Statistics Norway

Statistisk sentralbyrå

Vidar Halvorsen og John Erik Sjørbotten

Eksempler i Easytrieve Plus

Avdeling for personstatistikk/Gruppe for EDB

Kongsvinger 15. desember 1993

Forord

Easytrieve Plus brukes i dag av et stort antall personer i Byrået. Hovedtyngden går fortsatt på stormaskin, men vi har også en PC-utgave både for DOS og for Windows. DOS-versjonen kan startes fra hovedmenyen, Windows-utgaven ventes å komme på nettet med det første. Easytrieve for UNIX finnes foreløpig kun i Beta-versjon og er ennå ikke tilgjengelig for oss. (Det er heller ikke sikkert at Byrået vil kjøpe den inn om den blir ferdig)

Dette notatet er rettet mot personer som har litt erfaring med Easytrieve, og er ment som et oppslagsverk med eksempler på hvordan ting *kan* gjøres. Det er laget for å bedre forståelsen av programmering og for å øke effektiviteten. Beskrivelse av programmenene ligger foran hvert enkelt eksempel. Vi har lagt stor vekt på matche-problematikken, da denne kan fortone seg vanskelig for mange.
Alle eksemplene er fra stormaskin-miljøet unntatt de to siste, som er skrevet i CA-Easytrieve for PC. Disse to er tatt med for at det skal bli lettere å komme igang med PC-utgaven og for å vise at vi her får flere muligheter enn det vi er vant til fra stormaskinen.

Alle program-eksemplene ligger på datasettet **EZT.STD.MODULER** som hver enkelt bruker kan kopiere fra og bruke som utgangspunkt for program de selv skal lage.

Notatet blir også brukt i forbindelse med kurs som holdes internt i Byrået.

John Erik Sjørbotten

Vidar Teppa Halvorsen

Gruppe for EDB
Personstatistikk (GEP)
Kongsvinger

Innhold

PROGRAMOPPBYGGING OG NOEN STANDARDER VI HAR BRUKT.

Et Easytrieve-program må ha en spesiell rekkefølge i oppbyggingen:
Først kommer gjerne en blokk med kommentarer om hva programmet gjør, så eventuelt et PARM-parameter, deretter defineringen av fil- og arbeids-feltene og så selve jobb-aktiviteten, JOB eller SORT. Hvis vi innen JOB-aktiviteten vil utføre en ting flere ganger, eller vi av andre grunner vil skille den ut, skriver vi den for seg, bak selve hovedrutina i en prosedyre (PROC). Eventuelle rapporter kommer helt til slutt i aktiviteten. Etterpå kan gjerne flere JOB- og/eller SORT-aktiviteter følge i samme program.
Prosedyrene kan vi utføre fra hvor som helst i aktiviteten ved hjelp av PERFORM. Vi kan også utføre prosedyrer ved hjelp av START og FINISH i JOB INPUT-setningen. Prosedyren vi kaller opp med START blir utført ved initieringen av aktiviteten (brukes til nullstilling av arbeidsfelt, ol.). Prosedyren vi kaller opp med FINISH blir utført når aktiviteten avsluttes normalt (brukes til utskrift av summer, ol.).

Det er lurt å bruke noen standarder når en programmerer. Vi har f.eks. latt feltnavn begynne med spesielle bokstaver:

I	hvis det er fra en inn-fil. (I1, I2, osv. hvis flere inn-filer),
U	fra ut-fil,
W	på arbeidsfelt (work) og
X	på indekser (index).

Da kan vi med engang se hvor feltene hører hjemme. Vi har unntak fra denne standarden når vi bruker MOVE LIKE i programmet. Da må vi bruke like navn på fra- og til-feltene. For å gjøre programmene mer oversiktlige har vi brukt innrykk ved IF, DO WHILE o.l. og passet på å plassere END-IF rett under den IF den avslutter og END-DO under sin DO.

Programoppbygging:

```
PARM -
FILE -
JOB -
  IF -
    IF -
      DO WHILE -
        DO WHILE -
        END-DO
      END-DO
    END-IF
  ELSE
  END-IF
PROC
END-PROC
REPORT -
  REPORT-PROC
  END-PROC

SORT -
JOB -
```

Eksempler på PARM-parametre:

`PARM SYNTAX` Tar kun syntaks-sjekk av programmet uten å eksekvere det.

