

Af

Fiskeridirektoratet
Biblioteket

Norsk Oseanografisk Datasenter

(NOD)

23 SEPT. 1976

SORTERING OG LAGRING AV OSEANOGRAFISKE
MÅLEDATA VED NOD

PROGRAMBESKRIVELSE

av

Øyvin Strand

Nr. 14

MAI 1976



FISKERIDIREKTORATETS HAVFORSKNINGSINSTITUTT

BERGEN

NORSK OSEANOGRAFISK DATASENTER

SORTERING OG LAGRING AV OSEANOGRAFISKE
MÅLEDATA VED NOD

PROGRAMBESKRIVELSE

av

Øyvin Strand

Nr. 14

MAI 1976

FISKERIDIREKTORATETS HAVFORSKNINGSINSTITUTT
BERGEN

PROGRAMPAKKE FOR SORTERING OG LAGRING AV HYDROGRAFISKE DATA.

Pakken består av ialt atten forskjellige programdeler, hvorav sytten er beskrevet her. Programmene arbeider med data som er, eller blir, sortert etter Marsdensystemet, hvor jorden deles inn i ruter på $10^{\circ} \times 10^{\circ}$ (i programbeskrivelsen kalt MSQ-ruter). Programmene opererer med to systemfiler. Den ene inneholder både Marsdenrutetabellene med fortegnelse over hvilke filer som er lagret på de forskjellige magnetbånd, og magnetbåndtabellen, som inneholder nødvendige opplysninger for montering av båndene, samt antall filer og blokker på disse. Denne filen er gitt navnet REGFIL2. Den andre filen inneholder engradersrutetabellene som består av en tabell med hundre plasser for hver linje i Marsdenrutetabellene. Denne filen er gitt navnet REGFIL3.

Pakken består av tre hovedprogrammer:

- MSQDATA sorterer ut måledata tilhørende ønskede MSQ-ruter fra en originalfil.
- MSQDATA-INNH sorterer ut måledata fra en originalfil og lager innholdsregister for de nye måledatafilene.
- STYREPROGRAM er hovedprogram for rutinene:
 - CONT lager innholdsregister for måledatafiler.
 - SORTER sorterer en innholdsregisterfil etter år, skip, dato og stasjonsnummer.
 - MHSORT sorterer en innholdsregisterfil etter oppgitt rekkefølge for engradersruter.
 - OMSKR skriver om en måledatafil etter rekkefølgen i den tilhørende (sorterte) innholdsregisterfil. Adressene i innholdsregisterfilen endres i overensstemmelse med den nye rekkefølgen i måledatafilen.
 - IMIX fletter sammen to (sorterte) innholdsregisterfiler.
 - DMIX fletter sammen to (sorterte) måledatafiler.

Som en vil se, kan utsortering av nye måledatafiler og innholdsregistre gjøres på to måter: Ved bruk av MSQDATA og CONT, eller ved MSQDATA-INNH. Hvilken metode som er mest effektiv, må avgjøres ut fra kjennskap til originalfilens sammensetning (MSQDATA har større "kapasitet" enn MSQDATA-INNH).

Programmet benytter følgende rutiner som ikke er gitt egen dokumentasjon:

- RNAVN (RNAVN (N1, N2, N3, N4)) er en subrutine som gir ut et "navn" på inntil seks karakterer i N3 (høyrejustert). Dette navnet brukes som filnavn på platelagerfiler. I denne programpakken er N1 forutsatt å være MSQ-rutennummeret, N2 registertypenummer og N4 et eget navntypenummer.
 - ERTRAN (ERTRAN ("liste"), NERTRN ("liste")) er to systemrutiner i FORTRAN V. De kan utføre mange forskjellige funksjoner, men bare tre er brukt her:
 - "liste" = 2): Jobben avbrytes.
 - "liste" = 6, NA): NA er et endimensjonalt array som inneholder en styresetning som utføres (f.eks. tilordner en fil eller et magnetbånd).
 - "liste" = 9, ID, IK): ID viser dato (): aabbcc hvor aa er måned, bb dag og cc to siste siffre i årstallet), IK viser klokkeslett.
- NERTRN kalles som funksjon, og funksjonsverdien er en statusvariabel som en kan teste på (se UNIVAC's manualer). (Ved feilstatus i ERTRAN stopper programmet.)

-NTRAN (NTRAN (IO, "liste")) er en systemrutine i FORTRAN V som opererer på filer (skriver, leser og spoler etc.). IO er I/O-enhetsnummer. Listen kan bestå av en eller flere ordrer, og hver ordre igjen av et antall parametre.

Følgende ordrer er brukt:

- "1,NW,NDIM,L"): Skriver en blokk.
- "2,NW,NDIM,L"): Leser en blokk.
- "7,N"): Spoler over N blokker (magnetbånd).
- "8,N"): Spoler over N EOF-merker (magnetbånd).
- "9"): Setter EOF-merke (magnetbånd)..
- "10"): Posisjonerer lese/skrivehodet til begynnelsen av filen (platelager) eller magnetbåndet.
- "18"): Fraordner filen.
- "22"): Avslutt alle pågående NTRAN-operasjoner.

NW er blokk lengde, NDIM er et array som inneholder blokken, og L en statusvariabel (L = NW): lesing/skriving ferdig. L = -1): lesing/skriving ikke ferdig, L = -2): EOF-merke påtruffet, L = -3 eller -4): feil). Rutinen har den egenskapen at den kan arbeide parallelt (i tid) med det kallende program. En kan derved korte ned den tiden et program beslaglegger ressurser. En del av programmene benytter slik parallellprosessering ved innlesing av data. Disse programmene har da et innlesebuffer som rommer to blokker. Når den ene halvdel av bufferet brukes av det kallende program, overfører NTRAN data til den andre halvdel (dobbelbuffering).

Pakken er skrevet i FORTRAN V (for UNIVAC 1110). Maskinen benytter en ordlengde på 36 bits, og karakterkoden er på seks binære sifre. Et ord rommer dermed seks karakterer.

NTFEIL

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Dato : 20/5-75
Utført av : Ø. Strand
Formål : NTFEIL kontrollerer den status-
variable ved lesing/skriving i
NTRAN.

Kall: NTFEIL (NW, L, NUN, K, \$)

Generelt:

Ved lesing/skriving har NTRAN en variabel som til en hver tid angir status for pågående operasjon. Hvis lese/skriveoperasjonene er fullført, er status lik blokk lengden. De øvrige verdier er:

- 1): lesing/skriving pågår.
- 2): EOF-merke påtruffet.
- 3, -4): Feil.

NTFEIL kontrollerer at status har en gitt verdi. Hvis den statusvariable har en annen verdi enn forventet, utskrives diverse feilmeldinger, og programmet avbrytes. Hvis EOF-merke er nådd, returneres til ønsket linjenummer i kallende program.

Formelle parametre:

NAVN:	TYPE:	BRUK:
NW	INTEGER	Ønsket status.
L	INTEGER	Statusvariabel for NTRAN.
NUN	INTEGER	I/O-enhetsnummer for filen/-båndet som er brukt.
K	INTEGER	Blokknummer (eller annen indikator hvis blokknummer ikke er kjent).
\$	Linjenummer	Retur ved EOF-merke.

```
000001 000 C
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C
000008 000 C
000009 000 C
000010 000 C
000011 000 C
000012 000 10
000013 000 20
000014 000 30
000015 000
000016 000
000017 000
000018 000
000019 000
000020 000
000021 000
```

```
*****
*           *
*   NTFEIL   *
*           *
*****
```

```
      SUBROUTINE NTFEIL(NW,L,NUN,K,I)
      FORMAT(///4X,'LESE/SKRIVEFEIL INTRAN,UNIT ',I2)
      FORMAT(4X,'STATUSORD=',I5)
      FORMAT(4X,'BLOKK ',I5)
      IF (L.EQ.NW) RETURN          & I ORDEN
      IF (L.EQ.-2) RETURN 5       & EOF-MERKE PATRUFFET
      WRITE(6,10) NUN
      WRITE(6,20) L
      WRITE(6,30) K
      CALL ERTRAN(2)              & FEILAVBRUDD
      END
```

LESDB

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Dato : 18/12-75
Utført av : Ø. Strand
Formål : LESDB er en hjelperutine som be-
sørger innlesing av måledata ved
dobbelbuffring.

Kall: LESDB (INN, N, NUNIT, K, J, LSTAT, \$)

Generelt:

LESDB styrer innlesing av en måledatafil ved hjelp av dobbelbuffring. Rutinen opererer med et buffer som rommer to blokker () : med 200 linjer (poster)). Når data i første halvdel av bufferet (A) brukes, pågår innlesing av data til andre halvdel (B), og omvendt. Rutinen har en teller som økes med en for hvert kall. Når telleren når verdien 101, avsluttes innlesing av bufferdel B, og ny innlesing av bufferdel A initialiseres. Ved verdien 201 skiftes på ny, og telleren gis samtidig verdien 1 (telleren kan brukes som linjenummer). Rutinen har variabel postlengde.

Formelle parametre:

NAVN:	TYPE:	BRUK:
\$	Linjenummer	Retur hvis EOF-merke er påtruffet.
N	INTEGER	Postlengde.
INN(N, 200)	INTEGER	Måledatabuffer.
J	INTEGER	Teller som styrer skifte av blokk (kan brukes som linjenummer).
K	INTEGER	Blokknummer.
NUNIN	INTEGER	I/O-enhet for måledatafil.
LSTAT	INTEGER	Statusvariabel for NTRAN.

Eksterne rutiner : NTFEIL

Variable:

NAVN:	TYPE:	BRUK:
LESH	INTEGER	Hjelpevariabel.
NW	INTEGER	Blokklengde.

000001 002 C
000002 002 C
000003 002 C
000004 002 C
000005 002 C
000006 002 C
000007 002 C
000008 002 C
000009 002 C
000010 002 C
000011 002 C
000012 002 C
000013 002 C
000014 002 C
000015 002 C
000016 000
000017 002 C
000018 000
000019 000
000020 001
000021 000
000022 000
000023 000
000024 001
000025 001
000026 000
000027 000
000028 000
000029 001 200
000030 002 C
000031 002 C
000032 002 C
000033 001
000034 001
000035 000

```
*****  
*                               *  
*   LESDB                       *  
*                               *  
*****
```

LESDB STYRER INNLESING VED HJELP AV DOBBELBUFFRING.

NØDVENDIGE RUTINER: NTFEIL

SUBROUTINE LESDB(INN,N,NUNIN,K,J,LSTAT,I)

```
DIMENSION INN(N,200)  
J=J+1  
IF ((J.NE.101).AND.(J.NE.201)) RETURN  
Nw=N*100  
IF (LSTAT.NE.Nw) CALL NTRAN(NUNIN,22)  
K=K+1  
CALL NTFEIL(Nw,LSTAT,NUNIN,K,1200)  
LESH=J-100  
CALL NTRAN(NUNIN,2,Nw,INN(1,LESH),LSTAT)  
IF (J.EQ.201) J=1  
RETURN  
CONTINUE  
EOF-MERKE ER PATRUFFET.  
K=K-1  
RETURN 6  
END
```


MONTIN

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Entrepunkter : MONTER, TBAAND
Dato : 26/2-76
Utført av : Ø. Strand
Formål : MONTIN med entrepunkt brukes til å montere/tilordne alle typer Marsdensorterte filer på magnetbånd og platelager, og dessuten til å montere magnetbånd for skriving.

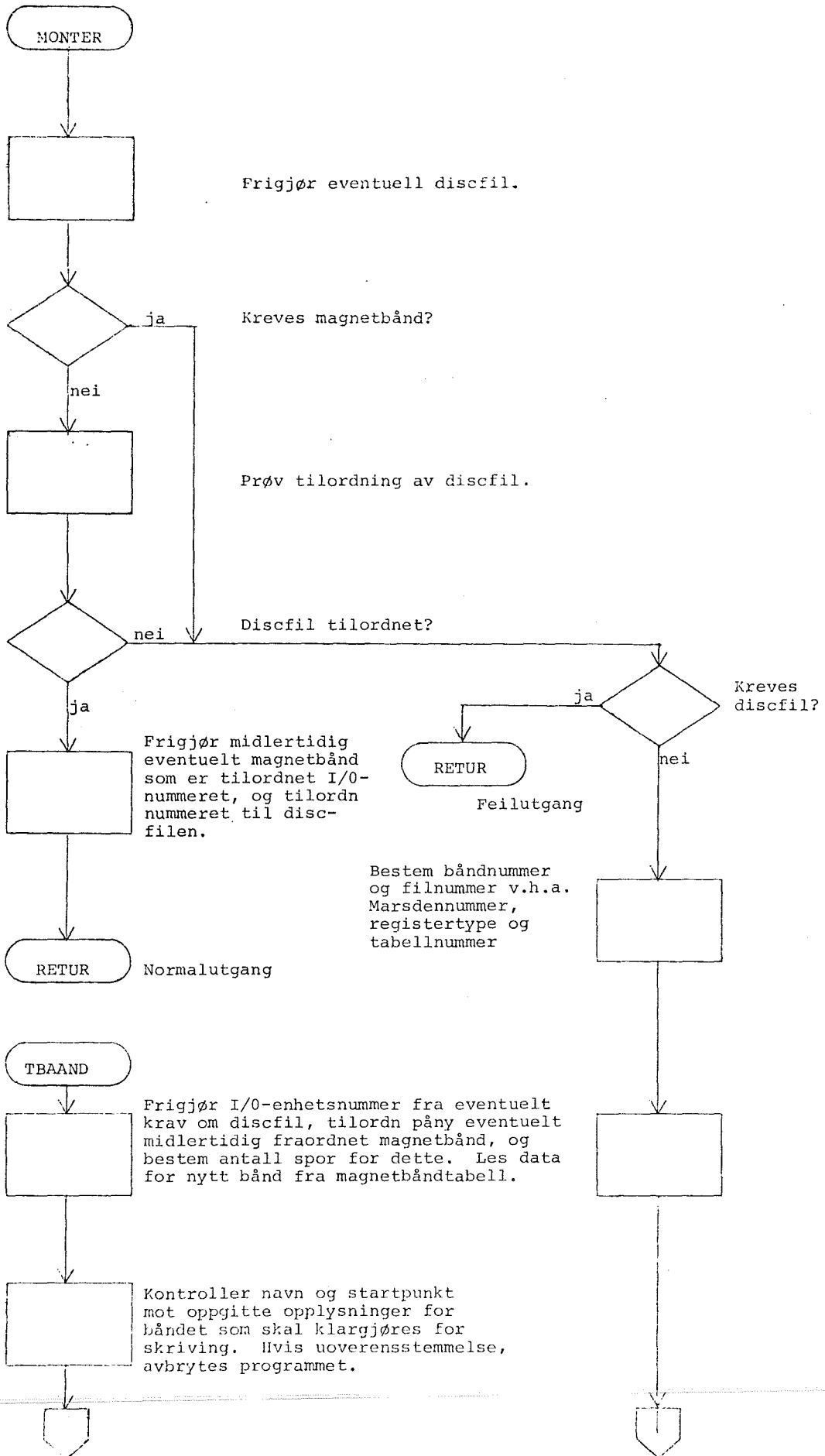
Kall: MONTIN(IDAPO)
MONTER(NRT, NRF, NUN, MSQ, \$, IREG, TAPE, NMOD, LOTAPE, LODISC, ITAB)
TBAAND(NRT, NRF, NUN, TAPE, NRTNY, NRFNY, OVER, ITDS)

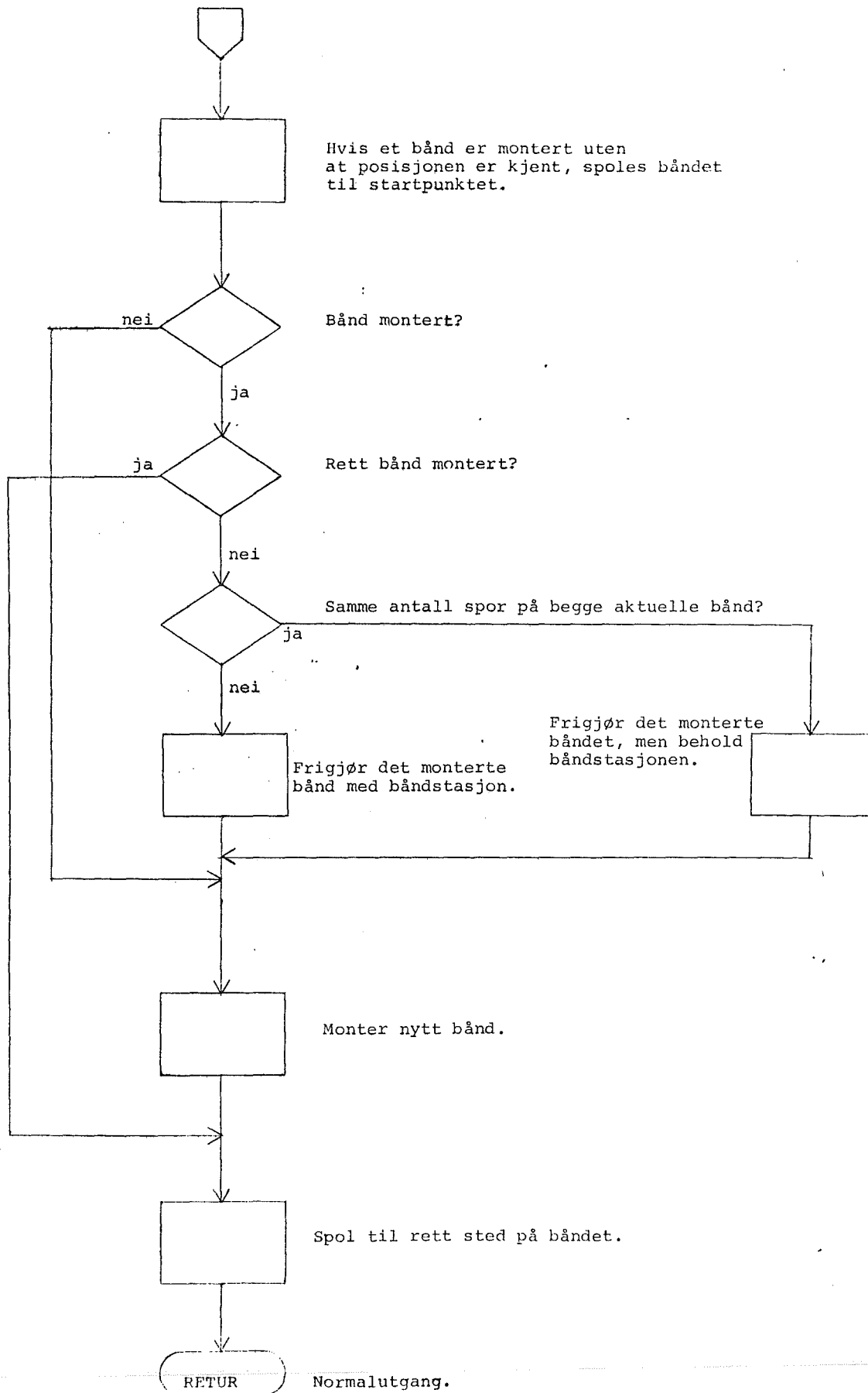
Generelt:

Rutinen brukes ved tilordning av eksisterende datafiler på platelager (disc) eller magnetbånd, samt ved oppretting av nye filer på magnetbånd. MONTIN leser inn initialverdier (lengde magnetbåndtabell og Marsdenrute-tabeller, oppdateringsdatoer og fortegnelse over Marsdenrutenes plass i tabellene) fra tabellfilen (REGFIL2). MONTER (se prinsippskjema) tilordner utfra oppgitte kriterier en discfil eller magnetbåndfil med data (hvis en slik finnes). TBAAND (se prinsippskjema) monterer et bånd slik at dette kan brukes til skriving.

Formelle parametre:

NAVN:	TYPE:	BRUK:
IDAPO	INTEGER	Antall poster i tabellfilen.
NRT	INTEGER	Nummer på det monterte magnetbånd. Verdien 0 betyr ingen bånd monterte.
NRF	INTEGER	Filnummer på eventuelt magnetbånd.
NUN	INTEGER	I/O-enhetsnummer for den fil/-magnetbånd som er/skal "monteres".
MSQ	INTEGER	Marsdenrute-nummer.
\$	Linjenummer	Retur fra rutinen hvis ingen fil er monterte-tilordnet.
IREG	INTEGER	Registertypenummer.
TAPE	LOGICAL	Indikator for filtype som er tilordnet det aktuelle I/O-enhetsnummer. Verdien TRUE angir magnetbåndfil eller ingen fil, FALSE platelagerfil.





NAVN:	TYPE:	BRUK:
NMOD	INTEGER	Typenummer for valg av navn på eventuell platelagerfil.
LOTAPE	LOGICAL	Indikator for tillatt filtype. Verdien TRUE medfører at den ønskede fil skal være en magnetbåndfil.
LODISC	LOGICAL	Indikator for tillatt filtype. Verdien TRUE medfører at den ønskede fil skal være en platelagerfil.
ITAB	INTEGER	Marsdenrutetabellnummer.
NRTNY	INTEGER	Nytt båndnummer (ved skriving).
NRFNY	INTEGER	Filnummer (start skriving).
OVER	LOGICAL	Angir hvorvidt det skal være tillatt (TRUE) å skrive over eksisterende data på et bånd.
ITDS(6)	INTEGER	Inneholder en fullstendig ordre for tilordning av magnetbåndet (skriving). Kontrolleres mot tilsvarende opplysninger i magnetbåndtabell.

COMMON variable:

RDATA:

NAVN:	TYPE:	BRUK:
MSQTAB(72)	INTEGER	Inneholder Marsdenrutenummer. Peger til tilhørende linje i Marsdenrutetabellene.
LBT	INTEGER	Magnetbåndtabellens lengde.
LMSQT	INTEGER	Marsdenrutetabellenes lengde.
IDATBT	INTEGER	Siste oppdateringsdato for magnetbåndtabell.
IDATMT(5)	INTEGER	Siste oppdateringsdato for Marsdenrutetabell nr. 1-5.

Variable:

NAVN:	TYPE:	BRUK:
NA(6)	INTEGER	Hjelpearray for rutinene ERTRAN og NERTRN (inneholder styresetn.).
K	INTEGER	Hjelpevariabel.

NAVN:	TYPE:	BRUK:
KA	INTEGER	Hjelpevariabel.
NAVN	INTEGER	Inneholder filnavn for platelagerfil.
IFAC	INTEGER	Statusvariabel fra NERTRN.
KB	INTEGER	Hjelpvariabel.
L	INTEGER	Hjelpevariabel.
J	INTEGER	Hjelpevariabel.
N	INTEGER	Hjelpevariabel.
NFILE	INTEGER	Nytt filnummer.
NR	INTEGER	Nytt båndnummer.
NTAPE(6)	INTEGER	Inneholder tilordningssetning fra magnetbåndtabell.
IGSPOR	INTEGER	Antall spor på montert magnetbånd.
IAS	INTEGER	Antall spor på magnetbånd som skal monteres.
IAFIL	INTEGER	Antall filer på et bånd (leses fra magnetbåndtabell).
IABL	INTEGER	Hjelpevariabel.
III	INTEGER	Hjelpevariabel.

Eksterne rutiner: RNAVN (Navngir eventuell discfil).

Interne rutiner: HJELP (Tilordner I/0-enhetsnummer til magnetbånd).

LEST (Leser data fra magnetbåndtabell)

INNG (Leser data for et eventuelt midlertidig fraordnet magnetbånd og tilordner dette påny. Frigjør I/0-enhetsnummeret fra discfilkrav.)

Faste I/0-enhetsnummer: 2 (Magnetbånd og Marsdenrutetabeller m.m.).

```
000001 007 C
000002 007 C
000003 007 C
000004 007 C
000005 007 C
000006 007 C
000007 007 C
000008 007 C
000009 007 C
000010 007 C
000011 007 C
000012 007 C
000013 007 C
000014 007 C
000015 007 C
000016 007 C
000017 007 C
000018 007 C
000019 007 C
000020 007 C
000021 007 C
000022 007 C
000023 007 C
000024 007 C
000025 007 C
000026 007 C
000027 007 C
000028 007 C
000029 007 60
000030 007 C
000031 007 C
000032 007 C
000033 007 C
000034 007 C
000035 007 C
000036 007 C
000037 007 C
000038 007 C
000039 010 C
000040 007 C
000041 007 C
000042 007 C

*****
*
*   MONTIN   *
*   MONTER  *
*   TBAAND  *
*
*****

MONTIN MED ENTREPUNKTER TILORDNER,UTFRA OPPGITTE PARAMETRE,
EN FIL, DENNE KAN VARE ENTEN EN MAGNETBANDFIL (FINNES VED
HJELP AV MSG-REGISTERET), ELLER DISCFIL (FINNES VED HJELP
AV OPPGITTE NAVNEPARAMETRE OG RUTINEN RNAVN).

NØDVENDIGE RUTINER ; RNAVN

SUBROUTINE MONTIN(IDAPO)
COMMON/RDATA/MSGTAB(72),LBT,LMSGT,IDATBT,IDATMT(5)
DEFINE FILE 2(IDAPO,48,L,III)
READ(2'1) LBT,LMSGT,IDATBT,IDATMT
DO 60 KA=2,10
KB=8*(KA-1)
N=KB-7
READ(2'KA) (MSGTAB(L),L=N,KB)
CONTINUE
RETURN
ENTRY MONTER(NRT,NRF,NUN,MSQ,I,IREG,TAPE,NMOD,LOTAPE,LODISC,
ITAB)

TILORDNER ØNSKET FIL.

LOGICAL LODISC,LOTAPE,TAPE
DIMENSION NA(6),NTAPE(6)
DIMENSION ITDS(6)
LOGICAL OVER
IF(TAPE) GO TO 70
ENCODE(50,NA) NUN
WRITE(6,40) NA

IFAC=NERTRN(6,NA)      6 FRIGJØR GAMMEL FIL.
IF (IFAC.NE.0) WRITE(6,53) IFAC
CONTINUE
IF (LOTAPE) GO TO 100      6KREVES MAGNETBAND?
IF (NMOD.EQ.0) GO TO 100

TILORDNING AV DISCFIL FORSØKES.

FORMAT(8HGFREE,A ,I2,4H. . )
CALL RNAVN(MSG,IREG,NAVN,NMOD)
FORMAT(8HGASG,AX ,A6,4H. . )
FORMAT(5H&USE ,I2,2H.,A6,4H. . )
FORMAT(4X,'FAC,STAT. ',0I2)
ENCODE(51,NA)NAVN
WRITE(6,40)NA
IFAC=NERTRN(6,NA)
IF( IFAC.NE.0) WRITE(6,53)IFAC
IF (IFAC.LT.0) GO TO 100      6DISCFIL UTILGJENGELIG?
IF (.NOT.TAPE) GO TO 80
TAPE=.FALSE.
ENCODE(50,NA)NUN      6FRIGJØR MIDLERTIDIG
WRITE(6,40)NA      6EVENTUELT MAGNETBAND FRA
IFAC=NERTRN(6,NA)      6DET OPPGITTE I/O-ENHETSNUMMER.
IF( IFAC.NE.0) WRITE(6,53)IFAC
IF (IFAC.EQ.0) CALL NTRAN(NUN,18)
CONTINUE
ENCODE(52,NA)NUN,NAVN
WRITE(6,40) NA
CALL ERTRAN(6,NA)
CALL NTRAN(NUN,10,22)
RETURN
CONTINUE

MAGNETBAND

IF (LODISC) GO TO 600      6KREVES DISCFIL?(INGEN FIL TILORDNES)

FINN RETT LINJE I MSG-TABELLEN.

DO 110 L=LMSGT,1,-1
IF (MSGTAB(L).EQ.MSG) GO TO 120
CONTINUE
GO TO 600
FORMAT(8H&FREE,S ,I2,4H. . )
```

```
000087 007 15 FORMAT(6H6FREE ,12,4H. . )
000088 007 40 FORMAT(2X,'***',6A6,'LLL')
000089 007 120 CONTINUE
000090 007 41 FORMAT(2I3)
000091 007 IF (ITAB.EQ.0) CALL ERTRAN(2)
000092 007 KA=10+LBT+LMSQT*(ITAB-1)+L
000093 007 READ(2,KA) NA
000094 007 DECODE(41,NA(IREG)) NR,NFILE
000095 007 IF (NR.EQ.0) GO TO 600
000096 007 CALL INNG
000097 007 CALL LEST(NR,IAS)
000098 007 GO TO 150
000099 007 C
000100 007 ENTRY TBAAND(NRT,NRF,NUN,TAPE,NRTNY,NRFNY,OVER,ITDS)
000101 007 C
000102 007 C TBAAND KLARGJØR ET MAGNETBAND FOR SKRIVING ();MONTERER OG
000103 007 C SPOLER TIL RETT STED)
000104 007 C
000105 007 30 FORMAT(4X,'START SKRIVING,BAND:',13,' FIL:',13,5X,'OVER?',L1)
000106 007 WRITE(6,30) NRTNY,NRFNY,OVER
000107 011 IF ((NRTNY.EQ.0).OR.(NRFNY.EQ.0)) CALL ERTRAN(2)
000108 007 CALL INNG
000109 007 CALL LEST(NRTNY,IAS)
000110 007 DO 125 KA=1,6
000111 007 IF (NTAPE(KA).NE.ITDS(KA)) CALL ERTRAN(2)
000112 007 125 CONTINUE
000113 007 NR=NRTNY
000114 007 NFILE=NRFNY
000115 007 IF ((NRFNY.LE.IAFIL).AND.(.NOT.OVER)) CALL ERTRAN(2)
000116 007 150 CONTINUE
000117 007 IF(NRT.EQ.0) GO TO 180 6MAGNETBAND IKKE MONTERET?
000118 007 IF(NRF.EQ.0) CALL NTRAN(NUN,10,22) 6SPOL TIL START HVIS
000119 007 6FILNR. ER UOPPGITT.
000120 007 IF(NRF.EQ.0)NRF=1
000121 007 180 CONTINUE
000122 007 TAPE=.TRUE.
000123 007 IF(NRT.EQ.0)GO TO 200
000124 007 IF(NR.EQ.NRT)GO TO 500
000125 007 C
000126 007 C DET MONTERTE BANDET FJERNES.
000127 007 C
000128 007 IF(IGSPOR.EQ.IAS)ENCODE(10,NA)NUN
000129 007 IF(IGSPCR.NE.IAS)ENCODE(15,NA)NUN
000130 007 CALL ERTRAN(6,NA)
```

```
000131 007 WRITE(6,40)NA
000132 007 200 CONTINUE
000133 007 C
000134 007 C NYTT BAND MONTERES.
000135 007 C
000136 007 WRITE(6,40)NTAPE
000137 007 IFAC=NERTRN(6,NTAPE)
000138 007 IF (IFAC.NE.0) WRITE(6,53) IFAC
000139 007 IF (IFAC.LT.0) CALL ERTRAN(2)
000140 007 CALL HJELP
000141 007 CALL NTRAN(NUN,10,22)
000142 007 NRT=NR
000143 007 NRF=1
000144 007 500 CONTINUE
000145 007 C
000146 007 C SPOL TIL RETT FIL
000147 007 C
000148 007 IF(NFILE.NE.1)GO TO 520
000149 007 NRF=1
000150 007 CALL NTRAN(NUN,10,22)
000151 007 RETURN
000152 007 520 CONTINUE
000153 007 IF (NRF.EQ.C) CALL NTRAN(NUN,10,22)
000154 007 IF (NRF.EQ.0) NRF=1
000155 007 IF (NFILE.EQ.NRF) GO TO 550
000156 007 IF(NFILE.LT.NRF)K=NFILE-NRF-1
000157 007 IF(NFILE.GT.NRF)K=NFILE-NRF
000158 007 CALL NTRAN(NUN,8,K)
000159 007 IF(NFILE.LT.NRF)CALL NTRAN(NUN,8,1)
000160 007 NRF=NFILE
000161 007 RETURN
000162 007 550 CONTINUE
000163 007 CALL NTRAN(NUN,8,-1)
000164 007 CALL NTRAN(NUN,8,1)
000165 007 NRF=NFILE
000166 007 RETURN
000167 007 600 CONTINUE
000168 007 C
000169 007 C
000170 007 C ØNSKET FIL ER UTILGJENGELIG
000171 007 C
000172 007 25 FORMAT(40X,'MSQ NR. ',13,' ER IKKE TILGJENGELIG')
000173 007 WRITE(6,25) MSQ
000174 007 RETURN 5
```

```
000175 007 C
000176 007 SUBROUTINE HJELP
000177 007 C
000178 007 C HJELP TILORDNER ET I/O-ENHETSNR. TIL ET GITT MAGNETBAND.
000179 007 C
000180 007 20 FORMAT(5HGUSE ,I2,5H,, )
000181 007 40 FORMAT(2X,'*'*',6A6,'LLL')
000182 007 ENCODE(20,NA)NUN
000183 007 DO 220 J=3,6
000184 007 NA(J)=NTAPE(J-1)
000185 007 220 CONTINUE
000186 007 DO 240 J=4,6
000187 007 DO 240 K=0,30,6
000188 007 N=FLD(K,6,NA(J))
000189 007 IF (N.EQ.46) GO TO 250
000190 007 240 CONTINUE
000191 007 STOP FEIL
000192 007 250 CONTINUE
000193 007 FLD(K,6,NA(J))=5
000194 007 K=K+6
000195 007 IF (K.EQ.36) J=J+1
000196 007 IF (K.EQ.36) K=0
000197 007 FLD(K,6,NA(J))=61
000198 007 K=K+6
000199 007 IF (K.EQ.36) J=J+1
000200 007 IF (K.EQ.36) K=0
000201 007 FLD(K,6,NA(J))=5
000202 007 WRITE(6,40) NA
000203 007 CALL ERTRAN(6,NA)
000204 007 RETURN
000205 007 C
000206 007 SUBROUTINE LEST(NRTAPE,IASPOR)
000207 007 C
000208 007 C LESER DATA FRA MAGNETBANDTABELL
000209 007 C
000210 007 31 FORMAT(2I5,IX,I1,6A6)
000211 007 IF (NRTAPE.GT.LBT) CALL ERTRAN(2)
000212 007 L=NRTAPE+10
000213 007 READ(2'L,31) IAFIL,IABL,IASPOR,NTAPE
000214 007 RETURN
000215 007 C
000216 007 SUBROUTINE INNG
000217 010 C
000218 010 C FRIGJØR I/O-ENHETSNUMMERET FRA DISCFILKRAV.

000219 010 C TILORDNER ET EVENTUELT MIDLERTIDIG FRAORDNET MAGNETBAND.
000220 010 C OG LESER INN PARAMETRE FOR DETTE FRA MAGNETBANDTABELLEN.
000221 009 IF(NRT.NE.0) CALL LEST(NRT,IGSPCR)
000222 007 IF (TAPE) RETURN
000223 007 CALL NTRAN(NUN,18)
000224 007 IF (NRT.EQ.C) RETURN
000225 007 CALL HJELP
000226 007 RETURN
000227 007 END
```


OPPD2

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Entrepunkter : RNREG, RNTAPE, SLUTT2
Dato : 2/3-76
Utført av : Ø. Strand
Formål : Brukes ved oppdatering av sorterings-
systemets Marsdenrute-tabeller og
magnetbåndtabell (REGFIL2).

Kall: OPPD2
RNREG(MSQ, IREG, NRB, NRF, ITAB)
RNTAPE(NR, NF, NB, MOD, NFRA)
SLUTT2

Generelt:

Sorteringssystemets magnetbåndtabell og Marsdenrutetabellene er lagret på en direkte - aksess fil. OPPD2 med entrepunkter brukes ved oppdatering av disse tabellene.

OPPD2 skriver ut dato m.m. for de eksisterende tabellene.

RNREG retter Marsdenrutetabellene.

RNTAPE retter magnetbåndtabellen.

SLUTT2 fjerner datoangivelsen for Marsdenrutetabeller som er tomme, samt skriver de nye datoangivelsene m.m. tilbake til registerfilen. Initialverdiene (tabellengder, datoer m.m.) må være innlest før OPPD2 kalles.

Formelle parametre:

NAVN:	TYPE:	BRUK:
MSQ	INTEGER	Marsdenrute-nummer.
IREG	INTEGER	Registertypenummer.
NRB	INTEGER	Nytt båndnummer.
NRF	INTEGER	Nytt filnummer.
NR	INTEGER	Magnetbåndnummer.
NF	INTEGER	1. Antall nye filer på et bånd. 2. Totalt antall filer på et bånd.
NB	INTEGER	Antall nye blokker på et bånd
MOD	INTEGER	Angir korreksjonstype for magnetbånd- tabell. 0): Filantall erstattes med oppgitt verdi, blokkantall justeres i over- ensstemmelse med gitte opplysning- er for første nye filnummer på båndet. (): Hvis endel av de gamle filene på båndet er overskrevet, av- merkes dette i magnetbåndtabellen). 1): Filantall/blokkantall adderes til gammel sum.

NAVN:	TYPE:	BRUK:
		2): Filantall/blokkantall settes inn istedenfor gamle verdier. 3): Filantall erstattes med oppgitt verdi, blokkantall økes.
ITAB	INTEGER	Marsdenrutetabellnummer (p. t. fem tabeller).
NFRA	INTEGER	Angir laveste nye filnummer på et bånd.

COMMON variable:

RDATA:

NAVN:	TYPE:	BRUK:
MSQTAB(72)	INTEGER	Inneholder Marsdenrutenummer. Viser tilhørende linje i tabellene.
LBT	INTEGER	Magnetbåndtabellens lengde.
LMSQT	INTEGER	Marsdenrutetabellenes lengde.
IDATBT	INTEGER	Siste oppdateringsdato for magnetbåndtabellen.
IDATMT(5)	INTEGER	Siste oppdateringsdato for Marsdenrutetabell nr. 1-5.

Variabel:

NAVN:	TYPE:	BRUK:
I	INTEGER	Hjelpevariabel.
II	INTEGER	Hjelpevariabel.
IM	INTEGER	Hjelpevariabel (ved listing av dato).
ISPOR	INTEGER	Antall spor (magnetbånd).
ISTAR	INTEGER	Merke hvis blokkantall i magnetbåndtabellene er feil (): for høyt). Brukes hvis det nøyaktige antall overskrevne blokker på et bånd er ukjent.
IY	INTEGER	Hjelpevariabel (ved listing av dato).
J	INTEGER	Hjelpevariabel.
L	INTEGER	Hjelpevariabel.
LL	INTEGER	Hjelpevariabel.

NAVN:	TYPE:	BRUK:
NA(6)	INTEGER	Brukes ved korreksjon av Marsden-rutetabellene. Inneholder båndnummer/filnummer for registertype 1-6.
NBG	INTEGER	Gammelt blokkantall innlest fra magnet-båndtabell.
NFG	INTEGER	Gammelt filantall innlest fra magnet-båndtabell.
TOM(5)	LOGICAL	Hjelpetabell som brukes til å angi hvorvidt adresseangivelser i magnet-båndtabellene er fjernet. Hvis så er tilfelle, blir vedkommende tabell(er) undersøkt til slutt. Hvis det da viser seg at de(n) er tomme, nullstilles datoangivelsen.

Faste I/O-enhetsnumre: 2(D.A. fil for tabellene)

```

2
4 000001 003 C
000002 003 C
6 000003 003 C *****
000004 003 C *
8 000005 003 C * OPPD2 *
000006 003 C * RNREG *
10 000007 003 C * RNTAPE *
000008 003 C * SLUTT2 *
12 000009 003 C *
000010 003 C *****
14 000011 003 C
000012 003 C
16 000013 003 C OPPD2 MED ENTREPUNKTER OPPDATERER MAGNETBANDTABELLEN
000014 003 C OG MARSDENRUTE-TABELLENE
18 000015 003 C
000016 003 C
20 000017 003 C FASTE I/O-ENHETSNR. : 2 (TABELLENE)
000018 003 C
22 000019 003 C
000020 003 C SUBROUTINE OPPD2
24 000021 003 C
000022 003 C UTSKRIFT AV DIVERSE OPPLYSNINGER
26 000023 003 C
000024 003 DIMENSION NA(6)
28 000025 003 COMMON/RDATA/MSQTAB(72),LBT,LMSQT,IDATBT,IDATMT(5)
000026 003 LOGICAL TOM(5)/5*.FALSE./
30 000027 003 12 FORMAT(4X,'REGFIL2 OPPDATERES;')
000028 005 13 FORMAT(4X,'MAGNETBANDTABELL SIST OPPDATERT: ',I2,IH/,J2,IH-,I2)
32 000029 005 14 FORMAT(4X,'MARSDENRUTETABELL NR.',I2,' SIST OPPDATERT: ',I2,IH/,
000030 005 *J2,IH-,I2)
34 000031 003 WRITE(6,12)
000032 003 16 FORMAT(4X,'FØLGENDE TABELLER ER I BRUK;')
36 000033 005 18 FORMAT(3I2)
000034 003 WRITE(6,16)
38 000035 003 IF(LBT.EQ.0) CALL ERTRAN(2)
000036 005 DECODE(18,IDATBT) IM,ID,IY
40 000037 007 WRITE(6,13)ID,IM,IY @DATO,MAGNETBANDTABELL
42 000038 007 C
000039 007 C DATO,MARSDENRUTETABELL.
44 000040 007 C
000041 003 DO 100 L=1,5
000042 003 IF(IDATMT(L).EQ.0) GO TO 100

2 000043 005 DECODE(18,IDATMT(L)) IM,ID,IY
4 000044 003 WRITE(6,14)L,ID,IM,IY
6 000045 003 100 CONTINUE
000046 003 RETURN
8 000047 003 C
000048 003 ENTRY RNREG(MSQ,IREG,NRB,NRF,ITAB)
10 000049 003 C
000050 003 C RETTER MARSDENRUTE-TABELLENE
12 000051 003 C
000052 004 IF (ITAB.EQ.0) CALL ERTRAN(2)
14 000053 003 DO 200 L=LMSQT,1,-1
000054 003 IF(MSQTAB(L).EQ.MSQ) GO TO 210
16 000055 003 200 CONTINUE
000056 003 CALL ERTRAN(2)
18 000057 003 210 CONTINUE
000058 007 J=10+LBT+LMSQT*(ITAB-1)+L @POSTNUMMER I D.A. FILEN,
20 000059 003 READ(2'J)NA
000060 003 10 FORMAT(2I3)
22 000061 003 ENCODE(10,NA(IREG))NRB,NRF
000062 007 CALL ERTRAN(9,IDATBT,ITAB),L) @DATOANGIVELSEN RETTES,
24 000063 003 IF(NRB.NE.0) GO TO 220
000064 003 NA(IREG)=6H
26 000065 003 TOM(ITAB)=.TRUE.
000066 003 220 CONTINUE
000067 003 WRITE(2'J) NA
000068 003 RETURN
28 000069 003 C
000070 005 ENTRY RNTAPE(NR,NF,NB,MOD,NFRA)
30 000071 003 C
000072 003 C RETTER MAGNETBANDTABELLEN
32 000073 003 C
000074 007 L=10+NR @POSTNUMMER I D.A. FILEN,
34 000075 003 READ(2'L,11)NFG,NBG,ISTAR,ISPOR,NA
000076 003 11 FORMAT(2I5,A1,I1,6A6)
36 000077 007 CALL ERTRAN(9,IDATBT,J) @DATOANGIVELSEN RETTES,
000078 005 IF (MOD.NE.0) GO TO (250,260,280),MOD
38 000079 005 IF(NFRA.EQ.1) GO TO 260
000080 006 IF (NFRA.LE.NFG) GO TO 280
40 000081 005 GO TO 285
000082 003 250 CONTINUE
42 000083 003 C
000084 003 C FILANTALL/BLOKKANTALL ØKES
44 000085 003 C
000086 003 NFG=NFG+NF

```

```
000087 003 NBG=NBG+NB
000088 003 IF(NB.LT.0) ISTAR=IH
000089 003 WRITE(2'L,11)NFG,NBG,ISTAR,ISPOR,NA
000090 003 RETURN
000091 003 260 CONTINUE
000092 003 C
000093 003 C FILANTALL/BLOKKANTALL ERSTATTES MED OPPGITTE VERDIER
000094 003 C
000095 003 ISTAR=IH
000096 003 WRITE(2'L,11)NF,NB,ISTAR,ISPOR,NA
000097 003 RETURN
000098 003 280 CONTINUE
000099 003 C
000100 005 C FILANTALL ERSTATTES MED OPPGITT VERDI
000101 003 C BLOKKANTALL ØKES
000102 003 C
000103 003 ISTAR=IH*
000104 005 285 CONTINUE
000105 003 NBG=NBG+NB
000106 003 WRITE(2'L,11)NF,NBG,ISTAR,ISPOR,NA
000107 003 RETURN
000108 003 C
000109 003 ENTRY SLUTT2
000110 003 C
000111 003 C OPPDATERING AVSLUTTES.
000112 003 C
000113 003 15 FORMAT(1H1,3X,'REGFIL2 ER OPPDATERT')
000114 003 WRITE(6,15)
000115 003 C
000116 003 C DATOANGIVELSE FOR SISTE OPPDATERING FJERNES FOR EVENTUELLE
000117 003 C TØMTE MARS DENRUTE-TABELLER,
000118 003 C
000119 003 DO 400 J=1,5
000120 003 IF(.NOT.TOM(J)) GO TO 400
000121 003 L=LBT+10+LMSQT*(J-1)+1
000122 003 LL=L+LMSQT-1
000123 003 DO 350 I=L,LL
000124 003 READ(2'1)NA
000125 003 DO 300 II=1,6
000126 003 IF(NA(II).NE.6H ) GO TO 400
000127 003 300 CONTINUE
000128 003 350 CONTINUE
000129 003 IDATMT(J)=0
000130 003 WRITE(6,17)J

000131 003 400 CONTINUE
000132 003 WRITE(6,16)
000133 005 DECODE(18,IDATBT) IM,ID,IY
000134 003 WRITE(6,13)ID,IM,IY
000135 003 17 FORMAT(4X,'MARS DENRUTETABELL NR:',12,' ER TOM')
000136 003 DO 500 J=1,5
000137 003 IF(IDATMT(J).EQ.0) GO TO 500
000138 005 DECODE(18,IDATMT(J)) IM,ID,IY
000139 003 WRITE(6,14)J,ID,IM,IY
000140 003 500 CONTINUE
000141 003 WRITE(2'1)LBT,LMSQT,IDATBT,IDATMT
000142 003 RETURN
000143 003 END
```

NY3

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Entrepunkter : OPPD3
Dato : 26/2-76
Utført av : Ø. Strand
Formål : Brukes ved oppdatering av sorterings-
systemets fortegnelse over antall stasjoner
pr. engradersrute (engradersrutetabellene,
REGFIL3).

Kall: NY3(KOPG)
OPPD3(MSQ, LINJE, NTELL, MOD, BEGGE)

Generelt:

Fortegnelsen over antall stasjoner pr. engradersrute er lagret i et antall poster á hundre ord på en direkte aksessfil. Hver post inneholder fortegnelse for en Marsdenrute. Rekkefølgen av postene er den samme som i Marsdenrutetabellene. Oppdatering skjer ved at korreksjoner (og fortolkningsparametre) gis til rutinen som så danner en ny fil med oppdatert innhold. Hvis ønskelig, kan korreksjonene også lagres uendret på en egen ekstrasfil.

Formelle parametre:

NAVN:	TYPE:	BRUK:
KOPG	LOGICAL	Angir (.TRUE.) om den gamle tabellen skal kopieres over til den nye i sin helhet.
MSQ	INTEGER	Marsdenrutenummer.
LINJE	INTEGER	Linjenummer i Marsdenrutetabell. Må oppgis hvis ikke Marsdenrutenummer er oppgitt.
NTELL (100)	INTEGER	Inneholder korreksjoner til en Marsdenrute.
MOD	INTEGER	Fortolkningsparameter: =0): Korreksjonen erstatter eventuelle gamle verdier. =1): Korreksjonen adderes til eksisterende innhold i ny fil. =2): Korreksjonen adderes sammen med vedkommende rutes gamle tabell (som må hentes fra gammel fil).
BEGGE	LOGICAL	Angir (.TRUE.) at korreksjonen også skal lagres uendret på en ekstrasfil.

