNA$=' '
0560 END DEF
```

## PRPRO

```
0100 ! ** Programm aus dem Memory ausdrucken **
0110 INTEGER R,A,D,I,J,K,L,S,Z,M,N,P,Q
0120 DIM R$[256],E$[1],H$[128],P$[3],S$[4],D$[128],A$[64],R1$[16],F$[32]
0130 E$=CHR$(27) @ S$=' '
0140 ! - Eingaben -
0150 INPUT 'Programmname: '; R1$ @ R1$=UPRC$(R1$) @ IF R1$='DEVFILE' OR R1$='IOFILE' THEN 570
0152 IF R1$='CALCPROG' OR R1$='KEYS' OR R1$='APPT' THEN 570
0155 I=1354 @ S$=' ' @ R$=R1$&S$&S$ @ R1$=R$[1,8]
0156 IF PEKE(I)=0 THEN I=I+1 @ GOTO 156
0157 R$=''
0160 FOR J=I TO I+255 @ R$=R$&CHR$(PEKE(J)) @ NEXT J
0170 P=POS(R$,R1$) @ IF P=0 THEN I=I+256 @ GOTO 160
0175 A=NUM(R$[P-7,P-7])*256+NUM(R$[P-8,P-8])
0180 P=NUM(R$[P-9,P-9])*256+NUM(R$[P-10,P-10])
0190 ! - Drucker einstellen -
0200 PWIDTH 130 @ PRINT E$&'!'&CHR$(6);
0205 PRINT E$&'!'&CHR$(36)&R1$&' ':'&E$&'!'&CHR$(6)
0210 ! - Beginn 1.Rec -
0215 DISP 'Wahl: '&CHR$(196)&'ez,'&CHR$(200)&'ex,'&CHR$(193)&'SCII: ';
0216 INPUT ''; Q$ @ Q$=UPRC$(Q$) @ D1,D2,D3=0
0217 IF POS(Q$,'D')#0 THEN D1=1
0218 IF POS(Q$,'H')#0 THEN D2=1
0219 IF POS(Q$,'A')#0 THEN D3=1
0220 FOR K=P TO P+A STEP 32 @ R$=''
0230 FOR J=K TO K+31
0240 IF J>P+A THEN R$=R$&CHR$(255) @ GOTO 260
0250 R$=R$&CHR$(PEKE(J-32768))
0260 NEXT J
0310 ! - Druckzeilen (dez,hex,ASCII) aufbauen -
0320 D$,H$,A$=''
0330 FOR I=1 TO 32
0340 D=NUM(R$[I,I])
0350 D$=D$&FNF$(STR$(D))
```

## PRPRO

```
0360 H$=H$&FNF$(FNH1$(D))
0370 A$=A$&FNA$(D) & ' '
0380 NEXT I
0390 ! - 3 Zeilen drucken -
0400 IF D1=1 THEN PRINT D$
0401 IF D2=1 THEN PRINT H$
0402 IF D3=1 THEN PRINT E$&'!'&CHR$(36);' ';A$
0410 PRINT E$&'!'&CHR$(6);
0440 NEXT K
0450 END
0460 ! - Zeichenumwandlung dez/hex -
0470 DEF FNH1$(D) = FNH0$(D\16)&FNH0$(MOD(D,16))
0480 DEF FNH0$(Z) = CHR$(Z+48+7*(Z>9))
0490 ! - Ausdruck formatieren -
0500 DEF FNF$(Z$)
0510 L=4-LEN(Z$) @ FNF$=S$[1,L]&Z$
0520 END DEF
0530 ! - ASCII-Zeichen -
0540 DEF FNA$(D)
0550 IF D>31 AND D<127 THEN FNA$=CHR$(D) ELSE FNA$=' '
0560 END DEF
0570 FOR J=1 TO LEN(R1$) @ R1$[J,J]=CHR$(NUM(R1$[J,J])+32) @ NEXT J
0580 GOTO 155
```

## SETTING

```
0100 ! ** Abfrage von Statusgroessen **
0110 DIM A$[7]
0120 SHORT I
0130 INTEGER J
0200 ! - Eingabe der Groesse -
0210 INPUT 'Groesse: '; A$ @ A$=UPRC$(A$)
0220 IF A$='PWIDTH' THEN 260
0230 IF A$='WIDTH' THEN 280
0240 IF A$='MARGIN' THEN 300
0250 IF A$='DELAY' THEN 320 ELSE DISP 'falsche Groesse' @ BEEP @ WAIT 2 @ GOTO 200
0260 ! - Unterprogramm fuer PWIDTH -
0270 I=PEKE(716)-(PEKE(716)=0) @ GOTO 400
0280 ! - Unterprogramm fuer WIDTH -
0290 I=PEKE(715)-(PEKE(715)=0) @ GOTO 400
0300 ! - Unterprogramm fuer MARGIN -
0310 I=PEKE(944) @ GOTO 400
0320 ! - Unteroutine fuer DELAY -
0330 I=0 @ FOR J=0 TO 3 @ I=I+PEKE(J+648)*256^J @ NEXT J
0340 I=I/64
0400 DISP A$; @ IF I<0 THEN DISP ' inf' ELSE DISP '='&STR$(I) @ STOP
```

## ANLEX