`PARM DEBUG (NODMAP NOCLIST NOXREF STATE) LIST (NOPARM)`
Tar vekk mye av maskin-utskriften bak programlista:

`DMAP`	maskinprodusert oversikt over filer og felt
`CLIST`	kondensert liste av det eksekverbare programmet
`REF`	kryssreferanse mellom alle felt
`STATE`	hvis progr. avsluttes unormalt, skrives nr. på linja i programmet som forårsaket bruddet
`LIST (PARM)`	liste over parametre brukt i syntakssjekken

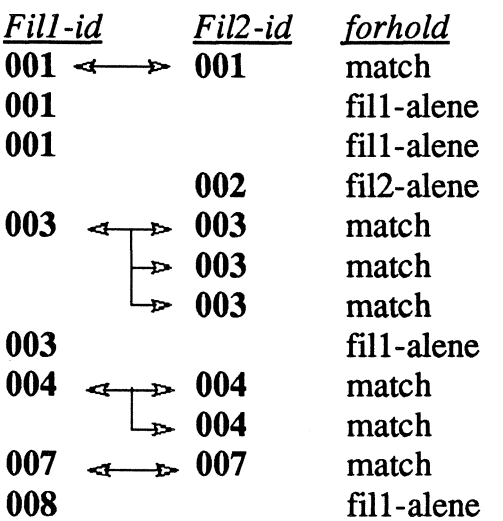
MATCHE-LOGIKKEN I EASYTRIEVE.

Filene som skal matches må være sortert stigende på de feltene det skal matches på (matchebegrepet), og feltene i begrepet må være definert likt på begge filer, begge alfanumeriske, begge numeriske, osv. Det kan gjerne inngå flere felter i matchebegrepet. Eksempel: fødselsnr * utdanning skrives slik:

```
JOB INPUT (INN1 KEY (I1-FNR I1-UTDAN) +
           INN2 KEY (I2-FNR I2-UTDAN)) FINISH SLUTT
```

Lesingen fra fil-1 og fil-2 foregår i følgende rekkefølge: Først leses en record fra fil-1, så en fra fil-2. Er matche-begrepet på fil-2 mindre eller lik matche-begrepet på fil-1, leses fil-2 inntil denne blir større enn fil-1. Først da leses en ny record på fil-1. Derfor får eventuelle dubletter på fil-1 aldri match mot fil-2, mens eventuelle dubletter på fil-2 kan få match mot fil-1.
Er det dubletter på begge filer ved en match, kobles alle dublettene på fil-2 til 1. dublett på fil-1 hvis vi ikke legger inn spesielle tester (DUPLICATE, FIRST-DUP, LAST-DUP).

Eksempel:



Her er et eksempel som viser hvordan hvordan Easytrieve definerer dubletter:

dublett-begrep:		
-		
0101		
0103	NOT DUPLICATE	
0104	DUPLICATE AND FIRST-DUP	(NOT LAST-DUP)
0104	DUPLICATE AND NOT FIRST-DUP	(NOT LAST-DUP)
0104	DUPLICATE AND LAST-DUP	(NOT FIRST-DUP)
0107		
-		

Vi ser at også den første recorden med likt dublettbegrep (0104) blir definert som dublett (DUPLICATE).

GEPMATCH er et typisk matcheprogram. Filen INN2 er "hovedfilen", det er denne vi skal ha ut med påførte opplysninger fra den andre filen, INN1. Selv om INN2 inneholder dubletter vil alle disse få overført opplysninger fra INN1 (hvis INN1 har samme matchebegrep). Alle recordene på INN2 blir skrevet ut.

Vi tester først om recordene fra begge filene har likt matchebegrep (linje 35). Har de det, overfører vi opplysninger fra begge filene til UT-record og skriver denne. Hvis testen ikke slår til, vet vi at recorden med lavest matchebegrep enten *bare* er på INN2-filen eller *bare* på INN1-filen.

Vi tester her først om den fins på INN2 (linje 41). Gjør den det, blanker vi ut feltet som skulle inneholdt opplysninger fra INN1 før vi skriver den ut. Dette for å unngå at opplysninger fra forrige record "henger" igjen i feltet.