COMMON variable:

RDATA:

NAVN:	TYPE:	BRUK:
MSQTAB (72)	INTEGER	Inneholder Marsdenrutenummer, peker til linjenummer i Marsdenrutetabellene.
LBT	INTEGER	Lengde på magnetbåndtabell.
LMSQT	INTEGER	Lengde på Marsdenrutetabell.

Verdier til RDATA må leses inn av andre programdeler.

DATT:

TILTAA	LOGICAL	Angir (. TRUE.) hvorvidt ekstrafilen er definert for direkte aksessbruk.
--------	---------	--

Variable:

NAVN:	TYPE:	BRUK:
I	INTEGER	Hjelpevariabel ved definisjon av direkte aksessfil. Brukes ikke andre steder i programmet.
II	INTEGER	Som I.
III	INTEGER	Som I.
DEFG	LOGICAL	Angir (. TRUE.) om originalfilen er definert for direkte aksessbruk.
J	INTEGER	Hjelpevariabel.
NDIV (100)	INTEGER	Hjelpetabell ved oppdatering.
Faste I/O-enhetsnumre:		3 (Originalutgave (gammel utgave) av engradersfortegnelsen). 26 (Ny utgave) 28 (Ekstrafil)
Interne rutiner:		NULL (Nullstiller en hel fil).

4 000001 005 C
 000002 005 C
 6 000003 005 C
 000004 005 C
 8 000005 005 C
 000006 005 C
 10 000007 005 C
 000008 005 C
 12 000009 005 C
 000010 005 C
 14 000011 005 C
 000012 005 C
 16 000013 005 C
 000014 005 C
 18 000015 005 C
 000016 005 C
 20 000017 005 C
 000018 005 C
 22 000019 005 C
 000020 005 C
 24 000021 005 C
 000022 005 C
 26 000023 005 C
 000024 005 C
 28 000025 005 C
 000026 005 C
 30 000027 005 C
 000028 008 C
 000029 005 10 C
 000030 005 C
 000031 005 C
 000032 009 C
 000033 009 C
 000034 005 C
 000035 009 C
 000036 005 C
 000037 009 C
 000038 009 C
 000039 009 C
 000040 005 C
 000041 005 C
 000042 005 C

```
*****
*
*      NY3      *
*      OPPD3    *
*
*****
```

NY3 MED ENTREPUNKTER OPPDATERER TABELLENE MED FORTEGNELSE
 OVER ANTALL STASJONER PR. ENGRADERSRUTE.

FASTE I/O-ENHETSNUMRE : 28 (EKSTRA TABELL)
 3 (TABELLENE SOM SKAL RETTES)
 26 (NY UTGAVE AV TABELLENE)

SUBROUTINE NY3(KOPG)

INITIALISERING.

```
COMMON/RDATA/MSQTAB(72),LBT,LMSGT
DIMENSION NDIV(100),NTELL(100)
LOGICAL KOPG,DEFG,TILTA,BEGGE
COMMON/DATT/TILTA
FORMAT(//20X,'*** ADVARSEL; MULIG FEIL I OPPD3 (' ,L1,I2,')')//)
DEFG=.FALSE.
TILTA=.FALSE.
DEFINE FILE 26(72,100,V,I) 6DEFINERER NY FIL
IF (.NOT.KOPG) CALL NULL(26) 6NULLSTILLER NY FIL
IF (.NOT.KOPG) RETURN
DEFINE FILE 3(72,100,U,II) 6ORIGINALFIL DEFINERES
DEFG=.TRUE.

DO 85 J=1,72
READ(3'J) NDIV
WRITE(26'J) NDIV
```

000043 005 85
 000044 005
 000045 005 C
 000046 005
 000047 005 C
 000048 005 C
 000049 005 C
 000050 005
 000051 005 C
 000052 009 C
 000053 005 C
 000054 005
 000055 005
 000056 005 140
 000057 005
 000058 005 150
 000059 005
 000060 006
 000061 006
 000062 009
 000063 009
 000064 006
 000065 006 155
 000066 009
 000067 006 160
 000068 005
 000069 005 C
 000070 005 C
 000071 005 C
 000072 005
 000073 005
 000074 005 250
 000075 005 C
 000076 005 C
 000077 005 C
 000078 005 C
 000079 005
 000080 005
 000081 005 260
 000082 005
 000083 005
 000084 005 270
 000085 005
 000086 005

```
CONTINUE
RETURN

ENTRY OPPD3(MSQ,LINJE,NTELL,MOD,BEGGE)

OPPDATERER TABELLEN FOR EN MSG.

IF(MSQ.EQ.0)GO TO 150

BESTEM POSTNUMMER.

DO 140 LINJE=LMSQT,1,-1
IF (MSQTAB(LINJE).EQ.MSQ) GO TO 150
CONTINUE
CALL ERTRAN(2)
CONTINUE
MSQ=MSQTAB(LINJE)
IF (.NOT.BEGGE) GO TO 160
IF (TILTA) GO TO 155
DEFINE FILE 28(72,100,V,III) 6EKSTRAFIL DEFINERES
CALL NULL(28) 6EKSTRAFIL NULLSTILLES
TILTA=.TRUE.
CONTINUE
WRITE(28'LINJE) NTELL 6TIL EKSTRAFIL
CONTINUE
IF (MOD.NE.0) GO TO (250,300),MOD

INNHOLDET I NTELL DANNER NY TABELL FOR VEDKOMMENDE MARSDEN=RUTE.

WRITE(26'LINJE) NTELL
RETURN
CONTINUE

NTELL ADDERES TIL ALLEREDE EKSISTERENDE INNHOLD I DEN
NYE VERSJON AV TABELLENE.

IF (.NOT.KOPG) WRITE(6,10) KOPG,MOD
READ(26'LINJE) NDIV
CONTINUE
DO 270 J=1,100
NDIV(J)=NDIV(J)+NTELL(J)
CONTINUE
WRITE(26'LINJE) NDIV
RETURN
```



```
2 000087 005 300 CONTINUE
3 000088 005 C
4 000089 005 C NTELL ADDERES TIL VEDKOMMENDE RUTES GAMLE TABELL,
5 000090 005 C
6 000091 005 IF (.NOT.DEFG) DEFINE FILE 3(50,100,U,11)
7 000092 005 IF (.NOT.DEFG) DEFG=.TRUE.
8 000093 005 IF (KOPG) WRITE(6,10) KOPG,MOG
9 000094 005 READ(3,LINJE) NDIV
10 000095 005 GO TO 260
11 000096 006 C
12 000097 006 SUBROUTINE NULL(NUN)
13 000098 006 C
14 000099 006 C NULLSTILLER EN FIL
15 000100 006 C
16 000101 007 DO 100 J=1,100
17 000102 006 NDIV(J)=0
18 000103 006 100 CONTINUE
19 000104 006 DO 150 J=1,72
20 000105 006 WRITE(NUN,J) NDIV
21 000106 006 150 CONTINUE
22 000107 006 RETURN
23 000108 005 END
24
26
28
30
32
34
36
38
40
42
44
46
```

ASGUP

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Entrepunkter : ASGAX, USE, FREE, FREEA, FREEB,
FREES
Dato : 1/3-76
Utført av : Ø. Strand
Formål : ASGUP med entrepunkter er en
samling rutiner som oppretter nye
filer, tilordner gamle, tilordner
I/0-enhetsnumre samt frigjør filer.

Kall: ASGUP (NUN, NAVN, TEMP)
ASGAX (NUN, NAVN)
USE (NUN, NAVN)
FREEA (NUN, TEMP)
FREE (NUN, NAVN)
FREEB(NUN)
FREES(NUN)

Generelt:

ASGUP oppretter en ny fil (temporær eller katalogisert). Hvis en fil med det aktuelle navn allerede eksisterer, tilordnes denne. Filen tilordnes også et I/0-enhetsnummer.

ASGAX tilordner eksisterende filer og gir disse et I/0-enhetsnummer.

USE tilordner et I/0-enhetsnummer til en fil.

FREEA frigjør katalogiserte filer fra kjøringen (som FREEB). For temporære filer frigjøres I/0-enhetsnummeret fra filen.

FREE frigjør en fil fra kjøringen.

FREEB frigjør et I/0-enhetsnummer fra en fil, og frigjør også filen fra kjøringen hvis denne ikke er tilordnet flere I/0-numre.

FREES demonterer et magnetbånd, men reserverer magnetbåndstasjonen for fortsatt bruk.

Formelle parametre:

NAVN:	TYPE:	BRUK:
NUN	INTEGER	I/0-enhetsnummer.
NAVN	INTEGER	Inneholder navn på filen som skal opprettes/tilordnes/frigjøres (inntil seks tegn, høyrejustert). I enkelte tilfeller kan NAVN angi et I/0-enhetsnummer (ASGUP, USE)
TEMP	LOGICAL	Angir hvorvidt den aktuelle fil er temporær eller katalogisert.

Variable:

NAVN:	TYPE:	BRUK:
I	INTEGER	Statusvariabel for NERTRN.
NA(6)	INTEGER	Inneholder styresetning ved kall av ERTRAN.

```

2
4 000001 006 C
000002 006 C
6 000003 006 C *****
000004 006 C *
8 000005 006 C *   ASGUP   *
000006 006 C *   ASGAX   *
10 000007 006 C *   USE     *
000008 006 C *   FREEA   *
12 000009 006 C *   FREE    *
000010 006 C *   FREEB   *
14 000011 006 C *   FREES   *
000012 006 C *           *
16 000013 006 C *****
000014 006 C
18 000015 006 C
000016 006 C   ASGUP MED ENTREPUNKTER DANNER EN SAMLING RUTINER SOM,
20 000017 009 C   VED HJELP AV ERTRAN,TILORDNER OG FRIGJØR TROMMEL,
000018 009 C   DISC OG MAGNETBANDFILER.
22 000019 006 C
000020 006 C
24 000021 006 C   SUBROUTINE ASGUP(NUN,NAVN,TEMP)
000022 006 C
26 000023 006 C   OPPRETTER EN NY FIL
000024 006 C
28 000025 006 C   DIMENSION NA(4)
000026 006 C   LOGICAL TEMP
000027 006 10  FORMAT(8HGASG,UP ,A6,14H.,F40///400 . )
000028 006 11  FORMAT(7HGASG,T ,A6,13H.,F///1000 . )
000029 006 12  FORMAT(7HGASG,T ,I2,13H.,F///1000 . )
000030 006 20  FORMAT(8HGASG,AX ,A6,4H. . )
000031 006 30  FORMAT(5HGUSE ,I2,2H.,,A6,4H. . )
000032 006 40  FORMAT(6HGFREE ,A6,4H. . )
000033 006 41  FORMAT(6HGFREE ,I2,4H. . )
000034 006 50  FORMAT(8HGFREE,A ,I2,4H. . )
000035 006 60  FORMAT(8HGFREE,B ,I2,4H. . )
000036 006 61  FORMAT(8HGFREE,S ,I2,4H. . )
000037 006 70  FORMAT(4X,'LLL',6A6,'LLL')
000038 006 75  FORMAT(4X,'FEIL I ASGAX ',0I2)
000039 006 76  FORMAT(4X,'FAC.STAT.=',0I2)
000040 006   IF (.NOT.TEMP) ENCODE(10,NA)NAVN
000041 009   IF (TEMP.AND.(NAVN.EQ.NUN)) ENCODE(12,NA) NUN   &FILNAVN=I/O-NR.
000042 006   IF (TEMP.AND.(NAVN.NE.NUN)) ENCODE(11,NA) NAVN

000043 006   WRITE(6,70)NA
000044 006   I=NERTRN(6,NA)
000045 006   IF(I.EQ.0) GO TO 250   & FILEN OPPRETTET?
000046 006   WRITE(6,76) I
000047 006   IF (I.GT.0) GO TO 250
000048 006   IF (NUN.EQ.NAVN) CALL ERTRAN(2)
000049 006 C
000050 006   ENTRY ASGAX(NUN,NAVN)
000051 006 C
000052 006 C   TILORDNER EN FIL
000053 006 C
000054 006   ENCODE(20,NA)NAVN
000055 006   WRITE(6,70)NA
000056 006   I=NERTRN(6,NA)
000057 006   IF(I.EQ.0) GO TO 250
000058 006   WRITE(6,75)I
000059 006   IF (I.GT.0) GO TO 250   & FILEN TILORDNET?
000060 006   CALL ERTRAN(2)   & NEI! (FEILAVBRUDD)
000061 006 C
000062 006   ENTRY USE(NUN,NAVN)
000063 006 C
000064 006 C   EN FIL TILORDNES ET I/O-ENHETSNR.
000065 006 C
000066 006 250 CONTINUE
000067 006   IF (NUN.EQ.NAVN) RETURN
000068 006   ENCODE(30,NA)NUN,NAVN
000069 006   WRITE(6,70)NA
000070 006   CALL ERTRAN(6,NA)
000071 006   RETURN
000072 006 C
000073 006   ENTRY FREEA(NUN,TEMP)
000074 006 C
000075 009 C   FRIGJØR I/O-ENHETSNUMMERET FRA TEMPORÆRE FILER.
000076 009 C   FRIGJØR KATALOGISERTE FILER.
000077 006 C
000078 006   IF (.NOT.TEMP) GO TO 300
000079 006   ENCODE(50,NA) NUN
000080 006   WRITE(6,70)NA
000081 010   I=NERTRN(6,NA)
000082 010   IF (I.NE.0) WRITE(6,76) I
000083 006   RETURN
000084 006 C
000085 006   ENTRY FREE(NUN,NAVN)
000086 006 C

```

```
000087 006 C FRIGJØR EN FIL
000088 006 C
000089 006 IF ((NUN.GT.0).AND.(NUN.LT.31)) ENCODE(41,NA) NUN
000090 006 IF ((NUN.LT.1).OR.(NUN.GT.30)) ENCODE(40,NA)NAVN
000091 006 WRITE(6,70)NA
000092 006 I=NERTRN(6,NA)
000093 006 IF (I.NE.0) WRITE(6,76) I
000094 006 RETURN
000095 006 C
000096 006 ENTRY FREEB(NUN)
000097 006 C
000098 006 C FRIGJØR ET I/O-ENHETSNR. FRA EN FIL,OG FRIGJØR
000099 006 C FILEN HVIS DENNE IKKE ER TILORDNET FLERE I/O-NR.
000100 006 C
000101 006 300 CONTINUE
000102 006 ENCODE(60,NA) NUN
000103 006 WRITE(6,70) NA
000104 006 CALL ERTRAN(6,NA)
000105 006 RETURN
000106 006 C
000107 006 ENTRY FREES(NUN)
000108 006 C
000109 008 C FRIGJØR ET MAGNETBAND,MEN BEHOLDER MAGNETBANDSTASJONEN
000110 006 C
000111 006 ENCODE (61,NA) NUN
000112 007 WRITE(6,70) NA
000113 006 I=NERTRN(6,NA)
000114 006 IF (I.NE.0) WRITE(6,76) I
000115 006 RETURN
000116 006 END
```

MSQDATA

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Hovedprogram
Dato : 11/12-75
Utført av : Ø. Strand
Formål : MSQDATA sorterer ut alle hydrografiske måledata tilhørende ønskede MSQ-ruter fra en gitt fil, og til nye måledatafiler.

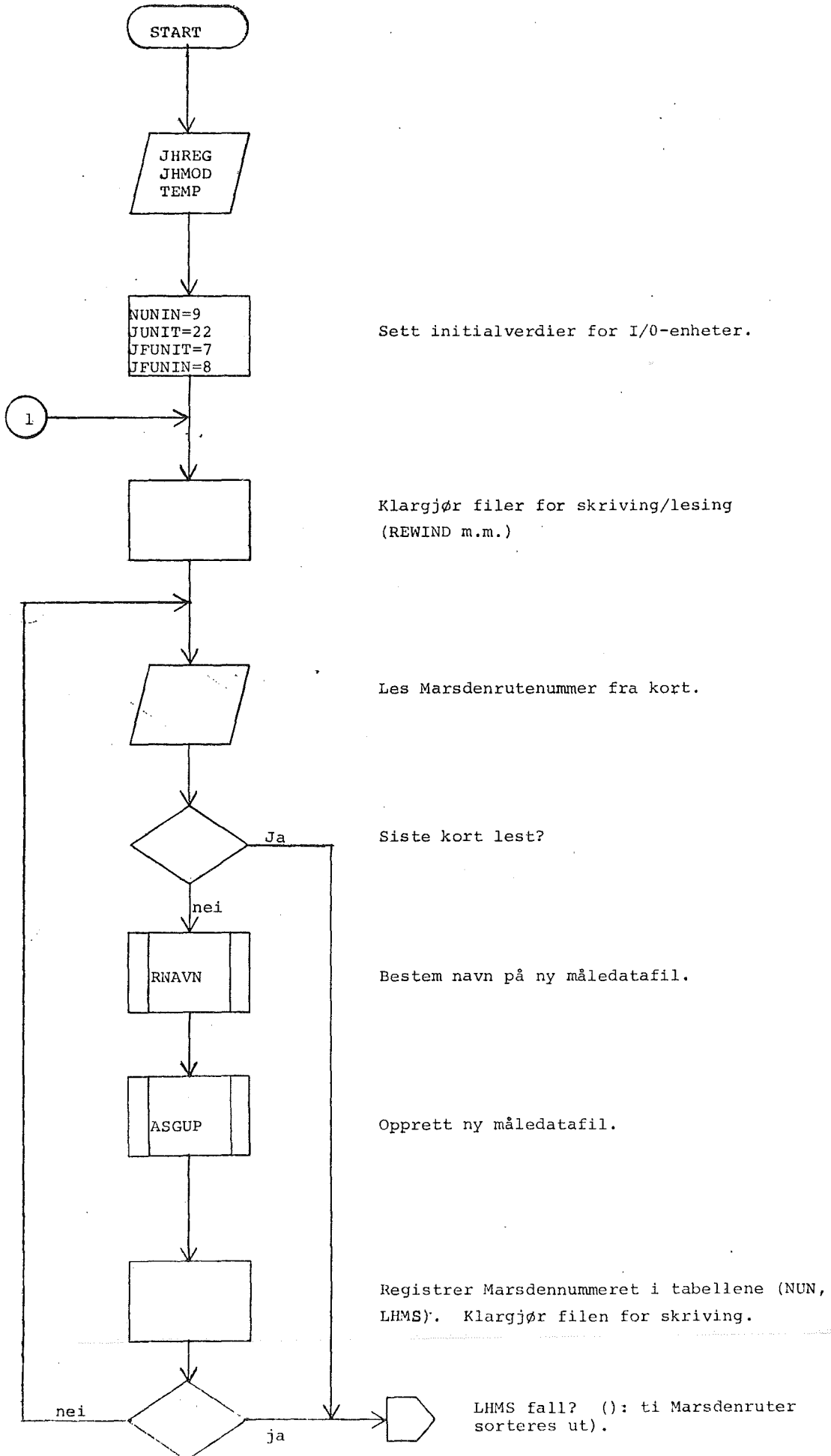
Generelt:

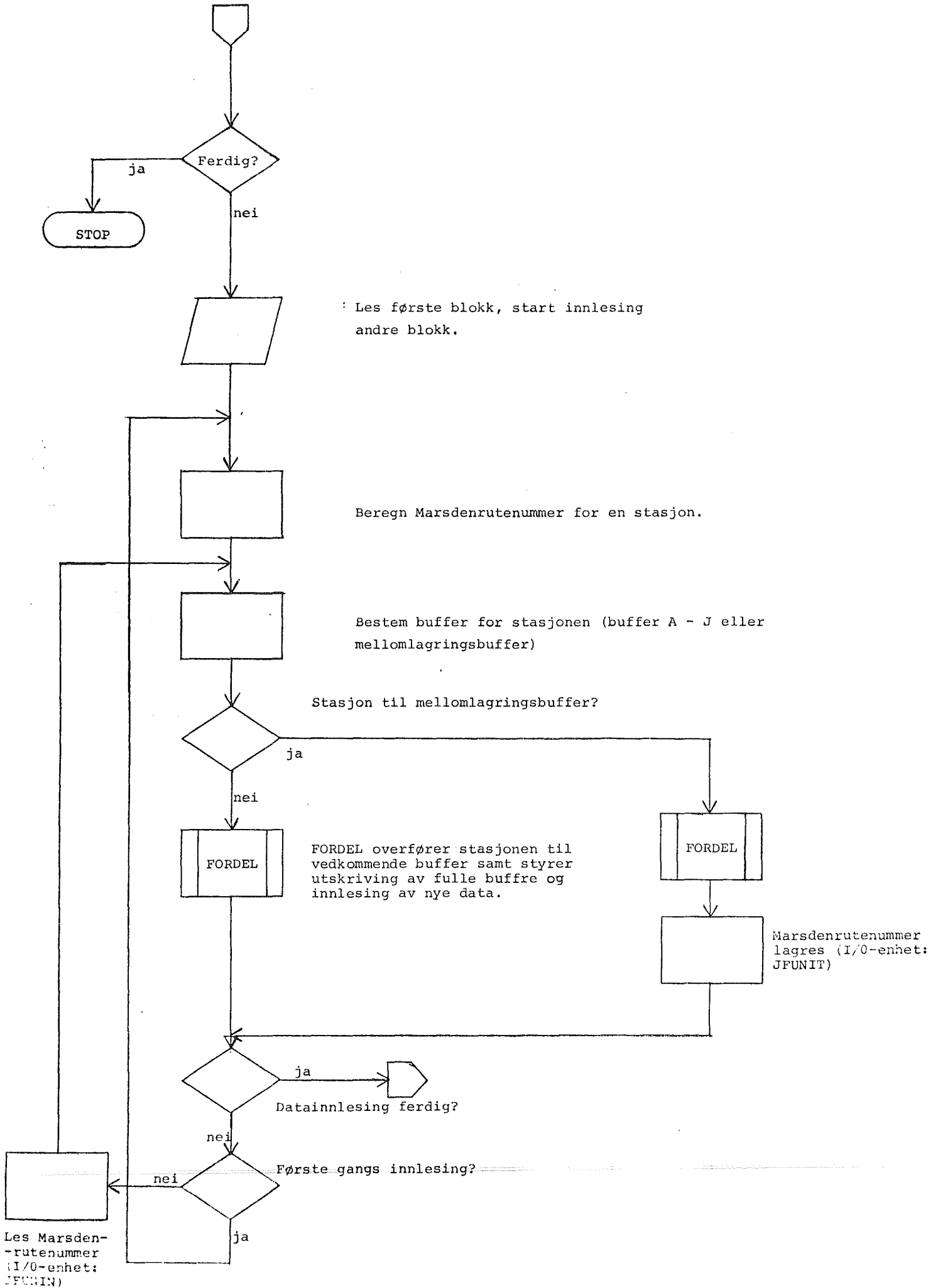
MSQDATA brukes ved utsortering av måledata tilhørende gitte MSQ-ruter fra en originalfil.

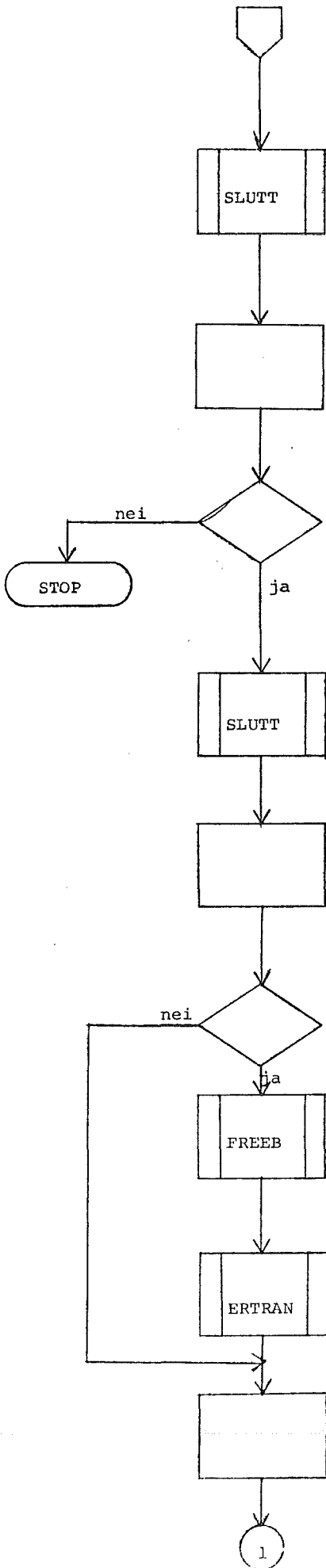
De ønskede MSQ-rutenumre (maksimum ti) leses inn fra kort, og nye filer opprettes. Under lesing av originalfilen (dobbelbuffering) beregnes MSQ-nummer for hver stasjon, og ved hjelp av tabeller plasseres stasjonen i rett buffer. Hvis en stasjon viser seg å tilhøre en MSQ-rute som ikke er under utsortering, lagres denne på en egen mellomagringsfil (I/O-enhet 22), mens MSQ-nummeret lagres på en annen fil. Når originalfilen er ferdig lest, avsluttes de nye måledatafilene, og originalfilen frigjøres. Deretter opprettes en ny, temporær, fil med samme I/O-enhet som originalfilen (9), og nye MSQ-numre kan leses inn. Ved annengangs utsortering, leses måledata og MSQ-numre for hver stasjon fra første innlesings mellomagringsfil (derved unngås ny beregning av MSQ-numre). Den nye, temporære, filen brukes nå som mellomagringsfil for måledata, mens de tilhørende MSQ-numre lagres på en egen fil. Ved senere utsortering, veksler de to filene (9 og 22) om å være henholdsvis innlesefil og mellomagringsfil. (Grunnen til at de samme I/O-numre brukes om igjen, er at NTRAN ikke tillater bruk av mer enn tolv forskjellige I/O-numre i hvert program.)

Variable:

NAVN:	TYPE:	BRUK:
NA(14, 100)	INTEGER	Buffer A.
NB(14, 100)	INTEGER	Buffer B.
NC(14, 100)	INTEGER	Buffer C.
ND(14, 100)	INTEGER	Buffer D.
NE(14, 100)	INTEGER	Buffer E.
NF(14, 100)	INTEGER	Buffer F.
NG(14, 100)	INTEGER	Buffer G.
NH(14, 100)	INTEGER	Buffer H.
NI(14, 100)	INTEGER	Buffer I.







Skriver ut siste blokk fra buffer A- J, nullstiller parametre.

Nye måledatafiler frigjøres.

Data i mellomlagringsfilen?

Skriver ut siste blokk til mellomlagringsfil, nullstiller parametre.

Bufferpekertabellen nullstilles.

Første gangs innlesing av innlesefil?

Frigjør originalfil.

Opprett temporær mellomlagringsfil.

Bytt om innlese og mellomlagringsfiler (NUNIN → JUNIT, JFUNIT → JFUNIN)

NAVN:	TYPE:	BRUK:
NJ(14, 100)	INTEGER	Buffer J.
NUT(14, 100)	INTEGER	Buffer for mellomlagringsfil.
NERHJ(3)	INTEGER	Brukes ved oppretting av ny, temporær, mellomlagringsfil (I/O-enhet 9).
LHMS(10)	INTEGER	Lagrer de (inntil) ti MSQ-numre som er under utsortering.
TEMP	LOGICAL	Angir type (temporær/katalogisert) for utskriftsfiler.
NUN(936)	INTEGER	Indikator som viser måledatabuffer for hver MSQ-rute (0): mellomlagringsbuffer, 1 - 10): Buffer A - J).
KA	INTEGER	Blokknummer, måledatafil A.
KB	INTEGER	Blokknummer, måledatafil B.
KC	INTEGER	Blokknummer, måledatafil C.
KD	INTEGER	Blokknummer, måledatafil D.
KE	INTEGER	Blokknummer, måledatafil E.
KF	INTEGER	Blokknummer, måledatafil F.
KG	INTEGER	Blokknummer, måledatafil G.
KH	INTEGER	Blokknummer, måledatafil H.
KI	INTEGER	Blokknummer, måledatafil I.
KJ	INTEGER	Blokknummer, måledatafil J.
JA	INTEGER	Linjenummer, buffer A.
JB	INTEGER	Linjenummer, buffer B.
JC	INTEGER	Linjenummer, buffer C.
JD	INTEGER	Linjenummer, buffer D.
JE	INTEGER	Linjenummer, buffer E.
JF	INTEGER	Linjenummer, buffer F.
JG	INTEGER	Linjenummer, buffer G.
JH	INTEGER	Linjenummer, buffer H.
JI	INTEGER	Linjenummer, buffer I.
JJ	INTEGER	Linjenummer, buffer J.

NAVN:	TYPE:	BRUK:
KUT	INTEGER	Blokknummer i mellomlagringsfil.
LUT	INTEGER	Linjenummer i mellomlagringsbuffer.
JHREG	INTEGER	Registertypenummer for nye måledatafiler.
JHMOD	INTEGER	Typenummer for valg av navn på nye måledatafiler.
JUNIT	INTEGER	I/O-enhet for mellomlagringsfil (måledata).
JFUNIT	INTEGER	I/O-enhet, lagring av MSQ-numre for mellomlagringsfil.
JFUNIN	INTEGER	I/O-enhet, innlesning av MSQ-numre for mellomlagrede stasjoner.
LABEL	INTEGER	Linjenummervelger etter overføring av en stasjon (): Beregning av MSQ-nummer for neste stasjon (originalfil)/innlesing av beregnet MSQ-nummer for neste stasjon.
LTELL	INTEGER	Teller antall innlesefiler som er brukt (): originalfil pluss innleste mellomlagringsfiler).
LUN	INTEGER	Teller for I/O-numre for nye datafiler (): viser høyeste I/O-nummer som er brukt ved den pågående innlesing av data).
IHMS	INTEGER	Teller i LHMS (tabell over MSQ-numre som er under utsortering).
MSQ	INTEGER	MSQ-nummer.
NAVN	INTEGER	Inneholder navn på ny måledatafil.
NT	INTEGER	Hjelpevariabel ved kodebestemmelse (vanlig kode/fjorddatakode).
NQ	INTEGER	Kvadrantnummer.
LAT	INTEGER	Viser (etter endel beregninger) avstand fra ekvator (i hele tigrader): 0-8).
LATH	INTEGER	Hjelpevariabel ved beregning av LAT.
LONG	INTEGER	Viser (etter endel beregninger) avstanden (i hele tigrader) fra nullmeridianen (): 0-17 (øst) 0-18 (vest)).
LONGH	INTEGER	Hjelpevariabel ved beregning av LONG.
L	INTEGER	Hjelpevariabel.
I	INTEGER	Hjelpevariabel.

COMMON variable:

DATA:

NAVN:	TYPE:	BRUK:
INN(14,200)	INTEGER	Innlesebuffer.
J	INTEGER	Linjenummer i innlesebuffer.
K	INTEGER	Blokknummer, innlesefil.
NUNIN	INTEGER	I/O-enhet, innlesefil.
LSTAT	INTEGER	Statusvariabel (NTRAN), innlesefil.

Eksterne rutiner

: RNAVN
ASGUP (FREEB, FREEA)
NTFEIL
FORDEL(SLUTT)

Interne rutiner

: HJELP (Brukes ved beregning av avstand, i hele tigrader, fra ekvator eller nullmeridianen, samt korrigerer kvadrantnummeret hvis dette er feil (ved grensene).

Faste I/O-numre

: 9,22 (innlesing/mellomlagring av måledata).
7,8 (innlesing/lagring (eller omvendt) av beregnede MSQ-numre).
10-19 (Nye måledatafiler).

```

000001      000  C
000002      000  C
000003      000  C
000004      000  C
000005      000  C
000006      000  C
000007      000  C
000008      000  C
000009      000  C
000010      000  C
000011      000  C
000012      000  C
000013      004  C
000014      003  C
000015      003  C
000016      000  C
000017      000  C
000018      000  C
000019      000  C
000020      000  C
000021      000  C
000022      000  C
000023      000  C
000024      000  C
000025      000  C
000026      000  C
000027      000  C
000028      000  C
000029      000  C
000030      000  C
000031      000  C
000032      000  C
000033      000  C
000034      002  C
000035      000  C
000036      000  C
000037      000  C
000038      000  C
000039      000  C
000040      000  C
000041      000  C
000042      002  C

          *****
          *
          *      MSQDATA      *
          *
          *      *****

MSQDATA SORTERER UT ALLE DATA TILHØRENDE ØNSKEDE MSQ-RUTER
FRA EN ORIGINALFIL TIL NYE DATAFILER. PROGRAMMET SORTERER UT MAKSIMUM
TI RUTER I HVERT GJENNOMLØP, MEN DE IKKE UTSORTERTE DATA
MELLOMLAGRES PÅ EN EGEN FIL, DET SAMME GJELDER OGSÅ DET
BEREGNEDE MSQ-RUTENUMMERET.
DISSE MELLOMLAGRINGSFILENE BRUKES VED NESTE GANGS INNLESING.

          NØDVENDIGE RUTINER : RNAVN
                                ASGUP
                                NTFEIL
                                FORDEL

          RESERVERTE I/O-NUMRE : 9,22 (INNLESEFIL/MELLOMLAGRINGSFIL)
                                7,8 (LESING/MELLOMLAGRING BEREGNEDE VERDIER)
                                10-19 (NYE MALEDATAFILER)

DIMENSION NA(14,100),NB(14,100),NC(14,100),ND(14,100),NE(14,100)
DIMENSION NF(14,100),NG(14,100),NH(14,100),NI(14,100),NJ(14,100)
DIMENSION NUT(14,100),NERHJ(3)
COMMON /DATA/INN(14,200),J,K,NUNIN,LSTAT
DIMENSION NUN(934),LHMS(10)
LOGICAL TEMP
DATA JA,JB,JC,JD,JE,JF,JG,JH,JI,JJ/10*1/,KA,KB,KC,KD,KE/5*0/
DATA KF,KG,KH,KI,KJ/5*0/
DATA NERHJ/14HGASG,T 9.,F . /
          10  FORMAT(2I3)
          20  FORMAT(1H0,4X,15,7HBLOKKER)
          21  FORMAT(4X,14,' BLOKKER MED ',I3,' LINJER I SISTE BLOKK',
          * ' GJENSTAR')
          30  FORMAT(L1)

000043      000  35  FORMAT(5X,11)
000044      000  36  FORMAT(2X,11)
000045      000  37  FORMAT(12)
000046      000  38  FORMAT(1X,11)
000047      000  39  FORMAT(12,13)
000048      000  41  FORMAT(2X,11,13)
000049      000  50  FORMAT(4X,'REGISTER NR:',I3,' NAVNMODUS NR:',I3)
000050      000  C
000051      000  C
000052      000  C
000053      000  C
000054      000  C
000055      002  C
000056      000  C
000057      003  C
000058      000  C
000059      000  C
000060      000  C
000061      000  C
000062      000  C
000063      000  C
000064      000  C
000065      000  C
000066      000  C
000067      000  C
000068      000  C
000069      000  C
000070      000  C
000071      000  C
000072      000  C
000073      000  C
000074      000  C
000075      000  C
000076      000  C
000077      000  C
000078      000  C
000079      000  C
000080      002  C
000081      000  C
000082      000  C
000083      000  C
000084      000  C
000085      000  C
000086      000  C

          BESTEM STARTPARAMETRE.

          READ(5,10) JHREG,JHMOD
          WRITE(6,50) JHREG,JHMOD
          READ(5,30) TEMP
          NUNIN=9
          ASSIGN 111 TO LABEL      RETURHOPP ETTER UTSORTERING AV EN STASJON
          JFUNIT=7
          JFUNIN=8
          JUNIT=22
          LTELL=0
          KUT=0
          LUT=1
          80  CONTINUE
          C
          C
          FORBERED INNLESING AV NYE MSQ-NUMRE.
          C
          LUN=9
          CALL NTRAN(NUNIN,10,22)
          IHMS=0
          REWIND JFUNIT
          CALL NTRAN(JUNIT,10,22)
          90  CONTINUE
          C
          C
          LES INN NYE MSQ-NUMRE, OG OPPRETT NYE FILER.
          C
          READ(5,10,END=95) MSQ
          LUN=LUN+1
          CALL RNAVN(MSG,JHREG,NAVN,JHMOD)
          CALL ASGUP(LUN,NAVN,TEMP)
          NUN(MSG)=LUN-9
          CALL NTRAN(LUN,10,22)
          IHMS=IHMS+1
          LHMS(IHMS)=MSG
          IF (IHMS.EQ.10) GO TO 100
          GO TO 90

```

```
000087 000 95 CONTINUE
000088 000 IF (IHMS.EQ.0) GO TO 1000 6FERDIG?
000089 000 100 CONTINUE
000090 000 C
000091 000 C LES INN FØRSTE BLOKK, START INNLESING AV ANDRE BLOKK.
000092 000 C
000093 001 CALL NTRAN(NUNIN,2,1400,INN(1,1),LSTAT,22)
000094 000 K=1
000095 000 CALL NTFEIL(1400,LSTAT,NUNIN,K,1)
000096 000 J=1
000097 001 CALL NTRAN(NUNIN,2,1400,INN(1,101),LSTAT)
000098 000 GO TO LABEL
000099 000 111 CONTINUE
000100 000 C
000101 004 C FØRSTE GANGS INNLESING AV MALEDATAFIL;
000102 000 C NY STASJON, BEREGN MSQ-NUMMER OG KVADRANTNUMMER.
000103 004 C MSQ-NUMMER BEREGNES VED HJELP AV BREDD OG
000104 004 C LENGDE I HELE TIGRADER.
000105 000 C
000106 000 NT=FLD(0,6,INN(14,J))
000107 003 IF (NT.EQ.57) GO TO 115 6FJORDDATAKODE?
000108 003 C
000109 003 C VANLIG KODE.
000110 003 C
000111 000 DECODE(35,INN(3,J)) NQ 6KVADRANTNUMMER.
000112 000 GO TO 120
000113 000 115 CONTINUE
000114 000 NQ=0
000115 000 120 CONTINUE
000116 000 IF (NQ.GT.1) GO TO 130
000117 000 DECODE(36,INN(2,J)) LAT
000118 000 121 CONTINUE
000119 000 IF (NQ.GT.0) GO TO 125
000120 000 C
000121 000 C KVADRANT 0.
000122 000 C
000123 000 IF (NT.NE.57) DECODE(37,INN(3,J)) LONG
000124 000 IF (NT.EQ.57) DECODE(38,INN(3,J)) LONG
000125 000 123 CONTINUE
000126 000 MSQ=36*(LAT+1)-LONG
000127 000 IF (MSQ.LT.289) GO TO 250
000128 000 MSQ=936-LONG
000129 000 GO TO 250
000130 000 125 CONTINUE

000131 000 C
000132 000 C KVADRANT 1.
000133 000 C
000134 000 DECODE(39,INN(3,J)) LONG, LONGH
000135 000 CALL HJELP(LONG, LONGH, 1123, 1)
000136 000 MSQ=36*LAT+LONG+1
000137 000 IF (LAT.LT.8) GO TO 250
000138 000 MSQ=901+LONG
000139 000 GO TO 250
000140 000 130 CONTINUE
000141 000 DECODE(41,INN(2,J)) LAT, LATH
000142 000 CALL HJELP(LAT, LATH, 1121, 2)
000143 000 IF (NQ.GT.2) GO TO 140
000144 000 C
000145 000 C KVADRANT 2.
000146 000 C
000147 000 DECODE(37,INN(3,J)) LONG
000148 000 135 CONTINUE
000149 000 MSQ=335+36*LAT-LONG
000150 000 GO TO 250
000151 000 140 CONTINUE
000152 000 C
000153 000 C KVADRANT 3.
000154 000 C
000155 000 DECODE(39,INN(3,J)) LONG, LONGH
000156 000 CALL HJELP(LONG, LONGH, 1135, 1)
000157 000 MSQ=300+36*LAT+LONG
000158 000 GO TO 250
000159 000 200 CONTINUE
000160 000 C
000161 000 C NY STASJON, LES INN BEREGNET MSQ-NUMMER.
000162 000 C
000163 000 READ(JFUNIN,END=400) MSQ
000164 000 250 CONTINUE
000165 000 C
000166 000 C BESTEM BUFFER FOR STASJONEN.
000167 000 C
000168 000 L=NUN(MSQ)
000169 000 IF (L.NE.0) GO TO (300,305,310,315,320,325,330,335,340,345),L
000170 000 C
000171 000 C MELLOPLAGRING.
000172 000 C
000173 000 WRITE(JFUNIT) MSQ
000174 000 CALL FORDEL(NT,LUT,KUT,JUNIT,1400)
```

```
000175      000      GO TO LABEL
000176      000      C
000177      000      C      BUFFER A-J.
000178      000      C
000179      000      300     CONTINUE
000180      000      CALL FORDEL(NA,JA,KA,10,1400)
000181      000      GO TO LABEL
000182      000      305     CONTINUE
000183      000      CALL FORDEL(NB,JB,KB,11,1400)
000184      000      GO TO LABEL
000185      000      310     CONTINUE
000186      000      CALL FORDEL(NC,JC,KC,12,1400)
000187      000      GO TO LABEL
000188      000      315     CONTINUE
000189      000      CALL FORDEL(ND,JD,KD,13,1400)
000190      000      GO TO LABEL
000191      000      320     CONTINUE
000192      000      CALL FORDEL(NE,JE,KE,14,1400)
000193      000      GO TO LABEL
000194      000      325     CONTINUE
000195      000      CALL FORDEL(NF,JF,KF,15,1400)
000196      000      GO TO LABEL
000197      000      330     CONTINUE
000198      000      CALL FORDEL(NG,JG,KG,16,1400)
000199      000      GO TO LABEL
000200      000      335     CONTINUE
000201      000      CALL FORDEL(NH,JH,KH,17,1400)
000202      000      GO TO LABEL
000203      000      340     CONTINUE
000204      000      CALL FORDEL(NI,JI,KI,18,1400)
000205      000      GO TO LABEL
000206      000      345     CONTINUE
000207      000      CALL FORDEL(NJ,JJ,KJ,19,1400)
000208      000      GO TO LABEL
000209      000      400     CONTINUE
000210      003      C
000211      003      C      INNLESEFIL FERDIGBEHANDLET.
000212      003      C
000213      000      LTELL=LTELL+1
000214      000      WRITE(6,20)K
000215      000      C
000216      000      C      AVSLUTT BUFFER A-J.
000217      000      C
000218      000      CALL SLUTT(NA,JA,KA,10,LUN,1450)

000219      000      CALL SLUTT(NB,JB,KB,11,LUN,1450)
000220      000      CALL SLUTT(NC,JC,KC,12,LUN,1450)
000221      000      CALL SLUTT(ND,JD,KD,13,LUN,1450)
000222      000      CALL SLUTT(NE,JE,KE,14,LUN,1450)
000223      000      CALL SLUTT(NF,JF,KF,15,LUN,1450)
000224      000      CALL SLUTT(NG,JG,KG,16,LUN,1450)
000225      000      CALL SLUTT(NH,JH,KH,17,LUN,1450)
000226      000      CALL SLUTT(NI,JI,KI,18,LUN,1450)
000227      000      CALL SLUTT(NJ,JJ,KJ,19,LUN,1450)
000228      000      450     CONTINUE
000229      000      C
000230      000      C      FRIGJØR DE NYE MALEDATAFILER.
000231      000      C
000232      000      CALL NTRAN(NUNIN,22)
000233      000      DO 500 I=10,LUN
000234      002      CALL FREEA(I,TEMP)
000235      000      500     CONTINUE
000236      000      I=KUT+1
000237      000      L=LUT-1
000238      000      WRITE(6,21) I,L
000239      000      IF ((KUT.EQ.0).AND.(LUT.EQ.1)) GO TO 1000      6FERDIG7
000240      000      CALL SLUTT(NUT,LUT,KUT,JUNIT,29,11)
000241      000      C
000242      000      C      NULLSTILL BUFFERPEKERTABELLEN.
000243      000      C
000244      000      DO 540 I=1,IHMS
000245      000      L=LHMS(I)
000246      000      NUN(L)=0
000247      000      540     CONTINUE
000248      000      ENDFILE JFUNIT
000249      000      IF (LTELL.NE.1) GO TO 560
000250      000      C
000251      000      C      FRIGJØR ORIGINALFIL,OPPRETT NY,TEMPORÆR,MELLOMLAGRINGSFIL (1/0-9).
000252      000      C
000253      000      CALL FREEB(NUNIN)
000254      000      CALL ERTRAN(6,NERHJ)
000255      000      560     CONTINUE
000256      000      C
000257      000      C      BYTT OM 1/0-NUMRE MELLOM INNLESE OG MELLOMLAGRINGSFILER.
000258      000      C
000259      000      L=NUNIN
000260      000      NUNIN=JUNIT
000261      000      JUNIT=L
000262      000      REWIND JFUNIT
```

```
2 000263 000 L=JFUNIT
000264 000 JFUNIT=JFUNIN
4 000265 000 JFUNIN=L
000266 000 C
6 000267 000 C SETT LINJENUMMERPEKER SLIK AT BEREGNEDE DATA LESES INN I NESTE
000268 000 C GJENNOMLØP.
8 000269 000 C
000270 000 ASSIGN 200 TO LABEL
000271 000 GO TO 80
10 000272 000 I CALL ERTRAN(2) 6FEIL!
12 000273 000 1000 CONTINUE
000274 000 STOP MSQDA
14 000275 000 C
000276 000 SUBROUTINE HJELP(LHJA,LHJB,I,KORR)
16 000277 000 C
000278 000 C HJELP BEREGNER AVSTAND FRA EKVATOR/NULLMERIDIAN I HELE TIGRADER,
18 000279 000 C SAMT KONTROLLERER KVADRANTNUMMER VED GRENSENE,
20 000280 004 C (STASJONER SOM LIGGER PÅ GRENSEN MELLOM TO TIGRADERSRUTER
22 000281 004 C TILHØRER PR. DEFINISJON DEN NORDLIGSTE/ØSTLIGSTE AV DISSE)
24 000282 000 C
26 000283 000 IF (LHJB.NE.0) RETURN
28 000284 000 IF (LHJA.NE.0) GO TO 100
30 000285 003 C
32 000286 004 C FEIL KVADRANTNUMMER I FORHOLD TIL OPPGITT POSISJON
34 000287 003 C
36 000288 000 NQ=NQ-KORR
38 000289 000 RETURN 3
40 000290 000 100 CONTINUE
42 000291 000 LHJA=LHJA-I
44 000292 000 RETURN
46 000293 000 END
```

FORDEL

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Entrepunkt : SLUTT
Dato : 11/12-75
Utført av : Ø. Strand
Formål : FORDEL overfører en hydrografisk stasjon (måledata) fra et innlesebuffer til et utskriftsbuffer, samt styrer innlesing og utskrivning av data.

Kall: FORDEL(NUT, NU, KN, NUNIT, \$)
SLUTT(NUT, NU, KN, NUNIT, LUN, \$)

Generelt:

FORDEL overfører en hydrografisk stasjon (måledata, vanlig kode eller fjorddatakode) fra et innlesebuffer til et utskrivningsbuffer. Rutinen styrer innlesing av nye data ved hjelp av rutinen LESDB (dobbelbuffring), samt skriver ut fulle utskriftsbuffer.

SLUTT (brukes av MSQDATA-INNH og MSQDATA) fyller ut resten av et utskriftsbuffer med nuller, og overfører dette til datafilen. Til slutt "nullstilles" endel parametre (blokknummer, linjenummer), og rutinen returnerer til ønsket sted i det kallende program.