```
0100 ! ** Analyse von LEX-Files im Memory des Rechners **
0110 INTEGER N,L,P,F,T,S,R,K,A,J,I
0120 DIM E$(1),A1$(200),A$(8),R$(256),F$(32),N$(8),H$(200),S$(4)
0130 E$=CHR$(27)
0140 ! - Positionieren des Bandes auf Directory -
0150 INPUT 'Programmname: '; A1$ @ A1$=UPRC$(A1$) & '
0160 A$=A1$(1,8)
0170 I=1354 @ S$='
0180 IF PEKE(I)=0 THEN I=I+1 @ GOTO 180
0190 R$='
0200 FOR J=I TO I+255 @ R$=R$&CHR$(PEKE(J)) @ NEXT J
0210 P=POS(R$,A$) @ IF P=0 THEN I=I+256 @ GOTO 190
0220 A=NUM(R$(P-7,P-7))*256+NUM(R$(P-8,P-8))
0230 P=NUM(R$(P-9,P-9))*256+NUM(R$(P-10,P-10))
0240 R$='
0250 FOR J=P TO P+255 @ R$=R$&CHR$(PEKE(J-32768)) @ NEXT J
0260 I=1
0270 A1$=CAT$(I) @ IF POS(A1$,A$)=0 THEN I=I+1 @ GOTO 270
0280 IF A1$(12,12)='#L' THEN DISP 'kein LEX-File' @ STOP
0290 F=VAL(A1$(14,17))
0300 A$=A1$(25,26)&'.'&A1$(28,29)&'.'&A1$(31,32)
0310 A1$=A1$(1,8)
0350 PWIDTH 75 @ PRINT E$&'!'&CHR$(8)
0360 IMAGE 3x,17a,5d,'Bytes ',3x,8a,/
0370 PRINT USING 360 ; A1$,F,A$
0380 PRINT @ PRINT
0390 PWIDTH 130 @ PRINT E$&'!'&CHR$(22);
0400 PRINT 'Befehle:'
0410 M=NUM(R$(12,12))*256+NUM(R$(11,11))+1
0420 A2=NUM(R$(2,2)) @ A1=NUM(R$(1,1))
0430 K=NUM(R$(6,6))*256+NUM(R$(5,5))+1
0440 J=NUM(R$(10,10))*256+NUM(R$(9,9))+1
0450 IF R$(K,K)=CHR$(255) THEN PRINT 'keine' @ GOTO 580
```

## ANLEX

```
0460 H$=R$[K,J] @ D=0
0470 FOR I=1 TO POS(H$,CHR$(255))
0480 IF NUM(H$[I,I])>128 THEN D=D+1
0490 NEXT I
0500 FOR I=1 TO D-1
0510 PRINT 180;A1;A2;I;' ';
0520 FOR B=1 TO POS(H$,CHR$(255))
0530 IF NUM(H$[B,B])>127 THEN 550
0540 NEXT B @ DISP 'Falsche Codierung !!' @ STOP
0550 A1$=H$[1,B] @ A1$[B,B]=CHR$(NUM(A1$[B,B])-128)
0560 PRINT A1$ @ H$=H$[B+1,LEN(H$)]
0570 NEXT I
0580 PRINT 'Fehlermeldungen:'
0590 IF R$[J,J]=CHR$(255) THEN PRINT 'keine' @ STOP
0600 H$=R$[J+1,M] @ D=0
0605 IF NUM(H$[1,1])>127 THEN H$=H$[2,LEN(H$)]
0607 H$=H$[1,POS(H$,CHR$(255)) ]
0610 FOR I=1 TO LEN(H$)
0620 IF NUM(H$[I,I])>128 THEN D=D+1
0630 NEXT I
0640 FOR I=1 TO D-1
0650 FOR B=1 TO LEN(H$)
0660 IF NUM(H$[B,B])>127 THEN 680
0670 NEXT B @ DISP 'Falsche Codierung !!' @ STOP
0680 A1$=H$[1,B] @ A1$[B,B]=CHR$(NUM(A1$[B,B])-128)
0690 PRINT A1$ @ H$=H$[B,LEN(H$)]
0700 NEXT I
0710 PRINT @ PRINT
0720 STOP
0730 DEF FNH1$(D) = FNH0$(D\16)&FNH0$(MOD(D,16))
0740 DEF FNH0$(Z) = CHR$(Z+48+7*(Z>9))
```



## CONVERS

```
0010  ! ** Zahlen-Basis-Wandler **
0020  DIM A$,A0$[24]
0030  INPUT 'Basis 1, Basis 2, Zahl: '; A,A0,A$ @ A$=UPRC$(A$) @ A1=0
0040  FOR I=1 TO LEN(A$)
0050  A2=NUM(A$[I,I])-64
0060  IF A2<=0 THEN A2=A2+7
0070  A2=A2+9 @ A1=A1*A+A2
0080  NEXT I
0090  A0$='' @ A2,A3=A1
0100  A3=A3\A0 @ A2=RMD(A2,A0)-9
0110  IF A2<=0 THEN A2=A2+7
0120  A2=A2+64 @ A0$=CHR$(A2)&A0$
0130  IF A3 THEN A2=A3 @ GOTO 100
0140  DISP A$;' (';STR$(A);')=';A0$;' (';STR$(A0);') '
0150  K$=UPRC$(KEY$)
0160  IF K$='' THEN 150 ELSE K=NUM(K$)
0170  IF K#13 THEN 150 ELSE 30
```