Hvis ingen av disse testene har slått til, må recorden med lavest matchebegrep være *bare* på INN1 (linje 46). Den har vi ikke bruk for i dette tilfelle, men vi teller den opp likevel.

RECORD-COUNT er opptellingsfelt Easytrieve selv definerer for hver fil. Disse inneholder hele tiden antall recorder som er lest eller skrevet fra/til den aktuelle filen. Disse feltene er definert på 10 posisjoner. For å få skrevet ut våre opptellingsfelt rett under disse har vi også definert våre felt med 10 posisjoner.

```
1  ************************* EASYTRIEVE PLUS ******************************
2  *  PROD.NR.: 000
3  *  PROSJEKT: EASYTRIEVE-EKSEMPLER
4  ************************************************************************
5  *  PROGRAM : GEPMATCH
6  *  LAGET   : 16.03.93 AV JOHN ERIK SJØRBOTTEN
7  ************************************************************************
8  *  BESKRIV.: MATCHING AV 2 FILER. ALLE RECORDENE FRA INN2 OVERFØRES,
9  *            MENS OPPLYSNINGER FRA INN1 HEKTES PÅ ALLE MAKKERE
10 *            INKLUSIV EVENTUELLE DUBLETTER FRA INN2.
11 *            INN1-REC UTEN MAKKER OVERFØRES IKKE.
12 *  REF.    :
13 ************************************************************************
14
15  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
16  FILE INN1
17    I1-FNR      1  11 A
18    I1-OPPL    43   6 A
19
20  FILE INN2
21    I2-REC      1  34 A
22    I2-FNR     23  11 A
23
24  FILE UT
25    U-REC2      1  34 A
26    U-OPPL1    35   6 A
27
28  W-ANTMAKKER   W  10 N 0
29  W-ANT1ALENE   W  10 N 0
30  W-ANT2ALENE   W  10 N 0
31
32
33  JOB INPUT (INN1 KEY (I1-FNR) +
34             INN2 KEY (I2-FNR))   FINISH SLUTT
35    IF MATCHED
36      W-ANTMAKKER = W-ANTMAKKER + 1
37      U-REC2  = I2-REC
38      U-OPPL1 = I1-OPPL
39      PUT UT
40    ELSE
41      IF INN2
42        W-ANT2ALENE = W-ANT2ALENE + 1
43        U-REC2  = I2-REC
44        U-OPPL1 = '      '
45        PUT UT
46      ELSE
47        W-ANT1ALENE = W-ANT1ALENE + 1
48      END-IF
49    END-IF
50
51  SLUTT. PROC
52    DISPLAY NEWPAGE +
53            '- GEPMATCH - PÅFØRING AV OPPLYSN FRA REG. -'
54    DISPLAY INN1:RECORD-COUNT 'REC FRA INN1'
55    DISPLAY INN2:RECORD-COUNT 'REC FRA INN2'
56    DISPLAY W-ANTMAKKER       'REC SOM MATCHET'
57    DISPLAY W-ANT1ALENE       'REC KUN FRA INN1'
58    DISPLAY W-ANT2ALENE       'REC KUN FRA INN2'
59    DISPLAY UT:RECORD-COUNT   'REC SKREVET UT'
60    DISPLAY '----------------------------------------'
61  END-PROC
```

GEPMATCA er et et matcheprogram som har mange likheter med GEPMATCH. Filen INN2 er "hovedfilen", det er denne som skal skrives ut med opplysninger hentet fra den andre filen, INN1. Her skal vi i tillegg fjerne eventuelle dubletter fra INN2.