Formelle parametre:

NAVN:	TYPE:	BRUK:
NUT(14, 100)	INTEGER	Utskriftsbuffer.
NU	INTEGER	Linjenummer i utskriftsbuffer.
KN	INTEGER	Blokknummer i utskrivningsfil.
NUNIT	INTEGER	I/O-enhet for utskrivningsfil.
\$	Linjenummer	FORDEL: Retur hvis innlesning av data er ferdig. (): siste stasjon i innlesefilen).
\$	Linjenummer	SLUTT: Retur hvis I/O-enhet (utskrift) har spesifisert verdi (LUN).
LUN	INTEGER	Hvis I/O-enhet for utskrift er lik LUN, returneres via eget uthopp (\$).

COMMON variable:

DATA:

NAVN:	TYPE:	BRUK:
INN(14,200)	INTEGER	Innlesebuffer.
J	INTEGER	Linjenummer i innlesebuffer.
K	INTEGER	Blokknummer i innlesefil.
NUNIN	INTEGER	I/O-enhet for innlesefil.
LSTAT	INTEGER	Statusvariabel (NTRAN).

Eksterne rutiner	: NTFEIL LESDB
------------------	-------------------

Variable:

NAVN:	TYPE:	BRUK:
I	INTEGER	Hjelpevariabel.
L	INTEGER	Hjelpevariabel.

```

000001 000 C
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C
000008 000 C
000009 000 C
000010 000 C
000011 000 C
000012 000 C
000013 000 C
000014 000 C
000015 000 C
000016 000 C
000017 000 C
000018 000 C
000019 000 C
000020 000 C
000021 000 C
000022 000 C
000023 000 C
000024 000 C
000025 000 C
000026 000 C
000027 000 10
000028 000 90
000029 000
000030 000
000031 000 100
000032 000
000033 000
000034 000
000035 000
000036 000
000037 000
000038 000 110
000039 001
000040 000 120
000041 000 C
000042 000 C

```

```

*****
*                               *
*   FORDEL                       *
*   SLUTT                         *
*                               *
*****

FORDEL MED ENTREPUNKT ER EN HJELPERUTINE SOM OVERFØRER
EN STASJON (HYDROGRAFISKE MALEDATA) FRA
INNLESEBUFFERET TIL ET DATABUFFER (NUT) SAMT STYRER LESING
AV NYE DATA OG UTSKRIVING AV FULLE BUFFRE.

NØDVENDIGE RUTINER : NTFEIL
                     LESDB

SUBROUTINE FORDEL(NUT,NU,KN,NUNIT,I)
EN STASJON OVERFØRES TILDATABUFFERET.
DIMENSION NUT(14,100)
COMMON /DATA/INN(14,200),J,K,NUNIN,LSTAT
FORMAT(2X,'LUNIT',I3,' ):','I3,' BLOKKER MED ',I6,' RECORDS')
CONTINUE
DO 100 I=1,14
NUT(I,NU)=INN(I,J)
CONTINUE
NU=NU+1
IF(NU.LE.100) GO TO 110
NU=1
CALL NTRAN(NUNIT,I,1400,NUT,L,22)
KN=KN+1
CALL NTFEIL(1400,L,NUNIT,KN,11000)
CONTINUE
CALL LESDB(INN,14,NUNIN,K,J,LSTAT,11000)
CONTINUE
NYTT KORTBILDE.

```

```

000043 000 C
000044 000
000045 002
000046 002
000047 000 C
000048 000 C
000049 000 C
000050 000
000051 000
000052 000 C
000053 000 C
000054 000 C
000055 000 C
000056 000 C
000057 000 C
000058 000
000059 000
000060 000
000061 000 150
000062 002
000063 000
000064 000
000065 000
000066 000
000067 002
000068 002
000069 000
000070 000
000071 000 1000
000072 000 C

```

```

L=FLD(6,6,INN(14,J))
IF (L.EQ.51) GO TO 90      63 I KOLONNE 80?
IF (L.EQ.57) GO TO 90      69 I KOLONNE 80?

STASJONEN FERDIG.

IF (L.EQ.0) RETURN 5      6 FILEN FERDIGLEST.
RETURN

ENTRY SLUTT(NUT,NU,KN,NUNIT,LUN,I)

AVSLUTNING ETTER AT EN FIL ER FERDIG GJENNOMLEST.
PARAMETRE NULLSTILLES.

DO 150 I=NU,100
DO 150 L=1,14
NUT(L,I)=0
CONTINUE
CALL NTRAN(NUNIT,I,1400,NUT,L,22)      65 ISTE BLOKK SKRIVES UT
NU=100+KN+NU-1
KN=KN+1
CALL NTFEIL(1400,L,NUNIT,KN,11000)
WRITE(6,10)NUNIT,KN,NU
KN=0      66 BLOKKNUMMER
NU=1      67 LINJENUMMER
IF (NUNIT.EQ.LUN) RETURN 6
RETURN
CALL ERTRAN(2)
END

```

MSQDATA - INNH

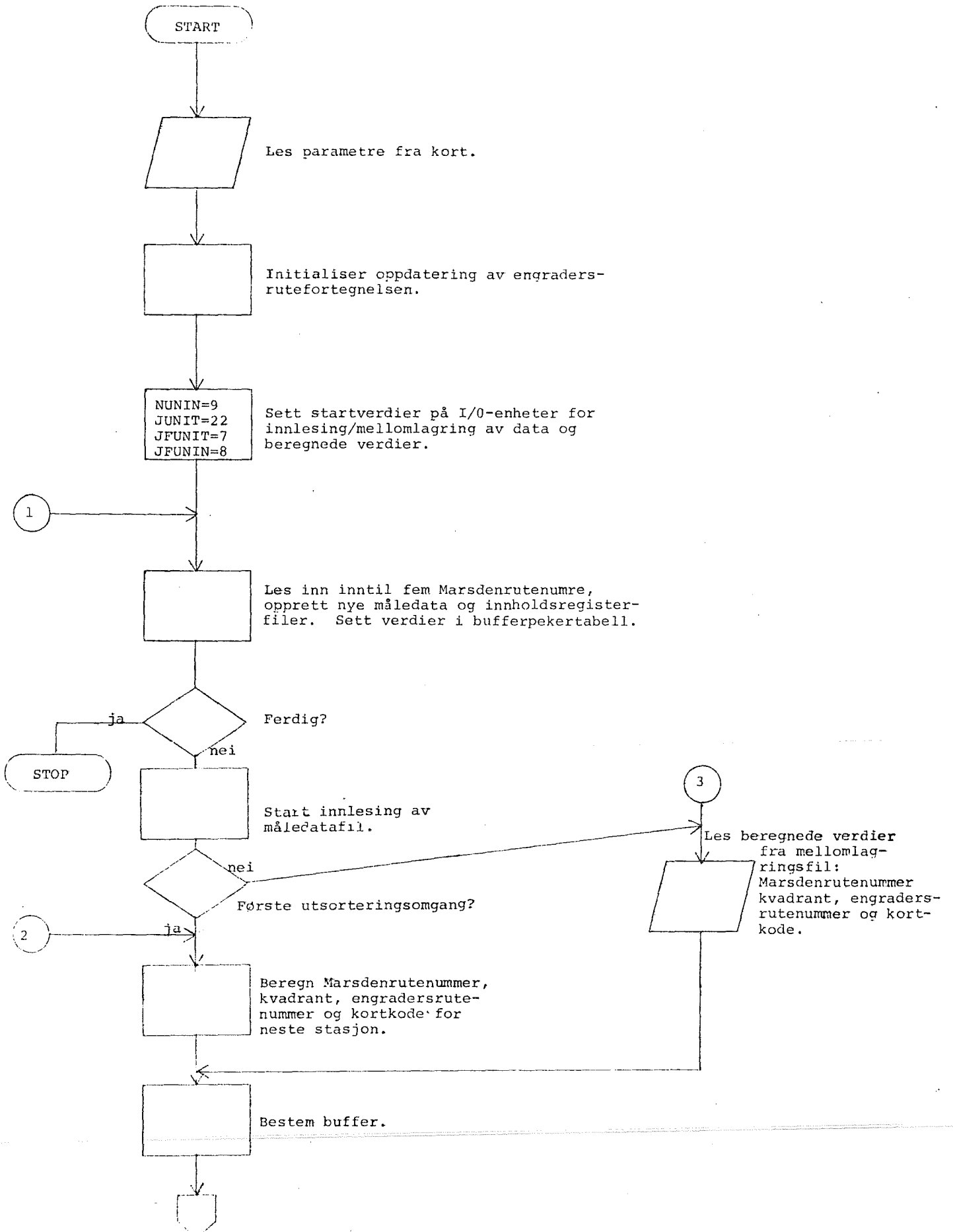
Programmeringsspråk : FORTRAN V (UNIVAC)
 Programtype : Hovedprogram
 Dato : 26/2-76
 Utført av : Ø. Strand
 Formål : MSQDATA-INNH sorterer ut alle hydrografiske måledata tilhørende ønskede MSQ-ruter fra en gitt fil, og til nye måledatafiler. Samtidig lages innholdsregisterfiler og engradersrutefortegnelser. Rekkefølgen i de nye måledatafilene kontrolleres.

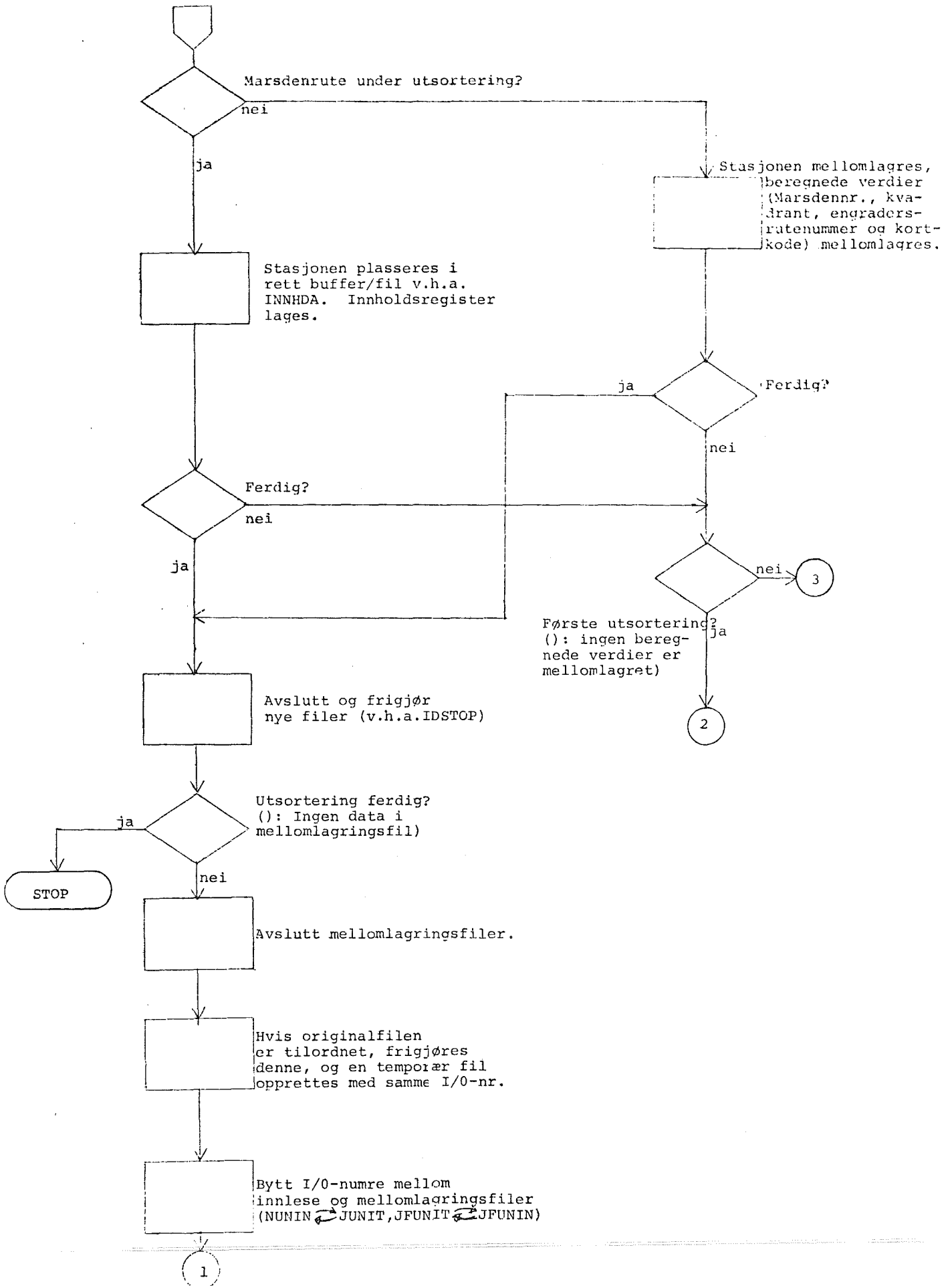
Generelt:

MSQDATA-INNH brukes ved utsortering av måledata fra en originalfil. Programmet sorterer ut alle data tilhørende gitte MSQ-ruter (maksimum fem) samt lager innholdsregisterfiler for de nye måledatafilene. Innlesefilen leses ved dobbelbuffering. For hver ny stasjon bestemmes MSQ-nummer, kvadrant, engradersrutenummer og kortkode (vanlig eller fjorddatakode) i hovedprogrammet før kontrollen overlates til rutinen INNHDA som besørger det videre arbeid med utsortering og kontroll av rekkefølgen etc. Data som ikke tilhører en av MSQ-rutene som er under utsortering, plasseres i en mellomagringsfil. Samtidig lagres de bestemte størrelsene for vedkommende stasjon i en annen fil. Når første gangs utsortering er avsluttet, frigjøres originalfilen og de nyopprettede filene. Deretter opprettes en ny, temporær, fil med samme I/O-nummer som originalfilen, og nye MSQ-rutenummer leses inn. Ved andre (og senere) gangs utsortering, brukes foregående utsorterings mellomagringsfil som innlesefil, samtidig som de nødvendige parametre, som alt er beregnet, leses inn fra lagringsfilen. Programmet besørger også registrering av antall stasjoner pr. engradersrute i de dertil brukte tabeller (REGFIL3). (Grunnen til at de samme I/O-numre brukes omigjen, er at NTRAN ikke tillater bruk av mer enn tolv forskjellige I/O-numre i hvert program.)

Variable:

NAVN:	TYPE:	BRUK:
NA(14, 100)	INTEGER	Måledatabuffer A.
NB(14, 100)	INTEGER	Måledatabuffer B.
NC(14, 100)	INTEGER	Måledatabuffer C.
ND(14, 100)	INTEGER	Måledatabuffer D.
NE(14, 100)	INTEGER	Måledatabuffer E.
NIA(10, 100)	INTEGER	Innholdsregisterbuffer A.
NIB(10, 100)	INTEGER	Innholdsregisterbuffer B.





NAVN:	TYPE:	BRUK:
NIC(10, 100)	INTEGER	Innholdsregisterbuffer C.
NID(10, 100)	INTEGER	Innholdsregisterbuffer D.
NIE(10, 100)	INTEGER	Innholdsregisterbuffer E.
NTA(100)	INTEGER	Engradersrutetabell A.
NTB(100)	INTEGER	Engradersrutetabell B.
NTC(100)	INTEGER	Engradersrutetabell C.
NTD(100)	INTEGER	Engradersrutetabell D.
NTE(100)	INTEGER	Engradersrutetabell E.
NUT(14, 100)	INTEGER	Buffer for mellomlagringsfil.
NERHJ(3)	INTEGER	Brukes ved oppretting av ny, temporær, mellomlagringsfil (I/O-enhet 9).
JA	INTEGER	Linjenummer, måledatabuffer A.
JB	INTEGER	Linjenummer, måledatabuffer B.
JC	INTEGER	Linjenummer, måledatabuffer C.
JD	INTEGER	Linjenummer, måledatabuffer D.
JE	INTEGER	Linjenummer, måledatabuffer E.
IA	INTEGER	Linjenummer, registerbuffer A.
IB	INTEGER	Linjenummer, registerbuffer B.
IC	INTEGER	Linjenummer, registerbuffer C.
ID	INTEGER	Linjenummer, registerbuffer D.
IE	INTEGER	Linjenummer, registerbuffer E.
KA	INTEGER	Blokknummer, måledatafil A.
KB	INTEGER	Blokknummer, måledatafil B.
KC	INTEGER	Blokknummer, måledatafil C.
KD	INTEGER	Blokknummer, måledatafil D.
KE	INTEGER	Blokknummer, måledatafil E.
LA	INTEGER	Blokknummer, innholdsregisterfil A.
LB	INTEGER	Blokknummer, innholdsregisterfil B.

NAVN:	TYPE:	BRUK:
LC	INTEGER	Blokknummer, innholdsregisterfil C.
LD	INTEGER	Blokknummer, innholdsregisterfil D.
LE	INTEGER	Blokknummer, innholdsregisterfil E.
IREG	INTEGER	Registertypenummer for nye innholdsregisterfiler.
JHREG	INTEGER	Registertypenummer for nye måledatafiler.
IMOD	INTEGER	Typenummer for valg av navn på nye innholdsregisterfiler.
JHMOD	INTEGER	Typenummer for valg av navn på nye måledatafiler.
JUNIT	INTEGER	I/O-enhet for mellomlagringsfil (måledata).
JFUNIT	INTEGER	I/O-enhet, lagring av beregnede parametre for stasjoner i mellomlagringsfilen (MSQ-nummer, kvadrant-nummer, engradersrute-nummer og kortkode).
JFUNIN	INTEGER	I/O-enhet, innlesning av beregnede parametre.
LABEL	INTEGER	Linjenummervelger (bestemmer fremgangsmåte ved ny stasjon (beregning/innlesning av parametre)).
LTELL	INTEGER	Teller antall innlesefiler som er brukt.
LUN	INTEGER	Teller for I/O-numre for nye måledatafiler (): viser høyeste I/O-nummer som er brukt ved den pågående innlesning av data).
KUT	INTEGER	Blokknummer, mellomlagringsfil.
LUT	INTEGER	Linjenummer, mellomlagringsbuffer.
IUN	INTEGER	Teller for I/O-numre for nye innholdsregisterfiler.
IHMS	INTEGER	Teller i LHMS (tabell over MSQ-numre under utsortering).
MSQ	INTEGER	MSQ-nummer.
NAVN	INTEGER	Navn på nye filer.
LAT	INTEGER	Avstand fra ekvator i hele tigrader.
LATH	INTEGER	Avstand (i hele grader) fra laveste breddegrad i en MSQ-rute (brukes ved beregning av engradersrutenummer).

NAVN:	TYPE:	BRUK:
LONG	INTEGER	Åvstand (i hele tigrader) fra nullmeridianen.
LONGH	INTEGER	Avstand (i hele grader) fra laveste lengdegrad i en MSQ-rute.
NIAH(10, 101)	INTEGER	Hjelpebuffer ved kontroll av rekkefølgen i innholdsregisterfil A. Linje 2-101 inneholder innholdsregisterbufferet for fil A (NIA), mens linje 1 inneholder siste post fra forrige buffer.
NIBH(10, 101)	INTEGER	Hjelpebuffer ved kontroll av rekkefølgen i innholdsregisterfil B.
NICH(10, 101)	INTEGER	Hjelpebuffer ved kontroll av rekkefølgen i innholdsregisterfil C.
NIDH(10, 101)	INTEGER	Hjelpebuffer ved kontroll av rekkefølgen i innholdsregisterfil D.
NIEH(10, 101)	INTEGER	Hjelpebuffer ved kontroll av rekkefølgen i innholdsregisterfil E.
LRA	LOGICAL	Angir korrekt/ukorrekt rekkefølge i fil A.
LRB	LOGICAL	Angir korrekt/ukorrekt rekkefølge i fil B.
LRC	LOGICAL	Angir korrekt/ukorrekt rekkefølge i fil C.
LRD	LOGICAL	Angir korrekt/ukorrekt rekkefølge i fil D.
LRE	LOGICAL	Angir korrekt/ukorrekt rekkefølge i fil E.
ITEMP	LOGICAL	Angir om de nye innholdsregisterfilene skal være temporære (TRUE) eller katalogisert.
HTEMP	LOGICAL	Angir om de nye måledatafilene skal være temporære (TRUE) eller katalogisert.
KOPG	LOGICAL	Angir hvorvidt eksisterende engradersrute- fortegnelse skal kopieres over til den nye utgaven før oppdateringen starter. Brukes av NY3.
IDAPO	INTEGER	Antall poster i tabellfilen (magnetbåndtabell, Marsdenrutetabell).
III	INTEGER	Hjelpevariabel.
LL	INTEGER	Hjelpevariabel.
LLL	INTEGER	Hjelpevariabel.
I	INTEGER	Hjelpevariabel.

NAVN:	TYPE:	BRUK:
L	INTEGER	Hjelpevariabel.
N	INTEGER	Hjelpevariabel.

COMMON variable:

blank:

NAVN:	TYPE:	BRUK:
NUN(936)	INTEGER	Indikator tabell som angir buffer for MSQ-rute (0): mellomlagringsrute, 1-5): måledatabuffer A-E).
LHMS(5)	INTEGER	Lagrer de (inntil) fem MSQ-numre som er under utsortering.
NQ	INTEGER	Kvadrantnummer.
NQH	INTEGER	Engradersrutenummer.
NT	INTEGER	Kodetype (vanlig kode/fjorddatakode).
BEGGE	LOGICAL	Brukes ved oppdatering av engradersrute- fortegnelsen (OPPD3). Angir hvorvidt (TRUE) de utsorterte engradersrute- tegnelsene skal lagres på en egen fil i tillegg til oppdateringen av den eksisterende fortegnelse.

DATA:

NAVN:	TYPE:	BRUK:
INN(14,200)	INTEGER	Innlesebuffer.
J	INTEGER	Linjenummer i innlesebuffer.
K	INTEGER	Blokknummer, innlesefil.
NUNIN	INTEGER	I/O-enhet, innlesefil.
LSTAT	INTEGER	Statusvariabel (NTRAN), for innlesefil.

RDATA:

NAVN:	TYPE:	BRUK:
MSQTAB(72)	INTEGER	Tabell over brukte Marsdenrutenumre med peker til linjenummer i Marsdenrutetabellene. Leses inn fra fil med magnetbåndtabell og Marsdenrutetabeller.

NAVN:	TYPE:	BRUK:
LBT	INTEGER	Magnetbåndtabellens lengde.
LMSQT	INTEGER	Marsdenrutetabellenes lengde.

Eksterne rutiner

: NY3
RNAVN
ASGUP(FREEB)
NTFEIL
FORDEL(SLUTT)
INNHDA(IDSTOP)

Interne rutiner

: HJELP (Brukes ved beregning av avstand (i hele tigrader) fra ekvator eller null-meridianen og avstand (i hele grader) fra laveste bredde/lengde i en MSQ-rute. Kvadrantnummeret kontrolleres i grensetilfellene).

Faste I/O-numre

: 9, 22 (innlesing/mellomlagring av måledata)
7, 8 (lagring/innlesing av beregnede parametre)
10-14 (nye måledatafiler)
17-21 (nye innholdsregisterfiler)
2, 3, 26, 28 (brukes av NY3)

```

4 000001 003 C
000002 003 C
6 000003 003 C
000004 003 C
8 000005 003 C
000006 003 C
10 000007 003 C
000008 003 C
12 000009 003 C
000010 003 C
14 000011 003 C
000012 003 C
16 000013 003 C
000014 003 C
18 000015 003 C
000016 003 C
2 000017 003 C
000018 003 C
22 000019 003 C
000020 003 C
24 000021 003 C
000022 003 C
26 000023 003 C
000024 003 C
28 000025 003 C
000026 003 C
30 000027 003 C
000028 003 C
000029 003 C
000030 003 C
000031 003 C
000032 003 C
000033 003 C
000034 003 C
000035 003 C
000036 003 C
000037 003 C
000038 003 C
000039 006 C
000040 003 C
000041 006 C
000042 006 C

```

```

*****
*
*      MSQDATA-INNH
*
*****

```

```

MSQDATA-INNH SORTERER UT ALLE DATA TILHØRENDE ØNSKEDE
MSQ-RUTER FRA EN DATAFIL, OG PRODUSERER INNHOLDSREGISTERFILER
FOR DE NYE DATAFILENE, PROGRAMMET BEHANDLER INNTIL FEM MSQ-RUTER
PA EN GANG, MEN LESING KAN GJENTAS FOR SA MANGE RUTER SOM
ØNSKET, DE IKKE UTSORTERTE DATA OVERFØRES TIL EN NY FIL HVOR
DE KAN LESES FRA VED NESTE GJENNOMLØP (DE VIKTIGSTE PARAMETRE
LAGRES, OG BEHØVER IKKE Å BEREGNES PÅ NY),

```

```

NØDVENDIGE RUTINER : NY3
                     RNAVN
                     ASGUP
                     NTFEIL
                     FORDEL
                     INNHDA

```

```

RESERVERTE I/O-NUMRE : 9,22 (INNLESE/MELLOMLAGRINGSFIL)
                      7,8 (LESING/MELLOMLAGRING BEREGNEDE VERDIER)
                      10-14 (NYE MALEDATAFILER)
                      17-21 (NYE INNHOLDSREGISTERFILER)
                      4 (LAGRING AV MARSDEN-NUMMER OG OPPLYSNINGER OM
                        KORREKT/IKKE KORREKT REKKEFØLGE)

```

```

DIMENSION NA(14,100),NB(14,100),NC(14,100),ND(14,100),NE(14,100)
DIMENSION NIA(10,100),NIB(10,100),NIC(10,100),NID(10,100),
*NIE(10,100)
DIMENSION NTA(100),NTB(100),NTC(100),NTD(100),NTE(100)
COMMON/RDATA/MSGTAB(72),LBT,LMSQT
DIMENSION NUT(14,100),NERHJ(3)
LOGICAL KOPG,BEGGE
COMMON NUN(936),LHMS(5),NQ,NQH,NT,BEGGE

```

```

000043 003 C
000044 003 C
000045 003 C
000046 003 C
000047 003 C
000048 003 C
000049 005 C
000050 003 C
000051 003 C
000052 003 C
000053 003 C
000054 003 10
000055 003 20
000056 003 21
000057 003 C
000058 005 22
000059 003 35
000060 003 36
000061 003 37
000062 003 38
000063 003 39
000064 003 41
000065 006 42
000066 003 50
000067 003 C
000068 003 C
000069 003 C
000070 006 C
30 000071 006 C
000072 006 C
35 000073 003 C
000074 003 C
35 000075 003 C
000076 003 C
36 000077 005 C
000078 003 C
37 000079 006 C
000080 006 C
38 000081 006 C
000082 006 C
39 000083 006 C
000084 006 C
39 000085 006 C
000086 006 75

```

```

COMMON /DATA/INN(14,200),J,K,NUNIN,LSTAT
DIMENSION NIAH(10,101),NIBH(10,101),NICH(10,101),NIDH(10,101),
*NIEH(10,101)
EQUIVALENCE (NIA,NIAH(1,2)),(NIB,NIBH(1,2)),(NIC,NICH(1,2)),
*(NID,NIDH(1,2)),(NIE,NIEH(1,2))
LOGICAL LRA,LRB,LRC,LRD,LRE
LOGICAL ITEMP,HTEMP
DATA LRA,LRB,LRC,LRD,LRE/5*,TRUE./
DATA JA,JB,JC,JD,JE,IA,IB,IC,IE/10*1/,KA,KB,KC,KD,KE/5*1/
DATA LA,LB,LC,LD,LE/5*0/
DATA NERHJ/14H#ASG,T 9.,F . /
FORMAT(2I3)
FORMAT(1H0,4X,15,7HBLOKKER)
FORMAT(4X,14,' BLOKKER MED ',13,' LINJER I SISTE BLOKK',
*' GJENSTAR')
*FORMAT(2L1)
FORMAT(5X,11)
FORMAT(2X,2I1)
FORMAT(12,11)
FORMAT(1X,2I1)
FORMAT(12,11,12)
FORMAT(2X,2I1,12)
FORMAT(4X,'KOPG=',L1,' BEGGE=',L1)
FORMAT(4X,'REGISTER NR:',13,' NAVNMODUS NR:',13)

```

```

BESTEM STARTPARAMETRE.

```

```

READ(5,10) ICAPO
READ(5,22) KOPG,BEGGE
WRITE(6,42) KOPG,BEGGE
READ(5,10) IREG,IMOD
WRITE(6,50) IREG,IMOD
READ(5,10) JHREG,JHMOD
WRITE(6,50) JHREG,JHMOD
READ(5,22) ITEMP,HTEMP
REWIAD 4
CALL NY3(KOPG)
DEFINE FILE 2(1DAPO,48,L,111)
READ(2'1) LBT,LMSQT
DO 75 L=2,10
LL=8*(L-1)
LLL=LL-7
READ(2'L) (MSGTAB(NT),NT=LLL,LL)
CONTINUE

```

```

000087 003 NUNIN=9
000088 003 ASSIGN 111 TO LABEL
000089 003 JFUNIT=7
000090 003 JFUNIN=8
000091 003 JUNIT=22
000092 003 LTELL=0
000093 003 KUT=0
000094 003 LUT=1
000095 003 80 CONTINUE
000096 003 C
000097 003 C FORBERED INNLESING AV NYE MSG-NUMRE.
000098 003 C
000099 003 LUN=9
000100 003 IUN=16
000101 003 CALL NTRAN(NUNIN,10,22)
000102 003 IHMS=1
000103 003 REWIND JFUNIT
000104 003 CALL NTRAN(JUNIT,10,22)
000105 003 90 CONTINUE
000106 003 C
000107 003 C LES INN NYE MSG-NUMRE,OG OPPRETT NYE FILER.
000108 003 C
000109 003 READ(5,10,END=95) MSG
000110 003 LUN=LUN+1
000111 003 IUN=IUN+1
000112 003 CALL RNAVN(MSG,IREG,NAVN,IMOD)
000113 005 CALL ASGUP(IUN,NAVN,ITEMP)
000114 003 CALL RNAVN(MSG,JHREG,NAVN,JHMOD)
000115 005 CALL ASGUP(LUN,NAVN,HTEMP)
000116 003 NUN(MSG)=LUN-9
000117 003 CALL NTRAN(LUN,10,22)
000118 003 CALL NTRAN(IUN,10,22)
000119 007 LHMS(IHMS)=MSG 6MARSDENNUMMERET LAGRES.
000120 003 IHMS=IHMS+1
000121 003 IF (IHMS.EQ.6) GO TO 100
000122 003 GO TO 90
000123 003 95 CONTINUE
000124 003 IF (IHMS.EQ.1) GO TO 1000
000125 003 100 CONTINUE
000126 003 C
000127 003 C LES INN FØRSTE BLOKK,START INNLESING AV ANDRE BLOKK.
000128 003 C
000129 004 CALL NTRAN(NUNIN,2,1400,INN(1,1),LSTAT,22)
000130 003 K=1

000131 003 CALL NTFEIL(1400,LSTAT,NUNIN,K,11)
000132 003 J=1
000133 003 CALL NTRAN(NUNIN,2,1400,INN(1,101),LSTAT)
000134 003 GO TO LABEL
000135 003 111 CONTINUE
000136 003 C
000137 003 C NY STASJON,BEREGN MSG-NUMMER,KVADRANT,ENGRADERSRUTENUMMER
000138 003 C OG KCRTKODE.
000139 003 C
000140 003 NT=FLD(C,6,INN(14,J))
000141 003 IF (NT.EQ.57) GO TO 115
000142 003 NT=1
000143 003 DECODE(35,INN(3,J)) NQ
000144 003 GO TO 120
000145 003 115 CONTINUE
000146 003 NT=9
000147 003 NQ=0
000148 003 120 CONTINUE
000149 003 IF (NQ.GT.1) GO TO 130
000150 003 DECODE(36,INN(2,J)) LAT,LATH
000151 003 121 CONTINUE
000152 003 IF (NQ.GT.0) GO TO 125
000153 003 C
000154 003 C KVADRANT 0.
000155 003 C
000156 003 IF (NT.EQ.1) DECODE(37,INN(3,J)) LONG,LONGH
000157 003 IF (NT.EQ.9) DECODE(38,INN(3,J)) LONG,LONGH
000158 003 123 CONTINUE
000159 003 MSG=36*(LAT+1)-LONG
000160 003 IF (MSG.LT.289) GO TO 250
000161 003 MSG=936-LONG
000162 003 GO TO 250
000163 003 125 CONTINUE
000164 003 C
000165 003 C KVADRANT 1.
000166 003 C
000167 003 DECODE(39,INN(3,J)) LONG,LONGH,L
000168 003 CALL HJELP(LONG,LONGH,L,123,1)
000169 003 MSG=36*LAT+LONG+1
000170 003 IF (LAT.LT.8) GO TO 250
000171 003 MSG=901+LONG
000172 003 GO TO 250
000173 003 130 CONTINUE
000174 003 DECODE(41,INN(2,J)) LAT,LATH,L

```

```
000175 003 CALL HJELP(LAT,LATH,L,112,2)
000176 003 IF (NQ.GT,2) GO TO 140
4 000177 003 C
000178 003 C KVADRANT 2.
6 000179 003 C
000180 003 DECODE(37,INN(3,J)) LONG, LONGH
8 000181 003 135 CONTINUE
000182 003 MSQ=335+36*LAT-LONG
10 000183 003 GO TO 250
000184 003 140 CONTINUE
12 000185 003 C
000186 003 C KVADRANT 3.
14 000187 003 C
000188 003 DECODE(39,INN(3,J)) LONG, LONGH, L
16 000189 003 CALL HJELP(LONG, LONGH, L, 1135, 1)
000190 003 MSQ=300+36*LAT+LONG
000191 003 GO TO 250
000192 003 200 CONTINUE
20 000193 003 C
000194 003 C NY STASJON, LES INN MSQ=NUMMER, KVADRANT, ENGRADERSRUTENUMMER
22 000195 003 C OG KORTKODE.
000196 003 C
24 000197 003 READ(JFLNIN, END=400) MSQ, NQ, NQH, NT
000198 003 GO TO 255
26 000199 003 250 CONTINUE
000200 003 NQH=LONGH+10*LATH
28 000201 003 255 CONTINUE
000202 007 C
000203 007 C BESTEM BUFFER FOR STASJONEN.
000204 007 C
000205 003 N=NUN(MSQ)
000206 003 IF (N.NE.0) GO TO (300,310,320,330,340),N
4 000207 003 C
000208 003 C MELLOMLAGRING.
6 000209 003 C
8 000210 003 WRITE(JFUNIT) MSQ, NQ, NQH, NT
10 000211 003 CALL FORDEL(NUT, LUT, KUT, JUNIT, 1400)
12 000212 003 GO TO LABEL
14 000213 003 C
16 000214 003 C BUFFER A-E.
18 000215 003 C
20 000216 003 300 CONTINUE
000217 003 CALL INNHDA(NA, JA, KA, IO, NIA, IA, LA, 17, NTA, 1400, NIAH, LRA)
000218 003 GO TO LABEL

000219 003 310 CONTINUE
000220 003 CALL INNHDA(NB, JB, KB, 11, NIB, IB, LB, 18, NTB, 1400, NIBH, LRB)
000221 003 GO TO LABEL
000222 003 320 CONTINUE
000223 003 CALL INNHDA(NC, JC, KC, 12, NIC, IC, LC, 19, NTC, 1400, NICH, LRC)
000224 003 GO TO LABEL
000225 003 330 CONTINUE
000226 003 CALL INNHDA(ND, JD, KD, 13, NID, ID, LD, 20, NTD, 1400, NIDH, LRD)
000227 003 GO TO LABEL
000228 003 340 CONTINUE
000229 003 CALL INNHDA(NE, JE, KE, 14, NIE, IE, LE, 21, NTE, 1400, NIEH, LRE)
000230 003 GO TO LABEL
000231 003 400 CONTINUE
000232 003 LTELL=LTELL+1
16 000233 003 WRITE(6,20)K
18 000234 003 C
20 000235 003 C AVSLUTT BUFFER A-E.
22 000236 003 C
24 000237 003 CALL IDSTOP(NA, JA, KA, IO, LUN, 1450, NIA, IA, LA, 17, NTA, NIAH, LRA)
26 000238 003 CALL IDSTOP(NB, JB, KB, 11, LUN, 1450, NIB, IB, LB, 18, NTB, NIBH, LRB)
28 000239 003 CALL IDSTOP(NC, JC, KC, 12, LUN, 1450, NIC, IC, LC, 19, NTC, NICH, LRC)
30 000240 003 CALL IDSTOP(ND, JD, KD, 13, LUN, 1450, NID, ID, LD, 20, NTD, NIDH, LRD)
32 000241 003 CALL IDSTOP(NE, JE, KE, 14, LUN, 1450, NIE, IE, LE, 21, NTE, NIEH, LRE)
34 000242 003 450 CONTINUE
000243 003 CALL NTRAN(NUNIN, 22)
36 000244 003 C
38 000245 003 C FRIGJØR DE NYE FILENE.
40 000246 003 C
42 000247 003 DO 500 I=10, LUN
44 000248 005 CALL FREEA(I, ITEMP)
000249 003 N=I+7
000250 005 CALL FREEA(N, HTEMP)
34 000251 003 500 CONTINUE
000252 003 I=KUT+1
36 000253 003 L=LUT-1
000254 003 WRITE(6,21) I, L
38 000255 003 IF ((KUT.EQ.0).AND.(LUT.EQ.1)) GO TO 1000 6FERDIG?
40 000256 003 CALL SLUTT(NUT, LUT, KUT, JUNIT, 29, 11)
42 000257 003 ENDFILE JFUNIT
44 000258 003 IF (LTELL.NE.1) GO TO 560
000259 003 C
000260 003 C FRIGJØR ORIGINALFIL, OPPRETT NY, TEMPORÆR, MELLOMLAGRINGSFIL (I/O-9),
44 000261 003 C
000262 003 CALL FREEB(NUNIN)
```

```
000263 003 CALL ERTRAN(6,NERHJ)
000264 003 560 CONTINUE
000265 003 C
000266 003 C BYTT OM I/O=NUMRE MELLOM INNLESE OG MELLOMLAGRINGSFILER.
000267 003 C
000268 003 L=NUNIN
000269 003 NUNIN=JUNIT
000270 003 JUNIT=L
000271 003 REWIND JFUNIT
000272 003 L=JFUNIT
000273 003 JFUNIT=JFUNIN
000274 003 JFUNIN=L
000275 003 C
000276 003 C SETT LINJENUMMERPEKER SLIK AT BEREGNEDE DATA LESES INN I NESTE
000277 003 C GJENNOMLØP.
000278 003 C
000279 003 ASSIGN 200 TO LABEL
000280 003 GO TO 80
000281 003 1 CALL ERTRAN(2)
000282 003 1000 CONTINUE
000283 003 ENDFILE 4
000284 003 STOP MCDATA
000285 003 C
000286 003 SUBROUTINE HJELP(LHJA,LHJB,LHJC,I,KORR)
000287 003 C
000288 003 C HJELP BEREGNER AVSTAND FRA EKVATOR/NULLMERIDIAN I HELE
000289 003 C TIGRADER, OG AVSTAND FRA LAVESTE BREDD/LENGDE I HELE
000290 003 C GRADER. KVADRANTNUMMERET KONTROLLERES VED GRENSENE.
000291 003 C
000292 003 IF (LHJC.NE.0) RETURN
000293 003 IF (LHJB.NE.0) GO TO 150
000294 003 IF (LHJA.NE.0) GO TO 100
000295 003 NQ=NQ-KORR
000296 003 RETURN 4
000297 003 100 CONTINUE
000298 003 LHJA=LHJA-1
000299 003 LHJB=LHJB-1
000300 003 RETURN
000301 003 150 CONTINUE
000302 003 LHJB=LHJB-1
000303 003 RETURN
000304 003 END
```

INNHDA

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Entrepunkt : IDSTOP
Dato : 20/5-75
Utført av : Ø. Strand
Formål : INNHDA er hjelperutine for MSQDATA-INNH. Rutinen overfører en stasjon til et måledatabuffer, lager innholdsregister og kontrollerer rekkefølgen. Rutinen styrer innlesing og utskriving av data, samt nullstiller diverse parametre etter bruk.

Kall: INNHDA(NUT, NU, KN, NUNIT, IUT, IU, KU, NUNUT, NTU, \$, IHNUT, REKKE)
IDSTOP(NUT, NU, KN, NUNIT, LUN, \$, IUT, IU, KU, NUNUT, NTU, IHNUT, REKKE)

Generelt:

INNHDA overfører en stasjon (hydrografiske måledata) fra et innlesebuffer til et måledatabuffer, og lager innholdsfortegnelse for den nye måledatafilen. Rutinen styrer innlesning av nye data til innlesefilen, utskriving av fulle buffere til de nye filene og kontrollerer rekkefølgen i filene.

IDSTOP fyller ut resten av de siste buffrene med nuller, og skriver disse ut på de respektive filer. Videre nullstilles endel parametre (for neste utsorteringsomgang), engradersrutetabellene oppdateres og Marsdenrutenummer lagres på en egen fil sammen med opplysninger om korrekt/ukorrekt rekkefølge.

Formelle variable:

NAVN:	TYPE:	BRUK:
NUT(14, 100)	INTEGER	Utskriftsbuffer for måledata.
NU	INTEGER	Linjenummer i måledatabuffer.
KN	INTEGER	Blokknummer i ny måledatafil.
NUNIT	INTEGER	I/O-enhet for ny måledatafil.
IUT(10, 100)	INTEGER	Buffer for innholdsregister.
IU	INTEGER	Linjenummer i innholdsregister.
KU	INTEGER	Blokknummer i ny innholdsregisterfil.
NUNUT	INTEGER	I/O-enhet for ny innholdsregisterfil.

NAVN:	TYPE:	BRUK:
NTU(100)	INTEGER	Tabell over antall stasjoner pr. engraders-rute.
\$	Linjenummer	INNDA: Retur hvis innlesefilen er ferdig behandlet.
LUN	INTEGER	Hvis måledatafilens I/O-enhet er lik LUN, returneres via eget uthopp.
\$	Linjenummer	IDSTOP: Spesielt uthopp (se LUN).
IHNUT(10, 101)	INTEGER	Linje 2-101 har samme innhold som IUT, linje 1 inneholder skipsnummer, stasjonsnummer, år og mnd.-dag fra siste post i foregående blokk. (Brukes ved kontroll av rekkefølgen).
REKKE	LOGICAL	Angir hvorvidt rekkefølgen er i overensstemmelse med vanlig sorteringsorden (TRUE).

COMMON variable:

blank:

NAVN:	TYPE:	BRUK:
NUN(936)	INTEGER	Hjelpevariabel i MSQDATA-INNH (nullstilles i IDSTOP).
LHMS(5)	INTEGER	Inneholder MSQ-numre.
NQ	INTEGER	Kvadrantnummer for stasjonen som behandles.
NQH	INTEGER	Engradersrutenummer for stasjonen som behandles.
NT	INTEGER	Kodetype (1 = vanlig kode, 9 = fjorddatakode).
BEGGE	LOGICAL	Parameter som brukes ved oppdatering av engradersrutefortegnelse (se: NY3). (Angir registrering på I/O-enhet 28).

DATA:

NAVN:	TYPE:	BRUK:
INN(14, 200)	INTEGER	Innlesebuffer.
J	INTEGER	Linjenummer i innlesebuffer.
K	INTEGER	Blokknummer, innlesefil.
NUNIN	INTEGER	I/O-enhet, innlesefil.

NAVN:	TYPE:	BRUK:
LSTAT	INTEGER	Statusvariabel (NTRAN), innlesefil.

Eksterne rutiner

- : LESDB
- NTFEIL
- NY3(OPPD3)

Interne rutiner

- : HJELP (Overfører en linje til måledata-bufferet, og skriver dette ut når det er fullt). TESTR (Kontrollerer rekkefølgen).