I programmet starter vi med å teste om INN2-recorden er en dublett og samtidig *ikke* er den første dubletten. Hvis det er en "fra-og-med-nr.2-dublett" fjerner vi denne og programmet går opp igjen til JOB INPUT uten å teste på match mot INN1.
Ellers, og bare ellers, utføres videre testing på match/ikke match som i forrige program GEPMATCH.

```
1  ************************ EASYTRIEVE PLUS ******************************
2  *  PROD.NR.: 000
3  *  PROSJEKT: EASYTRIEVE-EKSEMPLER
4  ***********************************************************************
5  *  PROGRAM : GEPMATCA
6  *  LAGET   : 16.03.93 AV JOHN ERIK SJØRBOTTEN
7  ***********************************************************************
8  *  BESKRIV.: MATCHING AV 2 FILER.
9  *            DUBLETTER FJERNES FRA INN2, MENS DE ANDRE INN2-RECORDENE
10 *            OVERFØRES I SIN HELHET OG OPPLYSNINGER FRA INN1 HEKTES PÅ
11 *            INN1-REC UTEN MAKKER OVERFØRES IKKE.
12 *  REF.    :
13 ***********************************************************************
14
15  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
16  FILE INN1
17    I1-FNR       1  11 A
18    I1-OPPL     43   6 A
19
20  FILE INN2
21    I2-REC       1  34 A
22    I2-FNR      23  11 A
23
24  FILE UT
25     U-REC2      1  34 A
26     U-OPPL1    35   6 A
27
28  W-ANTMAKKER    W  10 N 0
29  W-ANT1ALENE    W  10 N 0
30  W-ANT2ALENE    W  10 N 0
31  W-ANT2DUB      W  10 N 0
32
33
34  JOB INPUT (INN1 KEY (I1-FNR) +
35            INN2 KEY (I2-FNR))   FINISH SLUTT
36    IF DUPLICATE INN2 AND NOT FIRST-DUP INN2
37 *                                    DUBLETTER PÅ INN2 TELLES KUN OPP
38      W-ANT2DUB = W-ANT2DUB + 1
39    ELSE
40      IF MATCHED
41        W-ANTMAKKER = W-ANTMAKKER + 1
42        U-REC2  = I2-REC
43        U-OPPL1 = I1-OPPL
44        PUT UT
45      ELSE
46        IF INN2
47          W-ANT2ALENE = W-ANT2ALENE + 1
48          U-REC2  = I2-REC
49          U-OPPL1 = '      '
50          PUT UT
51        ELSE
52          W-ANT1ALENE = W-ANT1ALENE + 1
53        END-IF
54      END-IF
55    END-IF
56
57  SLUTT. PROC
58    DISPLAY NEWPAGE +
59            '- GEPMATCA - PÅFØRING AV OPPLYSN. FRA REG. -'
60    DISPLAY INN1:RECORD-COUNT 'REC FRA INN1'
61    DISPLAY INN2:RECORD-COUNT 'REC FRA INN2'
62    DISPLAY W-ANT2DUB         'DUBL PÅ INN2 (LEST FORBI)'
63    DISPLAY W-ANTMAKKER       'REC SOM MATCHET'
64    DISPLAY W-ANT1ALENE       'REC KUN FRA INN1'
65    DISPLAY W-ANT2ALENE       'REC KUN FRA INN2'
66    DISPLAY UT:RECORD-COUNT   'REC SKREVET UT'
67    DISPLAY '-----------------------------------------'
68  END-PROC
```

GEPMATCHD matcher 2 filer. Alle recorder fra INN1 og alle ikke-dubletter fra INN2 skal være med ut på UT-fil. Foruten å fjerne eventuelle dubletter på INN2 skal vi i tillegg telle dem opp fordelt på match/ikke match.

Vi starter her med selve matchingen, IF MATCHED (linje 41). Hvis denne testen får tilslag, tester vi videre på om recorden fra INN2 er en "ikke-dublett" eller en "nr.1-dublett". Hvis ja, kobler vi og skriver recorden ut. Hvis ikke, er det en reell dublett som vi bare teller opp.

Samme dublett-test gjentas for recorder som fins bare på INN2-fil (linje 53).

Hvis ingen av disse disse foregående testene har slått til, fins recorden med dette matchebegrepet bare på INN1. Det er da enten en record med matchebegrep som ikke fins på INN2 eller det kan også være en dublett (fra-og-med-nr.2-dublett, disse får jo aldri match).

Når en record fins bare på INN1 må vi huske på å få med matchebegrepet (her fødselsnr) fra INN1, da vi ikke har noen INN2-record og hente det fra.