Variable:

NAVN:	TYPE:	BRUK:
IC	INTEGER	Teller antall kort i stasjonen.
IOX	INTEGER	Teller antall oksygenobservasjoner i stasjonen.
IFOS	INTEGER	Teller antall fosfatobservasjoner i stasjonen.
KF	INTEGER	Blokknummer i ny måledatafil for første linje i stasjonen.
LI	INTEGER	Linjenummer for første linje i stasjonen (i blokk KF).
IDYP	INTEGER	Største måledyp.
I	INTEGER	Hjelpevariabel.
L	INTEGER	Hjelpevariabel.
N	INTEGER	Hjelpevariabel.
LL	INTEGER	Hjelpevariabel.
IHMAX	INTEGER	Hjelpevariabel.

```

4 000001      005 C
6 000002      005 C
8 000003      005 C
10 000004      005 C
12 000005      005 C
14 000006      005 C
16 000007      005 C
18 000008      005 C
20 000009      005 C
22 000010      005 C
24 000011      005 C
26 000012      005 C
28 000013      009 C
30 000014      009 C
32 000015      005 C
34 000016      005 C
36 000017      005 C
38 000018      005 C
40 000019      005 C
42 000020      005 C
44 000021      005 C
46 000022      005 C
48 000023      005 C
50 000024      005 C
52 000025      005 C
54 000026      005 C
56 000027      009 C
58 000028      009 C
60 000029      009 C
62 000030      009 C
64 000031      009 C
66 000032      005 C
68 000033      005 C
70 000034      005 C
72 000035      005 C
74 000036      005 C
76 000037      005 C
78 000038      005 C
80 000039      005 C
82 000040      005 C
84 000041      005 C
86 000042      008 C

      *****
      *
      *      INNHDA      *
      *      IDSTOP      *
      *
      *****

      INNHDA MED ENTREPUNKT ER EN HJELPERUTINE FOR HOVEDPROGRAMMET
      MSQDATA-INNH,RUTINEN OVERFØRER EN STASJON FRA INNLESEBUFFERET (INN)
      TIL ET DATABUFFER (NUT),LAGER EN INNHOLDSFORTEGNELSE (IUT),
      OG KONTROLLERER REKKEFØLGEN I FILEN,
      RUTINEN BESØRGER OGSÅ INNLESING/UTSKRIVING AV DE RESPEKTIVE
      BUFFRE NAR DISSE ER TOMME/FULLE,
      RUTINEN OPPDATERER TABELLEN OVER ANTALL STASJONER PR. ENGRADERSRUTE,
      SAMT NULLSTILLER DIVERSE PARAMETRE ETTER AT EN GJENNOMGANG AV
      INNLESEFILEN ER FERDIG.

      NØDVENDIGE RUTINER : LESDB
                          NTFEIL
                          NY3

      RESERVEVE 1/0-NUMRE : 4 (LAGRING AV OPPLYSNINGER OM MARSDEN-
                          RUTENUMRE OG KORREKT/UKORREKT REKKE-
                          =FØLGE)

      SUBROUTINE INNHDA(NUT,NU,KN,NUNIT,IUT,IU,KU,NUNUT,NTU,I,IHNUT,
      *REKKE)

      INNHDA SORTERER UT EN STASJON FRA INNLESEBUFFERET TIL DATABUFFERET,
      SAMT PRODUSERER ET INNHOLDSREGISTER.
      RUTINEN BESØRGER OGSÅ INNLESING/UTSKRIVING.

      DIMENSION NUT(14,100),IUT(10,100),NTU(100)
      DIMENSION IHNUT(10,101)
      LOGICAL REKKE
      LOGICAL BEGGE

000043      007 COMMON NUN(936),LHMS(5),NQ,NQH,NT,BEGGE
000044      005 COMMON /DATA/INN(14,200),J,K,NUNIN,LSTAT
000045      005 10 FORMAT(2X,'UNIT',I3,' '):',I3,' BLOKKER MED ',I6,' RECORDS')
000046      005 20 FORMAT(14,4X,I4,I5,1X,I3,I4)
000047      005 21 FORMAT(14,4X,I5,I5,I3,I4)
000048      005 24 FORMAT(/4X,'MARSDENRUTE NR:',I3)
000049      005 22 FORMAT(I3,L1)
000050      005 23 FORMAT(4X,'MSG NR :',I3,' HAR RIKTIG REKKEFØLGE')
000051      005 IF(NT.EQ.9) GO TO 100
000052      005 C
000053      005 C      STASJONSKORTET DEKODES (VANLIG KODE).
000054      005 C
000055      005 DECODE(20,INN(1,J))IUT(1,IU),IUT(6,IU),IUT(7,IU),IUT(3,IU),IUT
000056      005 * (4,IU)
000057      005 IUT(6,IU)=10*IUT(6,IU)
000058      005 IUT(7,IU)=10*IUT(7,IU)
000059      005 IF(NQ.GT.1)IUT(6,IU)=-IUT(6,IU)
000060      005 IF((NQ.EQ.1).OR.(NQ.EQ.3))IUT(7,IU)=-IUT(7,IU)
000061      005 GO TO 110
000062      005 100 CONTINUE
000063      005 C
000064      005 C      STASJONSKORTET DEKODES (FJORDDATAKODEN).
000065      005 C
000066      005 DECODE(21,INN(1,J))IUT(1,IU),IUT(6,IU),IUT(7,IU),IUT(3,IU),IUT
000067      005 * (4,IU)
000068      005 110 CONTINUE
000069      005 IUT(3,IU)=IUT(3,IU)+1000
000070      005 C
000071      005 C      STASJONEN REGISTRERES I ENGRADERSRUTETABELLEN.
000072      009 C
000073      005 NTU(NQH+1)=NTU(NQH+1)+1
000074      005 IC=1      GTELLER ANTALL KORT.
000075      005 IOX=0      GTELLER ANTALL OKSYGENOBSERVASJONER.
000076      005 IFOS=0      GTELLER ANTALL FOSFATOBSERVASJONER.
000077      005 30 FORMAT(4X,I4)
000078      005 31 FORMAT(4X,R4)
000079      005 DECODE(30,INN(1,J),ERR=130)IUT(2,IU)
000080      005 GO TO 140
000081      005 130 CONTINUE
000082      005 DECODE(31,INN(1,J))IUT(2,IU)
000083      005 140 CONTINUE
000084      005 C
000085      005 C      ADRESSEPARAMETRE.
000086      005 C

```

```
2 000087 005 KF=KN
3 000088 005 LI=NU
4 000089 005 CALL HJELP
5 000090 005 CALL LESDB(INN,14,NUNIN,K,J,LSTAT,11000)
6 000091 005 150 CONTINUE
7 000092 005 C
8 000093 005 C NYTT KORT.
9 000094 005 C
10 000095 005 N=FLD(6,6,INN(14,J))
11 000096 005 IF (N.EQ.51) GO TO 160
12 000097 005 IF (N.EQ.57) GO TO 160
13 000098 005 GO TO 300
14 000099 005 160 CONTINUE
15 000100 005 C
16 000101 005 C NYTT DATAKORT (RECORD).
17 000102 005 C
18 000103 005 IC=IC+1
19 000104 005 N=FLD(30,6,INN(13,J))
20 000105 005 IF (N.EQ.52) IFOS=IFOS+1
21 000106 005 N=FLD(18,6,INN(10,J))
22 000107 005 IF (N.NE.5) IOX=IOX+1
23 000108 005 CALL HJELP
24 000109 005 CALL LESDB(INN,14,NUNIN,K,J,LSTAT,11000)
25 000110 005 GO TO 150
26 000111 005 300 CONTINUE
27 000112 005 C
28 000113 005 C STASJONEN FERDIG.
29 000114 005 C
30 000115 005 L=NU-1
31 000116 005 IF (L.NE.0) DECODE(40,NUT(5,L))IDYP &MAKSIMALDYPET.
32 000117 005 40 FORMAT(3X,14)
33 000118 005 41 FORMAT(J2,12,12)
34 000119 005 42 FORMAT(15,13,14)
35 000120 005 43 FORMAT(J2,14)
36 000121 005 C
37 000122 005 C SETT INN RESTEN AV PARAMETRENE I INNHOLDSFORTEGNELSEN.
38 000123 005 C
39 000124 005 ENCODE(41,IUT(5,IU))NT,IOX,IFOS
40 000125 005 ENCODE(42,IUT(8,IU))KF,LI,IC
41 000126 005 ENCODE(43,IUT(10,IU))NGH,IDYP
42 000127 005 IU=IU+1
43 000128 005 IF (IU.LT.101) GO TO 315
44 000129 005 C
45 000130 005 C SKRIV UT ET INNHOLDSBUFFER.

2 000131 005 C
3 000132 005 IU=1
4 000133 005 KU=KU+1
5 000134 005 CALL NTRAN(NUNUT,1,1000,IUT,L,22)
6 000135 005 CALL NTFEIL(1000,L,NUNUT,KU,11000)
7 000136 005 CALL TESTR(100)
8 000137 005 315 CONTINUE
9 000138 005 IF (N.NE.0) RETURN
10 000139 005 C
11 000140 005 C SLUTT PA DATA I INNLESEFILEN.
12 000141 005 C
13 000142 005 RETURN 10
14 000143 005 C
15 000144 005 ENTRY IDSTOP(NUT,NU,KN,NUNIT,LUN,1,IUT,IU,KU,NUNUT,NTU,IHNUT,
16 000145 005 *REKKE)
17 000146 005 C
18 000147 005 C IDSTOP SKRIVER UT BUFFERNE NAR INNLESING ER AVSLUTTET,
19 000148 009 C NULLSTILLER PARAMETRE,LAGRER ENGRADERSFORTEGNELSEN SAMT
20 000149 009 C LAGRER MARSDENRUTENUMMER MED OPPLYSNINGER OM KORREKT/UKORREKT
21 000150 009 C REKKEFOLGE PA EN EGEN FIL (1/0=ENHET 4).
22 000151 005 C
23 000152 005 DO 350 I=NU,100
24 000153 005 DO 350 L=1,14
25 000154 005 NUT(L,I)=0
26 000155 005 350 CONTINUE
27 000156 009 CALL NTRAN(NUNIT,1,1400,NUT,L,22) &SISTE BLOKK TIL MALEDATAFIL
28 000157 005 NU=100*(KN-1)+NU-1
29 000158 005 CALL NTFEIL(1400,L,NUNIT,KN,11000)
30 000159 005 WRITE(6,10)NUNIT,KN,NU
31 000160 005 KN=1
32 000161 005 NU=1
33 000162 005 DO 400 I=IU,100
34 000163 005 DO 400 L=1,10
35 000164 005 IUT(L,I)=0
36 000165 005 400 CONTINUE
37 000166 005 IU=IU-1
38 000167 005 CALL TESTR(IU)
39 000168 005 DO 410 L=1,4
40 000169 005 IHNUT(L,I)=0
41 000170 005 410 CONTINUE
42 000171 005 IU=100+KU+IU
43 000172 005 KU=KU+1
44 000173 009 CALL NTRAN(NUNUT,1,1000,IUT,L,22) &SISTE BLOKK TIL INNHOLDSREGISTERFIL
45 000174 005 CALL NTFEIL(1000,L,NUNUT,KU,11000)
```

```

2 000175 005 WRITE(6,10)NUNUT,KU,IU
000176 005 KU=0
4 000177 005 IU=1
000178 005 L=NUNIT-9
8 000179 009 N=LHMS(L) %MARSDENRUTENUMMER
000180 009 NUN(N)=0 %BUFFERPEKERTABELL (BRUKES I MSQDATA-INNH) NULLSTILLES
8 000181 005 WRITE(6,24) N
000182 006 WRITE(4,22) N,REKKE %OPPLYSNINGER LAGRES FOR SENERE BRUK
10 000183 005 IF (REKKE) WRITE(6,23) N
000184 005 REKKE=.TRUE.
12 000185 007 CALL OPPD3(N,L,NTU,1,BEGGE)
000186 005 DO 450 L=1,100
14 000187 005 NTU(L)=0
000188 005 450 CONTINUE
16 000189 005 IF (NUNIT.EQ.LUN) RETURN 6
000190 005 RETURN
18 000191 005 1000 CALL ERTRAN(2) %FEILAVSLUTTING.
000192 005 C
20 000193 005 SUBROUTINE HJELP
000194 005 C
22 000195 005 C HJELP OVERFØRER DATA TIL DATABUFFERET OG SKRIVER DETTE UT
000196 005 C NAR DET ER FULLT.
24 000197 005 C
000198 005 10 FORMAT(3X,14)
26 000199 005 DO 100 I=1,14
000200 005 NUT(I,NU)=INN(I,J)
28 000201 005 100 CONTINUE
000202 005 NU=NU+1
000203 005 IF(NU.LE.100) RETURN
000204 005 NU=1
32 000205 005 KN=KN+1
000206 005 CALL NTRAN(NUNIT,1,1400,NUT,L,22)
34 000207 005 CALL NTFEIL(1400,L,NUNIT,KN,1999)
000208 005 DECODE(10,NUT(5,100))IDYP
36 000209 005 RETURN
000210 005 999 CALL ERTRAN(2)
38 000211 005 C
40 000212 005 SUBROUTINE TESTR(MAX)
42 000213 005 C
44 000214 009 C TESTR TESTER OM REKKEFØLGEN I IHNUT (MAX LINJER PLUSS SISTE
46 000215 009 C POST FRA FØRRIGE BLOKK) ER I OVERENSSTEMMELSE MED VANLIG
48 000216 009 C SORTERINGSORDEN.
50 000217 005 C
000218 006 IF (.NOT.REKKE) RETURN %FEIL REKKEFØLGE ER REGISTRERT TIDLIGERE

000219 006 % I:VIDERE TEST ER UNØDVENDIG
000220 005 IF (MAX.EQ.0) RETURN
000221 005 10 FORMAT(4X,'FEIL: ',4I5,2X,A6,2I9,2X,2A6,2X,A6)
000222 005 15 FORMAT(4X,'FEIL: ',15,1X,R4,2I5,2X,A6,2I9,2X,2A6,2X,A6)
000223 005 IHMAX=MAX+1
000224 005 DO 300 L=2,IHMAX
000225 005 LL=IHNUT(3,L)-IHNUT(3,L-1)
000226 005 IF (LL) 500,110,300
10 000227 005 110 CONTINUE
000228 005 LL=IHNUT(1,L)-IHNUT(1,L-1)
12 000229 005 IF (LL) 500,120,300
000230 005 120 CONTINUE
14 000231 005 LL=IHNUT(4,L)-IHNUT(4,L-1)
000232 005 IF (LL) 500,130,300
16 000233 005 130 CONTINUE
000234 005 LL=IHNUT(2,L)-IHNUT(2,L-1)
18 000235 005 IF (LL) 500,150,300
000236 005 150 CONTINUE
20 000237 005 IF (IHNUT(2,L).GT.9999) GO TO 200
000238 005 WRITE(6,10) (IHNUT(LL,L-1),LL=1,10)
22 000239 005 WRITE(6,10) (IHNUT(LL,L),LL=1,10)
000240 005 GO TO 500
24 000241 005 200 CONTINUE
000242 005 WRITE(6,15) (IHNUT(LL,L-1),LL=1,10)
26 000243 005 WRITE(6,15) (IHNUT(LL,L),LL=1,10)
000244 005 GO TO 500
28 000245 005 300 CONTINUE
000246 005 IHNUT(1,1)=IHNUT(1,IHMAX)
30 000247 005 IHNUT(2,1)=IHNUT(2,IHMAX)
000248 005 IHNUT(3,1)=IHNUT(3,IHMAX)
32 000249 005 IHNUT(4,1)=IHNUT(4,IHMAX)
000250 005 RETURN
34 000251 005 500 CONTINUE
000252 005 REKKE=.FALSE.
36 000253 005 RETURN
000254 005 END

```

38
40
42
44

STYREPROGRAM

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Hovedprogram
Dato : 1/5-76
Utført av : Ø. Strand
Formål : Hovedprogram for rutinene i sorterings- og lagringspakken.

Generelt:

STYREPROGRAM er hovedprogram for de deler av sorterings- og lagringspakken som ikke dekkes av MSQDATA og MSQDATA-INNH, det vil si rutinene CONT, SORTER, MHSORT, OMSKR, IMIX og DMIX med tilhørende hjelperutiner. Programmet kan dessuten kopiere filer.

INNLESTE VARIABLE:

De variable i programmet som, direkte eller inndirekte, gis verdier ved innlesing fra kort, kan grovt deles i fire grupper:

Hovedtabellene.
Rutineavhengige filspesifikasjoner.
Andre styrevariable.
Hjelpevariable ved innlesing.

Variable i den siste gruppen beskrives sammen med de øvrige hjelpevariable.

Hovedtabellene er følgende:

IDEV(3, 12)	(INTEGER)
LDEV(3, 12)	(LOGICAL)
IMSQ(3, 30)	(INTEGER)
LMSQ(30)	(LOGICAL)

IDEV og LDEV hører sammen og kan betraktes som en tabell med tolv linjer, hver på seks felter. Tabellene beskriver den generelle (rutine og innholdsuavhengige) del av de brukte filer (I/O-enhetsnumre). Programmet opererer med fire typer filer:

1. Magnetbåndfiler.
2. Katalogiserte discfiler.
3. Temporære discfiler med fast innhold.
Disse tre filtypene er uavhengig av I/O-enhetsnummeret og kan løses fra dette og gis et nytt nummer.
4. Kladdefiler er temporære filer med navn lik I/O-enhetsnummeret.
Disse filene kan brukes til skriving flere ganger, og en kan derved spare discplass.

Feltene i IDEV og LDEV har følgende betydning:

IDEV(1, i)	: I/O-enhetsnummer.
IDEV(2, i)	: Magnetbåndnummer.
IDEV(3, i)	: Filnummer (på bånd).

LDEV(1,i) : TRUE angir (mulig) kladdefil.
LDEV(2,i) : FALSE betyr: discfil.
TRUE betyr: magnetbånd eller ingen fil.
LDEV(3,i) : Hvis kladdefil er mulig: TRUE betyr at filen er klar til bruk. FALSE betyr at filen ikke kan brukes som den er (): inneholder ikke data/kan være en annen filtype). Hvis kladdefil er utelukket: TRUE betyr at en eventuell nyopprettet discfil skal være temporær, mens FALSE angir katalogisert fil.

IMSQ og LMSQ kan betraktes som en tabell med tredve linjer, hver på fire felter. Tabellene oppgir Marsdenrutenumre som skal behandles, sammen med en del andre opplysninger. Feltene har følgende betydning:

IMSQ(1,i) : Marsdenrutenummer.
IMSQ(2,i) : Navnmodus for eksisterende registerfil. (Gir sammen med Marsdenrutenummer og registertypenummeret filnavnet (v.h.a. rutinen RNAVN).)
IMSQ(3,i) : Navnmodus for eksisterende måledatafil.
LMSQ(i) : Angir korrekt/ukorrekt (TRUE/FALSE) rekkefølge for måledatafil.

Programpakkens hovedrutiner benytter, foruten mellomagringsfiler, fra to til fire datafiler. Disse deles inn i lese og skrivefiler, og hovedprogrammet inneholder et sett variable som beskriver hver enkelt av dem. Disse variable er delt inn i seks forskjellige typer som er gitt følgende formelle navn:

NAVN:	TYPE:	BRUK:
NYDEV	INTEGER	Lese og skrivefiler. Peger til linjenummer i IDEV og LDEV.
LOTAPE	LOGICAL	Skrivefiler: TRUE = magnetbånd FALSE = discfil Lesefiler: TRUE = krever magnetbåndfil.
LODISC	LOGICAL	Bare lesefiler. TRUE = krever discfil.
SLETT	LOGICAL	Bare lesefiler. TRUE angir at eventuelle discfiler skal frigjøres etter gjennomlesning.
IREG	INTEGER	Lese og skrivefiler. Registertypenummer.
NMOD	INTEGER	Bare skrivefiler. Navnmodus for eventuell ny discfil (ikke kladdefiler).

De aktuelle variable fremgår av tabelloversikten.

		NYDEV	LOTAPE	LODISC	SLETT	IREG	NMOD
CONT	Les måledata	MLCO	LTMLCO	LDMLCO	SLMLCO	MLRCO	
	Skriv innhold	ISCO	LTISCO			ISRCO	NAISCO
	Skriv måledata	MSCO	LTMSCO			MSRCO	NAMSCO
SORTER	Les innhold	ILSO	LTILSO	LDILSO	SLILSO	ILRSO	
	Skriv innhold	ISSO	LTISSO			ISRSO	NAISSO
MHSORT	Les innhold	ILMH	LTILMH	LDILMH	SLILMH	ILRMH	
	Skriv innhold	ISMH	LTISMH			ISRMH	NAISMH
OMSKR	Les innhold	ILOM	LTILOM	LDILOM	SLILOM	ILROM	
	Les måledata	MLOM	LTMLOM	LDMLOM	SLMLOM	MLROM	
	Skriv innhold	ISOM	LTISOM			ISROM	NAISOM
	Skriv måledata	MSOM	LTMSOM			MSROM	NAMSOM
IMIX	Les innhold A	ILAI	LTILAI	LDILAI	SLILAI	ILRAI	
	Les innhold B	ILBI	LTILBI	LDILBI	SLILBI	ILRBI	
	Skriv innhold	ISIM	LTISIM			ISRIM	NAISIM
DMIX	Les måledata A	MLAD	LTMLAD	LDMLAD	SLMLAD	MLRAD	
	Les måledata B	MLBD	LTMLBD	LDMLBD	SLMLBD	MLRBD	
	Skriv måledata	MSDM	LTMSDM			MSRDM	NAMSDM
KOPI	Les	KOL	LTLKO	LDLKO	SLLKO	KOLR	
	Skriv	KOS	LTSKO			KOSR	NASKO

De øvrige styrevariable er en samling variable som dels er generelle og dels knyttet til bestemte rutiner:

NAVN:

TYPE:

BRUK:

KOPG

LOGICAL

Brukes i rutinen NY3 (oppdatering/
opprettning av engradersrutefortegnelse),
(formell variabel: KOPG).

NAVN:	TYPE:	BRUK:
COPD2	LOGICAL	Angir at Marsdenrute- og magnetbåndfortegnelse (REGFIL2) skal oppdateres.
LMF	INTEGER	Angir (sammen med LMS) hvilke Marsdenområder som skal behandles (f.o.m. linje LMF t.o.m. LMS i tabellen IMSQ).
LMS	INTEGER	Se LMF.
IGOTO(9)	INTEGER	Angir behandlingsrekkefølgen (leses inn i klartekst i form av hovedrutinenavn, KOPI (filkopiering), START (ny innlesing av styrevariable) eller STOP (ferdig) som så omdannes til tall som fortolkes av hjelperutinen NESTE).
LOSP	LOGICAL	Brukes i hovedrutinen SORTER, (formell variabel: LOSP).
LHSORT	LOGICAL	Angir at de aktuelle filer er sortert etter oppgitt rekkefølge for engradersruter (brukes i IMIX og DMIX).
TOUT	LOGICAL	Brukes i hovedrutinen CONT (formell variabel: TOUT).
BEGGE	LOGICAL	Brukes i rutinen NY3 (entrepunkt OPPD3, formell variabel: BEGGE).
VDA	LOGICAL	Brukes i hovedrutinen OMSKR (formell variabel: VDA). TRUE angir omskrivning av måldatafil via mellomlagring på en direkte aksessfil.
MOP3CO	INTEGER	Brukes i rutinen NY3 (entrepunkt OPPD3, formell variabel: MOD). Angir oppdateringsmåte for engradersrutefortegnelse etter hovedrutinen CONT.
MTABS	INTEGER	Marsdenrutetabellnummer (i REGFIL2) for skrivefiler.
MTABL	INTEGER	Marsdenrutetabellnummer (i REGFIL2) for lesefiler (lesefil B i IMIX og DMIX).
MIXTAB	INTEGER	Marsdenrutetabellnummer (i REGFIL2) for lesefil A i IMIX og DMIX.
NUNDA	INTEGER	I/O-enhetsnummer for direkte aksessfil i MHSORT (formell variabel: NUNDA).
NAILAI	INTEGER	Navnmodus (gir sammen med andre variable filnavn fra rutinen RNAVN) for lesefil A i IMIX.

NAVN:	TYPE:	BRUK:
NAMLAD	INTEGER	Navnmodus for lesefil A i DMIX.
ITDSI(6)	INTEGER	Inneholder styresetning som brukes ved montering av magnetbånd i rutinen MONTIN (entrepunkt TBAAND). Båndet kan brukes til utskrivning av innholdsregisterfiler.
ISTNSK	INTEGER	Magnetbåndnummer (i magnetbåndtabellen) for båndet gitt i ITDSI.
ISFNSK	INTEGER	Filnummer hvor skriving skal starte på båndet gitt ved ITDSI og ISTNSK.
OVISSK	LOGICAL	Angir (TRUE) om det skal være tillatt å skrive over gammel informasjon på båndet gitt ved ITDSI og ISTNSK.
ITDSM(6)	INTEGER	Som ITDSI, men for måledatafiler.
MSTNSK	INTEGER	Som ISTNSK, men for måledatafiler.
MSFNSK	INTEGER	Som ISFNSK, men for måledatafiler.
OVMSK	LOGICAL	Som OVISSK, men for måledatafiler.
NDEFSSO(100)	INTEGER	Inneholder definisjon av sorteringsrekkefølge for engradersruter. (Formell variabel NDEFSSO i MHSORT, IMIX og DMIX).

INTERNE RUTINER:

Programmet benytter følgende interne rutiner:

MONSKR(LIDEV, LIM, IREG, NMOD, IM)
SKRB(LIDEV, LIM, IREG)
SKRBSL(LIDEV, NRFNY, KBLOKK)
MOLES(MTAB, LIDEV, LIM, LOTAPE, LODISC, IREG, NMOD)
KOPI(NUNIN, NUNUT, NDIM, NRL, KBK, \$)
LESLES(NYDEV, LOTAPE, LODISC, SLETT, IREG, \$)
LESSKR(NYDEV, LOTAPE, IREG, NMOD, \$)
NESTE(\$, \$, \$, \$, \$, \$, \$, \$, \$)
NYSKRB(NDIM, LSKRIV, \$, NR TNY, NRFNY, OVER)
NYLOGI(BOOL, LSKRIV, \$)
NYINT(INTEG, LSKRIV, \$)
HKLA(LLEDEV, LSKDEV, LIM, LOTAPE, NRL, KBK, \$, \$)

De formelle variable har følgende betydning:

NAVN:	TYPE:	BRUK:
NYDEV	INTEGER	Rutineavhengig filspesifikasjon (se tidligere definisjon).
LOTAPE	LOGICAL	Rutineavhengig filspesifikasjon (se tidligere definisjon).

NAVN:	TYPE:	BRUK:
LODISC	LOGICAL	Rutineavhengig filspesifikasjon (se tidligere definisjon).
SLETT	LOGICAL	Rutineavhengig filspesifikasjon (se tidligere definisjon).
IREG	INTEGER	Rutineavhengig filspesifikasjon (se tidligere definisjon).
NMOD	INTEGER	Navnmodus for discfiler (både lese og skrivefiler).
LIDEV	INTEGER	Som NYDEV.
LIM	INTEGER	Linjenummer i hovedtabellene IMSQ og LMSQ (): gir Marsdenrutenummer).
IM	INTEGER	Peker til feltnummer (nr. 2 eller 3) i IMSQ. (Peker til navnmodusfelt som skal endres).
NRTNY	INTEGER	Nummer på magnetbånd som skal klargjøres for skriving.
NRFNY	INTEGER	Startpunkt (filnummer) ved skriving på bånd (nr. NRTNY).
OVER	LOGICAL	Angir (TRUE) om overskriving av gammel informasjon på båndet NRTNY skal tillates.
NRL	INTEGER	Postlengde i den aktuelle fil (måledata: postlengde = 14, innhold: postlengde = 10).
NDIM(6)	INTEGER	I NYSKRB. Inneholder styresetning til bruk ved magnetbåndmontering.
NDIM(NRL, 200)	INTEGER	Databuffer i KOPI.
KBLOKK	INTEGER	Blokkantall (til bruk ved oppdatering av magnetbåndregister).
NUNIN	INTEGER	I/0-enhetsnummer for lesefil.
NUNUT	INTEGER	I/0-enhetsnummer for skrivefil.
KBK	INTEGER	Blokksteller.
LSKRIV	INTEGER	Inneholder tekst (i A-format) som skrives ut.
BOOL	LOGICAL	Tildeles verdi.
INTEG	INTEGER	Tildeles verdi.
LLEDEV	INTEGER	Som NYDEV (for lesefil).
LSKDEV	INTEGER	Som NYDEV (for skrivefil).

NAVN:	TYPE:	BRUK:
MTAB	INTEGER	Marsdenrutetabellnummer.
\$	INTEGER	Forskjellige uthopp fra rutinen.

MONSKR oppretter og klargjør discfiler for skriving, samt ajourfører hovedtabellene.

SKRB setter EOF-merke etter magnetbåndfiler, oppdaterer Marsdenrutetabellen og teller opp filnummeret.

SKRBSL avslutter skriving på magnetbånd. Magnetbåndtabellen oppdateres, NRFNY (startpunkt for eventuell fortsatt skriving) korrigeres, og en ekstra blokk skrives ut bak siste EOF-merke for å forebygge visse typer magnetbåndfeil.

MOLES tilordner og klargjør filer (alle typer) for lesing.

KOPI kopierer en fil. \$ er normal retur.

LESLES tilordner verdier til rutineavhengige filspesifikasjoner for lesefiler. \$ er normal retur.

LESSKR tilordner verdier til rutineavhengige filspesifikasjoner for skrivefiler. \$ er normal retur.

NESTE teller opp styretabellpekeren og gir uthopp til ønsket sted i programmet.

NYSKRB leser inn nye verdier til NDIM, NRTNY, NRFNY og OVER. \$ er normal retur.

NYLOGI tildeler ny verdi til BOOL. \$ er normal retur.

NYINT tildeler ny verdi til INTEG. \$ er normal retur.

HKLA brukes i forbindelse med hovedrutinene SORTER og OMSKR. Hvis rekkefølgen i den aktuelle fil allerede er korrekt, er sortering/omskrivning unødvendig. HKLA undersøker dette, og utfører eventuelt de nødvendige foranstaltninger. (Korrigerer hovedtabellen LDEV, og kopierer den aktuelle fil hvis skrivefilen er en magnetbåndfil). Første \$ er retur uten kopiering, mens siste \$ er retur etter kopiering.

INNLESNING AV VARIABLE:

For å gjøre innlesning av verdier til variable mest mulig generell, leses disse ved at den variables navn, eller en annen kode, oppgis på kortet. Variabelkoden blir fortolket i programmet, og de(n) oppgitte variable tildeles de(n) ønskede verdi(er).

HOVEDRUTINENE:

Hovedrutinene består av de eksterne rutinene CONT, SORTER, MHSORT, OMSKR, IMIX og DMIX samt den interne rutinen KOPI. Rundt hver av disse rutinene er det et sett hjelperutiner som tilordner og oppretter filer, monterer magnetbånd, frigjør filer samt oppdaterer de forskjellige registerfiler. I forbindelse med de to rutinene SORTER og OMSKR er det en rutine

(HKLA) som kontrollerer om den aktuelle fil allerede er korrekt sortert (fra hovedtabellen LMSQ).

Filer som er korrekt sortert, sorteres ikke påny, men hvis vedkommende rutines skrivefil er en magnetbåndfil, kopieres filen til båndet.

PRINSIPPBESKRIVELSE:

Det forenklete flytskjema viser at innlesing av kort foregår i tre adskilte programområder: Først en ikke-repeterbar del hvor nødvendige initialiseringer foretas, dernest en del hvor data leses inn i hovedtabellene, og tilslutt en del hvor de øvrige styrevariable kan gis verdier. Den siste delen styrer også direkte en del spesielle funksjoner: I tilfelle noen av hovedrutinene (): MHSORT og OMSKR) skal benytte direkte aksessfiler, blir disse filene opprettet (temporære) og definert for direkte aksessbruk samtidig som de tilhørende variable (NUNDA og VDA) tilordnes verdier. Videre kan en i denne delen av programmet beordre frigjøring av filer og demontering av magnetbånd (det siste slik at båndstasjonen reserveres for fortsatt bruk). Behandlingsrekkefølge mellom hovedrutinene bestemmes ved å oppgi de ønskede rutinenavn i rekkefølge på et kort. Navnene blir oversatt til tall som fortolkes av rutinen NESTE. Ved en spesiell startordre startes gjennomløpet av hovedrutinene. De Marsdenområder som skal behandles er gitt som: f.o.m. linje/t.o.m. linje i tabellen IMSQ. Etter valg kan behandling skje i parallell (): samtlige områder ferdigbehandles i en rutine før neste rutine tas i bruk) eller i serie (): hvert område ferdig behandles av samtlige rutiner før neste område behandles). Etter avslutning i hovedrutinene kan, om ønskelig, ny innlesing av styrevariable foretas.

COMMON variable:

DATA:

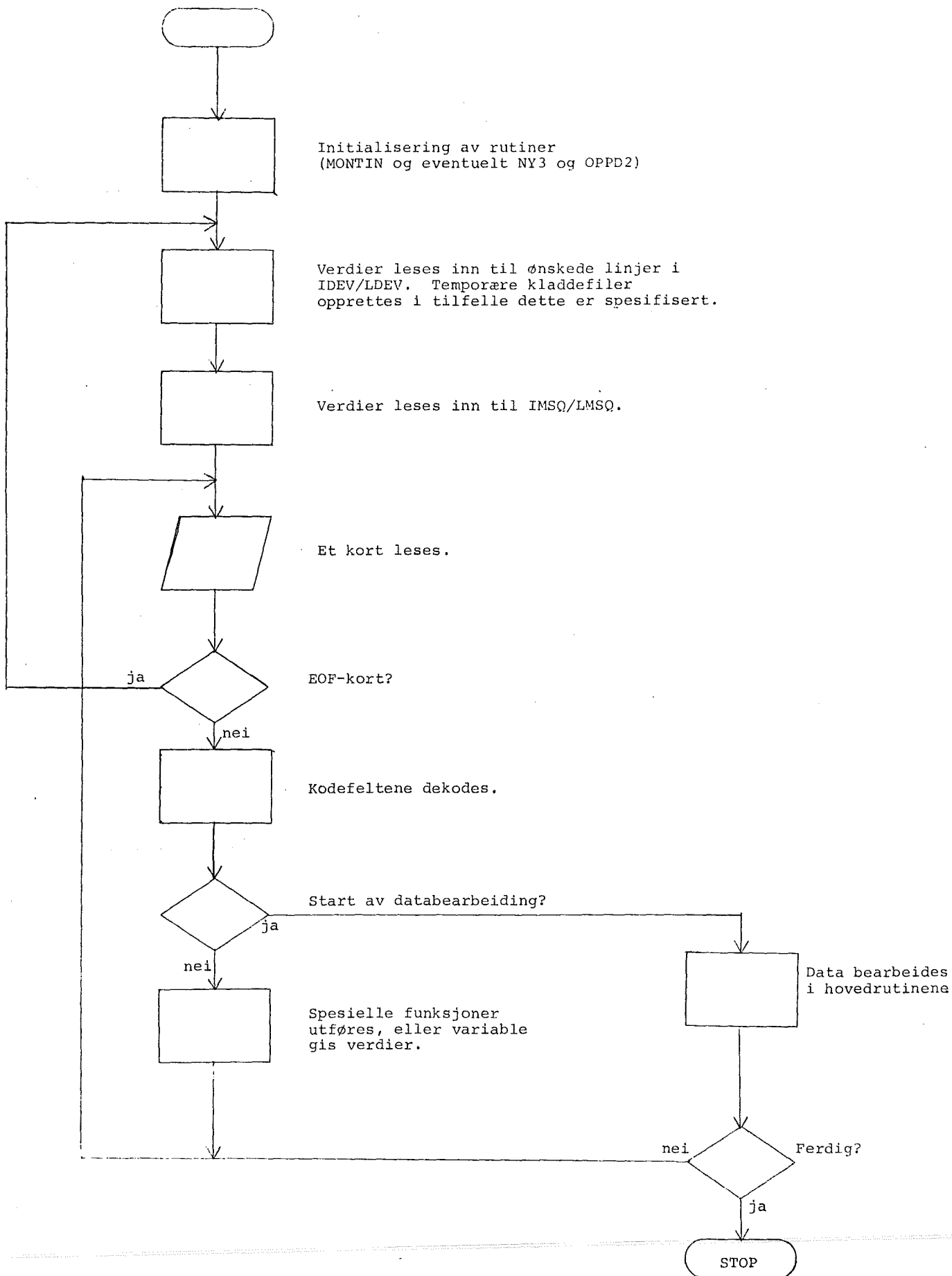
NAVN:	TYPE:	BRUK:
INN(2800)	INTEGER	Databuffer som brukes ved kopiering av filer (dobbelbuffring). Fellesområdet DATA brukes til mellomlagring i alle hovedrutinene. Derved spares mye plass i hurtighukommelsen.

RDATA:

NAVN:	TYPE:	BRUK:
MSQTAB(72)	INTEGER	Inneholder Marsdenrutennummer. Peker til linjenummer i Marsdenrutetabellene (REGFIL2).
LBT	INTEGER	Magnetbåndtabellens lengde (i linjer).
LMSQT	INTEGER	Marsdenrutetabellenes lengde (i linjer).

DATT:

NAVN:	TYPE:	BRUK:
TILTAA	LOGICAL	Viser (TRUE) om I/O-enhet nr. 28 er definert for direkte aksessbruk. (Filen inneholder engradersrutefortegnelse til bruk i MHSORT).



Variable:

NAVN:	TYPE:	BRUK:
IRPEK(9)	INTEGER	Hjelpetabell ved koding av IGOTO (se innleste variable). Tabellen inneholder hovedrutinenavn samt ordene STOP og START (STOP: medfører programavslutning, mens START medfører retur til området hvor kort leses).
IGOTOP	INTEGER	Peker i styretabellen IGOTO. Verdien økes med en for hver hovedrutine, og neste rutine finnes i IGOTO.
NYPAR(7)	INTEGER	Hjelpevariabel ved innlesing fra kort. Data leses inn i NYPAR på karakterformat og dekodes senere.
MRUT	INTEGER	Hjelpevariabel ved innlesning fra kort. Inneholder (hele eller deler av) kortkoden.
MT	INTEGER	Hjelpevariabel ved innlesning fra kort.
MB	INTEGER	Hjelpevariabel ved innlesning fra kort.
MF	INTEGER	Hjelpevariabel ved innlesning fra kort.
L	INTEGER	Hjelpevariabel (innlesning fra kort og andre formål).
LTONI	LOGICAL	Viser (TRUE) om I/O-enhet nr. 29 er definert for direkte aksessbruk (mellomlagringsfil i OMSKR).
LFIRE	LOGICAL	Viser (TRUE) om I/O-enhet nr. 4 har vært brukt. (Discfil brukt ved innlesning av Marsdenrutenumre og tilhørende rekkefølge).
LSERT	INTEGER	Teller (peker til) linjenummer i IMSQ ved seriebehandling i hovedrutinene.
LMSLA	INTEGER	Mellomlagrer LMS (se innleste variable) ved seriebehandling i hovedrutinene.
LMFLA	INTEGER	Mellomlagrer LMF (se innleste variable) ved seriebehandling i hovedrutinene.
NTELL(100)	INTEGER	Tabell over antall stasjoner pr. engradersrute.
JL	INTEGER	Hjelpevariabel i KOPI. Viser startadresse for den aktuelle halvdel av bufferet.
JS	INTEGER	Som JL.
NW	INTEGER	Blokk lengde (i KOPI).
JH	INTEGER	Hjelpevariabel.

NAVN:	TYPE:	BRUK:
LSTAT	INTEGER	Statusvariabel i NTRAN (skrivning i KOPI).
NAVN	INTEGER	Inneholder filnavn fra rutinen RNAVN.
LREC	INTEGER	Postlengde på kopierte filer.
IMHJ	INTEGER	Hjelpevariabel.
KBT	INTEGER	Teller antall blokker som skrives ut på et bånd.
KBMT	INTEGER	Teller antall blokker som skrives ut på et bånd (måledata).
KBIT	INTEGER	Teller antall blokker som skrives ut på et bånd (innholdsregisterfiler).
KB	INTEGER	Teller antall blokker.
KBT	INTEGER	Teller antall blokker (innholdsregisterfiler).
KBM	INTEGER	Teller antall blokker (måledatafiler).
IG	INTEGER	Hjelpevariabel.
I	INTEGER	Hjelpevariabel.
IADA	INTEGER	Postpeker i D. A. fil (I/O-enhet nr. 29).
IASMH	INTEGER	Hjelpevariabel (ved definisjon av D. A. fil).
IXX	INTEGER	Hjelpevariabel (ved definisjon av D. A. fil).
II	INTEGER	Hjelpevariabel.
J	INTEGER	Hjelpevariabel.
LL	INTEGER	Hjelpevariabel.
MRFMAX	INTEGER	Hjelpevariabel.
A	LOGICAL	Hjelpevariabel.
B	LOGICAL	Hjelpevariabel.
C	LOGICAL	Hjelpevariabel.
UTDISC	LOGICAL	Hjelpevariabel.

```
000001 009 C
000002 009 C
000003 009 C
000004 009 C
000005 009 C
000006 009 C
000007 009 C
000008 009 C
000009 009 C
000010 009 C
000011 009 C
000012 009 C
000013 009 C
000014 009 C
000015 009 C
000016 009 C
000017 009 C
000018 009 C
000019 009 C
000020 009 C
000021 009 C
000022 009 C
000023 009 C
000024 009 C
000025 009 C
000026 009 C
000027 009 C
000028 009 C
000029 009 C
000030 009 C
000031 009 C
000032 009 C
000033 009 C
000034 009 C
000035 009 C
000036 009 C
000037 009 C
000038 009 C
000039 009 C
000040 009 C
000041 009 C
000042 009 C

*****
*
*   STYREPROGRAM   *
*
*****

DIMENSION IDEV(3,12),IMSQ(3,30)
LOGICAL LDEV(3,12),LMSQ(30)
DIMENSION ITDSI(6),ITDSH(6)
DIMENSION NTELL(100),NDEFS0(100)
COMMON/DATA/ INN(2800)
LOGICAL TILYAA
COMMON/RDATA/MSGTAB(72),LBT,LMSGT
COMMON/CATT/TILYAA
DIMENSION NYPAR(7)
DIMENSION IRPEK(9)
DATA IRPEK/42HCONT DMIX IMIX KOPI MHSORTOMSKR SORTER,
  12HSTART STOP /
DIMENSION IGOTO(9)
LOGICAL LTONI/,FALSE./
LOGICAL LFIRE/,FALSE./
LOGICAL TOUT
LOGICAL A,B,C,COPD2
LOGICAL LOSP,UTDISC,VDA,LHSORT,KOPG,BEGGE
LOGICAL LTMLCO,LTISCO,LTMSCO,LTILSO,LTISSO,LTILMH ,LTISMH,LTILOM
LOGICAL LTMLOM,LTISOM,LTMSOM,LTILAI,LTILBI,LTISIM,LTMLAD,LTMLBD
LOGICAL LTMSDM,LDMLCO,LDILSO,LDILMH,LDILOM,LDMLOM,LDILAI,LDILBI
LOGICAL SLMLAD,SLMLBD,SLMLCO,SLILSO,SLILMH,SLILOM,SLMLOM,SLILAI
LOGICAL SLILBI,SLMLAD,SLMLBD
LOGICAL OVSSK,OVSSK
LOGICAL LTLKO,LTSKO,LDLKO,SLLKO
FORMAT(3R1,A6,1X,7A5)

10
C
C   INITIALISERING AV MONTERINGS OG OPPDATERINGSRUTINER.
C
TILYAA=,FALSE.
CALL MONTIN(270)
READ(5,13) L,MRUT,A,KOPG
IF (MRUT.NE.6HNY3 ) GO TO 375
```

```
000043 009 C
000044 009 C
000045 009 C
000046 009 C
000047 009 C
000048 009 C
000049 009 C
000050 009 C
000051 011 C
000052 009 C
000053 009 C
000054 009 C
000055 009 C
000056 009 C
000057 009 C
000058 009 C
000059 009 C
000060 009 C
000061 009 C
000062 009 C
000063 009 C
000064 009 C
000065 009 C
000066 009 C
000067 009 C
000068 009 C
000069 009 C
000070 009 C
000071 009 C
000072 009 C
000073 009 C
000074 009 C
000075 009 C
000076 009 C
000077 009 C
000078 009 C
000079 009 C
000080 009 C
000081 009 C
000082 009 C
000083 009 C
000084 009 C
000085 011 C
000086 011 C

IF (A) CALL NY3(KOPG)
READ(5,13) L,MRUT,COPD2
IF (MRUT.NE.6HOPPD2 ) GO TO 375
IF (COPD2) CALL OPPD2
CONTINUE
12
FORMAT(13,A6,1X,315,A1,3L1)
13
FORMAT(13,A6,1X,3L1)
C
C   INNLESING AV HOVEDTABELLER.
C
DO 130 I=1,13
READ(5,12,END=140) L,MRUT,MT,MB,MF,J,A,B,C
IF (J.NE.1H*) GO TO 375
IF (L.LT.1) GO TO 375
IF (L.GT.12) GO TO 375
IF (MRUT.NE.6HDEV ) CALL ERTRAN(2)
IDEV(1,L)=MT
LDEV(2,L)=MB
IDEV(3,L)=MF
LDEV(1,L)=A
LDEV(2,L)=B
LDEV(3,L)=C
IF(.NOT.A) GO TO 130
C
C   OPPRETNING AV KLADDEFIL.
C
CALL FREEA(MT,TRUE,.)
CALL ASGUP(MT,MT,TRUE,.)
LDEV(2,L)=,FALSE.
LDEV(3,L)=,FALSE.
130
CONTINUE
CALL ERTRAN(2)
140
CONTINUE
WRITE(6,14)
14
FORMAT(//13X,' I/O-NR. TAPENR. FILNR. KLADD TAPE TEMP'/)
15
FCRMT(4X,'NR.',12,' '); ',318,3L6)
DO 165 I=1,12
IF (IDEV(1,I).EQ.0) GO TO 165
WRITE(6,15) I,(IDEV(MT,I),MT=1,3),(LDEV(MB,I),MB=1,3)
165
CONTINUE
C
C   INNLESING AV MARSDENRUTENUMMER M.M.
C   MARSDENNUMMERET OG REKKEFØLGEN KAN LESES ENTEN
C   FRA KORT ELLER FRA EN DISCFIL (I/O-ENHET 4).
C
```



```

000087 009 C
000088 009 READ(5,12,END=300) L,MRUT,MT,MB
000089 009 MF=0
000090 009 IF (MRUT.EQ.6HIMSQR) MF=5 6MARSDENRUTENUMMER FRA KORT.
000091 009 IF (MRUT.EQ.6HIMSQDF) MF=4 6MARSDENRUTENUMMER FRA DISCFIL.
000092 009 IF (MF.EQ.0) GO TO 300
000093 009 IF ((.NOT.LFIRE).AND.(MF.EQ.4)) REWIND 4
000094 009 IF ((.NOT.LFIRE).AND.(MF.EQ.4)) LFIRE=.TRUE.
000095 009 16 FORMAT(13,L1)
000096 009 17 FORMAT(4X,12,' ): ','MARSDENRUTE NR:',13,' REKKE=',L1)
000097 009 18 FORMAT(//4X,'RUTETABELL;NAVNMODUS INNHOLD=',12,' ,NAVNMODUS',
000098 009 * MALEDATA=',12/)
000099 009 WRITE(6,18) MT,MB
000100 009 DO 170 I=1,31
000101 009 READ(MF,16,END=300) IMSQ(1,I),LMSQ(1)
000102 009 IMSQ(2,I)=MT
000103 009 IMSQ(3,I)=MB
000104 009 WRITE(6,17) I,IMSQ(1,I),LMSQ(1)
000105 009 170 CONTINUE
000106 012 IF (MF.NE.4) CALL ERTRAN(2)
000107 012 BACKSPACE 4
000108 012 GO TO 300
000109 009 200 CONTINUE
000110 009 C
000111 011 C TILORDNING AV VERDIER TIL DIVERSE STYREVARIABLE.
000112 011 C SPESIELLE FUNKSJONER UTFØRES.
000113 009 C
000114 009 IF (MRUT.NE.6HLMFLMS) GO TO 210
000115 009 20 FORMAT(15)
000116 009 DECODE(20,NYPAR(1)) LMF
000117 009 DECODE(20,NYPAR(2)) LMS
000118 009 11 FORMAT(4X,'BEHANDLING AV OMRÅDE ',12,' T.O.M.',12)
000119 009 WRITE(6,11) LMF,LMS
000120 009 GO TO 300
000121 009 210 CONTINUE
000122 009 IF (MRUT.NE.6HSTART ) GO TO 215
000123 009 IGOTOP=0
000124 009 GO TO 380
000125 009 215 CONTINUE
000126 009 IF (MRUT.NE.6HHOPP ) GO TO 235
000127 011 C
000128 011 C NY BEHANDLINGSREKKEFØLGE.
000129 011 C
000130 009 21 FORMAT(10X,9A6)

000131 011 READ(0,21)IGOTO 6KORTET LESES PANY MED NYTT FORMAT.
000132 009 31 FORMAT(4X,'REKKEFØLGE:',9(2H**,A6))
000133 009 WRITE(6,31) IGOTO
000134 009 DO 230 I=1,9
000135 009 IF (IGOTO(I).EQ.6H ) GO TO 225
000136 009 DO 220 L=1,9
000137 009 IF (IGOTO(I).GT.IRPEK(L)) GO TO 220
000138 009 IF (IGOTO(I).LT.IRPEK(L)) GO TO 375
000139 009 IGOTO(I)=L
000140 009 GO TO 230
000141 009 220 CONTINUE
000142 009 GO TO 375
000143 009 225 CONTINUE
000144 009 IGOTO(I)=0
000145 009 230 CONTINUE
000146 009 GO TO 300
000147 009 235 CONTINUE
000148 009 IF (MRUT.EQ.6HFJERN) GO TO 265
000149 009 IF (MRUT.EQ.6HFJERN ) GO TO 270
000150 009 IF (MRUT.EQ.6HLOSP ) CALL NYLOGI(LOSP,4HLOSP,1300)
000151 009 IF (MRUT.EQ.6HLHSORT) CALL NYLOGI(LHSORT,6HLHSORT,1300)
000152 009 IF (MRUT.EQ.6HTOUT ) CALL NYLOGI(TOUT,4HTOUT,1300)
000153 009 IF (MRUT.EQ.6HBEGGE ) CALL NYLOGI(BEGGE,5HBEGGE,1300)
000154 009 IF (MRUT.EQ.6HVDA ) CALL NYLOGI(VDA,3HVDA,1240)
000155 009 IF (MRUT.EQ.6HMOP3CO) CALL NYINT(MOP3CO,6HMOP3CO,1300)
000156 009 IF (MRUT.EQ.6HMNTABS ) CALL NYINT(MNTABS ,6HMNTABS ,1300)
000157 009 IF (MRUT.EQ.6HMNTABL ) CALL NYINT(MNTABL ,6HMNTABL ,1300)
000158 009 IF (MRUT.EQ.6HMIXTAB) CALL NYINT(MIXTAB,6HMIXTAB,1300)
000159 009 IF (MRUT.EQ.6HNUNDA ) CALL NYINT(NUNDA ,6HNUNDA ,1245)
000160 009 IF (MRUT.EQ.6HNAILAI) CALL NYINT(NAILAI,6HNAILAI,1300)
000161 009 IF (MRUT.EQ.6HNAMLAD) CALL NYINT(NAMLAD,6HNAMLAD,1300)
000162 009 IF (MRUT.EQ.6HITDSI ) CALL NYSKRB(ITDSI,5HITDSI,1300,1STNSK,1SFNSK
000163 009 * ,OVISSK)
000164 009 IF (MRUT.EQ.6HITDSM ) CALL NYSKRB(ITDSM,5HITDSM,1300,MSTNSK,MSFNSK
000165 009 * ,OVMSSK)
000166 009 GO TO 250
000167 009 240 CONTINUE
000168 009 IF (LTONT) GO TO 300
000169 009 IF (.NOT.VDA) GO TO 300
000170 010 C
000171 010 C NY,TEMPORAR,FIL (1/0=ENHET NR,29) OPPRETTES SOM
000172 010 C HELLOMLAGRINGSFIL FOR HOVEDRUTINEN OMSKR.
000173 010 C
000174 009 LTONT=.TRUE.

```

```
000175 009 CALL FREF(29,1)
000176 009 CALL ASGUP(29,29,.TRUE.)
000177 010 DECODE(20,NYPAR(2)) I 6MAKSIMAL FILSTØRRELSE (I POSTER).
000178 012 IF (1.EQ.0) I=5000 6MAKSIMAL FILSTØRRELSE UOPPGITT.
000179 009 DEFINE FILE 29(1,14,U,1ADA)
000180 009 GO TO 300
000181 009 245 CONTINUE
000182 010 C
000183 010 C NY,TEMPORÆR,FIL (I/O-ENHET NUNDA) OPPRETTES
000184 010 C (MELLOMLAGRINGSFIL I MHSORT).
000185 010 C
000186 009 CALL FREE(NUNDA,1)
000187 009 CALL ASGUP(NUNDA,NUNDA,.TRUE.)
000188 010 DECODE(20,NYPAR(2)) I 6MAKSIMAL FILSTØRRELSE (I POSTER).
000189 012 IF (1.EQ.0) I=1000 6MAKSIMAL FILSTØRRELSE UOPPGITT.
000190 009 DEFINE FILE NUNDA(1,10,U,IASMH)
000191 009 GO TO 300
000192 009 250 CONTINUE
000193 009 IF (MRUT.NE.6HNDEF50) GO TO 375
000194 010 C
000195 010 C REKKEFØLGE FOR ENGRADERSRUTER (VED GEOGRAFISK SORTERING)
000196 010 C LESES INN (FIRE KORT).
000197 010 C
000198 009 L=1
000199 009 LL=25
000200 009 23 FORMAT(25I3)
000201 009 24 FORMAT(4X,A6,' '); RUTE ',I2,'-',I3,' ':'',25I3)
000202 009 DO 260 I=1,4
000203 009 READ(5,23) (NDEF50(I),I=L,LL)
000204 009 WRITE(6,24) MRUT,L,LL,(NDEF50(I),I=L,LL)
000205 009 L=L+25
000206 009 LL=LL+25
000207 009 260 CONTINUE
000208 009 GO TO 300
000209 009 265 CONTINUE
000210 010 C
000211 012 C ET MAGNETBAND DEMONTERES.MAGNETBANDSTASJONEN RESERVERES
000212 012 C FOR FORTSATT BRUK.
000213 010 C
000214 010 DECODE(20,NYPAR(1)) I 6LINJE I IDEV.
000215 009 CALL FREES(IDEV(1,I))
000216 009 27 FORMAT(1H1,'ADVARSEL:MAGNETBAND ER IKKE TILORDNET ',
000217 009 *'I/O-ENHET ',I2//////////)
000218 009 IF(.NOT.LDEV(2,I)) WRITE(6,27) IDEV(1,I)

000219 009 IF(.NOT.LDEV(2,I)) GO TO 300
000220 009 IDEV(2,I)=0
000221 009 IDEV(3,I)=0
000222 009 GO TO 300
000223 009 270 CONTINUE
000224 010 C
000225 010 C EN FIL FRIGJØRES.
000226 010 C
000227 009 DECODE(20,NYPAR(1)) I
000228 009 CALL FREE(IDEV(1,I),I)
000229 009 IF(.NOT.LDEV(2,I)) WRITE(6,27) IDEV(1,I)
000230 009 IF(.NOT.LDEV(2,I)) GO TO 300
000231 009 IDEV(2,I)=0
000232 009 IDEV(3,I)=0
000233 009 GO TO 300
000234 009 300 CONTINUE
000235 009 C
000236 009 C INNLESING OG DEKODING AV STYRELVARIABLE OG
000237 009 C FILSPESIFIKASJONSVARIABLE.
000238 009 C
000239 009 READ(5,10,END=100) MT,MB,MF,MRUT,NYPAR
000240 009 DO 305 I=1,7
000241 009 IF (MRUT.GT.IRPEK(I)) GO TO 305
000242 009 IF (MRUT.LT.IRPEK(I)) GO TO 200
000243 009 GO TO 310
000244 009 305 CONTINUE
000245 009 GO TO 200
000246 009 310 CONTINUE
000247 009 IF (MT.EQ.14) GO TO 315
000248 009 IF (MT.EQ.18) GO TO 335
000249 009 CALL ERTRAN(2)
000250 009 315 CONTINUE
000251 009 IF (MB.EQ.24) GO TO 325
000252 009 IF (MB.NE.17) CALL ERTRAN(2)
000253 009 GO TO (375,375,316,317,318,319,320),I
000254 009 316 CONTINUE
000255 009 IF (MF.EQ.6) CALL LESLES(ILAI,LYLAI,LDILAI,SLILAI,ILRAI,1300)
000256 009 IF (MF.EQ.7) CALL LESLES(ILBI,LYLBI,LDILBI,SLILBI,ILRBI,1300)
000257 009 GO TO 375
000258 009 317 CALL LESLES(KOL,LYLKO,LDLKO,SLLKO,KCLR,1300)
000259 009 318 CALL LESLES(ILMH,LYLMH,LDILMH,SLILMH,ILRMH,1300)
000260 009 319 CALL LESLES(ILOM,LYLOM,LDILOM,SLILOM,ILROM,1300)
000261 009 320 CALL LESLES(ILSO,LYLSO,LDILSO,SLILSO,ILRSO,1300)
000262 009 325 CONTINUE
```

```
000263 009 326 GO TO (326,375,327,328,329,330,331),I
000264 009 326 CALL LESSKR(ISCO,LTISCO,ISRCO,NAISCO,1300)
000265 009 327 CALL LESSKR(ISIM,LTISIM,ISRIM,NAISIM,1300)
000266 009 328 CALL LESSKR(KOS,LTSKO,KOSR,NASKO,1300)
000267 009 329 CALL LESSKR(ISMH,LTISMH,ISRMH,NAISMH,1300)
000268 009 330 CALL LESSKR(ISOM,LTISOM,ISRSM,NAISOM,1300)
000269 009 331 CALL LESSKR(ISSO,LTISSO,ISRSO,NAISSO,1300)
000270 009 335 CONTINUE
000271 009 IF (MB.EQ.24) GO TO 340
000272 009 IF (MB.EE.17) GO TO 375
000273 009 GO TO (336,337,375,317,375,338,375),I
000274 009 336 CALL LESLES(MLCO,LTMLCO,LDMLCO,SLMLCO,MLRCO,1300)
000275 009 337 CONTINUE
000276 009 IF (MF.EQ.6) CALL LESLES(MLAD,LTMLAD,LDMLAD,SLMLAD,MLRAD,1300)
000277 009 IF (MF.EQ.7) CALL LESLES(MLBD,LTMLBD,LDMLBD,SLMLBD,MLRBD,1300)
000278 009 GO TO 375
000279 009 338 CALL LESLES(MLOM,LTMLOM,LDMLOM,SLMLOM,MLROM,1300)
000280 009 340 CONTINUE
000281 009 GO TO (341,342,375,328,375,343,375),I
000282 009 341 CALL LESSKR(MSCO,LTMSCO,MSRCO,NAMSCO,1300)
000283 009 342 CALL LESSKR(MSDM,LTMSDM,MSRDM,NAMSDM,1300)
000284 009 343 CALL LESSKR(MSOM,LTMSOM,MSROM,NAMSON,1300)
000285 009 375 CONTINUE
000286 009 C
000287 009 C FEILAVBRUDD.
000288 009 C
000289 009 25 FORMAT(1H1,5X,'** FEIL **',3R1,A6,1X,7A5)
000290 009 WRITE(6,25) MT,MB,MF,MRUT,NYPAR
000291 009 CALL ERTRAN(2)
000292 009 380 CONTINUE
000293 009 C
000294 009 C START DATABEARBEIDING.
000295 009 C
000296 009 IF (NYPAR(1).NE.5HSERIE) GO TO 388
000297 010 C
000298 010 C SERIEBEHANDLING.
000299 010 C
000300 009 IF (IGOTOP.GT.0) GO TO 385
000301 012 LMF=LMF 6BEHANDLINGSMRADENES
000302 012 LMSLA=LMS 6ADRESSER LAGRES.
000303 012 LSERT=LMF-1 6TELLER
000304 009 385 CONTINUE
000305 009 LSERT=LSERT+1
000306 009 IF (LSERT.GT.LMSLA) GO TO 386
```

```
000307 009 IGOTOP=0
000308 009 LMF=LSERT
000309 009 LMS=LSERT
000310 009 WRITE(6,11) LMF,LMS
000311 009 GO TO 389
000312 009 386 CONTINUE
000313 009 C
000314 009 C FERDIG MED SERIEBEHANDLING.LMF OG LMS FAR TILBAKE
000315 009 C SINE GAMLE VERDIER.
000316 012 C
000317 009 LMF=LMFLA
000318 009 LMS=LMSLA
000319 009 GO TO 300
000320 009 388 CONTINUE
000321 009 C
000322 009 C PARALLELLBEHANDLING.
000323 009 C
000324 009 IF (IGOTOP.GT.0) GO TO 300
000325 009 389 CONTINUE
000326 009 CALL NESTE(1390,1640,1590,1690,1490,1520,1440,1375,1900)
000327 009 390 CONTINUE
000328 009 C
000329 009 C CONT
000330 009 C
000331 009 IF (LTISCO)CALL TBAAND(IDEV(2,ISCO),IDEV(3,ISCO),IDEV(1,ISCO)
000332 009 +,LDEV(2,ISCO),ISTNSK,ISFNSK,OVSSK,ITDS1)
000333 009 IF (LTMSCO.AND.TOUT) CALL TBAAND(IDEV(2,MSCO),IDEV(3,MSCO),
000334 009 +IDEV(1,MSCO),LDEV(2,MSCO),MSTNSK,MSFNSK,OVMSK,ITDSM)
000335 009 KBMT = C
000336 009 KBIT = C
000337 009 DO 400 I=LMF,LMS
000338 009 CALL MOLES(MTABL,MLCO,I,LTMLCO,LDMLCO,MLRCO,IMSQ(3,I))
000339 009 IF (.NOT.LTISCO) CALL MONSKR(ISCO,I,ISRCO,NAISCO,2)
000340 009 IF ((.NOT.LTMSCO).AND.TOUT) CALL MONSKR(MSCO,I,MSRCO,NAMSCO,3)
000341 009 CALL CONT(IDEV(1,MLCO),IDEV(1,ISCO),IDEV(3,MLCO),NTELL,IMSQ
000342 009 + (1,I),KBM,KBI,TOUT,IDEV(1,MSCO),LMSQ(1))
000343 009 IF (SLMLCO.AND.(.NOT.LDEV(2,MLCO))) CALL FREE(IDEV(1,MLCO),I)
000344 009 CALL OPFD3(IMSQ(1,I),L,NTPELL,HOP3CO,BEGGE)
000345 009 IF (LTISCO) CALL SKRB(ISCO,I,ISRCO)
000346 009 IF (LTMSCO.AND.TOUT) CALL SKRB(MSCO,I,MSRCO)
000347 009 KBMT = KBMT + KBM
000348 009 KBIT = KBIT + KBI
000349 009 400 CONTINUE
000350 009 IF (LTISCO) CALL SKRBSL(ISCO,ISFNSK,KBIT)
```

```

000351 009 IF(LTMSCO.AND.TCUT) CALL SKRBSL(MSCG,MSFNK,KBT)
000352 009 CALL NESTE(1375,1640,1590,1690,1490,1375,1440,1380,1900)
000353 009 440 CONTINUE
000354 009 C
000355 009 C SORTER
000356 009 C
000357 009 IF(LTISSO) CALL TBAAND(IDEV(2,ISSO),IDEV(3,ISSO),IDEV(1,ISSO),LDEV
000358 009 *(2,ISSO),ISTNSK,ISFNK,OVISSK,ITDS1)
000359 009 KBT=0
000360 009 DO 450 I=LMF,LMS
000361 009 CALL MOLES(MTABL,ILSO,I,LTILSO,LDILSO,ILRSO,IMSQ(2,I))
000362 009 CALL HKLA(ILSO,ISSO,I,LTISSO,10,KB,1450,1445)
000363 009 IF(.NOT.LTISSO) CALL MONSKR(ISSO,I,ISRSO,NAISSO,2)
000364 009 UTDISC = .NOT.LTISSO
000365 009 CALL SORTER(IDEV(1,ILSO),IDEV(1,ISSO),IDEV(3,ISSO),LOSP,UTDISC,
000366 009 *LMSQ(1),KB)
000367 009 445 CONTINUE
000368 009 IF(SLILSO.AND.(.NOT.LDEV(2,ILSO))) CALL FREE(IDEV(1,ILSO),I)
000369 009 IF(LTISSO) CALL SKRB(ISSO,I,ISRSO)
000370 009 KBT=KBT+KB
000371 009 450 CONTINUE
000372 009 IF(LTISSO) CALL SKRBSL(ISSO,ISFNK,KBT)
000373 009 CALL NESTE(1375,1640,1590,1690,1490,1520,1375,1380,1900)
000374 009 490 CONTINUE
000375 009 C
000376 009 C MHSORT
000377 009 C
000378 009 IF(LTISMH) CALL TBAAND(IDEV(2,ISMH),IDEV(3,ISMH),IDEV(1,ISMH),LDEV
000379 009 *(2,ISMH),ISTNSK,ISFNK,OVISSK,ITDS1)
000380 009 KBT=0
000381 009 DO 500 I=LMF,LMS
000382 009 CALL MOLES(MTABL,ILMH,I,LTILMH,LDILMH,ILRMH,IMSQ(2,I))
000383 009 IF(.NOT.LTISMH) CALL MONSKR(ISMH,I,ISRMH,NAISMH,2)
000384 012 C
000385 013 C ENGRADERSRUTEFORTEGNEELSE FOR FILEN MA LESES INN.
000386 012 C (FRA I/O-ENHET 28).
000387 012 C
000388 009 IF (.NOT.TILTAA) DEFINE FILE 28(72,100,U,1XX)
000389 009 IF (.NOT.TILTAA) TILTAA=.TRUE.
000390 009 DO 495 L=LMSQT,1,-1
000391 009 IF (IMSQ(1,1).NE.MSQTAB(L)) GO TO 495
000392 009 GO TO 496
000393 009 495 CONTINUE
000394 009 CALL ERTRAN(2)

```

```

000395 009 496 CONTINUE
000396 009 READ(28*L) NTELL @ENGRADERSRUTEFORTEGNEELSE LESES INN.
000397 009 CALL MHSORT(IDEV(1,ILMH),IDEV(1,ISMH),NUNDA,NTELL,NDEFSO,IDEV(3,
000398 009 *ILMH),KB,LDEV(2,ILMH))
000399 009 IF(SLILMH.AND.(.NOT.LDEV(2,ILMH))) CALL FREE(IDEV(1,ILMH),I)
000400 009 IF(LTISMH) CALL SKRB(ISMH,I,ISRMH)
000401 009 KBT = KBT + KB
000402 009 500 CONTINUE
000403 009 IF(LTISMH) CALL SKRBSL(ISMH,ISFNK,KBT)
000404 009 CALL NESTE(1375,1640,1590,1690,1375,1520,1375,1380,1900)
000405 009 520 CONTINUE
000406 009 C
000407 009 C OMSKR
000408 009 C
000409 009 IF(LTISOM)CALL TBAAND(IDEV(2,ISOM),IDEV(3,ISOM),IDEV(1,ISOM)
000410 009 *,LDEV(2,ISOM),ISTNSK,ISFNK,OVISSK,ITDS1)
000411 009 IF(LTMSOM) CALL TBAAND(IDEV(2,MSOM),IDEV(3,MSOM),IDEV(1,MSOM
000412 009 *,LDEV(2,MSOM),MSTNSK,MSFNK,OVMSK,ITDSM)
000413 009 KBMT = 0
000414 009 KBIT = 0
000415 009 DO 550 I=LMF,LMS
000416 009 CALL MOLES(MTABL,MLOM,I,LTILOM,LDILOM,MLROM,IMSQ(3,I))
000417 009 CALL MOLES(MTABL,ILOM,I,LTILOM,LDILOM,ILROM,IMSQ(2,I))
000418 009 KBI=0
000419 009 KBM=0
000420 009 CALL HKLA(ILOM,ISOM,I,LTISOM,10,KBI,1540,1535)
000421 009 IF(.NOT.LTISOM) CALL MONSKR(ISOM,I,ISROM,NAISOM,2)
000422 009 IF(.NOT.LTMSOM) CALL MONSKR(MSOM,I,MSROM,NAMSON,3)
000423 009 CALL OMSKR(IDEV(1,ILOM),IDEV(1,MLOM),IDEV(1,ISOM),IDEV(1,MSOM),
000424 009 *KBI,KBM,LDEV(2,MLOM),VDA,IDEV(3,MLOM),IADA)
000425 009 535 CONTINUE
000426 009 IF(SLILCM.AND.(.NOT.LDEV(2,ILOM))) CALL FREE(IDEV(1,ILOM),I)
000427 009 540 CONTINUE
000428 009 CALL HKLA(MLOM,MSOM,I,LTMSOM,14,KBM,1545,1544)
000429 009 544 CONTINUE
000430 009 IF(SLMLCM.AND.(.NOT.LDEV(2,MLOM))) CALL FREE(IDEV(1,MLOM),I)
000431 009 545 CONTINUE
000432 009 IF(LTISOM) CALL SKRB(ISOM,I,ISROM)
000433 009 IF(LTMSOM) CALL SKRB(MSOM,I,MSROM)
000434 009 KBMT = KBMT + KBM
000435 009 KBIT = KBIT + KBI
000436 009 550 CONTINUE
000437 009 IF(LTISOM) CALL SKRBSL(ISOM,ISFNK,KBIT)
000438 009 IF(LTMSOM) CALL SKRBSL(MSOM,MSFNK,KBMT)

```

```

000439 009 CALL NESTE(1375,1640,1590,1690,1375,1375,1375,1380,1900)
000440 009 590 CONTINUE
000441 009 C
000442 009 C IMIX
000443 009 C
000444 009 IF(LTISIM) CALL TBAAND(IDEV(2,ISIM),IDEV(3,ISIM),IDEV(1,ISIM),LDEV
000445 009 * (2,ISIM),ISTNSK,ISFNSK,OVISSK,ITDSI)
000446 009 KBT=0
000447 009 DO 600 I=LMF,LMS
000448 009 CALL MOLES(MIXTAB,ILAI,I,LTILAI,LDILAI,ILRAI,NAILAI)
000449 009 CALL MOLES(MTABL,ILBI,I,LTILBI,LDILBI,ILRBI,IMSQ(2,I))
000450 009 IF(.NOT.LTISIM) CALL MONSKR(ISIM,I,ISRIM,NAISIM,2)
000451 009 CALL IMIX(IDEV(1,ILAI),IDEV(1,ILBI),IDEV(3,ILAI),IDEV(3,ILBI),
000452 009 * IDEV(1,ISIM),LHSORT,KB,NTELL,NDEFSO)
000453 009 CALL OPPD3(IMSQ(1,I),L,NTELL,1,BEGGE)
000454 009 IF(SLILAI.AND.(.NOT.LDEV(2,ILAI))) CALL FREE(IDEV(1,ILAI),I)
000455 009 IF(SLILBI.AND.(.NOT.LDEV(2,ILBI))) CALL FREE(IDEV(1,ILBI),I)
000456 009 IF(LTISIM) CALL SKRB(ISIM,I,ISRIM)
000457 009 KBT=KBT+KB
000458 009 600 CONTINUE
000459 009 IF(LTISIM) CALL SKRBSL(ISIM,ISFNSK,KBT)
000460 009 CALL NESTE(1375,1640,1375,1690,1375,1375,1375,1380,1900)
000461 009 640 CONTINUE
000462 009 C
000463 009 C DMIX
000464 009 C
000465 009 IF(LTMSDM) CALL TBAAND(IDEV(2,MSDM),IDEV(3,MSDM),IDEV(1,MSDM),LDEV
000466 009 * (2,MSDM),MSTNSK,MSFNSK,OVMSSK,ITDSM)
000467 009 KBT=0
000468 009 DO 650 I=LMF,LMS
000469 009 CALL MOLES(MIXTAB,MLAD,I,LTMLAD,LDMLAD,MLRAD,NAMLAD)
000470 009 CALL MOLES(MTABL,MLBD,I,LTMLBD,LDMLBD,MLRBD,IMSQ(3,I))
000471 009 IF(.NOT.LTMSDM) CALL MONSKR(MSDM,I,MSRDM,NAMSDM,3)
000472 009 CALL DMIX(IDEV(1,MLAD),IDEV(1,MLBD),IDEV(3,MLAD),IDEV(3,MLBD),
000473 009 * IDEV(1,MSDM),LHSORT,IMSQ(1,I),KB,NDEFSO)
000474 009 IF(SLMLAD.AND.(.NOT.LDEV(2,MLAD))) CALL FREE(IDEV(1,MLAD),I)
000475 009 IF(SLMLBD.AND.(.NOT.LDEV(2,MLBD))) CALL FREE(IDEV(1,MLBD),I)
000476 009 IF(LTMSDM) CALL SKRB(MSDM,I,MSRDM)
000477 009 KBT=KBT+KB
000478 009 650 CONTINUE
000479 009 IF(LTMSDM) CALL SKRBSL(MSDM,MSFNSK,KBT)
000480 009 CALL NESTE(1375,1375,1590,1690,1375,1375,1375,1380,1900)
000481 009 690 CONTINUE
000482 009 C

000483 009 C KOPIERING
000484 009 C
000485 009 IMHJ=2
000486 009 J=KOLR/2
000487 009 J=2*J
000488 009 IF (J.EQ.KOLR) IMHJ=3
000489 009 LREC=10
000490 009 IF (IMHJ.EQ.3) LREC=14
000491 009 IF(LTSKO.AND.(IMHJ.EQ.2)) CALL TBAAND(IDEV(2,KOS),IDEV(3,KOS),
000492 009 * IDEV(1,KOS),LDEV(2,KOS),ISTNSK,ISFNSK,OVISSK,ITDSI)
000493 009 IF(LTSKO.AND.(IMHJ.EQ.3)) CALL TBAAND(IDEV(2,KOS),IDEV(3,KOS),
000494 009 * IDEV(1,KOS),LDEV(2,KOS),MSTNSK,MSFNSK,OVMSSK,ITDSM)
000495 009 KBT=0
000496 009 DO 700 I=LMF,LMS
000497 009 CALL MOLES(MTABL,KOL,I,LTLKO,LDLKO,KOLR,IMSQ(IMHJ,I))
000498 009 IF(.NOT.LTSKO) CALL MONSKR(KOS,I,KOLR,NASKO,IMHJ)
000499 009 CALL KOPI(IDEV(1,KOL),IDEV(1,KOS),INN,LREC,KB,1695)
000500 009 695 CONTINUE
000501 009 IF(SLLKO.AND.(.NOT.LDEV(2,KOL))) CALL FREE(IDEV(1,KOL),I)
000502 009 IF(LTSKO) CALL SKRB(KOS,I,KOSR)
000503 009 KBT=KBT+KB
000504 009 700 CONTINUE
000505 009 IF(LTSKO.AND.(IMHJ.EQ.2)) CALL SKRBSL(KOS,ISFNSK,KBT)
000506 009 IF(LTSKO.AND.(IMHJ.EQ.3)) CALL SKRBSL(KOS,MSFNSK,KBT)
000507 009 CALL NESTE(1390,1640,1590,1690,1490,1520,1440,1380,1900)
000508 009 900 CONTINUE
000509 009 C
000510 009 C FERDIG.
000511 009 C
000512 009 IF (COPD2) CALL SLUTT2
000513 009 C
000514 009 SUBROUTINE MONSKR(LIDEV,LIM,IREG,NMOD,IM)
000515 009 C
000516 009 C OPPRETTER OG KLARGJØR DISCFILER FOR SKRIVING.
000517 009 C
000518 009 IF (.NOT.LDEV(1,LIDEV)) GO TO 200 6KLADEFIL UTELUKKET?
000519 012 IF (LDEV(3,LIDEV)) GO TO 190 6KLADEFIL KLAR TIL BRUK?
000520 012 CALL FREEA(IDEV(1,LIDEV),.TRUE.) 6FRIGJØR EVENTUELL ANNEN FIL
000521 012 6FRA 1,0-NUMMERET.
000522 009 LDEV(2,LIDEV)=.FALSE,
000523 009 LDEV(3,LIDEV)=.TRUE.
000524 009 190 CONTINUE
000525 009 CALL NTRAN(IDEV(1,LIDEV),10,22) 6KLADEFIL.
000526 009 RETURN

```

```
000527 009 200 CONTINUE
000528 009 C
000529 009 C ANDRE DISCFILER.
000530 009 C
000531 009 CALL FREEA(IDEV(1,LIDEV),LDEV(3,LIDEV)) &FRIGJØR EKSISTERENDE FIL.
000532 009 CALL RNAVN(IMSQ(1,LIM),IREG,NAVN,NMOD) &FILNAVN.
000533 009 CALL ASGUP(IDEV(1,LIDEV),NAVN,LDEV(3,LIDEV)) &NY FIL.
000534 009 CALL NTRAN(IDEV(1,LIDEV),10,22)
000535 009 LDEV(2,LIDEV) =.FALSE.
000536 009 IMSQ(1M,LIM)=NMOD
000537 009 RETURN
000538 009 C
000539 009 SUBROUTINE SKRB(LIDEV,LIM,IREG)
000540 009 C
000541 009 C SETTER EOF-MERKE OG OPPDATERER MARS DENRUTETABELL OG FILELLER.
000542 009 C
000543 009 IF (COPD2) CALL RNREG(IMSQ(1,LIM),IREG,IDEV(2,LIDEV),IDEV(3,
000544 009 *LIDEV),MTABS)
000545 009 CALL NTRAN(IDEV(1,LIDEV),9,22)
000546 009 IDEV(3,LIDEV)=IDEV(3,LIDEV)+1
000547 009 RETURN
000548 009 C
000549 009 SUBROUTINE SKRBSL(LIDEV,NRFNY,KBLOKK)
000550 009 C
000551 009 C AVSLUTTER (MIDLERTIDIG) SKRIVING PA BAND.
000552 009 C OPPDATERER MAGNETBANDTABELL.
000553 009 C
000554 009 CALL NTRAN(IDEV(1,LIDEV),1,600,INN,L,22) &EN EKSTRA BLOKK.
000555 009 &FOREBYGGER VISSSE FEIL.
000556 009 NRFMAX=IDEV(3,LIDEV)-1
000557 009 IF (COPD2) CALL RNTAPE(IDEV(2,LIDEV),NRFMAX,KBLOKK,0,NRFNY)
000558 009 NRFNY=IDEV(3,LIDEV) &NY STARTADRESSE VED EVENTUELL FORTSATT
000559 009 &SKRIVING PA BANDET
000560 009 RETURN
000561 009 C
000562 009 SUBROUTINE MOLES(MTAB,LIDEV,LIM,LOTAPE,LODISC,IREG,NMOD)
000563 009 C
000564 009 C TILORDNER FILER FOR LESING.
000565 009 C
000566 009 LOGICAL LOTAPE,LODISC
000567 009 IF (.NOT.LDEV(1,LIDEV)) GO TO 200 &KLADDEFIL UTELUKKET?
000568 009 IF (.NOT.LDEV(3,LIDEV)) GO TO 200 &KLADDEFIL TOM?
000569 009 CALL NTRAN(IDEV(1,LIDEV),10,22)
000570 009 RETURN
```

```
000571 009 200 CONTINUE
000572 009 CALL MONTER(IDEV(2,LIDEV),IDEV(3,LIDEV),IDEV(1,LIDEV),IMSQ(1,LIM),
000573 009 *1999,IREG,LDEV(2,LIDEV),NMOD,LOTAPE,LODISC,MTAB)
000574 009 RETURN
000575 009 999 CALL ERTRAN(2)
000576 009 C
000577 009 SUBROUTINE KOPI(NUNIN,NUNUT,NDIM,NRL,KBK,1)
000578 009 C
000579 012 C KOPIERER FILER VED HJELP AV DOBBELBUFFRING.
000580 009 C
000581 009 DIMENSION NDIM(NRL,200)
000582 009 10 FORMAT(4X,'I/O-ENHET ',I2,' KOPIERES TIL I/O-ENHET ',I2)
000583 009 15 FORMAT(4X,I6,' BLOKKER ER KOPIERT')
000584 009 WRITE(6,10) NUNIN,NUNUT
000585 009 Nw=100*NRL
000586 009 KBK=0
000587 009 CALL NTRAN(NUNIN,2,Nw,NDIM,L,22)
000588 009 CALL NTFEIL(Nw,L,NUNIN,1,1999)
000589 009 JL=101
000590 009 JS=1
000591 009 100 CONTINUE
000592 009 JH=JS+99
000593 009 IF (NDIM(1,JH).EQ.0) GO TO 200
000594 009 CALL NTRAN(NUNUT,1,Nw,NDIM(1,JS),LSTAT)
000595 009 CALL NTRAN(NUNIN,2,Nw,NDIM(1,JL),L)
000596 009 IF (LSTAT.NE.Nw) CALL NTRAN(NUNUT,22)
000597 009 KBK=KBK+1
000598 009 CALL NTFEIL(Nw,LSTAT,NUNUT,KBK,1999)
000599 009 IF (L.NE.Nw) CALL NTRAN(NUNIN,22)
000600 009 CALL NTFEIL(Nw,L,NUNIN,KBK+1,1999)
000601 009 JH=JS
000602 009 JS=JL
000603 009 JL=JH
000604 009 GO TO 100
000605 009 200 CONTINUE
000606 009 CALL NTRAN(NUNUT,1,Nw,NDIM(1,JS),LSTAT,22)
000607 009 KBK=KBK+1
000608 009 CALL NTFEIL(Nw,LSTAT,NUNUT,KBK,1999)
000609 009 CALL NTRAN(NUNIN,22)
000610 009 WRITE(6,15) KBK
000611 009 RETURN 6
000612 009 999 CALL ERTRAN(2)
000613 009 C
000614 009 SUBROUTINE LESLES(NYDEV,LOTAPE,LODISC,SLETT,IREG,1)
```

```
000615 009 C
000616 009 C      DEKODER RUTINEAVHENGIGE FILSPESIFIKASJONER FOR LESCIFILER.
000617 009 C
000618 009 LOGICAL LOTAPE,LODISC,SLETT
000619 009 10  FORMAT(4X,'***',3R1,'***',A6,5(5X,A5))
000620 009 11  FORMAT(1H0,5X,'TYPE  NAVN  NYDEV=',I2,'  LOTAPE=',L1,
000621 009 *'  LODISC=',L1,'  SLETT=',L1,'  IREG=',I2)
000622 009 12  FORMAT(15)
000623 009 13  FORMAT(4X,L1)
000624 009      WRITE(6,11)NYDEV,LOTAPE,LODISC,SLETT,IREG
000625 009      WRITE(6,10)MT,MB,MF,MRUT,(NYPAR(I),I=1,5)
000626 009      IF (NYPAR(1),NE,5H      ) DECODE(12,NYPAR(1))NYDEV
000627 009      IF (NYPAR(2),NE,5H      ) DECODE(13,NYPAR(2))LOTAPE
000628 009      IF (NYPAR(3),NE,5H      ) DECODE(13,NYPAR(3))LODISC
000629 009      IF (NYPAR(4),NE,5H      ) DECODE(13,NYPAR(4))SLETT
000630 009      IF (NYPAR(5),NE,5H      ) DECODE(12,NYPAR(5))IREG
000631 009      IF (NYDEV.GT.12) CALL ERTRAN(2)
000632 009      RETURN 6
000633 009 C
000634 009 SUBROUTINE LESSKR(NYDEV,LOTAPE,IREG,NMOD,I)
000635 009 C
000636 009 C      DEKODER RUTINEAVHENGIGE FILSPESIFIKASJONER FOR SKRIVEFILER.
000637 009 C
000638 009 LOGICAL LOTAPE
000639 009 10  FORMAT(1H0,5X,'TYPE  NAVN  NYDEV=',I2,'  LOTAPE=',L1,20X,
000640 009 *'  IREG=',I2,'  NMOD=',I2)
000641 009 11  FORMAT(4X,'***',3R1,'***',A6,2(5X,A5),20X,5(5X,A5))
000642 009 13  FORMAT(15)
000643 009 14  FORMAT(4X,L1)
000644 009      WRITE(6,10)NYDEV,LOTAPE,IREG,NMOD
000645 009      WRITE(6,11) MT,MB,MF,MRUT,NYPAR
000646 009      IF (NYPAR(1),NE,5H      ) DECODE(13,NYPAR(1)) NYDEV
000647 009      IF (NYPAR(2),NE,5H      ) DECODE(14,NYPAR(2)) LOTAPE
000648 009      IF (NYPAR(3),NE,5H      ) DECODE(13,NYPAR(3)) IREG
000649 009      IF (NYPAR(4),NE,5H      ) DECODE(13,NYPAR(4)) NMOD
000650 009      IF (LOTAPE.AND.LDEV(1,NYDEV)) CALL ERTRAN(2)
000651 009      IF (NYDEV.GT.12) CALL ERTRAN(2)
000652 009      RETURN 5
000653 009 C
000654 009 SUBROUTINE NESTE(I,I,I,I,I,I,I,I)
000655 009 C
000656 009 C      GIR NESTE STEG ETTER BEHANDLING I EN HOVEDRUTINE.
000657 009 C
000658 009      IGOTOP=IGOTOP+1

000659 009      IG=IGOTO(IGOTOP)
000660 009      RETURN IG
000661 009 C
000662 009 SUBROUTINE NYSKRB(NDIM,LSKRIV,I,NRTNY,NRFNY,OVER)
000663 009 C
000664 009 C      LESER INN NØDVENDIGE SPESIFIKASJONER FOR MAGNETBAND SOM SKAL
000665 009 C      BRUKES TIL SKRIVING.
000666 009 C
000667 009 DIMENSION NDIM(6)
000668 009 LOGICAL OVER
000669 009 10  FORMAT(10X,6A6)
000670 009 11  FORMAT(4X,A6,'=',6A6)
000671 009      READ(0,10) NDIM
000672 009      WRITE(6,11) LSKRIV,NDIM
000673 009 12  FORMAT(3X,A6,1X,2I5,4X,L1)
000674 009 13  FORMAT(4X,'NRTNY=',I3,'  NRFNY=',I3,'  OVER=',L1)
000675 009      READ(5,12) MRUT,NRTNY,NRFNY,OVER
000676 009      IF(MRUT.NE.6HTAPESK) CALL ERTRAN(2)
000677 009      WRITE(6,13) NRTNY,NRFNY,OVER
000678 009      RETURN 3
000679 009 C
000680 009 SUBROUTINE NYLOGI(BOOL,LSKRIV,I)
000681 009 C
000682 009 C      DEKODER LOGISKE VARIABLE.
000683 009 C
000684 009 LOGICAL BOOL
000685 009 10  FORMAT(4X,A6,'=',L1)
000686 009 11  FORMAT(4X,L1)
000687 009      WRITE(6,10) LSKRIV,BOOL
000688 009      DECODE(11,NYPAR(1)) BOOL
000689 009      WRITE(6,10) LSKRIV,BOOL
000690 009      RETURN 3
000691 009 C
000692 009 SUBROUTINE NYINT(INTEG,LSKRIV,I)
000693 009 C
000694 009 C      DEKODER HELTALLSVARIABLE.
000695 009 C
000696 009 10  FORMAT(4X,A6,'=',I5)
000697 009 11  FORMAT(15)
000698 009      WRITE(6,10) LSKRIV,INTEG
000699 009      DECODE(11,NYPAR(1)) INTEG
000700 009      WRITE(6,10) LSKRIV,INTEG
000701 009      RETURN 3
000702 009 C
```

```
000703 009 SUBROUTINE HKLA(LLEDEV,LSKDEV,LIM,LOTAPE,NRL,KBK,1,1)
000704 009 C
000705 009 C KONTROLLERER OM DATA TILHØRENDE VEDKOMMENDE MARS DENRUTE
000706 009 C ALLEREDE HAR KORREKT REKKEFØLGE, OG BEHANDLER I SA FALL
000707 009 C DISSE (MED TILKNYTTETE VARIABLE).
000708 009 C
000709 009 LOGICAL LOTAPE
000710 009 IF (.NOT.LMSQ(LIM)) RETURN
000711 009 IF (LOTAPE) GO TO 200
000712 009 IF (LDEV(1,LSKDEV)) LDEV(3,LSKDEV)=.FALSE. 6REGISTRERER AT SKRIVEFILEN
000713 012 6(KLADDEFIL) IKKE BLIR BRUKT.
000714 009 RETURN 7
000715 009 200 CONTINUE
000716 009 C
000717 009 C FILEN KOPIERES TIL MAGNETBAND.
000718 009 C
000719 009 CALL KOPI(IDEV(1,LLEDEV),IDEV(1,LSKDEV),INN,NRL,KBK,1300)
000720 009 300 RETURN 8
000721 009 END
```


CONT

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Dato : 31/12-75
Utført av : Ø. Strand
Formål : CONT produserer en innholdsregister-
fil for en måledatafil inneholdende
hydrografiske data (modifisert ICES
standardkode og NOD's fjord-
datakode),
samt registrerer antall stasjoner pr.
engradersrute.

Kall: CONT(NUNIT, NUTU, IF, NTELL, MSQ, K, KUT, TOUT, NUND, REKKE)

Generelt:

CONT produserer en innholdsregisterfil for en gitt måledatafil inneholdende hydrografiske data. Rutinen fortolker følgende typer data: ICES standardkode (med visse begrensninger), modifisert ICES standardkode og NOD's fjorddatakode. Hvis ønskelig, kan måledatafilen samtidig overføres til en annen fil. Rutinen tester også om rekkefølgen i måledatafilen er i overensstemmelse med standard sorteringsorden. Rutinen benytter dobbelbuffering ved innlesing av data. De aktuelle filer må tilordnes/opprettas i det kallende hovedprogram.

Formelle parametre:

NAVN:	TYPE:	BRUK:
NUNIT	INTEGER	I/O-enhetsnummer for måledatafil.
NUTU	INTEGER	I/O-enhetsnummer for innholdsregisterfil.
IF	INTEGER	Filnummer på eventuelt magnetbånd (måledatafil).
NTELL(100)	INTEGER	Fortegnelse over antall stasjoner pr. engradersrute (standard nummerering (0-99)).
MSQ	INTEGER	Marsdenrute-nummer.
K	INTEGER	Antall leste blokker fra måledatafilen.
KUT	INTEGER	Antall blokker i innholdsregisterfilen.
TOUT	LOGICAL	Logisk variabel som viser om måledatafilen skal skrives over til en annen fil.
NUND	INTEGER	I/O-enhetsnummer for eventuell ny måledatafil.
REKKE	LOGICAL	Angir om rekkefølgen i måledatafilen er korrekt. (): i overensstemmelse med standardsorteringsorden).

Eksterne rutiner : NTFEIL
 : LESDB

Interne rutiner : HJELP (Omregner en posisjon (bredde eller lengde) til hele antall grader relativt til laveste posisjon i MSQ-ruten. (Benyttes ved beregning av engradersrutenummer.)).

 TESTR (Kontrollerer rekkefølgen i filene).

COMMON variable:

DATA:

NAVN:	TYPE:	BRUK:
INN(14, 200)	INTEGER	Måledatabuffer.
IHNUT(10, 101)	INTEGER	Brukes ved kontroll av rekkefølgen. Inneholder innholdsregisterbufferet (post 2-101) samt siste post i forrige innholdsregisterblokk (post 1).

Variable:

NAVN:	TYPE:	BRUK:
NUT(10, 100)	INTEGER	Innholdsregisterbuffer.
JMIN	INTEGER	Nedre grense for den halvdel av bufferet som behandles i øyeblikket (=1 eller 101).
I	INTEGER	Linjenummer i registerbufferet.
ISTAT	INTEGER	Teller antall stasjoner.
J	INTEGER	Linjenummer i måledatabufferet.
IC	INTEGER	Antall poster i en stasjon.
IOX	INTEGER	Antall oksygenobservasjoner i en stasjon.
IFOS	INTEGER	Antall fosfatobservasjoner i en stasjon.
KF	INTEGER	Blokknummer for første linje i en stasjon.
LI	INTEGER	Linjenummer, første linje i en stasjon.
NT	INTEGER	Stasjonstypenummer (fjorddatakode eller vanlig kode).
MSQH	INTEGER	Engradersrutenummer.
IDYP	INTEGER	Siste dyp i en stasjon.
NQ	INTEGER	Kvadrant (0-3).