Vi blanker ut felt som skulle inneholde opplysninger fra fil som mangler. Dette for å unngå at data fra forrige record "henger" igjen (linje 57 og 64).

```
 1 ************************* EASYTRIEVE PLUS ******************************
 2 *  PROD.NR.: 000
 3 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
 4 ************************************************************************
 5 *  PROGRAM : GEPMATCD
 6 *  LAGET   : 16.03.93 AV JOHN ERIK SJØRBOTTEN
 7 ************************************************************************
 8 *  BESKRIV.: MATCHING AV 2 FILER.
 9 *            DUBLETTER PÅ INN2 FJERNES OG TELLES OPP FORDELT PÅ MED/
10 *            UTEN MAKKER.
11 *            ALLE DE ANDRE RECORDENE FRA INN2 OG ALLE FRA INN1 SKRIVES
12 *            UT PÅ UT-FIL.
13 *  REF.    :
14 ************************************************************************
15
16                          fortsetter...
```

```
17
18  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
19  FILE INN1
20    I1-FNR       1  11 A
21    I1-REC      12  10 A
22
23  FILE INN2
24    I2-REC       1  40 A
25    I2-FNR       1  11 A
26
27  FILE UT
28     U-2REC      1  40 A
29     U-FNR       1  11 A
30     U-1REC     41  10 A
31
32  W-ANTMAKKER    W  10 N 0
33  W-ANT1ALENE    W  10 N 0
34  W-ANT2ALENE    W  10 N 0
35  W-ANT2DUBM     W  10 N 0    . * ANT DUBL PÅ INN2 MED  MAKKER PÅ INN1
36  W-ANT2DUBA     W  10 N 0    . * ANT DUBL PÅ INN2 UTEN MAKKER PÅ INN1
37
38
39  JOB INPUT (INN1 KEY (I1-FNR) +
40            INN2 KEY (I2-FNR))   FINISH SLUTT
41    IF MATCHED
42      IF NOT DUPLICATE INN2  OR  FIRST-DUP INN2
43 *                                                ENTEN (IKKE DUBLETT) EL.
44 *                                                      (1. DUBLETT) PÅ I2
45        W-ANTMAKKER = W-ANTMAKKER + 1
46        U-2REC  = I2-REC
47        U-1REC  = I1-REC
48        PUT UT
49      ELSE
50        W-ANT2DUBM = W-ANT2DUBM + 1
51      END-IF
52    ELSE
53      IF INN2
54        IF NOT DUPLICATE INN2  OR  FIRST-DUP INN2
55          W-ANT2ALENE = W-ANT2ALENE + 1
56          U-2REC  = I2-REC
57          U-1REC  = ' '                . * SETTER FELTENE FOR INN1 BLANKE
58          PUT UT
59        ELSE
60          W-ANT2DUBA = W-ANT2DUBA + 1
61        END-IF
62      ELSE
63        W-ANT1ALENE = W-ANT1ALENE + 1
64        U-2REC = ' '                   . * SETTER FELTENE FOR INN2 BLANKE
65        U-FNR  = I1-FNR                . * FØDSELSNR HENTES HER FRA INN1
66        U-1REC = I1-REC
67        PUT UT
68      END-IF
69    END-IF
70
71  SLUTT. PROC
72    DISPLAY NEWPAGE +
73            '- GEPMATCD ----------- SUMMER --------------'
74    DISPLAY INN1:RECORD-COUNT 'REC FRA INN1'
75    DISPLAY INN2:RECORD-COUNT 'REC FRA INN2'
76    DISPLAY W-ANT2DUBM        'DUBL PÅ INN2 MED MAKKER (LEST FORBI)'
77    DISPLAY W-ANT2DUBA        'DUBL PÅ INN2 UTEN MAKKER(LEST FORBI)'
78    DISPLAY W-ANTMAKKER       'REC SOM MATCHET'
79    DISPLAY W-ANT1ALENE       'REC KUN FRA INN1'
80    DISPLAY W-ANT2ALENE       'REC KUN FRA INN2'
81    DISPLAY UT:RECORD-COUNT   'REC SKREVET UT'
82    DISPLAY '-----------------------------------------'
83  END-PROC
```

GEPMATCHZ er en litt spesiell match. Nå er det INN1 som er "hovedfilen". Det er denne vi skal skrive ut med opplysninger hentet fra INN2. Vi skal summere et felt (I2-OPPL) fra alle recordene med likt matchebegrep (dubletter eller single) på INN2 og summen skal vi overføre til INN1-recorden med samme matchebegrep.