NAVN:	TYPE:	BRUK:
LONG	INTEGER	Hjelpevariabel ved beregning av engradersrutenummer.
LAT	INTEGER	Hjelpevariabel ved beregning av engradersrutenummer.
N	INTEGER	Hjelpevariabel.
JMAX	INTEGER	Øvre grense for den halvdel av bufferet som behandles i øyeblikket (=100 eller 200).
LSTAT	INTEGER	Statusvariabel i NTRAN.
L	INTEGER	Hjelpevariabel.
LL	INTEGER	Hjelpevariabel.
IHMAX	INTEGER	Hjelpevariabel.

```
000001 008 C
000002 008 C
6 000003 008 C *****
000004 008 C *
8 000005 008 C *   CONT   *
000006 008 C *
10 000007 008 C *****
000008 008 C
12 000009 008 C
000010 008 C   CONT PRODUSERER EN INNHOLDSREGISTERFIL FOR EN DATAFIL,
14 000011 010 C   SAMT REGISTRERER ANTALL STASJONER PR. ENGRADERSRUTE
000012 010 C   OG KONTROLLERER OM REKKEFØLGEN ER KORREKT.
16 000013 010 C   HVIS ØNSKELIG, OVERFØRES MALEDATAFILEN TIL EN
000014 010 C   NY FIL (F. EKS. MAGNETBAND)
18 000015 008 C
000016 008 C
20 000017 008 C   NØDVENDIGE RUTINER ; NTFEIL
000018 008 C   LESDB
22 000019 008 C
000020 008 C
24 000021 011 SUBROUTINE CONT(NUNIT,NUTU,IF,NTELL,MSQ,K,KUT,TOUT,NUND,REKKE)
000022 008 C
000023 008 C
000024 009   DIMENSION NUT(10,100),NTELL(100)
000025 009   COMMON /DATA/INN(14,200),IHNUT(10,101)
000026 008   EQUIVALENCE (NUT,IHNUT(1,2))
000027 008   LOGICAL TOUT,REKKE
000028 012 20   FORMAT(4X,14,12H BLOKKER MED,15,' RECORDS ER LEST,','
000029 008   114HDET BLE FUNNET,15,10H STASJONER)
000030 008 30   FORMAT(4X,14)
000031 008 35   FORMAT(4X,R4)
000032 008 10   FORMAT(14,4X,14,15,1X,13,14)
000033 008 11   FORMAT(14,4X,15,15,13,14)
000034 008 14   FORMAT(3X,14)
000035 008 16   FORMAT(J2,14)
000036 008 15   FORMAT(J2,12,12)
000037 008 17   FORMAT(15,13,14)
000038 008 18   FORMAT(2X,14,15,11)
000039 008 25   FORMAT(4X,'MALEDATAFILEN OVERFØRES TIL 1/0-ENHET ',12)
000040 008 26   FORMAT(4X,'KVADRANT ',11)
000041 008 27   FORMAT(/4X,'MSQ NR: ',13,' HAR RIKTIG REKKEFØLGE'/)
000042 008   K = 1

000043 008   KUT=0
000044 008   I = 1
000045 008   ISTAT = 0
000046 008   JMIN=1
000047 008   JMAX=100
000048 008   REKKE=.TRUE.
000049 008   IHNUT(1,1)=0
000050 008   IHNUT(2,1)=0
000051 008   IHNUT(3,1)=0
000052 008   IHNUT(4,1)=0
000053 008   IF (TOUT) WRITE(6,25) NUND
000054 008 C
000055 008 C   FØRSTE BLOKK LESES INN.
000056 008 C
000057 009   CALL NTRAN(NUNIT,2,1400,INN(1,1),LSTAT,22)
000058 009   CALL NTFEIL(1400,LSTAT,NUNIT,1,11000)
000059 008 C
000060 008 C   INNLESING AV ANDRE BLOKK PABEGYNNES.
000061 008 C
000062 009   CALL NTRAN(NUNIT,2,1400,INN(1,101),LSTAT)
000063 008 C
000064 008 C   KVADRANTNUMMER BESTEMMES.
000065 008 C
000066 008   IF ((INN(14,1).EQ.2H0J).OR.(INN(14,1).EQ.2H01))DECODE(18,INN(2,1))
000067 008   *LAT, LONG, NG
000068 008   IF ((LAT.EQ.0).AND.(NG.GT.1)) NG=NG-2
000069 008   IF ((LONG.EQ.0).AND.(NG.EQ.1)) NG=0
000070 008   IF ((LONG.EQ.0).AND.(NG.EQ.3)) NG=2
000071 008   IF ((INN(14,1).EQ.2H9J).OR.(INN(14,1).EQ.2H91))NG=0
000072 008   WRITE(6,26) NG
000073 008   DO 100 L = 1,100
000074 008   NTELL(L) = 0
000075 008 100 CONTINUE
000076 008   GO TO 115
000077 008 110 CONTINUE
000078 009   CALL LESDB(INN,14,NUNIT,K,JMAX,LSTAT,11000)
000079 008   JMIN=JMAX
000080 008   JMAX=JMAX+99
000081 008 115 CONTINUE
000082 008 C
000083 008 C   EN BLOKK GJENNOMSØKES.
000084 008 C
000085 008   DO 500J = JMIN,JMAX
000086 008 C
```

```
000087 008 C NYTT KORTBILDE.
000088 008 C
000089 008 N=FLD(6,6,INN(14,J))
000090 008 IF ((N,NE,15).AND.(N,NE,49)) GO TO 490
000091 008 C
000092 008 C NY STASJON.
000093 008 C
000094 008 ISTAT = ISTAT+1
000095 008 IF(ISTAT.EQ.1) GO TO 250
000096 008 C
000097 008 C SETT INN DE SISTE PARAMETRE I FORRIGE STASJONS INNHOLDSRECORD.
000098 008 C
000099 008 N=J-1
00100 008 IF((N,NE,0).AND.(N,NE,100)) DECODE(14,INN(5,N))IDYP
00101 008 ENCODE(15,NUT(5,1))NT,IOX,IFOS
00102 008 ENCODE(17,NUT(8,1))KF,LI,IC
00103 008 ENCODE(16,NUT(10,1))MSQH,IDYP
00104 008 I = I+1
00105 008 IF (I.LE.100) GO TO 250
00106 008 CALL NTRAN(NUTU,1,1000,NUT,L,22)
00107 008 KUT=KUT+1
00108 008 CALL NTFEIL(1000,L,NUTU,KUT,11000)
00109 008 I = I
00110 008 CALL TESTR(100)
00111 008 250 CONTINUE
00112 008 C
00113 008 C EN NY STASJON BEREGNES.
00114 008 C
00115 008 IC=1 GANTALL KORTBILDER,
00116 008 IOX=0 GANTALL OKSYGENOBSERVASJONER,
00117 008 IFOS=0 GANTALL FOSFATOBSERVASJONER,
00118 008 C
00119 008 C ADRESSEPARAMETRE.
00120 008 C
00121 008 KF=K
00122 008 LI=J-JMIN+1
00123 008 NT=FLD(0,6,INN(14,J))
00124 008 IF(NT.EQ.57) GO TO 260
00125 008 C
00126 008 C VANLIG KORTKODE.
00127 008 C
00128 008 NT=1
00129 008 DECODE(10,INN(1,J))NUT(1,1),NUT(6,1),NUT(7,1),NUT(3,1),NUT(4,1)
00130 008 NUT(6,1)=10*NUT(6,1)

000131 008 NUT(7,1)=10*NUT(7,1)
000132 008 IF((NQ,EQ,1).OR.(NQ,EQ,3))NUT(7,1)=-NUT(7,1)
000133 008 IF(NQ.GT.1)NUT(6,1)=-NUT(6,1)
000134 008 GO TO 265
000135 008 260 CONTINUE
000136 008 C
000137 008 C FJORDDATAKODEN.
000138 008 C
000139 008 NT=9
000140 008 DECODE(11,INN(1,J))NUT(1,1),NUT(6,1),NUT(7,1),NUT(3,1),NUT(4,1)
000141 008 265 CONTINUE
000142 008 DECODE(30,INN(1,J),ERR=270)NUT(2,1)
000143 008 GO TO 275
000144 008 270 CONTINUE
000145 008 DECODE(35,INN(1,J))NUT(2,1)
000146 008 275 CONTINUE
000147 008 NUT(3,1)=NUT(3,1)+1000
000148 008 C
000149 008 C ENGRADERSRUTENUMMER BEREGNES OG REGISTRERES.
000150 008 C
000151 008 LAT=NUT(6,1)
000152 008 LONG=NUT(7,1)
000153 008 CALL HJELP(LAT)
000154 008 CALL HJELP(LONG)
000155 008 MSQH=LONG+10*LAT
000156 008 NTELL(MSQH+1)=NTELL(MSQH+1)+1
000157 008 GO TO 500
000158 008 490 CONTINUE
000159 008 C
000160 008 C DYBDEKORTBILDE.
000161 008 C
000162 008 IF (N.EQ.C) GO TO 800 G FERDIG?
000163 008 IC=IC+1
000164 008 C
000165 008 C FOSFAT.
000166 008 C
000167 008 N=FLD(30,6,INN(13,J))
000168 008 IF(N.EQ.52) IFOS=IFOS+1
000169 008 C
000170 008 C OKSYGEN.
000171 008 C
000172 008 N=FLD(18,6,INN(10,J))
000173 008 IF(N,NE,5) IOX=IOX+1
000174 008 500 CONTINUE
```

```
2 000175 008 DECODE(14,INN(5,JMAX))IDYP 6SISTE DYP I BLOKKEN.
000176 009 IF (,NOT,TOUT) GO TO 110
4 000177 010 C
000178 010 C MALEDATA OVERFØRES TIL NY FIL.
6 000179 010 C
000180 009 CALL NTRAN(NUND,1,1400,INN(1,JMIN),L,22)
8 000181 009 CALL NTFEIL(1400,L,NUND,K,11000)
000182 008 GO TO 110
10 000183 008 800 CONTINUE
000184 008 C
12 000185 008 C FERDIG,SISTE PARAMETRE SETTES INN I SISTE STASJON.
000186 008 C
14 000187 008 J=J-1
000188 008 IF (J,EQ,100) GO TO 810
16 000189 008 IF (J,EQ,0) GO TO 810
000190 008 DECODE(14,INN(5,J))IDYP
18 000191 008 810 CONTINUE
000192 008 IF (J,GE,100) J=J-100
20 000193 008 ENCODE(15,NUT(5,1))NT,IOX,IFOS
000194 008 ENCODE(17,NUT(8,1))KF,LI,IC
22 000195 008 ENCODE(16,NUT(10,1))MSQH,IOYP
000196 008 J=J+100*(K-1)
24 000197 008 WRITE(6,20)K,J,ISTAT
000198 009 IF (TOUT) CALL NTRAN(NUND,1,1400,INN(1,JMIN),L,22)
26 000199 009 IF (TOUT) CALL NTFEIL(1400,L,NUND,K,11000)
000200 008 I=I+1
28 000201 008 IF (I,NE,101) GO TO 850
000202 008 CALL NTRAN(NUTU,1,1000,NUT,L,22)
30 000203 008 KUT=KUT+1
000204 008 CALL NTFEIL(1000,L,NUTU,KUT,11000)
32 000205 008 I=1
000206 008 CALL TESTR(100)
34 000207 008 850 CONTINUE
000208 008 DO 860 J=1,100
36 000209 008 DO 860 L=1,10
000210 008 NUT(L,J)=0
38 000211 008 860 CONTINUE
000212 008 870 CONTINUE
40 000213 008 C
000214 008 C SISTE BLOKK.
42 000215 008 C
000216 008 CALL NTRAN(NUTU,1,1000,NUT,L,22)
44 000217 008 KUT=KUT+1
000218 008 CALL NTFEIL(1000,L,NUTU,KUT,11000)
46

2 000219 008 I=I+1
000220 008 CALL TESTR(I)
4 000221 009 CALL NTRAN(NUNIT,22)
000222 010 IF (LSTAT,EQ,-2) IF=IF+1 6EOF-MERKE PASSERT?
6 000223 008 IF (REKKE) WRITE(6,27) MSQ
000224 008 RETURN
8 000225 008 1000 CALL ERTRAN(2) 6FEILAVBRUDD.
000226 008 C
10 000227 008 SUBROUTINE HJELP(JHJELP)
12 000228 008 C
000229 008 C HJELP BRUKES VED BEREGNING AV ENGRADERSRUTENUMMER.
14 000230 008 C
000231 008 JHJELP=JHJELP+1
16 000232 008 JHJELP=JHJELP/1000
000233 008 L=JHJELP/10
18 000234 008 L=10*L
000235 008 JHJELP=JHJELP-L
20 000236 008 JHJELP=ABS(JHJELP)
000237 008 RETURN
22 000238 008 C
000239 008 SUBROUTINE TESTR(MAX)
24 000240 008 C
000241 010 C TESTR TESTER OM REKKEFØLGEN I FILENE ER I OVERENSSTEMMELSE
000242 008 C MED VANLIG SORTERINGSORDEN.
26 000243 008 C
000244 010 IF (,NOT,REKKE) RETURN 6FEIL REKKEFØLGE ER ALLEREDE REGISTRERT
28 000245 008 IF (MAX,EQ,0) RETURN
000246 008 10 FORMAT(4X,'FEIL: ',4I5,2X,A6,2I9,2X,2A6,2X,A6)
30 000247 008 15 FORMAT(4X,'FEIL: ',15,1X,R4,2I5,2X,A6,2I9,2X,2A6,2X,A6)
000248 008 IHMAX=MAX+1
32 000249 008 DO 300 L=2,IHMAX
000250 008 LL=IHNUT(3,L)-IHNUT(3,L-1)
34 000251 008 IF (LL) 500,110,300
000252 008 110 CONTINUE
36 000253 008 LL=IHNUT(1,L)-IHNUT(1,L-1)
000254 008 IF (LL) 500,120,300
38 000255 008 120 CONTINUE
000256 008 LL=IHNUT(4,L)-IHNUT(4,L-1)
40 000257 008 IF (LL) 500,130,300
000258 008 130 CONTINUE
42 000259 008 LL=IHNUT(2,L)-IHNUT(2,L-1)
000260 008 IF (LL) 500,150,300
44 000261 008 150 CONTINUE
000262 008 IF (IHNUT(2,L).GT.9999) GO TO 200
46
```

```
2 000263 008 WRITE(6,10) (IHNUT(LL,L-1),LL=1,10)
000264 008 WRITE(6,10) (IHNUT(LL,L),LL=1,10)
4 000265 008 GO TO 500
000266 008 200 CONTINUE
6 000267 008 WRITE(6,15) (IHNUT(LL,L-1),LL=1,10)
000268 008 WRITE(6,15) (IHNUT(LL,L),LL=1,10)
8 000269 008 GO TO 500
000270 008 300 CONTINUE
10 000271 008 IHNUT(1,1)=IHNUT(1,IHMAX)
000272 008 IHNUT(2,1)=IHNUT(2,IHMAX)
12 000273 008 IHNUT(3,1)=IHNUT(3,IHMAX)
000274 008 IHNUT(4,1)=IHNUT(4,IHMAX)
14 000275 008 RETURN
000276 008 500 CONTINUE
16 000277 008 REKKE=,FALSE.
000278 008 RETURN
18 000279 008 END
```

20

22

24

26

28

30

32

34

36

38

40

42

44

46

DMIX

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Dato : 18/12-75
Utført av : Ø. Strand
Formål : DMIX fletter sammen to filer inne-
holdende sorterte, hydrografiske,
måledata.

Kall: DMIX(NUNA, NUNB, IFA, IFB, NUNUT, LHSORT, MSQ, KU, NDEFSSO)

Generelt:

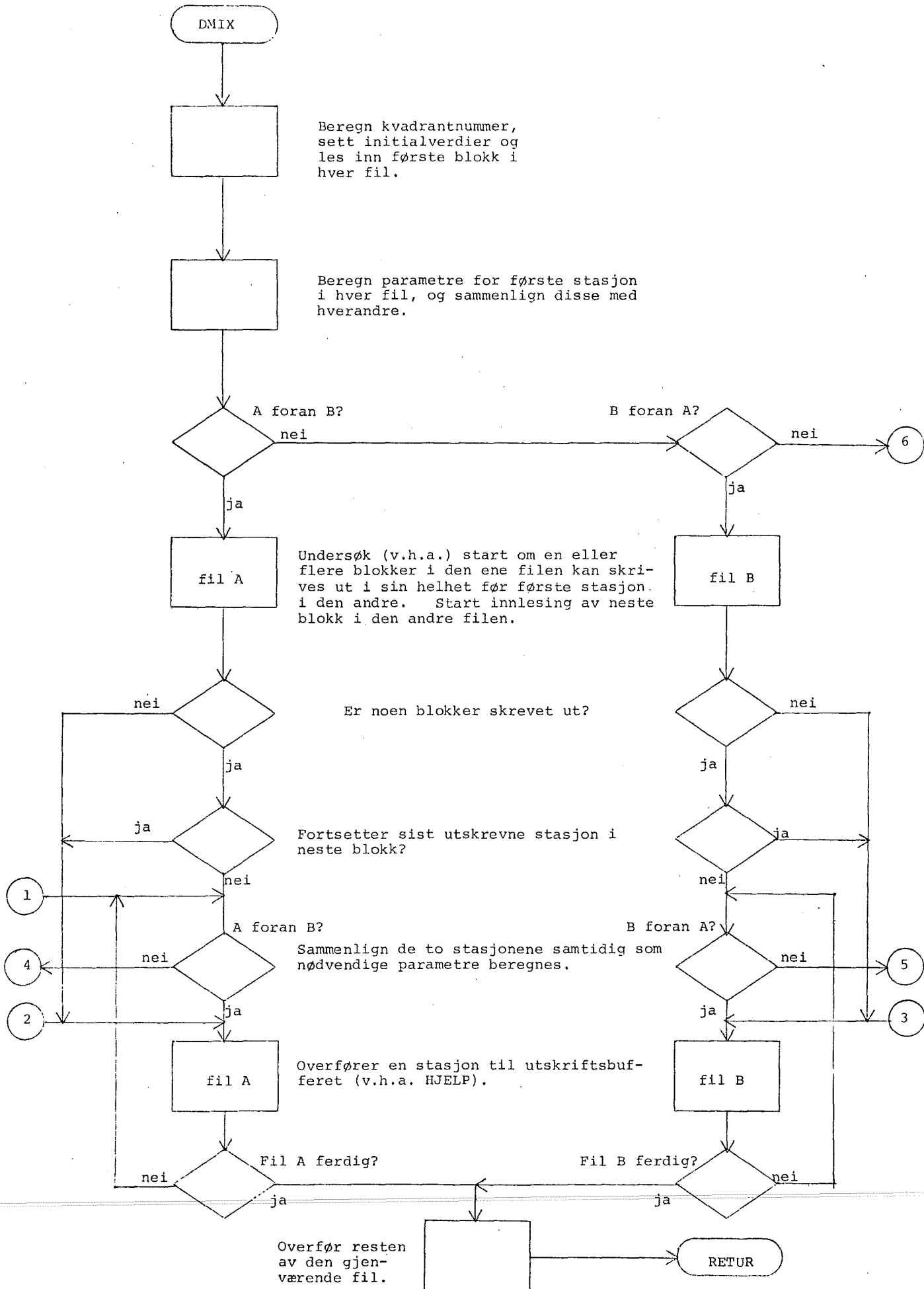
DMIX fletter sammen to sorterte måledatafiler (A og B) (hydrografiske data) slik at den nye fellesfilen også blir korrekt sortert. Rutinen benytter dobbelbuffering ved innlesing av data. Ved sammenligning av to stasjoner benyttes følgende parametre:

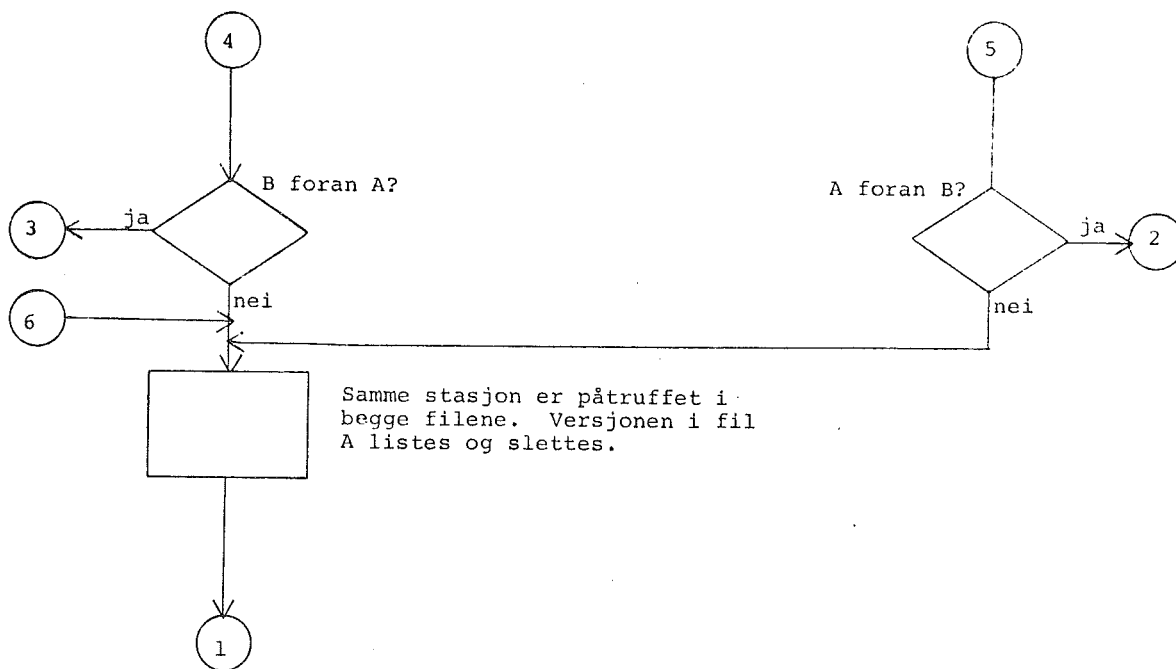
1. Sorteringsnummer fra oppgitt tabell.
2. Årstall (avledet).
3. Skipsnummer.
4. Dato (mnd - dag).
5. Stasjonsnummer.

Det første benyttes bare ved sortering etter geografiske kriterier. Ved fletting sammenlignes de fem (fire) parametrene for de to stasjonene etter tur, og ved ulikhet går den stasjonen som har laveste verdi på vedkommende parameter foran. Beregning/dekoding av parametre er tidkrevende, og verdiene for stasjonen som blir igjen etter en sammenligning lagres derfor til neste gang (hvis ikke alle parametre i denne er beregnet før ulikhet påtreffes, må de resterende beregnes separat). Rutinen vil først undersøke om en eller flere blokker i den ene filen i sin helhet går foran første stasjon i den andre filen. Deretter flettes de to filene sammen inntil den ene er ferdig, og tilslutt overføres resten av siste fil til fellesbufferet.

Formelle parametre:

NAVN:	TYPE:	BRUK:
NUNA	INTEGER	I/O-enhet, fil A.
NUNB	INTEGER	I/O-enhet fil B.
IFA	INTEGER	Filnummer på eventuelt magnetbånd (fil A).
IFB	INTEGER	Filnummer på eventuelt magnetbånd (fil B).
NUNUT	INTEGER	I/O-enhet for ny måledatafil.





NAVN:	TYPE:	BRUK:
LHSORT	LOGICAL	Angir om filene er sortert kronologisk (FALSE) eller etter oppgitt rekkefølge av engradersruter (TRUE).
MSQ	INTEGER	Marsdenrutenummer.
KU	INTEGER	Teller antall blokker i den sammenflettede fil.
NDEF50(100)	INTEGER	Inneholder sorteringsrekkefølge for engradersruter (ved behandling av ikke-kronologisk sorterte filer).

Eksterne rutiner : NTFEIL

Interne rutiner : HJELP (Overfører en stasjon fra et innlesebuffer til fellesbufferet, samt styrer innlesing (ved dobbelbuffering) og utskrivning av nye data).

HTEST (Sammenligner to stasjoner etter sorteringsnummer).

TEST (Sammenligner år, skip, dato og stasjonsnummer for to stasjoner).

START (Undersøker om en eller flere blokker i en gitt fil (A eller B) kan skrives ut i sin helhet før første stasjon i den andre innlesefilen).

BEREGN (Beregner skipsnummer, dato og stasjonsnummer for en stasjon).

COMMON variable:

DATA:

NAVN:	TYPE:	BRUK:
INNA(14, 200)	INTEGER	Buffer A.
INNB(14, 200)	INTEGER	Buffer B.
NUT(14, 100)	INTEGER	Fellesbuffer.

Variable:

NAVN:	TYPE:	BRUK:
BEGGE	LOGICAL	Angir hvorvidt begge filene er under behandling (TRUE), eller om den ene er ferdig.
N	INTEGER	Hjelpevariabel.
NQ	INTEGER	Kvadrant.

NAVN:	TYPE:	BRUK:
LU	INTEGER	Linjenummer i fellesbufferet.
LA	INTEGER	Linjenummer i buffer A.
LB	INTEGER	Linjenummer i buffer B.
L	INTEGER	Hjelpevariabel.
LABA	INTEGER	Linjenummerpeker (viser neste trinn etter utsortering av en stasjon fra buffer A).
LABB	INTEGER	Linjenummerpeker (viser neste trinn etter utsortering av en stasjon fra buffer B).
IGA	INTEGER	Sorteringsnummer for stasjon i fil A.
IGB	INTEGER	Sorteringsnummer for stasjon i fil B.
IYA	INTEGER	Tre siste sifre i årstallet (på karakterformat) for en stasjon i buffer A.
IYB	INTEGER	Som IYA (buffer B).
ISA	INTEGER	Skipsnummer for en stasjon i buffer A.
ISB	INTEGER	Skipsnummer for en stasjon i buffer B.
IDA	INTEGER	Dato (): måned, dag som firesifret tall) for en stasjon i buffer A.
IDB	INTEGER	Dato for en stasjon i buffer B.
ISHA	INTEGER	Stasjonsnummer for en stasjon i buffer A.
ISHB	INTEGER	Stasjonsnummer for en stasjon i Buffer B.
LSTATA	INTEGER	Statusvariabel (NTRAN) for fil A.
LSTATB	INTEGER	Statusvariabel (NTRAN) for fil B.
NC	INTEGER	Kode, kolonne 80 (viser masterpost/dybdepost).
J	INTEGER	Hjelpevariabel.
I	INTEGER	Hjelpevariabel.
JA	INTEGER	Hjelpevariabel.
ILVEKK	INTEGER	Teller antall slettede linjer.
ISVEKK	INTEGER	Teller antall slettede stasjoner.
ISVEKK	INTEGER	Teller antall slettede stasjoner.

NAVN:	TYPE:	BRUK:
JG	INTEGER	Hjelpevariabel.
JH	INTEGER	Hjelpevariabel.

Beskrivelse av hjelperutinene:

HJELP(NDIM, LZ, NUN, \$, LS, LVEKK)

HJELP overfører en stasjon fra buffer NDIM til fellesbufferet (NUT) samt styrer innlesing og utskrivning av data. LZ er linjenummer i NDIM. NUN er I/O-enhetsnummer for den tilhørende innlesefil, og LS er statusvariabel (NTRAN). \$ er retur hvis filen er ferdigbehandlet. LVEKK er en logisk variabel som angir hvorvidt stasjonen skal fjernes.

HTEST(ND, LH, IGG, IGN, \$, \$)

HTEST brukes ved fletting av geografisk sorterte filer. Rutinen sammenligner sorteringsnummer for to stasjoner, en i buffer ND (linje LH, sorteringsnummer: IGN), og en hvor nummeret (IGG) er beregnet på forhånd. Hvis stasjonen i ND går foran den beregnede stasjonen, returneres via første \$, i motsatt fall via siste \$. Hvis de to stasjonene har samme nummer, skjer retur fra rutinen på vanlig måte.

TEST(ND, LH, IYG, IYN, ISG, ISN, IDG, IDN, ISHG, ISHN, \$, \$, \$, \$)

TEST sammenligner to stasjoner etter år, skip, dato og stasjonsnummer. Rutinen virker på samme måte som HTEST: En stasjon (buffer ND, linje LH) sammenlignes med parametre for en allerede beregnet stasjon (IYG, ISG, IDG, ISHG). Første \$): Retur hvis stasjonen i buffer ND kommer først. Andre \$): Retur hvis den beregnede stasjonen kommer først, og alle parametre for stasjonen i buffer ND (IYN, ISN, IDN, ISHN) ikke er beregnet (disse må da beregnes ferdig). Tredje \$): Retur hvis den beregnede stasjonen kommer først, og alle parametre for stasjonen i buffer ND er beregnet (): de to stasjonene har samme årstall, skipsnummer og dato). Fjerde \$): Feilutgang.

START(ND, NUN, IGG, IGN, IYG, IYN, ISG, ISN, IDG, IDN, ISHG, ISHN, \$, LS)

START undersøker om det er mulig å skrive ut en eller flere blokker fra den ene filen (buffer: ND) før første stasjon i den andre filen (som må være beregnet). IGG, IGN, IYG, IYN, ISH, ISN, IDG, IDN, ISHG og ISHN har samme betydning som i TEST og HTEST. NUN er I/O-enhetsnummer for innlesefilen, og LS er statusvariabel (NTRAN). \$): Retur hvis ingen blokker er overført, eller siste stasjon i sist overførte blokk fortsetter i neste blokk.

BEREGN(ND, LH, ISN, IDN, ISHN, \$)

BEREGN beregner skipsnummer (ISN), dato (IDN) og stasjonsnummer (ISHN) for linje LH i buffer ND. Deretter returneres via \$.

```
000001 004 C
000002 004 C
000003 004 C
000004 004 C
000005 004 C
000006 004 C
000007 004 C
000008 004 C
000009 004 C
000010 004 C
000011 004 C
000012 004 C
000013 004 C
000014 004 C
000015 008 C
000016 008 C
000017 004 C
000018 004 C
000019 004 C
000020 004 C
000021 004 C
000022 004 C
000023 004 C
000024 004 C
000025 004 C
000026 004 25
000027 004 30
000028 004 C
000029 004 C
000030 004 C
000031 004 C
000032 004 C
000033 004 C
000034 004 C
000035 004 C
000036 004 C
000037 004 105
000038 004 C
000039 004 C
000040 004 C
000041 004 115
000042 004 C

          *****
          *
          *      DMIX      *
          *
          *****

DMIX FLETTER SAMMEN TO SORTERTE DATAFILER,
SORTERINGSORDEN ER: AR,SKIP,DATO OG STASJONSNUMMER,ELLER
OPPGITT REKKEFOLGE AV ENGRADERSRUTER,AR,SKIP,DATO OG
STASJONSNUMMER();GEOGRAFISK SORTERING)
RUTINEN UTELATER DEN ENE VERSJONEN AV EVENTUELLE DOUBLETTER,
(DETTE FORUTSETTER AT UTGAVENE ER IDENTISKE I ALLE AKTUELLE
SORTERINGSKRITERIER)

NØDVENDIGE RUTINER : NTFEIL

SUBROUTINE DMIX(NUNA,NUNB,IFA,IFB,NUNUT,LHSORT,MSQ,KU,NDEFSO)
COMMON /DATA/ INNA(14,200),INNB(14,200),NUT(14,100)
DIMENSION NDEFSO(100)
LOGICAL LHSORT,BEGGE
FORMAT(4X,14,' BLOKKER MED ',13,' LINJER I SISTE BLOKK')
FORMAT(4X,15,' STASJONER MED ',16,' LINJER ER FJERNET')

      KVADRANT (NQ) BESTEMMES.

      IF (MSQ.GT.900) GO TO 115
      IF (MSQ.GT.288) GO TO 125
      N=MSQ-1
      N=N/36
      N=36*N
      N=MSQ-N
      CONTINUE
      NQ=0
      IF (N.LT.19) NQ=1
      GO TO 130
      CONTINUE
      N=MSQ-900

GO TO 105
CONTINUE
N=MSQ-300
N=N/36
N=36*N
N=MSQ-300-N
NQ=2
IF (N.LT.18) NQ=3
CONTINUE
BEGGE=.TRUE.
KU=0
LU=1
LA=1
LB=1
ILVEKK=0
ISVEKK=0

      LES INN FØRSTE BLOKK I HVER FIL.

      CALL NTRAN(NUNA,2,1400,INNA,L,22)
      CALL NTFEIL(1400,L,NUNA,1,1999)
      CALL NTRAN(NUNB,2,1400,INNB,L,22)
      CALL NTFEIL(1400,L,NUNB,1,1999)
      ASSIGN 230 TO LABA
      ASSIGN 190 TO LABB
      IF(LHSORT) ASSIGN 220 TO LABA      @GEOGRAFISK
      IF(LHSORT) ASSIGN 180 TO LABB      @SORTERING

BEREGN PARAMETRE FOR FØRSTE STASJON I BUFFER A.

IF (LHSORT) CALL HTEST(INNA,1,-1,IGA,1135,1135)
CONTINUE
IYA=FLO(0,18,INNA(4,1))
CALL BEREGN(INNA,1,ISA,IDA,ISHA,1136)
CONTINUE

SAMPENLIGN FØRSTE STASJON I DE TO BUFFERE.

IF(LHSORT) CALL HTEST(INNB,1,IGA,IGB,1150,1138)
CALL TEST(INNB,1,IYA,IYB,ISA,ISB,IDA,IDB,ISHA,ISHB,1150,1140,
(14),1290)
CONTINUE
IYB=FLO(0,18,INNB(4,1))
CONTINUE
```

```
000087 004 CALL BEREGN(INNB,1,ISB,IDB,ISHB,1141)
000088 004 141 CONTINUE
000089 004 C
000090 004 C START INNLESING AV NESTE BLOKK I FIL B.
000091 004 C UNDERSØK OM EN ELLER FLERE BLOKKER I FIL A KAN SKRIVES UT
000092 004 C I SIN HELHET FØR FØRSTE STASJON I FIL B.
000093 004 C
000094 004 CALL NTRAN(NUNB,2,1400,INNB(1,101),LSTATB)
000095 004 CALL START(INNA,NUNA,1GB,1GA,IYB,IYA,ISB,ISA,IDB,IDA,
000096 004 *ISHB,ISHA,1300,LSTATA)
000097 004 GO TO LABA
000098 004 150 CONTINUE
000099 004 C
000100 004 C START INNLESING AV NESTE BLOKK I FIL A.
000101 004 C UNDERSØK OM EN ELLER FLERE BLOKKER I FIL B KAN SKRIVES UT
000102 004 C I SIN HELHET FØR FØRSTE STASJON I FIL A.
000103 004 C
000104 004 CALL NTRAN(NUNA,2,1400,INNA(1,101),LSTATA)
000105 004 CALL START(INNB,NUNB,1GA,1GB,IYA,IYB,ISA,ISB,IDA,IDB,
000106 004 *ISHA,ISHB,1400,LSTATB)
000107 004 GO TO LABB
000108 004 C
000109 004 C
000110 004 C BEGGE FILENE GJENNOMLESES,SAMMENLIGNES, OG SKRIVES UT PÅ FELLES-
000111 004 C FILEN INNTIL DEN ENE ER FERDIG.
000112 004 C
000113 004 C
000114 004 180 CONTINUE
000115 004 C
000116 004 C FORRIGE UTPLUKKEDE STASJON KOM FRA BUFFER B.
000117 004 C
000118 004 CALL HTEST(INNB,LB,1GA,1GB,1400,1200)
000119 004 190 CONTINUE
000120 004 CALL TEST(INNB,LB,IYA,IYB,ISA,ISB,IDA,IDB,ISHA,ISHB,1400,1210,
000121 004 *1300,1280)
000122 004 200 CONTINUE
000123 004 C
000124 004 C STASJONEN I BUFFER A KOMMER FØRST,OG DE RESTERENDE
000125 004 C PARAMETRE FOR STASJONEN I BUFFER B MA BEREGNES.
000126 004 C
000127 004 IYB=FLD(0,18,INNB(4,LB))
000128 004 210 CONTINUE
000129 004 CALL BEREGN(INNB,LB,ISB,IDB,ISHB,1300)
000130 004 220 CONTINUE

000131 004 C
000132 004 C FORRIGE UTPLUKKEDE STASJON KOM FRA BUFFER A.
000133 004 C
000134 004 CALL HTEST(INNA,LA,1GB,1GA,1300,1240)
000135 004 230 CONTINUE
000136 004 CALL TEST(INNA,LA,IYB,IYA,ISB,ISA,IDB,IDA,ISHB,ISHA,1300,1250,
000137 004 *1400,1280)
000138 004 240 CONTINUE
000139 004 C
000140 004 C STASJONEN I BUFFER B KOMMER FØRST,OG DE RESTERENDE
000141 004 C PARAMETRE FOR STASJONEN I BUFFER A MA BEREGNES.
000142 004 C
000143 004 IYA=FLD(0,18,INNA(4,LA))
000144 004 250 CONTINUE
000145 004 CALL BEREGN(INNA,LA,ISA,IDA,ISHA,1400)
000146 004 280 CONTINUE
000147 004 C
000148 004 C SAMME STASJON ER PATRUFFET I BEGGE FILENE,
000149 004 C OG STASJONEN I FIL A LISTES OG SLETTES.
000150 004 C
000151 004 ISVEKK=ISVEKK+1
000152 004 CALL HJELP(INNA,LA,NUNA,1320,LSTATA,.TRUE.)
000153 004 GO TO LABA
000154 006 290 CONTINUE
000155 006 C FØRSTE STASJON I FILENE ER IDENTISKE,OG INNLESING AV DE NESTE
000156 006 C BLOKKENE MA STARTES SEPARAT.
000157 006 C
000158 007 CALL NTRAN(NUNA,2,1400,INNA(1,101),LSTATA)
000159 006 CALL NTRAN(NUNB,2,1400,INNB(1,101),LSTATB)
000160 006 GO TO 280
000161 004 300 CONTINUE
000162 004 C
000163 004 C EN STASJON FRA BUFFER A SKRIVES UT.
000164 004 C
000165 004 CALL HJELP(INNA,LA,NUNA,1320,LSTATA,.FALSE.)
000166 004 GO TO LABA
000167 004 320 CONTINUE
000168 004 C
000169 004 C FIL A ER FERDIG.
000170 004 C
000171 004 IF (.NOT.BEGGE) GO TO 500
000172 004 BEGGE=.FALSE.
000173 004 C
000174 004 C LINJENUMMERPEKEREN ENDRES SLIK AT RESTEN AV FIL B SKRIVES UT.
```

```
000175 004 C
000176 004 ASSIGN 400 TO LABB
000177 004 400 CONTINUE
000178 004 C
000179 004 C EN STASJON FRA BUFFER B SKRIVES UT.
000180 004 C
000181 004 CALL HJELP(INNB,LB,NUNB,1420,LSTATB,.FALSE.)
000182 004 GO TO LABB
000183 004 420 CONTINUE
000184 004 C
000185 004 C
000186 004 C FIL B ER FERDIG.
000187 004 C
000188 004 IF (.NOT.BEGGE) GO TO 500
000189 004 BEGGE=.FALSE.
000190 004 C
000191 004 C LINJENUMMERPEKEREN ENDRES SLIK AT RESTEN AV FIL A SKRIVES UT.
000192 004 C
000193 004 ASSIGN 300 TO LABA
000194 004 GO TO LABA
000195 004 500 CONTINUE
000196 004 C
000197 004 C FERDIG.
000198 004 C
000199 004 CALL NTRAN(NUNA,22)
000200 004 CALL NTRAN(NUNB,22)
000201 004 IF (LSTATA.EQ.-2) IFA=IFA+1
000202 004 IF (LSTATB.EQ.-2) IFB=IFB+1
000203 004 DO 520 J=LU,100
000204 004 DO 520 I=1,14
000205 004 NUT(I,J)=0
000206 004 520 CONTINUE
000207 004 CALL NTRAN(NUNUT,1,1400,NUT,L,22)
000208 004 KU=KU+1
000209 004 CALL NTFEIL(1400,L,NUNUT,KU,1999)
000210 004 L=LU-1
000211 004 WRITE(6,25) KU,L
000212 004 WRITE(6,30) ISVEKK,ILVEKK
000213 004 RETURN
000214 004 999 CALL ERTRAN(2)
000215 004 C
000216 004 SUBROUTINE BEREKN(ND,LH,ISN,IDN,ISHN,I)
000217 004 C
000218 004 C BEREKN DEKODER SKIPSNUMMER,DATO OG STASJONSNUMMER FRA ET

000219 004 C KORTBILDE.
000220 004 C
000221 004 DIMENSION ND(14,200)
000222 004 20 FORMAT(14)
000223 004 41 FORMAT(4X,14)
000224 004 42 FORMAT(4X,R4)
000225 004 44 FORMAT(3X,14)
000226 004 DECODE(20,ND(1,LH)) ISN
000227 004 DECODE(44,ND(4,LH)) IDN
000228 004 DECODE(41,ND(1,LH),ERR=100) ISHN
000229 004 RETURN 6
000230 004 100 CONTINUE
000231 004 DECODE(42,ND(1,LH)) ISHN
000232 004 RETURN 6
000233 004 C
000234 004 SUBROUTINE START(ND,NUN,IGG,IGN,IYG,IYN,ISG,ISN,IDG,IDN,
000235 004 *ISHG,ISHN,I,LS)
000236 004 C
000237 004 C START UNDERSØKER HVORVIDT SISTE FOREKOMMENDE STASJON
000238 004 C I ET BUFFER (ND) GAR FORAN EN ALLEREDE BEREGNET STASJON.
000239 004 C I SA FALL SKRIVES DETTE BUFFERET UT PA FELLESFILEN,OG NESTE
000240 004 C BLOKK LESES INN.(BRUKES TIL A UNDERSØKE OM DEN ENE AV
000241 004 C INNLESEFILENE INNEHOLDER EN ELLER FLERE BLOKKER SOM I SIN HELHET
000242 004 C GAR FORAN DEN FØRSTE STASJONEN I DEN ANDRE FILEN).
000243 004 C
000244 004 DIMENSION ND(14,200)
000245 004 100 CONTINUE
000246 005 IF(ND(I,100).EQ.0) GO TO 250
000247 004 C
000248 004 C FINN STARTEN PA SISTE STASJON I BUFFERET.
000249 004 C
000250 004 DO 120 J=100,1,-1
000251 004 NC=FLD(6,6,ND(14,J))
000252 004 IF(NC.EG.51) GO TO 120
000253 004 IF(NC.EG.57) GO TO 120
000254 004 GO TO 125
000255 004 120 CONTINUE
000256 004 CALL ERTRAN(2)
000257 004 125 CONTINUE
000258 004 C
000259 004 C SAMMENLIGN SISTE STASJON I BUFFERET MED FØRSTE STASJON
000260 004 C I DEN ANDRE FILEN.
000261 004 C
000262 004 IF(LHSORT) CALL HJEST(ND,J,IGG,IGN,1200,1250)
```



```
000263 004 CALL TEST(ND,J,IYG,IYN,ISG,ISN,IDG,IDN,ISHG,ISHN,1200,1250,1250,
000264 004 *1250)
000265 004 200 CONTINUE
000266 004 C
000267 004 C EN BLOKK SKRIVES UT,OG EN NY LESES INN.
000268 004 C
000269 004 KU=KU+1
000270 004 CALL NTRAN(NUNUT,1,1400,ND,L,22)
000271 004 CALL NTFEIL(1400,L,NUNUT,KU,1950)
000272 004 CALL NTRAN(NUN,2,1400,ND,L,22)
000273 004 CALL NTFEIL(1400,L,NUN,-30,1950)
000274 004 GO TO 100
000275 004 250 CONTINUE
000276 004 C
000277 004 C SLUTT PA OVERFØRING AV HELE BLOKKER.
000278 004 C START INNLESING AV NESTE BLOKK I FILEN.
000279 004 C
000280 004 CALL NTRAN(NUN,2,1400,ND(1,101),LS)
000281 004 IF(KU.EG.0) RETURN 13
000282 004 NC=FLD(6,6,ND(14,1))
000283 004 IF(NC.EG.51) RETURN 13
000284 004 IF(NC.EG.57) RETURN 13
000285 004 RETURN
000286 004 950 CALL ERTRAN(2)
000287 004 C
000288 004 SUBROUTINE TEST(ND,LH,IYG,IYN,ISG,ISN,IDG,IDN,ISHG,ISHN,1,1,1,1)
000289 004 C
000290 004 C TEST UNDERSØKER OM EN NY STASJON (BUFFER ;ND) SKAL KOMME FORAN
000291 004 C ELLER ETTER EN ALLEREDE BEREGNET STASJON I DET ANDRE BUFFERET.
000292 004 C
000293 004 DIMENSION ND(14,200)
000294 004 43 FORMAT(14)
000295 004 44 FORMAT(3X,14)
000296 004 41 FORMAT(4X,14)
000297 004 42 FORMAT(4X,R4)
000298 004 IYN=FLD(0,18,ND(4,LH))
000299 004 IF (IYN.LT.IYG) RETURN 11
000300 004 IF (IYN.GT.IYG) RETURN 12
000301 004 DECODE(43,ND(1,LH)) ISN
000302 004 IF (ISN.LT.ISG) RETURN 11
000303 004 IF (ISN.GT.ISG) RETURN 12
000304 004 DECODE(44,ND(4,LH)) IDN
000305 004 IF (IDN.LT.IDG) RETURN 11
000306 004 IF (IDN.GT.IDG) RETURN 12

000307 004 DECODE(41,ND(1,LH),ERR=100) ISHN
000308 004 GO TO 120
000309 004 100 CONTINUE
000310 004 DECODE(42,ND(1,LH)) ISHN
000311 004 120 CONTINUE
000312 004 IF (ISHN.LT.ISHG) RETURN 11
000313 004 IF (ISHN.GT.ISHG) RETURN 13
000314 004 RETURN 14
000315 004 C
000316 004 SUBROUTINE HTEST(ND,LH,IGG,IGN,1,1)
000317 004 C
000318 004 C HTEST BRUKES VED FLETING AV GEOGRAFISK SORTERTE FILER
000319 004 C RUTINEN AVGJØR HVORVIDT EN NY STASJON (BUFFER ;
000320 004 C ND) HAR LAVERE,SAMME ELLER HØYERE SORTERINGSNUMMER ENN EN ALLEREDE
000321 004 C BEREGNET STASJON I DET ANDRE BUFFERET.
000322 004 C
000323 004 DIMENSION ND(14,200)
000324 004 48 FORMAT(15)
000325 004 49 FORMAT(2X,11)
000326 004 47 FORMAT(2X,14)
000327 004 50 FORMAT(3X,11)
000328 004 C
000329 004 C ENGRADERSRUTENUMMER BEREGNES.
000330 004 C
000331 004 IF (NG.GT.1) GO TO 100
000332 004 DECODE(50,ND(2,LH)) IGN
000333 004 GO TO 110
000334 004 100 CONTINUE
000335 004 DECODE(47,ND(2,LH)) IGN
000336 004 IGN=(IGN-1)/100
000337 004 JH=IGN/10
000338 004 JH=10*JH
000339 004 IGN=IGN-JH
000340 004 110 CONTINUE
000341 004 IF (NG.GT.0) GO TO (125,120,125),NG
000342 004 120 CONTINUE
000343 004 DECODE(49,ND(3,LH)) JG
000344 004 GO TO 130
000345 004 125 CONTINUE
000346 004 DECODE(48,ND(3,LH)) JG
000347 004 JG=(JG-1)/100
000348 004 JH=JG/10
000349 004 JH=10*JH
000350 004 JG=JG-JH
```

```
000351 004 130 CONTINUE
000352 004 IGN=10*IGN
000353 004 IGN=IGN+JG ØENGRADERSRUTENUMMER
000354 004 IGN=NDEF50(IGN+1) ØSORTERINGSNUMMER
000355 004 IF (IGN.GT.IGG) RETURN 6
000356 004 IF (IGN.LT.IGG) RETURN 5
000357 004 RETURN
000358 004 C
000359 004 SUBROUTINE HJELP(NDIM,LZ,NUN,I,LS,LVEKK)
000360 004 C
000361 004 C HJELP OVERFØRER EN STASJON FRA INNLESEBUFFERET (NDIM) TIL
000362 004 C FELLESBUFFERET,SAMT STYRER INNLESING OG UTSKRIVING AV DATA.
000363 004 C
000364 004 DIMENSION NDIM(14,200)
000365 004 LOGICAL LVEKK
000366 004 10 FORMAT(4X,14A6)
000367 004 100 CONTINUE
000368 004 IF (LVEKK) GO TO 140
000369 004 DO 110 JA=1,14
000370 004 NUT(JA,LU)=NDIM(JA,LZ)
000371 004 110 CONTINUE
000372 004 LU=LU+1
000373 004 IF(LU.LT.101)GO TO 150
000374 004 C
000375 004 C FELLESBUFFERET ER FULLT.
000376 004 C
000377 004 LU=1
000378 004 KU=KU+1
000379 004 CALL NTRAN(NUNUT,1,1400,NUT,L,22)
000380 004 CALL NTFEIL(1400,L,NUNUT,KU,1999)
000381 004 GO TO 150
000382 004 140 CONTINUE
000383 004 C
000384 004 C STASJONER SOM SKAL FJERNES LISTES,OG ANTALL LINJER TELLES.
000385 004 C
000386 004 ILVEKK=ILVEKK+1
000387 004 WRITE(6,10) (NDIM(JA,LZ),JA=1,14)
000388 004 150 CONTINUE
000389 004 LZ=LZ+1
000390 004 IF ((LZ,NE.101).AND.(LZ,NE.201)) GO TO 200
000391 004 JH=LZ-100
000392 004 CALL NTRAN(NUN,22)
000393 004 CALL NTFEIL(1400,LS,NUN,-10,1999)
000394 004 CALL NTRAN(NUN,2,1400,NDIM(1,JH),LS)
000395 004 IF (LZ.EQ.201) LZ=1
000396 004 200 CONTINUE
000397 004 NC=FLD(6,6,NDIM(14,LZ)) Ø KORTKODEN (KOLONNE 80) BESTEMMES
000398 004 IF (NC.EQ.51) GO TO 100
000399 004 IF (NC.EQ.57) GO TO 100
000400 004 IF (NC.EQ.0) RETURN 4 Ø SLUTT PÅ DATA FRA VEDKOMMENDE FIL
000401 004 RETURN ØBY STASJON
000402 004 999 CALL ERTRAN(2) ØFEILAVBRUDD
000403 004 END
```

IMIX

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Dato : 18/12-75
Utført av : Ø. Strand
Formål : IMIX fletter sammen to sorterte
innholdsregisterfiler.

Kall: IMIX(NUNA, NUNB, IFA, IFB, NUNUT, LHSORT, KU, NTELL, NDEF SO)

Generelt:

IMIX fletter sammen to sorterte innholdsregisterfiler (A og B) til en ny fil. Adressene i den nye filen rettes slik at de viser til adressene i den tilhørende sammenflettede måledatafil. Virkemåten er følgende: Rutinen undersøker først om en eller flere blokker fra den ene filen i sin helhet kommer før første stasjon i den andre filen. Disse overføres i såfall direkte til den nye filen. Deretter sammenlignes stasjon for stasjon inntil en fil er ferdig gjennomført. Tilslutt overføres de resterende data fra den siste filen. Hvis to make stasjoner påtreffes (): samme årstall, skip, dato og stasjonsnummer) blir versjonen i fil A slettet og registrert i engradersrutetabellen (NTELL) (som senere kan brukes til å oppdatere REGFIL3).

Formelle parametre:

NAVN:	TYPE:	BRUK:
NUNA	INTEGER	I/O-enhet, fil A.
NUNB	INTEGER	I/O-enhet, fil B.
IFA	INTEGER	Filnummer på eventuelt magnetbånd (fil A).
IFB	INTEGER	Filnummer på eventuelt magnetbånd (fil B).
NUNUT	INTEGER	I/O-enhet, utskriftsfil (fellesfilen).
LHSORT	LOGICAL	Angir om filene er sortert geografisk (TRUE) eller etter år, skip, dato og stasjonsnummer (FALSE).
KU	INTEGER	Teller antall blokker i fellesfilen.
NTELL(100)	INTEGER	Ved retur viser NTELL antall slettede stasjoner i hver engradersrute. Antallet fremkommer som et negativt tall.
NDEF SO(100)	INTEGER	Inneholder sorteringsnummer for hver engradersrute (ved geografisk sortering).

Eksterne rutiner : NTFEIL

Interne rutiner : HJELP (entrepunkt: HOPP)
(Overfører en stasjon fra et innlesebuffer til utskriftsbufferet. Styrer utskrivning av fulle buffere og innlesing av nye data (ved dobbelbuffering). HOPP fjerner en stasjon.)

HTEST (entrepunkt: TEST)
(HTEST sammenligner to stasjoner sortert geografisk. TEST sammenligner to stasjoner sortert kronologisk).

START (Undersøker om en eller flere blokker i den ene av de to innlesefilene i sin helhet går foran første stasjon i den andre filen (disse overføres i såfall direkte til fellesfilen).

COMMON variable:

DATA:

NAVN:	TYPE:	BRUK:
INNHA(10, 200)	INTEGER	Innlesebuffer A.
INNHB(10, 200)	INTEGER	Innlesebuffer B.
NUT(10, 100)	INTEGER	Utskriftsbuffer.

Variable:

NAVN:	TYPE:	BRUK:
BEGGE	LOGICAL	Viser hvorvidt begge filene er under behandling (TRUE), eller den ene er ferdig.
LU	INTEGER	Linjenummer, fellesbuffer.
LA	INTEGER	Linjenummer, buffer A.
LB	INTEGER	Linjenummer, buffer B.
KBD	INTEGER	Blokknummer i den tilhørende sammenflettede måledatafil.
LIND	INTEGER	Linjenummer for første linje i ny stasjon i den tilhørende måledatafil.
I	INTEGER	Indikator som viser hvilken fil sist utsorterte stasjon kom fra (I>0): fil B, (I<0): fil A.
LABEL	INTEGER	Linjenummervelger (etter utsortering av en stasjon).

NAVN:	TYPE:	BRUK:
LSTATA	INTEGER	Statusvariabel (NTRAN), fil A.
LSTATB	INTEGER	Statusvariabel (NTRAN), fil B.
NQA	INTEGER	Sorteringsnummer for siste stasjon (fil A).
NQB	INTEGER	Sorteringsnummer for siste stasjon (fil B).
J	INTEGER	Hjelpevariabel.
JA	INTEGER	Hjelpevariabel.
JZ	INTEGER	Hjelpevariabel.
L	INTEGER	Hjelpevariabel.
JH	INTEGER	Hjelpevariabel.
ITVEKK	INTEGER	Teller antall slettede stasjoner.

Beskrivelse av hjelperutinene:

HJELP(NDIM, LZ, NUN, \$, LS)
HOPP(NDIM, LZ, NUN, \$, LS) (entrepunkt)

HJELP overfører en stasjon fra buffer NDIM (linjeLZ) til fellesbufferet. NUN er I/O-enhet for vedkommende fil (A eller B), og LS er statusvariabel for NTRAN. Rutinen styrer også innlesing og utskrivning av data. \$ er retur hvis den aktuelle filen er ferdig lest. HOPP sletter en stasjon i buffer NDIM.

HTEST(\$, \$, \$)
TEST(\$, \$, \$) (entrepunkt)

HEST sammenligner en stasjon i hvert av de to innlesebufferne (sortert etter sorteringsnummer, år, skip, dato og stasjonsnummer). Den ene stasjonen må ha sorteringsnummeret dekodet på forhånd (): ved forrige sammenligning). (Denne mekanismen styres av den globale variable I (se variabellisten).)
TEST sammenligner år, skip, dato og stasjonsnummer.

Første \$): Stasjonen i buffer A går foran.
Andre \$): Stasjonen i buffer B går foran.
Tredje \$): Feilutgang (): samme stasjon i begge buffere).

START(NDIM, NUN, LS, IND, \$, \$)

START undersøker om en eller flere blokker i den ene filen (A eller B) kan skrives ut i sin helhet før første stasjon i den andre filen. NDIM er buffer, NUN I/O-enhet og LS linjenummer i bufferet. (): LA eller LB). IND angir hvilken fil som skal prøves overført blokkvis: IND<0): fil A, (IND>0): fil B. Begge \$-tegn er feilutganger.

```

000001 003 C
000002 003 C
000003 003 C
000004 003 C
6 000005 003 C
000006 003 C
10 000007 003 C
000008 003 C
12 000009 003 C
000010 003 C
14 000011 003 C
000012 004 C
16 000013 004 C
000014 003 C
18 000015 003 C
000016 006 C
20 000017 007 C
000018 004 C
22 000019 004 C
000020 003 C
24 000021 003 C
000022 003 C
26 000023 003 C
000024 003 C
28 000025 003 C
000026 003 C
30 000027 003 C
000028 003 C
000029 003 C
000030 003 47
000031 003 25
000032 003 26
000033 003 27
000034 003 28
000035 003 29
000036 003 30
40 000037 003
000038 003 31
42 000039 003
000040 003
44 000041 003
000042 003

*****
*
*      IMIX      *
*
*****

IMIX FLETTER SAMMEN TO SORTERTE INNHOLDSREGISTERFILER,
SORTERINGSORDEN ; AR,SKIP,DATO OG STASJONSNUMMER,ELLER
OPPGITT REKKEFØLGE AV ENGRADERSRUTER,AR,SKIP,DATO OG
STASJONSNUMMER ();GEOGRAFISK SORTERING)
REGISTERETS ADRESSEFELTER ENDRES TIL A ANGI ADRESSENE I DEN
TILHØRENDE SAMMENFLETTEDE DATAFIL.
RUTINEN UTELATER DEN ENE VERSJONEN AV EVENTUELLE DOUBLETTER
( );STASJONER SOM ER IDENTISKE I ALLE AKTUELLE SORTERINGSKRITERIER),
SAMT REGISTRERER DISSE I EN TABELL SOM KAN BRUKES VED
OPPDATERING AV ENGRADERSRUTEFORTEGNELSEN.

NØDVENDIGE RUTINER ; NTFEIL

SUBROUTINE IMIX(NUNA,NUNB,IFA,IFB,NUNUT,LHSORT,KU,NTELL,NDEFSO)

COMMON /DATA/ INNHA(10,200),INNHB(10,200),NUT(10,100)
DIMENSION NTELL(100),NDEFSO(100)
LOGICAL BEGGE,LHSORT
FORMAT(12)
FORMAT(4X,14,' BLOKKER MED ',13,' LINJER I SISTE BLOKK')
FORMAT(5X,5HUTGAR,18,4X,R4,218,2X,A6,218,2X,2A6,2X,A6)
FORMAT(5X,5HUTGAR,418,2X,A6,218,2X,2A6,2X,A6)
FORMAT(10X,418,2X,A6,218,2X,2A6,2X,A6)
FORMAT(10X,18,4X,R4,218,2X,A6,218,2X,2A6,2X,A6)
FORMAT(4X,'MALEDATAFIL:',16,' BLOKKER MED ',13,' LINJER ',
*' I SISTE BLOKK')
FORMAT(4X,15,' STASJONER ER FJERNET')
BEGGE=.TRUE.
KU=0
LU=1
ITVEKK=0

DO 90 J=1,100
NTELL(J)=0
CONTINUE

LES INN FØRSTE BLOKK I HVER FIL.

CALL NTRAN(NUNA,2,1000,INNHA,L,22)
CALL NTFEIL(1000,L,NUNA,1,1999)
CALL NTRAN(NUNB,2,1000,INNHB(1,1),L,22)
CALL NTFEIL(1000,L,NUNB,1,1999)

UNDERSØK OM EN ELLER FLERE BLOKKER I DEN ENE FILEN I SIN
HELHET KOMMER FØR FØRSTE STASJON I DEN ANDRE FILEN.

KBD=1
LIND=1
I=1
LA=1
LB=1
DECODE(47,INNHA(10,1)) NQA
NQA=NQA+1
NQA=NDEFSO(NQA)
IF (LHSCRT) CALL HTEST(1100,1120,1140)
CALL TEST(1100,1120,1140)
CONTINUE

FØRSTE STASJON KOMMER FRA FIL A.

CALL START(INNHA,NUNA,LA,-1,1999)
GO TO 140
CONTINUE

FØRSTE STASJON KOMMER FRA FIL B.

CALL START(INNHB,NUNB,LB,1,1999)
CONTINUE

START INNLESING AV DE TO NESTE BLOKKENE.

CALL NTRAN(NUNA,2,1000,INNHA(1,101),LSTATA)
CALL NTRAN(NUNB,2,1000,INNHB(1,101),LSTATB)

SETT LINJENUMMERPEKER.

```

```
000087 00J ASSIGN 155 TO LABEL
000088 003 IF (,NOT,LHSORT) ASSIGN 159 TO LABEL
000089 003 IF (INNHA(9,100),EQ,0) ASSIGN 150 TO LABEL
000090 003 IF (INNHB(9,100),EQ,0) ASSIGN 150 TO LABEL
6 000091 003 GO TO LABEL
000092 003 150 CONTINUE
8 000093 003 IF (INNHA(9,LA),EQ,0) GO TO 340
000094 003 IF (INNHB(9,LB),EQ,0) GO TO 440
10 000095 003 IF (,NOT,LHSORT) GO TO 159
000096 003 155 CONTINUE
12 000097 003 C
000098 003 C SAMMENLIGN EN STASJON I HVERT BUFFER.
14 000099 003 C
000100 003 CALL HTEST(1300,1400,1180)
16 000101 003 159 CONTINUE
000102 003 CALL TEST(1300,1400,1180)
18 000103 003 180 CONTINUE
000104 003 C
20 000105 005 C TO MAKE STASJONER ER PATRUFFET, OG STASJONEN I BUFFER A LISTES OG
000106 005 C SLETTES.
22 000107 003 C
000108 003 ITVEKK=ITVEKK+1
24 000109 003 IF (INNHA(2,LA),GT,9999) WRITE(6,26) (INNHA(JZ,LA),JZ=1,10)
000110 003 IF (INNHA(2,LA),LT,10000) WRITE(6,27) (INNHA(JZ,LA),JZ=1,10)
26 000111 003 IF (INNHB(2,LB),LT,10000) WRITE(6,28) (INNHB(JZ,LB),JZ=1,10)
000112 003 IF (INNHB(2,LB),GT,9999) WRITE(6,29) (INNHB(JZ,LB),JZ=1,10)
28 000113 003 DECODE (47,INNHA(10,LA)) JH
000114 003 JH=JH+1
30 000115 005 NTELL(JH)=NTELL(JH)+1 6REGISTRERING I TABELL.
000116 003 I=-1
000117 003 CALL HOPP(INNHA,LA,NUNA,1320,LSTAT)
000118 003 GO TO LABEL
32 000119 003 999 CALL ERTRAN(2)
000120 003 300 CONTINUE
34 000121 003 C
000122 003 C EN STASJON OVERFØRES FRA BUFFER A.
36 000123 003 C
000124 003 CALL HJELP(INNHA,LA,NUNA,1320,LSTAT)
000125 003 GO TO LABEL
40 000126 003 320 CONTINUE
42 000127 003 C
000128 003 C SISTE BLOKK I FIL A ER LEST INN.
44 000129 003 C
000130 003 IF (BEGGE) ASSIGN 150 TO LABEL

2 000131 003 IF (BEGGE) GO TO LABEL
000132 003 ASSIGN 325 TO LABEL
000133 003 325 CONTINUE
000134 003 IF (INNHA(9,LA),EQ,0) GO TO 500
6 000135 003 GO TO 300
000136 003 340 CONTINUE
8 000137 003 C
000138 003 C FIL A ER FERDIG, OG LINJENUMMERPEKEREN ENDRES SLIK AT
10 000139 003 C RESTEN AV FIL B SKRIVES UT.
000140 003 C
12 000141 003 BEGGE=.FALSE.
000142 003 ASSIGN 425 TO LABEL
14 000143 003 IF ((LB,LT,101),AND,(INNHB(9,100),NE,0)) ASSIGN 400 TO LABEL
000144 003 IF ((LB,GT,100),AND,(INNHB(9,200),NE,0)) ASSIGN 400 TO LABEL
16 000145 003 GO TO LABEL
000146 003 400 CONTINUE
18 000147 003 C
000148 003 C EN STASJON OVERFØRES FRA BUFFER B.
20 000149 003 C
000150 003 CALL HJELP(INNHB,LB,NUNB,1420,LSTATB)
22 000151 003 GO TO LABEL
000152 003 420 CONTINUE
24 000153 003 C
000154 003 C SISTE BLOKK I FIL B ER LEST INN.
26 000155 003 C
000156 003 IF (BEGGE) ASSIGN 150 TO LABEL
28 000157 003 IF (BEGGE) GO TO LABEL
000158 003 ASSIGN 425 TO LABEL
30 000159 003 425 CONTINUE
000160 003 IF (INNHB(9,LB),EQ,0) GO TO 500
32 000161 003 GO TO 400
000162 003 440 CONTINUE
34 000163 003 C
000164 003 C FIL B ER FERDIG, OG LINJENUMMERPEKEREN ENDRES SLIK AT
36 000165 003 C RESTEN AV FIL A SKRIVES UT.
000166 003 C
38 000167 003 BEGGE=.FALSE.
000168 003 ASSIGN 325 TO LABEL
40 000169 003 IF ((LA,LT,101),AND,(INNHA(9,100),NE,0)) ASSIGN 300 TO LABEL
000170 003 IF ((LA,GT,100),AND,(INNHB(9,200),NE,0)) ASSIGN 300 TO LABEL
42 000171 003 GO TO LABEL
000172 003 500 CONTINUE
44 000173 003 C
000174 003 C FERDIG.
```

```
2 000175 003 C
000176 003 CALL NTRAN(NQA,22)
4 000177 003 CALL NTRAN(NUNB,22)
000178 003 IF (LSTATA.EQ.-2) IFA=IFA+1
6 000179 003 IF (LSTATB.EQ.-2) IFB=IFB+1
000180 003 DO 520 J=LU,100
8 000181 003 DO 520 L=1,10
000182 003 NUT(L,J)=0
10 000183 003 520 CONTINUE
000184 003 CALL NTRAN(NUNUT,1,1000,NUT,L,22)
12 000185 003 KU=KU+1
000186 003 CALL NTFEIL(1000,L,NUNUT,KU,1999)
14 000187 003 L=LU-1
000188 003 WRITE(6,25) KU,L
16 000189 003 L=LIND-1
000190 003 WRITE(6,30) KBD,L
18 000191 003 WRITE(6,31) ITVEKK
000192 003 RETURN
20 000193 003 C
000194 003 SUBROUTINE START(NDIM,NUN,LS,IND,I)
22 000195 003 C
000196 003 C START UNDERSØKER OM EN ELLER FLERE BLOKKER I DEN ENE FILEN
24 000197 003 C (BUFFER ; NDIM) KAN SKRIVES UT I SIN HELHET FØR FØRSTE
000198 003 C STASJON I DEN ANDRE FILEN, HVILKEN FIL SOM EVENTUELT SKAL
26 000199 003 C SKRIVES UT, MA VÆRE BESTEMT PÅ FORHAND (VED SAMMENLIGNING
000200 003 C AV FØRSTE STASJON I FILENE).
28 000201 003 C
000202 003 DIMENSION NDIM(10,200)
30 000203 003 11 FORMAT(15,13,14)
000204 003 10 FORMAT(15,13)
32 000205 003 LS=100
000206 003 100 CONTINUE
34 000207 003 IF (NDIM(9,100).EQ.0) GO TO 250 6SISTE BLOKK I FILEN?
000208 003 IF (LH5ORT) CALL HTEST(1150,1160,1250)
36 000209 003 CALL TEST(1150,1160,1250)
000210 003 115 RETURN 5
38 000211 003 150 CONTINUE
40 000212 003 IF (IND.GT.0) GO TO 250
42 000213 003 GO TO 200
000214 003 160 CONTINUE
44 000215 003 IF (IND.LT.0) GO TO 250
000216 003 200 CONTINUE
46 000217 003 C
000218 003 C EN BLOKK SKRIVES UT, OG EN NY LESES INN.

000219 003 C
000220 003 KU=KU+1
000221 003 CALL NTRAN(NUNUT,1,1000,NDIM,L,22)
000222 003 CALL NTFEIL(1000,L,NUNUT,KU,115)
000223 003 DECODE(11,NDIM(8,100)) KBD,LIND,L 6NØDVENDIG I TILFELLE DETTE
000224 003 6 ER SISTE POST I FILEN
000225 003 LIND=LIND+L
000226 003 IF (LIND.GT.100) KBD=KBD+1
10 000227 003 IF (LIND.GT.100) LIND=LIND-100
000228 003 CALL NTRAN(NUN,2,1000,NDIM,L,22)
12 000229 003 CALL NTFEIL(1000,L,NUN,-20,115)
000230 003 GO TO 100
14 000231 003 250 CONTINUE
000232 003 C
16 000233 003 C EVENTUELLE HELE BLOKKER ER OVERFØRT, OG PARAMETRE FOR VIDERE
000234 003 C FLETNING BESTEMMES.
18 000235 003 C
000236 003 I=IND
20 000237 003 LS=1
000238 003 IF (NDIM(8,1).NE.0) DECODE (10,NDIM(8,1))KBD,LIND
22 000239 003 RETURN
000240 003 C
24 000241 003 SUBROUTINE HTEST(I,1,1)
000242 003 C
26 000243 003 C HTEST SAMMENLIGNER TO STASJONER (INKLUDERT SAMMENLIGNING
000244 003 C AV ENGRADERSRUTENUMMER MED TILHØRENDE SORTERINGSORDEN).
28 000245 003 C
000246 003 47 FORMAT(12)
30 000247 003 IF (I.GT.0) GO TO 156
000248 003 DECODE(47,INNHA(10,LA)) NQA
32 000249 003 NQA=NQA+1
000250 003 NQA=NDEF50(NQA)
34 000251 003 GO TO 157
000252 003 156 CONTINUE
36 000253 003 DECODE(47,INNHB(10,LB)) NQB
000254 003 NQB=NQB+1
38 000255 003 NQB=NDEF50(NQB)
40 000256 003 157 CONTINUE
000257 003 I=NQA-NQB
42 000258 003 IF (I) 300,159,400
000259 003 C
000260 003 ENTRY TEST(I,1,1)
44 000261 003 C
000262 003 C TEST SAMMENLIGNER AR,SKIP,DATO OG STASJONSNUMMER FOR TO STASJONER.
```



```
2 000263 003 C
000264 003 159 CONTINUE
4 000265 003 I=INNHA(3,LA)-INNHB(3,LB)
000266 003 IF (I)300,160,400
6 000267 003 160 CONTINUE
000268 003 I=INNHA(1,LA)-INNHB(1,LB)
8 000269 003 IF (I)300,170,400
000270 003 170 CONTINUE
10 000271 003 I=INNHA(4,LA)-INNHB(4,LB)
000272 003 IF (I) 300,175,400
12 000273 003 175 CONTINUE
000274 003 I=INNHA(2,LA)-INNHB(2,LB)
14 000275 003 IF (I)300,180,400
000276 003 180 RETURN 3
16 000277 003 300 RETURN 1
000278 003 400 RETURN 2
18 000279 003 C
000280 003 SUBROUTINE HJELP(NDIM,LZ,NUN,I,LS)
20 000281 003 C
000282 003 C HJELP OVERFØRER EN LINJE FRA ET BUFFER (NDIM) TIL UTSKRIFTSBUFFERET
22 000283 003 C (NUT),SAMT STYRER INNLESING OG UTSKRIVING AV DATA.
000284 003 C
24 000285 003 20 FORMAT(2X,I4)
000286 003 10 FORMAT(15,13,I4)
26 000287 003 DIMENSION NDIM(10,200)
000288 003 DO 100 JA=1,7
28 000289 003 NUT(JA,LU)=NDIM(JA,LZ)
000290 003 100 CONTINUE
30 000291 003 NUT(10,LU)=NDIM(10,LZ)
000292 003 DECODE(20,NDIM(9,LZ))JA GANTALL LINJER I STASJONEN.
32 000293 003 ENCODE(10,NUT(8,LU)) KBD,LIND,JA
000294 003 LU=LU+1
34 000295 003 IF(LU.LT.101) GO TO 150
000296 003 LU=1
36 000297 003 KU=KU+1
000298 003 CALL NTRAN(NUNUT,1,1000,NUT,L,22)
000299 003 CALL NTFEIL(1000,L,NUNUT,KU,1999)
000300 003 150 CONTINUE
000301 003 LIND=LIND+JA
000302 003 IF (LIND.LT.101) GO TO 200
000303 003 KBD=KBD+1
000304 003 LIND=LIND-100
40 000305 003 200 CONTINUE
000306 003 C
```

```
000307 003 ENTRY HOPP(NDIM,LZ,NUN,I,LS)
000308 003 C
000309 003 C HOPP BRUKES SEPARAT NAR EN STASJON SKAL UTELATES.
000310 003 C
6 000311 003 LZ=LZ+1
000312 003 IF ((LZ.NE.101).AND.(LZ.NE.201)) RETURN
000313 003 CALL NTRAN(NUN,22)
000314 003 CALL NTFEIL(1000,LS,NUN,-10,1999)
8 000315 003 JH=LZ-100
000316 003 CALL NTRAN(NUN,2,1000,NDIM(1,JH),LS)
10 000317 003 IF (LZ.EQ.201) LZ=1
000318 003 JH=LZ+99
12 000319 003 IF (NDIM(9,JH).EQ.0) RETURN 4
000320 003 RETURN
14 000321 003 999 CALL ERTRAN(2)
16 000322 003 END
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
```

MHSORT

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Dato : 19/12-75
Utført av : Ø. Strand
Formål : MHSORT sorterer en innholdsregister-
fil etter ønsket rekkefølge av engraders-
ruter.

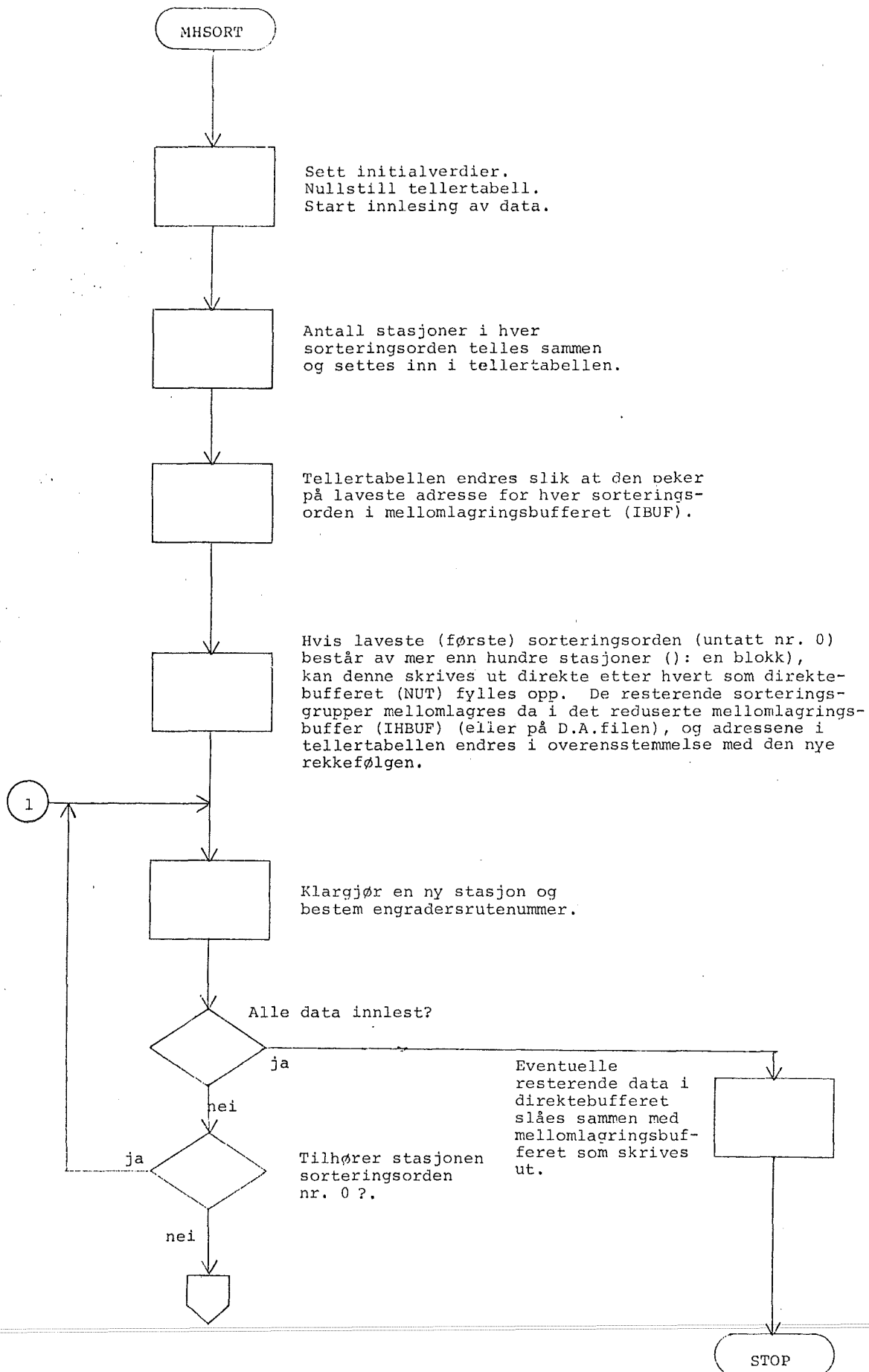
Kall: MHSORT(NUNI, NUNU, NUNDA, NTELL, NDEFSSO, IFIL, KU, LITAPE)

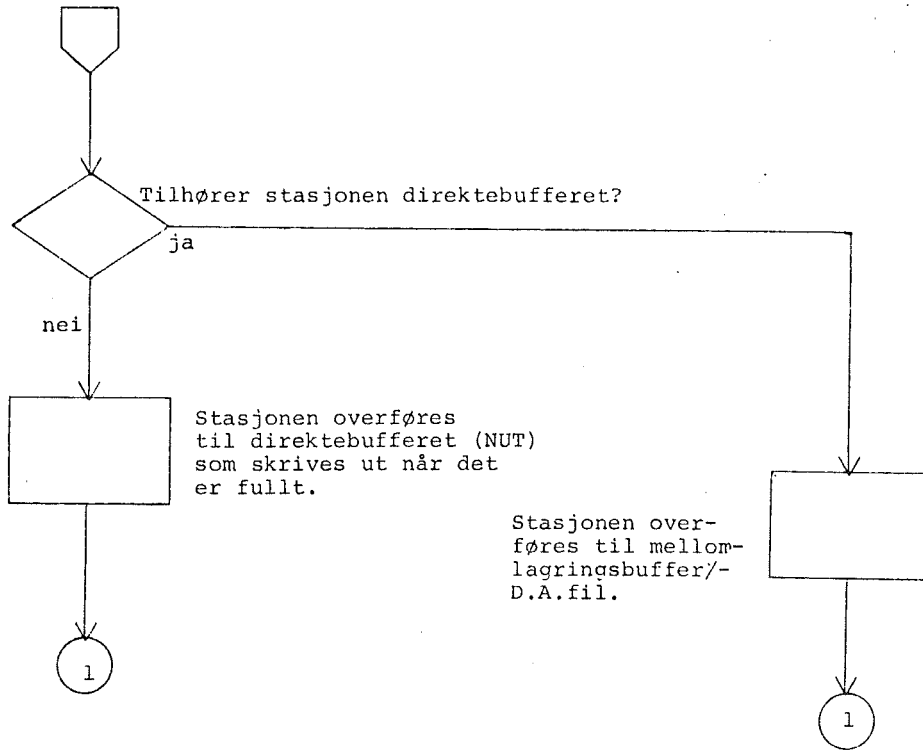
Generelt:

MHSORT brukes ved sortering av innholdsregisterfiler etter geografiske kriterier. Sorteringsgrunnlaget er engradersruter (standard nummerering innen hver Marsden-rute). Rutene samles i grupper (sorteringsordener) med en eller flere ruter i hver. Etter sortering, vil hver gruppe være samlet slik at den innbyrdes rekkefølgen i gruppen er uendret. Gruppene nummereres (0-100), og vil etter sortering være plassert i rekkefølge etter stigende nummer. (Sorteringsorden nr. 0 angir at stasjoner tilhørende vedkommende gruppe skal utelates). Ved sortering bruker rutinen et (eller flere) buffer(e) hvor stasjonene plasseres i riktig rekkefølge etter hvert som de leses inn. Hvis bufferet i primærlageret er for lite, "skjøtes" dette med en direkte aksessfil (D.A. fil).

Formelle parametre:

NAVN:	TYPE:	BRUK:
NUNI	INTEGER	I/0-enhetsnummer for innlesefil.
NUNU	INTEGER	I/0-enhetennummer for ny, sortert, fil.
NUNDA	INTEGER	I/0-enhetsnummer for direkte aksessfil (utvidet buffer).
NTELL(100)	INTEGER	Inneholder innlesefilens fordeling på engradersruter.
NDEFSSO(100)	INTEGER	Inneholder ønsket sorteringsrekkefølge for engradersrutene. Sorteringsrekkefølge 0-100 (maksimum). Verdien 0 angir at stasjoner tilhørende vedkommende rute skal fjernes.
IFIL	INTEGER	Filnummer (ved bruk av magnetbånd) for innlesefil.
KU	INTEGER	Teller antall blokker som skrives ut.
LITAPE	LOGICAL	Angir om innlesefilen er en magnetbåndfil (.TRUE.) eller platelagerfil.





COMMON variable:

DATA:

NAVN:	TYPE:	BRUK:
IBUF(10, 1100)	INTEGER	Stort mellomlagringsbuffer (1100 stasjoner).
INN(10, 200)	INTEGER	Innlesebuffer.
IHS(100)	INTEGER	Tellertabell.

Eksterne rutiner: NTFEIL
LESDB

Interne rutiner: FRABUF (Overfører data fra et mellomlagringsbuffer).
TILBUF (Overfører data til et mellomlagringsbuffer eller D.A. fil).

Variable:

NAVN:	TYPE:	BRUK:
NUT(10, 100)	INTEGER	Direktebuffer som brukes hvis laveste (første) forekommende sorteringsorden inneholder mer enn hundre stasjoner. Disse kan da overføres til den nye innholdsregisterfilen etter hvert som direktebufferet fylles opp, og opptar dermed mindre plass. Direktebufferet er identisk med de første hundre plassene i IBUF.
IHBUF(10, 1000)	INTEGER	Redusert mellomlagringsbuffer. Er identisk med de tusen siste plassene i IBUF. (Brukes hvis direktebufferet er i bruk).
IU	INTEGER	Linjenummer i direktebufferet.
K	INTEGER	Teller antall leste blokker.
LSTAT	INTEGER	Statusvariabel i NTRAN.
NTS	INTEGER	Viser høyeste sorteringsorden som er oppgitt i NDEFSSO.
MAXB	INTEGER	Teller antall stasjoner som skal til mellomlagringsbuffer/D.A. fil.
JMAX	INTEGER	Hjelpevariabel ved datainnlesing. Rutinen benytter dobbelbuffering ved innlesing, og JMAX viser øvre grense for den aktuelle halvdel av innlesebufferet.

NAVN:	TYPE:	BRUK:
JMIN	INTEGER	Hjelpevariabel som viser nedre grense for den aktuelle halvdel av innlesebufferet.
IP	INTEGER	Viser laveste sorteringsorden (untatt nr. 0) som ifølge NTELL inneholder data.
IDIR	INTEGER	Hjelpevariabel som brukes til å bestemme hvorvidt sorteringsorden IP kan skrives ut direkte.
LODIR	LOGICAL	Angir om (TRUE) direktebufferet er i bruk.
J	INTEGER	Hjelpevariabel.
L	INTEGER	Hjelpevariabel.
N	INTEGER	Hjelpevariabel.
M	INTEGER	Hjelpevariabel.
IS	INTEGER	Hjelpevariabel som viser nedre grense for data i mellomagringsbufferet ved utskrivning av data.
JJ	INTEGER	Hjelpevariabel.
LL	INTEGER	Hjelpevariabel.

```

4 000001 009 C
5 000002 009 C
6 000003 009 C
7 000004 009 C
8 000005 009 C
9 000006 009 C
10 000007 009 C
11 000008 009 C
12 000009 009 C
13 000010 009 C
14 000011 009 C
15 000012 009 C
16 000013 009 C
17 000014 009 C
18 000015 009 C
19 000016 009 C
20 000017 009 C
21 000018 007
22 000019 009 C
23 000020 007
24 000021 007
25 000022 007
26 000023 007
27 000024 007
28 000025 007
29 000026 007
30 000027 007
31 000028 007
32 000029 007
33 000030 007
34 000031 007
35 000032 007
36 000033 007
37 000034 007
38 000035 007
39 000036 007
40 000037 009
41 000038 007
42 000039 011 C
43 000040 011 C
44 000041 011 C
45 000042 007

          *****
          *
          *   MHSORT   *
          *
          *****

MHSORT SORTERER EN INNHOLDSREGISTERFIL ETTER OPPGITT
REKKEFØLGE FOR ENGRADERSRUTER.

          NØDVENDIGE RUTINER : NTFEIL
                              LESDB

SUBROUTINE MHSORT(NUNI,NUNU,NUNDA,NTELL,NDEF50,IFIL,KU,LITAPE)
DIMENSION NTELL(100),NDEF50(100)
COMMON/DATA/IBUF(10,100),INN(10,200),IHS(100)
DIMENSION NUT(10,100),IHBUF(10,1000)
LOGICAL LITAPE,LODIR
EQUIVALENCE(IBUF,NUT),(IHBUF,IBUF(1,101))
FORMAT(4X,'SORTERINGSGRUPPE: ',13,' HAR ',16,' STASJONER')
FORMAT(14X,'TOTALT: ',16,' STASJONER')
FORMAT(12)
FORMAT(4X,15,' BLOKKER PLUSS ',13,' LINJER DIREKTE,RESTEN',
*15,' POSTER')
FORMAT(4X,'FILNUMMER ER ØKET')
FORMAT(4X,'LEST: ',15,' BLOKKER MED ',13,' POSTER I SISTE BLOKK')
FORMAT(4X,'SKREVET: ',15,' BLOKKER MED ',13,
* ' POSTER I SISTE BLOKK')
KU=0
IU=1
K=0
CALL NTRAN(NUNI,2,1000,INN(1,1),LSTAT)  @START INNLESING
JMAX=200

TELLERTABELLEN NULLSTILLES.

DO 100 J=1,100

          IHS(J)=0
          CONTINUE

          ANTALL STASJONER I HVER SORTERINGSORDEN SETTES INN
          I TELLERTABELLEN.

          NTS=0
          DO 120 J=1,100
          N=NDEF50(J)
          IF(N.EQ.0) GO TO 120
          IF(NTELL(J).EQ.0) GO TO 120
          IF(N.GT.NTS)NTS=N
          IHS(N)=IHS(N)+NTELL(J)
          CONTINUE
          120

          TELLERTABELLEN ENDRES SLIK AT DEN PEKER PÅ LAVESTE ADRESSE
          FOR HVER SORTERINGSORDEN.

          MAXB=0
          DO 130 J=1,NTS
          WRITE(6,10)J,IHS(J)
          IF(MAXB.LT.100)IDIR=J
          IF(MAXB.EQ.0) IP=J
          M=IHS(J)
          IHS(J)=MAXB+1
          MAXB=MAXB+M  @ANTALL STASJONER I SORTERINGSORDEN I T.O.M. J.
          CONTINUE
          130
          WRITE(6,11)MAXB
          LODIR=.FALSE.
          IF(IDIR.NE.IP) GO TO 160

          SORTERINGSORDEN NR. IP KAN SKRIVES UT DIREKTE,OG ADRESSENE
          FOR DE ØVRIGE SORTERINGSGRUPPER ENDRES.

          LODIR=.TRUE.
          N=IP+1
          M=IHS(N)-1  @ANTALL STASJONER SOM SKRIVES UT DIREKTE.
          DO 150 J=N,NTS
          IHS(J)=IHS(J)-M
          CONTINUE
          150
          MAXB=MAXB-M  @ANTALL STASJONER TIL MELLOMLAGRINGSBUFFER/D.A. FIL
          CONTINUE
          160
          CALL LESDB(INN,10,NUNI,K,JMAX,LSTAT,1999)
          JM=N-JMAX

```

```
000087 007 JMAX=JMIN*99
000088 009 C
000089 009 C EN BLOKK FORDELES
000090 009 C
000091 007 DO 400 J=JMIN,JMAX
000092 007 IF(INN(8,J).EQ.0) GO TO 450
000093 009 DECODE(12,INN(10,J))N BENGRADESRUTENUMMER
000094 007 N=N+1
000095 007 M=NDEFSC(N)
000096 009 IF(M,EQ.0) GO TO 400 BSTASJONEN UTGAR
000097 007 IF((LODIR).AND.(M,EQ.IP)) GO TO 200
000098 009 C
000099 009 C STASJONEN OVERFØRES TIL MELLOMLAGRINGSBUFFERET/D.A. FIL
000100 011 C
000101 007 I=IHS(M)
000102 011 IHS(M)=I+1 BTELLERTABELLENS ADRESSEPEKER ØKES.
000103 007 IF(LODIR)CALL TILBUF(IHBUF,1000,J,1)
000104 007 IF(.NOT.LODIR)CALL TILBUF(IBUF,1100,J,1)
000105 007 GO TO 400
000106 007 200 CONTINUE
000107 009 C
000108 009 C STASJONEN SKRIVES UT DIREKTE.
000109 009 C
000110 007 DO 250 L=1,10
000111 007 NUT(L,1L)=INN(L,J)
000112 007 250 CONTINUE
000113 007 IU=IU+1
000114 007 IF(IU.LT.101) GO TO 400
000115 007 KU=KU+1
000116 007 IU=1
000117 007 CALL NTRAN(NUNU,1,1000,NUT,L,22)
000118 007 CALL NTFEIL(1000,L,NUNU,KU,1999)
000119 007 400 CONTINUE
000120 007 GO TO 160
000121 007 450 CONTINUE
000122 009 C
000123 009 C FERDIG MED INNLESING.
000124 009 C
000125 007 CALL NTRAN(NUNI,22)
000126 007 IF((LSTAT.EG.-2).AND.(LITAPE))IFIL=IFIL+1
000127 007 IF((LSTAT.EG.-2).AND.(LITAPE))WRITE(6,14)
000128 007 J=J-1
000129 009 IF (J.GE.100) J=J-100
000130 007 WRITE(6,15)K,J

000131 007 L=IU-1
000132 007 WRITE(6,13)KU,L,MAXB
000133 007 IS=1
000134 007 IF(.NOT.LODIR) GO TO 650
000135 009 IF(MAXB.LT.1) GO TO 900 BINGEN DATA I MELLOMLAGRINGSBUFFERET?
000136 007 IF(IU.EG.1) GO TO 520
000137 007 IF(IU.LT.51) GO TO 550
000138 009 C
000139 009 C DE FØRSTE STASJONENE I MELLOMLAGRINGSBUFFERET OVERFØRES TIL
000140 009 C DIREKTEBUFFERET.
000141 009 C
000142 007 DO 500 J=IS,MAXB
000143 007 DO 480 L=1,10
000144 007 NUT(L,1L)=IHBUF(L,J)
000145 007 480 CONTINUE
000146 007 IU=IU+1
000147 007 IF(IU.EG.101) GO TO 510
000148 007 500 CONTINUE
000149 007 GO TO 900
000150 007 510 CONTINUE
000151 007 IS=J+1
000152 007 IU=1
000153 007 CALL NTRAN(NUNU,1,1000,NUT,L,22)
000154 007 KU=KU+1
000155 007 CALL NTFEIL(1000,L,NUNU,KU,1999)
000156 007 520 CONTINUE
000157 009 C
000158 009 C DATA FRA MELLOMLAGRINGSBUFFERET.
000159 009 C
000160 007 CALL FRABUF(IHBUF,1000,IS,MAXB,1900)
000161 007 GO TO 700
000162 007 550 CONTINUE
000163 009 C
000164 009 C RESTEN AV DIREKTEBUFFERET SLÆS SAMMEN MED MELLOMLAGRINGSBUFFERET.
000165 009 C
000166 007 IU=IU-1
000167 007 DO 600 J=IU,1,-1
000168 007 M=100-J+1
000169 007 DO 580 L=1,10
000170 007 NUT(L,M)=NUT(L,J)
000171 007 580 CONTINUE
000172 007 600 CONTINUE
000173 007 MAXB=MAXB+100
000174 007 IS=101-IU
```



```
000175 007 IU=1
000176 007 CONTINUE
000177 009 C
000178 011 C UTSKRIVNING AV STORT BUFFER ( ):SAMTLIGE/RESTERENDE DATA ER PLASSERT
000179 010 C I BUFFERET ELLER PA D.A. FIL)
000180 010 C
000181 007 CALL FRABUF(1BUF,1100,15,MAXB,1900)
000182 007 700 CONTINUE
000183 007 IF (MAXB.LT.1) GO TO 900
000184 009 C
000185 010 C DATA FRA D.A. FIL
000186 009 C
000187 007 DO 750 J=1,MAXB
000188 007 READ(NUNDA,J)(NUT(L,IU),L=1,10)
000189 007 IU=IU+1
000190 007 IF (IU.LT.101) GO TO 750
000191 007 IU=1
000192 007 KU=KU+1
000193 007 CALL NTRAN(NUNU,1,1000,NUT,L,22)
000194 007 CALL NTFEIL(1000,L,NUNU,KU,1999)
000195 007 750 CONTINUE
000196 007 900 CONTINUE
000197 009 C
000198 009 C FERDIG!
000199 009 C
000200 007 DO 950 J=IU,100
000201 007 DO 950 L=1,10
000202 007 NUT(L,J)=0
000203 007 950 CONTINUE
000204 007 CALL NTRAN(NUNU,1,1000,NUT,L,22)
000205 007 KU=KU+1
000206 007 CALL NTFEIL(1000,L,NUNU,KU,1999)
000207 007 IU=IU-1
000208 007 WRITE(6,16)KU,IU
000209 007 RETURN
000210 007 999 CALL ERTRAN(2)
000211 009 C
000212 007 SUBROUTINE FRABUF(NDIM,IL,MIN,MAX,I)
000213 009 C
000214 011 C FRABUF OVERFØRER DATA FRA ET MELLOMLAGRINGSBUFFER.
000215 009 C
000216 007 DIMENSION NDIM(10,IL)
000217 007 IF (MAX.LT.MIN)RETURN 5
000218 007 L=MAX

000219 007 IF (L.GT.IL)L=IL
000220 007 J=L-MIN+1
000221 007 JJ=J/100
000222 007 M=MIN+100*JJ
000223 007 IF (M.GT.L) GO TO 110
000224 008 DO 100 J=M,L
000225 008 DO 95 LL=1,10
000226 007 NUT(LL,IU)=NDIM(LL,J)
000227 008 95 CONTINUE
000228 007 IU=IU+1
000229 007 100 CONTINUE
000230 007 110 CONTINUE
000231 007 IF (JJ.EQ.0) GO TO 160
000232 007 DO 150 LL=1,JJ
000233 007 M=MIN+100*(LL-1)
000234 007 CALL NTRAN(NUNU,1,1000,NDIM(1,M),LSTAT,22)
000235 007 KU=KU+1
000236 007 CALL NTFEIL(1000,LSTAT,NUNU,KU,1999)
000237 007 150 CONTINUE
000238 007 160 CONTINUE
000239 007 IF (IL.GE.MAX) RETURN 5
000240 007 MAX=MAX-IL
000241 007 RETURN
000242 007 999 CALL ERTRAN(2)
000243 009 C
000244 007 SUBROUTINE TILBUF(NDIM,IL,JZ,IZ)
000245 009 C
000246 009 C TILBUF OVERFØRER DATA TIL MELLOMLAGRINGSBUFFERET ELLER D.A.FILEN
000247 009 C
000248 007 DIMENSION NDIM(10,IL)
000249 007 IF (IZ.GT.IL) GO TO 150
000250 007 DO 100 L=1,10
000251 007 NDIM(L,IZ)=INN(L,JZ)
000252 007 100 CONTINUE
000253 007 RETURN
000254 007 150 CONTINUE
000255 007 IZ=IZ-IL
000256 007 WRITE(NUNDA,IZ)(INN(L,JZ),L=1,10)
000257 007 RETURN
000258 007 END
```

OMSKR

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Dato : 18/12-75
Utført av : Ø. Strand
Formål : OMSKR skriver om en måledatafil
etter rekkefølgen i en innholds-
registerfil.

Kall: OMSKR(NUNI, NUNH, JUNUNI, JUNUNH, KI, KU, HTAPE, VDA, IDFIL, IADA)

Generelt:

OMSKR lager, på grunnlag av en måledatafil og en tilhørende innholdaregisterfil (som kan være sortert, stokket om eller redusert) en ny måledatafil med tilhørende innholdsregisterfil (): med korrigerede adresser). Rutinen kan arbeide på to forskjellige måter: Måledatafilen kan i sin helhet mellomlagres på en direkte aksessfil (D.A.fil, I/O-enhet 29) og leses inn igjen fra denne i korrekt rekkefølge, eller den kan leses inn ved "frem-og-tilbakespoling" på originalfilen. Ved den første fremgangsmåten benyttes dobbelbuffering, og denne bør brukes hvis originalfilens rekkefølge skal endres sterkt. Direkte innlesing egner seg best hvis rekkefølgen bare skal gjennomgå mindre endringer.

Formelle parametre:

NAVN:	TYPE:	BRUK:
NUNI	INTEGER	I/O-enhetsnummer for innholdsregisterfil (leses).
NUNH	INTEGER	I/O-enhetsnummer for måledatafil (leses).
JUNUNI	INTEGER	I/O-enhetsnummer for ny, korrigert, innholdsregisterfil.
JUNUNH	INTEGER	I/O-enhetsnummer for ny måledatafil.
KI	INTEGER	Antall innleste blokker fra innholdsregisterfil.
KU	INTEGER	Antall blokker med måledata som skrives ut.
HTAPE	LOGICAL	Angir hvorvidt måledata leses fra magnetbånd (TRUE) eller platelagerfil.
VDA	LOGICAL	Angir hvorvidt måledatafilen skal mellomlagres på en D.A. fil (TRUE).
IDFIL	INTEGER	Filnummer (på eventuelt magnetbånd) for måledatafil (original).
IADA	INTEGER	Postnummer i D.A. filen.

COMMON variable:

DATA:

NAVN:	TYPE:	BRUK:
INNA(14, 200)	INTEGER	Innlesebuffer for måledata (ved dobbel- buffring).
INNH(10, 100)	INTEGER	Buffer for innholdsregister.
NUT(14, 100)	INTEGER	Buffer for ny måledatafil.

Variable:

NAVN:	TYPE:	BRUK:
INN(14, 100)	INTEGER	Innlesebuffer for måledata (er identisk med første halvdel av INNA).
KH	INTEGER	Peker til blokknummer i måledatafil (ved innlesing uten mellomlagring).
IU	INTEGER	Linjenummer i måledatabuffer.
LSTAT	INTEGER	Statusvariabel i NTRAN.
JZ	INTEGER	Blokknummer for første linje i en måle- dataserie som skal hentes frem.
MA	INTEGER	Linjenummer for første linje i måle- dataserie (i blokk JZ).
MC	INTEGER	Antall linjer i en stasjon (måledata).
I	INTEGER	Hjelpevariabel.
J	INTEGER	Hjelpevariabel.
JJ	INTEGER	Hjelpevariabel.
L	INTEGER	Hjelpevariabel.
MD	INTEGER	Hjelpevariabel.
MB	INTEGER	Hjelpevariabel.

Faste I/O-enhetsnumre:

29 (mellomlagringsfil. Må defineres
for direkte aksessbruk i hovedpro-
grammet).

```

000001 008 C
000002 008 C
000003 008 C
000004 008 C
000005 008 C
000006 008 C
000007 008 C
000008 008 C
000009 008 C
000010 008 C
000011 008 C
000012 008 C
000013 008 C
000014 008 C
000015 008 C
000016 008 C
000017 008 C
000018 008 C
000019 008 C
000020 008 C
000021 008 C
000022 008 C
000023 008 C
000024 005
000025 005
000026 008 C
000027 008 C
000028 005
000029 007
000030 006
000031 006
000032 005
000033 005 40
000034 005 45
000035 005
000036 005 56
000037 008 41
000038 008
000039 005
000040 008 C
000041 008 C
000042 008 C

```

```

*****
*
*   OMSKR   *
*
*****

```

```

OMSKR LAGER,PA GRUNNLAG AV EN MALEDATAFIL OG DEN TILHØRENDE
INNHOIDSREGISTERFIL (ELLER DELER AV DENNE),EN NY MALEDATAFIL
MED SAMME REKKEFØLGE SOM I INNHOIDSREGISTERFILEN.
SAMTIDIG LAGES EN NY INNHOIDSREGISTERFIL MED KORRIGERTE
ADRESSEPARAMETRE.

```

```

NØDVENDIGE RUTINER : NTFEIL
                    LESDB

```

```

RESERVERTE I/O-NUMRE : 29 (DIREKTE AKSESSFIL)

```

```

SUBROUTINE OMSKR(NUNI,NUNH,JUNUNI,JUKUNH,KI,KU,HTAPE,VDA,IDFIL,
*IADA)

```

```

LOGICAL HTAPE
DIMENSION INN(14,100)
COMMON /DATA/INNA(14,200),INNH(10,100),NUT(14,100)
EQUIVALENCE (INN,INNA)
LOGICAL VDA
FORMAT(4X,I3,' LINJER')
FORMAT(4X,I5,' BLOKKER(INNH)',10X,'OUTPUT:',I5,' BLOKKER',I3,
*' LINJER I SISTE BLOKK')
FORMAT(15,I3,I4)
FORMAT(4X,'***** OMSKR KALLES *****')
WRITE(6,41)
IF (.NOT.VDA) GO TO 110

```

```

MALEDATAFILEN OVERFØRES TIL EN DIREKTE AKSESSFIL (I/O-ENHET 29)

```

```

000043 005
000044 005
000045 005
000046 006
000047 005
000048 005
000049 005 100
000050 005
000051 005
000052 006
000053 005
000054 005 105
000055 005
000056 005
000057 005 57
000058 005
000059 005 110
000060 005
000061 005
000062 005
000063 005
000064 005 150
000065 005 C
000066 005 C
000067 005 C
000068 005
000069 005
000070 005
000071 005
000072 005
000073 009
000074 008
000075 005
000076 005
000077 005
000078 005 C
000079 005 C
000080 005 C
000081 005
000082 005
000083 005
000084 005
000085 005
000086 005

```

```

FIND(29'I)
CALL NTRAN(NUNH,2,1400,INN,L,22)
CALL NTFEIL(1400,L,NUNH,1,1999)
CALL NTRAN(NUNH,2,1400,INNA(1,101),LSTAT)
KH=1
J=1
CONTINUE
IF (INNA(1,J).EQ.0) GO TO 105
WRITE(29'IADA) (INNA(L,J),L=1,14)
CALL LESDB(INNA,14,NUNH,KH,J,LSTAT,1999)
GO TO 100
CONTINUE
CALL NTRAN(NUNH,22)
IF ((HTAPE).AND.(LSTAT.EQ.=2)) IDFIL=IDFIL+1
FORMAT(4X,'D.A. ',3I6)
WRITE(6,57) KH,IADA,LSTAT
CONTINUE
KI=0
KU=1
KH=0
IU=1
CONTINUE
EN BLOKK LESES INN FRA INNHOIDSREGISTERET.
CALL NTRAN(NUNI,2,1000,INNH,L,22)
KI=KI+1
CALL NTFEIL(1000,L,NUNI,KI,1999)
DO 500 J=1,100
IF (INNH(8,J).EQ.0) GO TO 590
DECODE(56,INNH(8,J)) JZ,MA,MC      6ADRESSEN DEKODES.
ENCODE(56,INNH(8,J)) KU,IU,MC     6NYE ADRESSER SETTES INN.
IF (VDA) GO TO 185
IF (KH.EQ.JZ) GO TO 200
IF (HTAPE) GO TO 180
LES INN RETT BLOKK (DISC).
I=JZ-KH
IF (I.LT.0) CALL NTRAN(NUNH,10,22)
IF (I.LT.0) I=JZ
DO 170 JJ=1,I
CALL NTRAN(NUNH,2,1400,INN,L,22)
CALL NTFEIL(1400,L,NUNH,JJ,1999)

```

```
000087 005 170 CONTINUE
000088 005 KH=JZ
000089 005 GO TO 200
000090 005 180 CONTINUE
000091 005 C
000092 005 C LES INN RETT BLOKK (MAGNETBAND).
000093 005 C
000094 005 I=JZ-KH-1
000095 005 IF(I.NE.0)CALL NTRAN(NUNH,7,1)
000096 005 CALL NTRAN(NUNH,2,1400,INN,L,22)
000097 005 KH=JZ
000098 005 CALL NTFEIL(1400,L,NUNH,KH,1999)
000099 005 GO TO 200
000100 005 185 CONTINUE
000101 005 C
000102 008 C MALEDATAFILEN ER MELLOMLAGRET PA EN DIREKTE AKSESSFIL.
000103 005 C
000104 005 L=100*(JZ-1)+MA
000105 005 FIND (29'L)
000106 005 DO 190 I=1,MC
000107 005 READ(29'TADA) (NUT(L,IU),L=1,14)
000108 005 IU=IU+1
000109 005 IF (IU.LT.101) GO TO 190
000110 005 IU=1
000111 005 CALL NTRAN(JUNUH,1,1400,NUT,L,22)
000112 005 CALL NTFEIL(1400,L,JUNUH,KU,1999)
000113 005 KU=KU+1
000114 005 190 CONTINUE
000115 005 GO TO 500
000116 005 200 CONTINUE
000117 005 C
000118 009 C STASJONEN OVERFØRES TIL UTSKRIVNINGSBUFFERET (NUT)
000119 005 C
000120 005 MD=MA+MC-1
000121 005 MB=MD
000122 005 IF (MB.GT.100) MB=100
000123 005 220 CONTINUE
000124 005 DO 300 I=MA,MB
000125 005 DO 250 L=1,14
000126 005 NUT(L,IU)=IKN(L,I)
000127 005 250 CONTINUE
000128 005 IU=IU+1
000129 005 IF(IU.LT.101)GO TO 300
000130 005 IU=1

000131 005 C
000132 005 C ET BUFFER SKRIVES UT (DATA).
000133 005 C
000134 005 CALL NTRAN(JUNUH,1,1400,NUT,L,22)
000135 005 CALL NTFEIL(1400,L,JUNUH,KU,1999)
000136 005 KU=KU+1
000137 005 300 CONTINUE
000138 005 IF (MB.NE.100) GO TO 500
000139 005 IF (MD.EQ.100) GO TO 500
000140 005 C
000141 005 C EN NY BLOKK LESES INN (DATA).
000142 005 C
000143 005 CALL NTRAN(NUNH,2,1400,INN,L,22)
000144 005 KH=KH+1
000145 005 CALL NTFEIL(1400,L,NUNH,KH,1999)
000146 005 MA=1
000147 005 MB=MD-100
000148 005 GO TO 220
000149 005 500 CONTINUE
000150 008 C
000151 008 C INNHOLDSREGISTERBUFFERET (MED KORRIGERTE ADRESSER)
000152 008 C SKRIVES UT.
000153 008 C
000154 005 CALL NTRAN(JUNUNI,1,1000,INNH,L,22)
000155 005 CALL NTFEIL(1000,L,JUNUNI,KI,1999)
000156 005 GO TO 150
000157 005 590 CONTINUE
000158 005 C
000159 005 C FERDIG.
000160 005 C
000161 005 CALL NTRAN(JUNUNI,1,1000,INNH,L,22)
000162 005 CALL NTFEIL(1000,L,JUNUNI,KI,1999)
000163 005 WRITE(6,40)J
000164 005 DO 650 I=1U,100
000165 005 DO 650 L=1,14
000166 005 NUT(L,I)=0
000167 005 650 CONTINUE
000168 005 CALL NTRAN(JUNUH,1,1400,NUT,L,22)
000169 005 CALL NTFEIL(1400,L,JUNUH,KU,1999)
000170 005 IU=IU-1
000171 005 WRITE(6,45)KI,KU,IU
000172 005 RETURN
000173 005 999 CALL ERTRAN(2)
000174 005 END
```

SORTER

Programmeringsspråk : FORTRAN V (UNIVAC)
Programtype : Subrutine
Dato : 1/5-76
Utført av : Ø. Strand
Formål : SORTER sorterer en innholdsregister-
fil etter år, skip, dato og stasjons-
nummer.

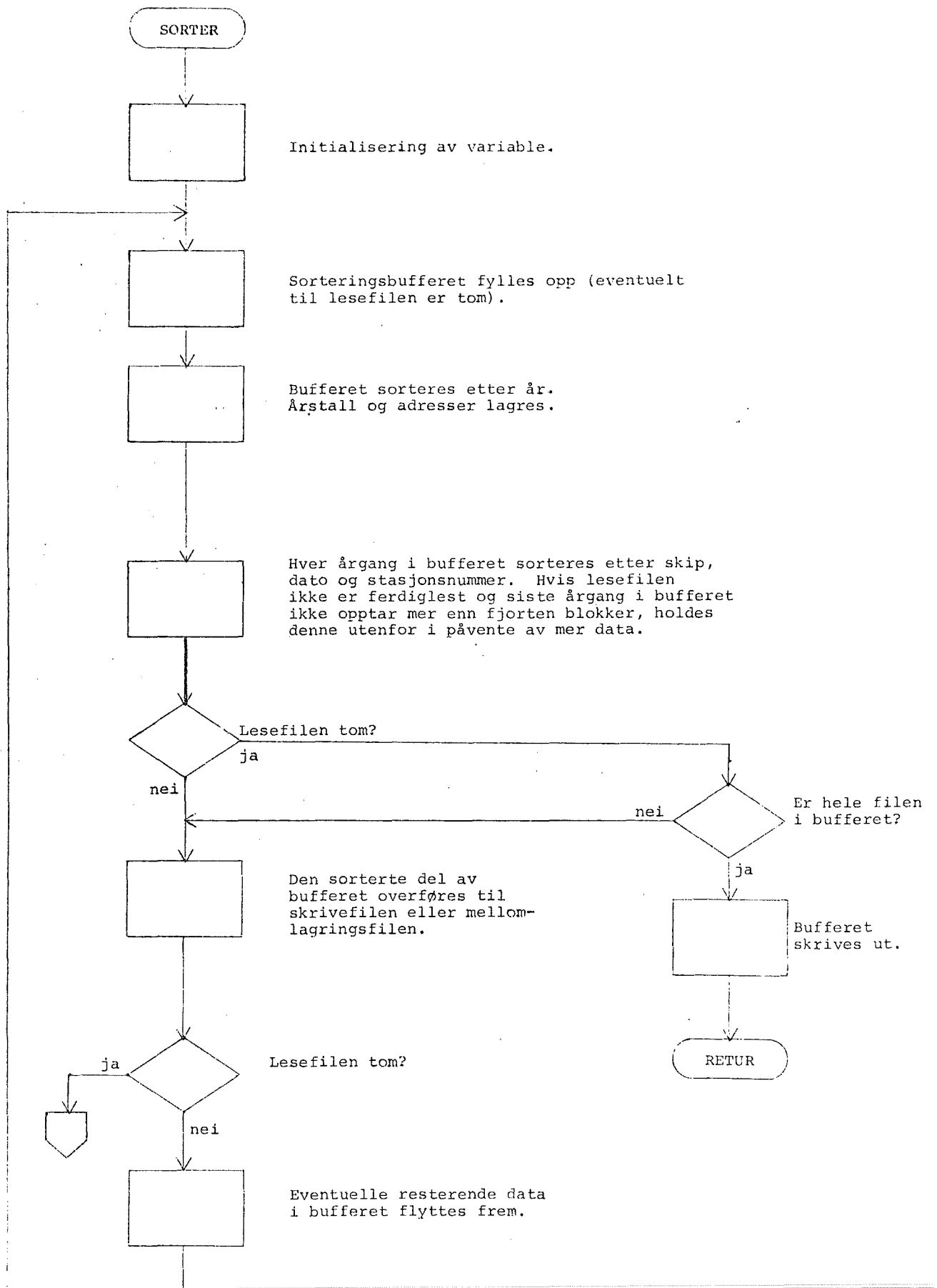
Kall: SORTER(NUNIT, NUTU, IUFIL, LOSP, UTDISC, REKKE, K)

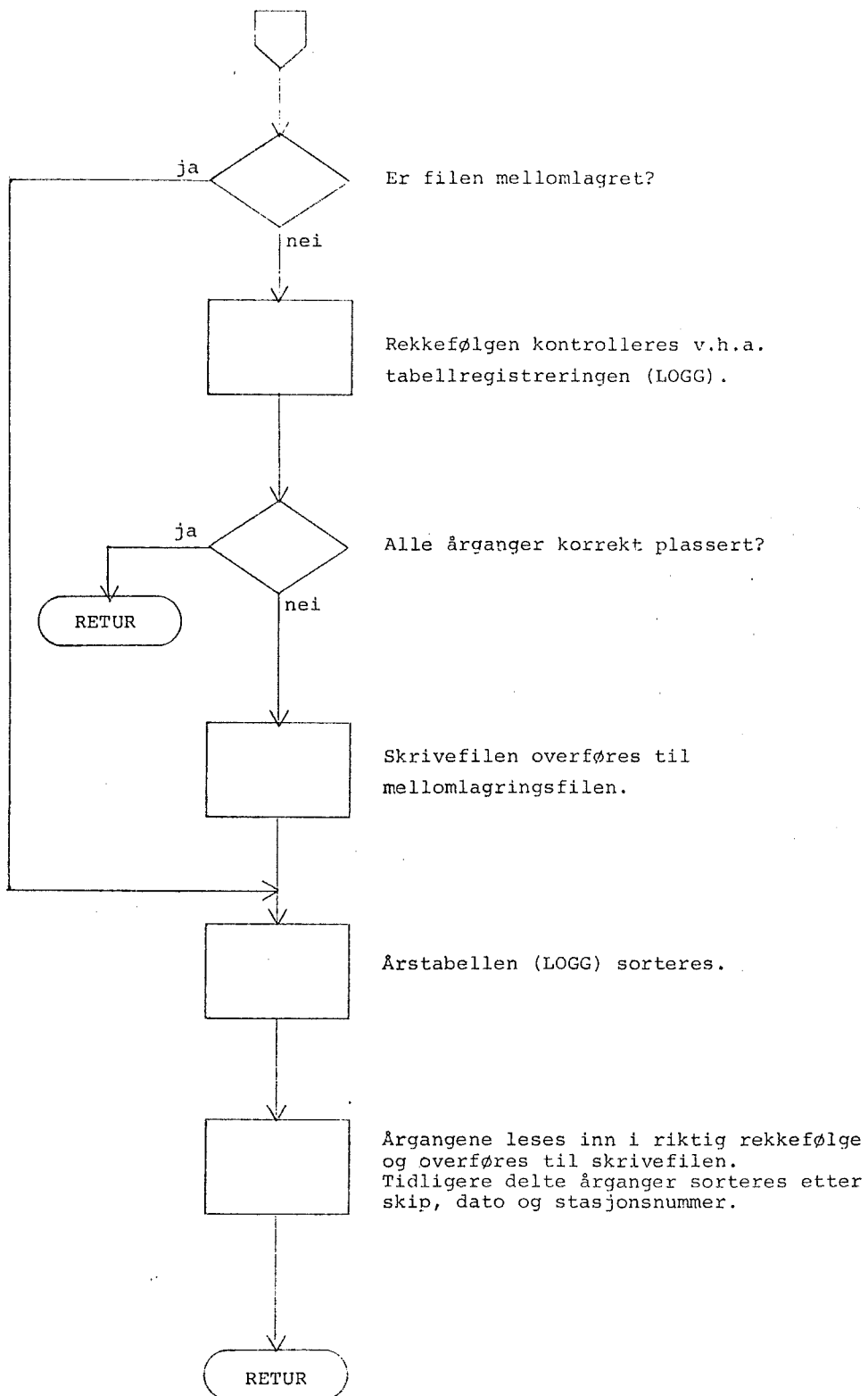
Generelt:

SORTER sorterer en innholdsregisterfil etter år, skipsnummer, dato og stasjonsnummer. Rutinen er primært laget for å sortere data som allerede ligger i noenlunde riktig rekkefølge. Rutinen arbeider ved hjelp av et sorteringsbuffer med plass til 1500 stasjoner. Dette bufferet sorteres først etter stigende årstall. Deretter sorteres hver forekommende årgang etter stigende skipsnummer, og tilslutt sorteres hvert skip etter dato og stasjonsnummer. Når bufferet er ferdig sortert, skrives det ut på en fil. Hvis der er mer data igjen i den usorterte filen, holdes siste årgang tilbake i påvente av eventuelle flere data tilhørende samme årgang. Dersom en fil inneholder mer enn 1500 stasjoner, kan den behandles på to forskjellige måter: Den kan overføres direkte fra sorteringsbufferet til utskriftsfilen etterhvert som dette er ferdig sortert, eller den kan overføres til en mellomlagringsfil (I/O-enhet 7). Etter direkte overføring til utskriftsfilen, kontrolleres årsgangsrekkefølgen. Hvis denne er feil, leses filen inn igjen og ut på mellomlagringsfilen. Fra mellomlagringsfilen. Hvis en årgang er skilt (): har forekommet svært spredt i utgangsmaterialet) leses denne inn samlet, og sorteres påny etter skip, dato og stasjonsnummer.

Formelle parametre:

NAVN:	TYPE:	BRUK:
NUNIT	INTEGER	I/O-enhet for usortert fil (lesefil).
NUTU	INTEGER	I/O-enhet for sortert fil (utskriftsfil).
IUFIL	INTEGER	Filnummer (på magnetbånd) for utskriftsfil.
LOSP	LOGICAL	Logisk variabel som angir utskriftsmåte etter første gangs behandling i sorteringsbufferet (mer enn 1500 stasjoner). Verdien TRUE medfører at data mellomlagres på I/O-enhet 7. Verdien FALSE medfører direkte utskrift til utskriftsfilen (med senere kontroll og, om nødvendig, ny innlesing).
UTDISC	LOGICAL	Angir filtype for utskriftsfilen (TRUE = platelagerfil, FALSE = magnetbånd).





NAVN:	TYPE:	BRUK:
REKKE	LOGICAL	Angir om rekkefølgen i innlesefilen er korrekt (): ingen endringer er gjort).
K	INTEGER	Antall blokker.

Eksterne rutiner : NTFEIL

Interne rutiner : HSORTA (sorterer bufferet etter år eller skip).
HJELP (sorterer etter dato og stasjonsnummer).

Faste I/O-enhetsnumre : 7 (mellomlagring).

COMMON variable:

DATA:

NAVN:	TYPE:	BRUK:
NSORT(10, 1500)	INTEGER	Sorteringsbuffer.
LOGG(2, 100)	INTEGER	Tabell over år og antall stasjoner som er skrevet ut på utskriftsfil/mellomlagringsfil.
NYEAR(2, 100)	INTEGER	Tabell over år og høyeste linjenummer for hvert år i et sorteringsbuffer (brukes ved sortering av et buffer etter år).
NSHIP(2, 100)	INTEGER	Tabell over skipsnummer og høyeste linjenummer for hvert skip i et år (sorteringsbufferet). Brukes ved sortering etter skipsnummer.
LOGGH(100)	INTEGER	Brukes ved innlesing og sortering av årganger fra mellomlagringsfilen. LOGGH viser opprinnelig rekkefølge for årgangene (mens LOGG sorteres).

Variable:

NAVN:	TYPE:	BRUK:
IL	INTEGER	Linjenummer i LOGG (1-100).
KN	INTEGER	Høyeste linjenummer i sorteringsbufferet.
KM	INTEGER	Hjelpevariabel.
JZZ	INTEGER	Hjelpevariabel.
MB	INTEGER	Hjelpevariabel.

NAVN:	TYPE:	BRUK:
J	INTEGER	Hjelpevariabel.
M	INTEGER	Hjelpevariabel.
IS	INTEGER	Hjelpevariabel.
KK	INTEGER	Antall årganger som behandles i sorteringsbufferet (=IY eller IY-1).
N	INTEGER	Antall skip i en årgang. Hjelpevariabel.
JI	INTEGER	Hjelpevariabel.
MA	INTEGER	Hjelpevariabel.
NV	INTEGER	Hjelpevariabel.
I	INTEGER	Hjelpevariabel.
JJ	INTEGER	Hjelpevariabel.
KA	INTEGER	Hjelpevariabel.
L	INTEGER	Hjelpevariabel.
II	INTEGER	Hjelpevariabel.
IZ	INTEGER	Hjelpevariabel.
JZ	INTEGER	Hjelpevariabel.
A	LOGICAL	Hjelpevariabel.
BOOL	LOGICAL	Ferdig/ikke ferdig med innlesing av data (TRUE): ferdig).
IY	INTEGER	Antall årganger i et buffer.
LESH	INTEGER	Startpunkt (linjenummer) i bufferet ved innlesing av en blokk.
LINY	INTEGER	Teller i LOGG.
KHUT	INTEGER	Hjelpevariabel. (linjepeker/blokkpeker i NSORT).
LUT	INTEGER	Hjelpevariabel.
NFLYTT(10)	INTEGER	Hjelpebuffer ved ombytting i NSORT.
NFMAX	INTEGER	Hjelpevariabel ved ombytting i NSORT.
NFMIN	INTEGER	Hjelpevariabel ved ombytting i NSORT.
IDEL	INTEGER	Hjelpevariabel ved ombytting i NSORT.
JTIL	INTEGER	Hjelpevariabel ved ombytting i NSORT.

NAVN:	TYPE:	BRUK:
JFRA	INTEGER	Hjelpevariabel ved ombytting i NSORT:
MM	INTEGER	Hjelpevariabel.

Beskrivelse av hjelperutinene.

Kall: HSORTA(NDIM, ND, LH, JPAR, LMIN, LMAX)
HJELP(LMIN, LMAX)

HSORTA sorterer fra og med linje LMIN til og med linje LMAX i sorteringsbufferet etter feltet JPAR i innholdsregisterposten (eks.:): JPAR = 1): skipsnummer, JPAR = 3): år). Rutinen er beregnet på å sortere etter parametre hvor samme verdi forekommer i store grupper (år, skipsnummer). Påtrufne verdier, og overste adresse for disse, registreres i tabellene NDIM (NSHIP eller NYEAR ved kall). Når en feilplassert verdi påtreffes, brukes denne tabellen til å bestemme hvor det feilplasserte området skal flyttes. LH er linjenummerpeker til NDIM. ND er største antall linjer i NDIM.

HJELP sorterer fra og med linje LMIN til og med linje LMAX i sorteringsbufferet etter dato og stasjonsnummer (bobblesortering).

```

000001 000 C
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C
000008 000 C
000009 000 C
000010 000 C
000011 000 C
000012 000 C
000013 000 C
000014 000 C
000015 000 C
000016 000 C
000017 000 C
000018 000 C
000019 001 C
000020 000 C
000021 000 C
000022 000 C
000023 000 C
000024 000 C
000025 000 C
000026 000 C
000027 000 C
000028 000 15
000029 000 C
000030 000 C
000031 000 C
000032 000 C
000033 000 C
000034 000 C
000035 000 C
000036 000 C
000037 000 C
000038 000 110
000039 000 C
000040 000 C
000041 000 C
000042 000 C

*****
*
*   SORTER   *
*
*****

SORTER SORTERER EN INNHOLDSREGISTERFIL ETTER AR,SKIPSNUMMER,
DATO OG STASJONSNUMMER.
ROUTINEN UNDERSØKER OGSÅ OM REKKEFØLGEN I INNLESEFILEN ER KORREKT.

NØDVENDIGE ROUTINER: NTFEIL

RESERVEVE I/O-ENHETSNUMRE : 7 (MELLOMLAGRINGSFIL)

SUBROUTINE SORTER(NUNIT,NUTU,IUFIL,LOSP,UTDISC,REKKE,K)
COMMON /DATA/ NSORT(10,1500),NYEAR(2,100),NSHIP(2,100),LOGG(2,100)
DIMENSION LOGGH(100)
EQUIVALENCE (NYEAR,LOGGH)
LOGICAL BOOL,LOSP,UTDISC,REKKE
FORMAT(2X,'*** SORTER KALLES ***')
WRITE(6,15)
LUT=0
IL=0
K=0
BOOL=.FALSE.
KN=0
KM=1
REKKE=.TRUE.
IF (LOSP) REWIND 7
CONTINUE
LESH=KN+1

INNLESING TIL SORTERINGSBUFFERET.

CALL NTRAN(NUNIT,2,1000,NSORT(1,LESH),L,22)
K=K+1
CALL NTFEIL(1000,L,NUNIT,K,11999)
KN=KN+100
IF(NSORT(9,KN).EQ.0) GO TO 200
IF (KN.LE.1400) GO TO 110
GO TO 220
200 CONTINUE
BOOL=.TRUE.
JZZ=KN-99
DO 210 J=JZZ,KN
IF(NSORT(9,J).NE.0) GO TO 210
KN=J-1
GO TO 220
210 CONTINUE
KN=JZZ-1
220 CONTINUE
IF (KN.LT,KM) N=KN
IF (KN.LT,KM) GO TO 621

BUFFERET SORTERES ETTER AR,(AR OG ADRESSER LAGRES I NYEAR)

CALL HSCRTA(NYEAR,100,IY,3,KM,KN)
FORMAT(2X,L1,' KM=',15,' KN=',15)
WRITE(6,50) BOOL,KM,KN
51 FORMAT(2X,10I11)
WRITE(6,51) LUT
WRITE(6,51) ((NYEAR(J,I),J=1,2),I=1,IY)
MM=KM
KK=IY
621 ARGANGER PATRUFFET
IF ((BOOL).OR.(KK.EQ.1)) GO TO 405
J=NYEAR(2,IY)-NYEAR(2,IY-1)
IF (LOSP) GO TO 400
N=NYEAR(2,IY-1)/100
6 LENGDE PA SISTE ARGANG
N=100*N
N=NYEAR(2,IY-1)-N
J=J+N
400 CONTINUE

HVIS SIST PATRUFNE ARGANG IKKE ER FOR STOR,
HOLDES DENNE UTENFOR VIDERE SORTERING I PAVENTE
AV EVENTUELLE FLERE DATA FRA SAMME AR.

IF (J.LT.1401) KK=KK-1

```

```
2 000087 000 405 CONTINUE
3 000088 000 C
4 000089 000 C SORTERING ETTER SKIP,DATO OG STASJONSNUMMER.
5 000090 000 C
6 000091 000 DO 600 J=1,KK
7 000092 000 M=NYEAR(2,J)
8 000093 000 C
9 000094 000 C HVER ARGANG SORTERES ETTER SKIPSNUMMER,(SKIPSNUMMER
10 000095 000 C OG ADRESSER LAGRES I NSHIP)
11 000096 000 C
12 000097 000 CALL HSORTA(NSHIP,100,N,1,MM,M)
13 000098 000 WRITE(6,10) NYEAR(1,J),NYEAR(2,J)
14 000099 000 10 FORMAT(//4X,3HAR ,14,' '); ',14/)
15 000100 000 C
16 000101 000 C HVERT FOREKOMMENDE SKIP SORTERES ETTER DATO OG STASJONSNUMMER
17 000102 000 C
18 000103 000 DO 500 IZ=1,N
19 000104 000 M=NSHIP(2,IZ)
20 000105 000 CALL HJELP(MM,M)
21 000106 000 WRITE(6,11) NSHIP(1,IZ),NSHIP(2,IZ)
22 000107 000 11 FORMAT(4X,'SKIP ',14,' '); ',14)
23 000108 000 MM=M+1
24 000109 000 500 CONTINUE
25 000110 000 600 CONTINUE
26 000111 000 IF(BOOL.AND,(K.LE.15)) GO TO 700
27 000112 000 C
28 000113 000 C DEN SORTERTE DEL AV BUFFERET OVERFØRES TIL UTSKRIFTSFILEN
29 000114 000 C ELLER MELLOMLAGRES PÅ EN DISCFIL(I/O=ENHET 7).
30 000115 000 C ARSTALL OG ANTALL STASJONER AVMERKES I EN TABELL (LOGG),
31 000116 000 C
32 000117 000 DO 620J=1,KK
33 000118 000 N=NYEAR(2,J)
34 000119 000 IL=IL+1
35 000120 000 IF(IL.GT.100) CALL ERTRAN(2)
36 000121 000 LOGG(1,IL)=NSORT(3,N) GAR
37 000122 000 IF(J.RE.1) M=NYEAR(2,J-1)
38 000123 000 IF(J.EQ.1) M=KM-1
39 000124 000 LOGG(2,IL)=N-M GANTALL STASJONER
40 000125 000 M=M+1
41 000126 000 IF (LOSP) WRITE(7)((NSORT(I,L),I=1,10),L=M,N) @MELLOMLAGRING
42 000127 000 620 CONTINUE
43 000128 000 621 CONTINUE
44 000129 000 KHUT=N+1
45 000130 000 IF (LOSP) GO TO 630

000131 000 KHUT=N/100
000132 000 IF (BOOL) KHUT=KHUT+1
000133 000 IF (KHUT.EQ.0) CALL ERTRAN(2)
000134 000 DO 625 J=1,KHUT
000135 000 I=100*(J-1)+1
000136 000 CALL NTRAN(NUTU,1,1000,NSORT(1,I),L,22) @DATA SKRIVES UT.
000137 000 CALL NTFEIL(1000,L,NUTU,-25,11999)
000138 000 625 CONTINUE
000139 000 L=100*KHUT
000140 000 LUT=N-L
000141 000 KHUT=L+1
000142 000 5054 FORMAT(4X,'REST=',13)
000143 000 WRITE(6,5054) LUT
000144 000 630 CONTINUE
000145 000 IF(BOOL) GO TO 661 @ FERDIG
000146 000 IF (KHUT.GT.KN) GO TO 660 @BUFFERET TOMT
000147 000 C
000148 000 C DEN RESTERENDE DEL AV BUFFERET FLYTTES FREM.
000149 000 C
000150 000 DO 650 J=KHUT,KN
000151 000 L=J-KHUT+1
000152 000 DO 650 I=1,10
000153 000 NSORT(I,L)=NSORT(I,J)
000154 000 650 CONTINUE
000155 000 660 CONTINUE
000156 000 KN=KN-KHUT+1
000157 000 KM=LUT+1
000158 000 GO TO 110
000159 000 661 CONTINUE
000160 000 C
000161 000 C FØRSTE GJENNOMGANG AV FILEN ER FERDIG.
000162 000 C
000163 000 IF (LOSP) GO TO 750
000164 000 C
000165 000 C ARSTABELLEN (LOGG) FOR DEN SORTERTE FILEN KONTROLLERES,
000166 000 C
000167 000 WRITE(6,12)
000168 000 12 FORMAT(1H1)
000169 000 13 FORMAT(4X,216)
000170 000 14 FORMAT(//4X,'*** EN ARGANG ER DELT/FEILPLASSERT ***')
000171 000 J=LOGG(1,1)-1
000172 000 DO 663 I=1,IL
000173 000 WRITE(6,13) LOGG(1,I),LOGG(2,I)
000174 000 IF (LOGG(1,I).GT.J) GO TO 662
```

```
000175 000 WRITE(6,14)
000176 000 GO TO 664
000177 000 662 CONTINUE
000178 000 J=LOGG(1,1)
000179 000 663 CONTINUE
000180 000 IF (REKKE) WRITE(6,1529)
000181 000 RETURN
000182 000 C
000183 000 C EN ELLER FLERE ARGANGER ER DELT/FEILPLASSERT.
000184 000 C FILEN OVERFØRES TIL TEMPORÆR FIL (I/O-ENHET 7)
000185 000 C FOR SENERE INNLESING I RIKTIG REKKEFØLGE.
000186 000 C
000187 000 664 CONTINUE
000188 000 IF ((UTDISC).OR.(IUFIL.EQ.1)) CALL NTRAN(NUTU,10,22)
000189 000 IF ((.NOT.UTDISC).AND.(IUFIL.GT.1)) CALL NTRAN(NUTU,8,-1)
000190 000 IF ((.NOT.UTDISC).AND.(IUFIL.GT.1)) CALL NTRAN(NUTU,8,1)
000191 000 LINY=1
000192 000 KN=1
000193 000 REWIND 7
000194 000 665 CONTINUE
000195 000 IF (KN.GT.1401) GO TO 670
000196 000 CALL NTRAN(NUTU,2,1000,NSORT(1,KN),L,22)
000197 000 CALL NTFEIL(1000,L,NUTU,-39,11999)
000198 000 KN=KN+100
000199 000 IF (NSORT(8,KN-1).NE.0) GO TO 665
000200 000 670 CONTINUE
000201 000 KN=KN-1
000202 000 LUT=0
000203 000 671 CONTINUE
000204 000 I=KN-LUT
000205 000 IF (LOGG(2,LINY).GT.1) GO TO 675
000206 000 I=LUT+1
000207 000 LUT=LUT+LOGG(2,LINY)
000208 000 WRITE(7) ((NSORT(JZ,L),JZ=1,10),L=1,LUT)
000209 000 LINY=LINY+1
000210 000 IF (LINY.GT.1L) GO TO 695
000211 000 GO TO 671
000212 000 675 CONTINUE
000213 000 JZ=LUT+1
000214 000 IF (JZ.GT.KN) GO TO 690
000215 000 DO 685 J=JZ,KN
000216 000 L=J-LUT
000217 000 DO 680 I=1,10
000218 000 NSORT(I,L)=NSORT(I,J)

000219 000 680 CONTINUE
000220 000 685 CONTINUE
000221 000 690 CONTINUE
000222 000 KN=KN-LUT+1
000223 000 GO TO 665
000224 000 695 CONTINUE
000225 000 IF ((UTDISC).OR.(IUFIL.EQ.1)) CALL NTRAN(NUTU,10,22)
000226 000 IF ((.NOT.UTDISC).AND.(IUFIL.GT.1)) CALL NTRAN(NUTU,8,-1)
000227 000 IF ((.NOT.UTDISC).AND.(IUFIL.GT.1)) CALL NTRAN(NUTU,8,1)
000228 000 GO TO 750
000229 000 700 CONTINUE
000230 000 C
000231 000 C FILER SOM FAR PLESS I BUFFERET I EN OMGANG(DVS. MINORE ENN
000232 000 C 1501 STASJONER),SKRIVES UT PA DISC HER,
000233 000 C
000234 000 DO 720 I=1,K
000235 000 LESH=100*(I-1)+1
000236 000 CALL NTRAN(NUTU,1,1000,NSORT(I,LESH),L,22)
000237 000 CALL NTFEIL(1000,L,NUTU,I,11999)
000238 000 720 CONTINUE
000239 000 IF (REKKE) WRITE(6,1529)
000240 000 RETURN
000241 000 750 CONTINUE
000242 000 C
000243 000 C ALLE DATA ER HELLOMLAGRET PA DISC.(MER ENN 1500 STASJONER
000244 000 C I FILEN) DE ENKELTE DELER LESES INN IGJEN I RETT REKKEFØLGE,
000245 000 C HVIS EN ARGANG ER DELT(): HAR FOREKOMMET SVÆRT SPREDT I
000246 000 C UTGANGSMATERIALET),LESES DENNE INN SAMLET OG SORTERES
000247 000 C ETTER SKIPSNUMMER,DATO OG STASJONSNUMMER.
000248 000 C
000249 000 ENDFILE 7
000250 000 REWIND 7
000251 000 1500 FORMAT(2X,1116)
000252 000 WRITE(6,1500) ((LOGG(I,J),I=1,2),J=1,1L)
000253 000 C
000254 000 C TABELLEN MED OVERSIKT OVER ARGANGER (LOGG) SORTERES.
000255 000 C OPPRINNELIG NUMMER PA ARGANGEN LAGRES I LOGGH.
000256 000 C
000257 000 DO 760 I=1,1L
000258 000 LOGGH(I)=1
000259 000 760 CONTINUE
000260 000 J=1L+1
000261 000 765 CONTINUE
000262 000 BOOL=.TRUE.
```

```
000263 000 J=J+1
000264 000 IF (J, EQ, 1) GO TO 780
000265 000 DO 775 I=2, J
000266 000 IF (LOGG(I, I), GE, LOGG(I, I=1)) GO TO 775
000267 000 REKKE=.FALSE.
000268 000 BOOL=.FALSE.
000269 000 DO 770 L=1, 2
000270 000 N=LOGG(L, I)
000271 000 LOGG(L, I)=LOGG(L, I=1)
000272 000 LOGG(L, I=1)=N
000273 000 770 CONTINUE
000274 000 N=LOGGH(I)
000275 000 LOGGH(I)=LOGGH(I=1)
000276 000 LOGGH(I=1)=N
000277 000 775 CONTINUE
000278 000 IF (.NOT. BOOL) GO TO 765
000279 000 780 CONTINUE
000280 000 C
000281 000 C ARGANGENE LESES INN (LINY ER TELLER I LOGG, IS ER
000282 000 C POSTNUMMERPEKER I MELLOMLAGRINGSFILEN)
000283 000 C
000284 000 LINY=1
000285 000 IS=1
000286 000 KN=0
000287 000 BOOL=.FALSE.
000288 000 800 CONTINUE
000289 000 IF (LINY.GT. IL) GO TO 875
000290 000 IF ((LINY.LT. IL).AND.(LOGG(I, LINY+1), EQ, LOGG(I, LINY))) GO TO 860
000291 000 810 CONTINUE
000292 000 L=1500-KN
000293 000 IF (LOGG(2, LINY), GT. L) GO TO 870
000294 000 I=LOGGH(LINY) -15
000295 000 IF (I.EQ.0) GO TO 830
000296 000 IF (I.GT.0) GO TO 820
000297 000 I=-I
000298 000 DO 815 J=1, I
000299 000 BACKSPACE 7
000300 000 815 CONTINUE
000301 000 GO TO 830
000302 000 820 CONTINUE
000303 000 DO 825 J=1, I
000304 000 READ(7)
000305 000 825 CONTINUE
000306 000 830 CONTINUE

000307 000 IS=LOGGH(LINY)+1
000308 000 I=KN+1
000309 000 KN=KN+LOGG(2, LINY)
000310 000 READ(7) ((NSORT(JZ, L), JZ=1, 10), L=I, KN)
000311 000 LINY=LINY+1
000312 000 IF (.NOT. BOOL) GO TO 800
000313 000 IF ((LINY.LE. IL).AND.(LOGG(I, LINY=1), EQ, LOGG(I, LINY))) GO TO 810
000314 000 BOOL=.FALSE.
000315 000 KM=LUT+1
000316 000 CALL HSORYA(NSHIP, 100, N, 1, KM, KN)
000317 000 JZ=KM
000318 000 DO 850 J=1, N
000319 000 L=NSHIP(2, J)
000320 000 CALL HJELP(JZ, L)
000321 000 JZ=L+1
000322 000 850 CONTINUE
000323 000 GO TO 800
000324 000 860 CONTINUE
000325 000 BOOL=.TRUE.
000326 000 GO TO 875
000327 000 870 CONTINUE
000328 000 IF (BOOL) WRITE(6, 1541) LOGG(I, LINY)
000329 000 IF (BOOL) CALL ERTRAN(2)
000330 000 1541 FORMAT(///4X, 'ARGANG ', 14, ' ER FOR STOR')
000331 000 875 CONTINUE
000332 000 L=KN/100
000333 000 IF (L.EQ.0) GO TO 900
000334 000 DO 890 J=1, L
000335 000 I=100*(J+1)+1
000336 000 CALL NTRAN(NUTU, 1, 1000, NSORT(I, I), JZ, 22)
000337 000 CALL NYFEIL(1000, JZ, NUTU, =41, 1999)
000338 000 890 CONTINUE
000339 000 900 CONTINUE
000340 000 L=100*L
000341 000 IF (IL.LT. LINY) GO TO 950
000342 000 LUT=KN=L
000343 000 IF (L.EQ.0) GO TO 930
000344 000 IF (LUT.EQ.0) KN=0
000345 000 IF (LUT.EQ.0) GO TO 930
000346 000 I=L+1
000347 000 DO 920 J=1, KN
000348 000 L=J+1+1
000349 000 DO 910 JZ=1, 10
000350 000 NSORT(JZ, L)=NSORT(JZ, J)
```

```

000351 000 910 CONTINUE
000352 000 920 CONTINUE
000353 000 KN=LUT
000354 000 930 CONTINUE
000355 000 IF (BOOL) GO TO 810
000356 000 GO TO 800
000357 000 950 CONTINUE
000358 000 IF (KN.EQ.1500) L=0
000359 000 IF (KN.EQ.1500) KN=0
000360 000 KN=KN+1
000361 000 L=L+100
000362 000 DO 960 J=KN,L
000363 000 DO 960 I=1,10
000364 000 NSORT(I,J)=0
000365 000 960 CONTINUE
000366 000 KN=L-99
000367 000 CALL NTRAN(NUTU,1,1000,NSORT(I,KN),JZ,22)
000368 000 CALL NTFEIL(1000,JZ,NUTU,=51,11999)
000369 000 1529 FORMAT(4X,'*****' RIKTIG REKKEFØLGE *****?)
000370 000 IF (REKKE) WRITE(6,1529)
000371 000 RETURN
000372 000 1999 CALL ERTRAN(2) 6FEILAVBRUDD
000373 000 C
000374 000 SUBROUTINE HSORTA(NDIM,ND,LH,JPAR,LMIN,LMAX)
000375 000 C
000376 000 C HSORTA ER EN HJELPERUTINE SOM BRUKES VED SORTERING ETTER
000377 000 C AR (JPAR=3) ELLER SKIP (JPAR=1).
000378 000 C SORTERING FRA OG MED LINJE LHIN TIL OG MED LMAX.
000379 000 C RUTINEN FØRER EN TABELL (NDIM) HVOR VERDIER MED TILHØRENDE
000380 000 C ØVRE ADRESSER SOM ER PATRUFFET, AVMERKES.
000381 000 C DENNE TABELLEN BRUKES VED EVENTUELL FLYTTING AV DATAOMRADER.
000382 000 C
000383 000 DIMENSION NDIM(2,ND)
000384 000 DIMENSION NFLYTT(10)
000385 000 I=LMIN
000386 000 LH=0
000387 000 100 CONTINUE
000388 000 LH=LH+1
000389 000 IF(LH.GT.ND) CALL ERTRAN(2)
000390 000 NDIM(1,LH)=NSORT(JPAR,I) 0 NY VERDI AVMERKES I TABELL.
000391 000 NV=NSORT(JPAR,I) 0 VERDI P.T.
000392 000 110 CONTINUE
000393 000 I=I+1 0 LINJENUMMER ØKES.
000394 000 IF(I.GY.LHAX) GO TO 500 0FERDIG?

000395 000 IF(NSORT(JPAR,I).EQ.NV) GO TO 110 0 INGEN ENDRING?
000396 000 NDIM(2,LH)=I-1
000397 000 IF(NSORT(JPAR,I).GT.NV) GO TO 100
000398 000 C
000399 000 C DEN NYE VERDI ER HINDRE ENN DEN FORRIGE, OG MÅ FLYTTES PÅ Plass.
000400 000 C
000401 000 REKKE=.FALSE.
000402 000 NV=NSORT(JPAR,I)
000403 000 DO 150 L=LH,1,-1
000404 000 IF(NDIM(1,L).EQ.NV) GO TO 200 0 SAMME VERDI PATRUFFET TIDLIGERE?
000405 000 IF(NDIM(1,L).LT.NV) GO TO 160 0 LAVERE VERDI ENN DEN NYE?
000406 000 150 CONTINUE
000407 000 L=0
000408 000 160 CONTINUE
000409 000 HA=L+2
000410 000 LH=LH+1
000411 000 IF(LH.GY.ND) CALL ERTRAN(2)
000412 000 C
000413 000 C Plass lages på rett sted i tabellen.
000414 000 C
000415 000 DO 170 L=LH,HA,-1
000416 000 NDIM(1,L)=NDIM(1,L-1)
000417 000 NDIM(2,L)=NDIM(2,L-1)
000418 000 170 CONTINUE
000419 000 L=HA-1
000420 000 NDIM(1,L)=NV
000421 000 HA=L+1
000422 000 IF(HA.NE.0) NDIM(2,L)=NDIM(2,HA)
000423 000 IF(HA.EQ.0) NDIM(2,L)=LMIN+1
000424 000 200 CONTINUE
000425 000 C
000426 000 C BESTEM LENGDEN PÅ OMRÅDET SOM SKAL FLYTTES.
000427 000 C
000428 000 DO 250 II=1,LHAX
000429 000 IF(NSORT(JPAR,II).NE.NV) GO TO 260
000430 000 250 CONTINUE
000431 000 II=LHAX+1
000432 000 260 CONTINUE
000433 000 II=II+1 0 ØVRE GRENSE FOR OMRÅDET SOM SKAL FLYTTES FREM,
000434 002 KA=NDIM(2,L)+1 0 NEDRE GRENSE FOR OMRÅDET SOM SKAL FLYTTES BAK,
000435 000 JJ=II-1+1 0 ANTALL LINJER SOM SKAL FLYTTES FREM,
000436 002 JJ=II-KA 0 ANTALL LINJER SOM SKAL FLYTTES BAK,
000437 001 NFMAX=MAX(JJ,JI)
000438 001 NFMIN=MIN(JJ,JI)

```



```
000439 001 270 CONTINUE
000440 001 MA=MOD(NFMAX,NFMIN)
000441 001 IF (MA,EQ,0) GO TO 280
000442 001 NFMAX=NFMIN
000443 001 NFMIN=MA
000444 001 GO TO 270
000445 001 280 CONTINUE
000446 001 MB=JJ+JI
000447 001 MB=(MB/NFMIN)+1
000448 001 NFMIN=NFMIN+KA-1
000449 001 IDEL=KA+JJ-1
000450 001 DO 350 JZ=KA,NFMIN
000451 001 DO 285 JZZ=1,10
000452 001 NFLYTT(JZZ)=NSORT(JZZ,JZ)
000453 001 285 CONTINUE
000454 001 JTIL=JZ
000455 001 DO 320 MA=1,MB
000456 001 IF (JTIL,LE,IDEL) JFRA=JTIL+JI
000457 001 IF (JTIL,GT,IDEL) JFRA=JTIL-JJ
000458 001 DO 300 JZZ=1,10
000459 001 NSORT(JZZ,JTIL)=NSORT(JZZ,JFRA)
000460 001 300 CONTINUE
000461 001 JTIL=JFRA
000462 001 320 CONTINUE
000463 001 DO 350 JZZ=1,10
000464 001 NSORT(JZZ,JTIL)=NFLYTT(JZZ)
000465 000 350 CONTINUE
000466 000 C
000467 000 C TABELLADRESSENE RETTES OPP,
000468 000 C
000469 000 DO 400 MA=LH,L,+1
000470 000 NDIM(2,MA)=NDIM(2,MA)+JJ
000471 000 400 CONTINUE
000472 000 I=I+1
000473 000 NV=NDIM(1,LH)
000474 000 GO TO 110
000475 000 500 CONTINUE
000476 000 NDIM(2,LH)=I-1
000477 000 RETURN
000478 000 C
000479 000 SUBROUTINE HJELP(LMIN,LMAX)
000480 000 C
000481 000 C HJELP SORTERER LINJE LMIN TIL OG MED LMAX I SORTERINGSBUFFERET
000482 000 C ETTER DATO OG STASJONSNUMMER,

000483 000 C
000484 000 LOGICAL A
000485 000 IF(LMIN,EQ,LMAX) RETURN
000486 000 MB=LMAX+1
000487 000 MA=LMIN+1
000488 000 100 CONTINUE
000489 000 A=,FALSE,
000490 000 MB=MB-1 6 ØVRE GRENSE FLYTTES NED EN PLASS FOR HVERT GJENNOMLØP
000491 000 IF(MB,LT,MA) RETURN
000492 000 DO 300 I=MA,MB
000493 000 IF(NSORT(4,I),GT,NSORT(4,I-1)) GO TO 300
000494 000 IF(NSORT(4,I),LT,NSORT(4,I-1)) GO TO 150
000495 000 IF(NSORT(2,I),GT,NSORT(2,I-1)) GO TO 300
000496 000 IF(NSORT(2,I),LT,NSORT(2,I-1)) GO TO 150
000497 000 10 FORMAT(4X,4I7,1X,A6,2I7,1X,2A6,1X,A6)
000498 000 11 FORMAT(4X,17,3X,R4,2I7,1X,A6,2I7,1X,2A6,1X,A6)
000499 000 C
000500 000 C FEILUTSKRIFTER,
000501 000 C
000502 000 IF(NSORT(2,1),GT,9999) GO TO 110
000503 000 WRITE(6,10)(NSORT(JZZ,I-1),JZZ=1,10)
000504 000 WRITE(6,10)(NSORT(JZZ,1),JZZ=1,10)
000505 000 GO TO 300
000506 000 110 CONTINUE
000507 000 WRITE(6,11)(NSORT(JZZ,I-1),JZZ=1,10)
000508 000 WRITE(6,11)(NSORT(JZZ,1),JZZ=1,10)
000509 000 GO TO 300
000510 000 150 CONTINUE
000511 000 REKKE=,FALSE,
000512 000 A=,TRUE,
000513 000 DO 250 II=1,10
000514 000 JJ=NSORT(II,1)
000515 000 NSORT(II,1)=NSORT(II,I-1)
000516 000 NSORT(II,I-1)=JJ
000517 000 250 CONTINUE
000518 000 300 CONTINUE
000519 000 IF(A) GO TO 100
000520 000 RETURN
000521 000 END
```