Vi starter med å summere det aktuelle feltet på INN2 inn i et arbeidsfelt (linje 36). Vi gjør det bare hvis feltet har en numerisk verdi (linje 35).

Etterpå tester vi på om INN1 og INN2 matcher (linje 40). Hvis de gjør det, sjekker vi videre om recorden er den siste på matchebegrepet fra INN2 ("single"-record eller LAST-DUP). Hvis den er det, skriver vi ut INN1-recorden med det arbeidsfeltet vi har summert opp. Vi må da også passe på å nullstille dette arbeidsfeltet, så det ligger klart for neste record.
Hvis recordene matcher, uten at recorden fra INN2 er den siste på gjeldende matchebegrep (ikke LAST-DUP), skriver vi ingenting ut, programmet går opp igjen til JOB INPUT og leser neste fra INN2. (Husk at Easytrieve leser fil-2 helt til matchebegrepet blir større enn på fil-1).

Hvis vi ikke får match, men matchebegrepet fins på INN1 (linje 49), skriver vi ut recorden med null i sumfeltet på recorden. Vi må *ikke* nullstille arbeidsfeltet her, for i det ligger det tall fra en INN2-record som allerede er lest og den vil kanskje matche neste INN1-record.

Hvis det en ident fra *bare* INN2, nullstiller vi arbeidsfeltet, så det er klart for neste ident som kanskje vil matche.

Alle arbeidsfelt som defineres numeriske, får automatisk null i oppstartverdi.

Eksempel.

Data inn til programmet:

Fil INN1:

```
          1    1    2    2    3
1...5....0....5....0....5....0
FØRSTEKONSULENT    01124711122
FULLMEKTIG         02034422233
SEKRETÆR           03055933344
FØRSTESEKRETÆR     05045244455
PLANLEGGER         06116055566
```

Fil INN2:

```
          1    1
1...5....0....5.
01124711122 0010
01124711122 0100
02034422233 0800
03055933344 0250
03055933344 0020
03055933344 0070
03055933344 0550
06116055566 0900
06116055566 0100
06116055566 0050
```

Data ut fra programmet (fil UT):

```
          1    1    2    2    3
1...5....0....5....0....5....0....
FØRSTEKONSULENT    011247111220110
FULLMEKTIG         020344222330800
SEKRETÆR           030559333440890
FØRSTESEKRETÆR     050452444550000
PLANLEGGER         061160555661050
```

```
1  ************************* EASYTRIEVE PLUS ******************************
2  *  PROD.NR.: 000
3  *  PROSJEKT: EASYTRIEVE-EKSEMPLER
4  ************************************************************************
5  *  PROGRAM : GEPMATCZ
6  *  LAGET   : 16.03.93 AV JOHN ERIK SJØRBOTTEN
7  ************************************************************************
8  *  BESKRIV.: MATCHING AV 2 FILER.
9  *            ANTALL TIMER SUMMERES PR PERSON FOR INN2 (INKL DUBLETTER)
10 *            SUMFELTET OVERFØRES SÅ TIL MAKKER PÅ INN1.
11 *            (HVIS INN1 HAR DUBLETTER, VIL IKKE DISSE FÅ MAKKER.)
12 *  REF.    :
13 ************************************************************************
14  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
15  FILE INN1                      . * HOVED-FIL
16    I1-REC       1  30 A
17    I1-FNR      20  11 A
18
19  FILE INN2                      . * TIME-FIL
20    I2-FNR       1  11 A
21    I2-OPPL     13   4 N
22
23  FILE UT
24     U-REC1      1  30 A
25     U-OPPL2    31   4 N
26
27  W-SUM           W   4 N        . * SUMFELT FOR ANTALL TIMER
28  W-ANTMAKKER     W  10 N 0
29  W-ANT1ALENE     W  10 N 0
30  W-ANT2ALENE     W  10 N 0
31
32
33  JOB INPUT (INN1 KEY (I1-FNR)  INN2 KEY (I2-FNR))  FINISH SLUTT
34    IF INN2
35      IF I2-OPPL NUMERIC
36        W-SUM = W-SUM + I2-OPPL        . * SUMMERER ANT TIMER (PR PERSON)
37      END-IF
38    END-IF
39
40    IF MATCHED
41      IF NOT DUPLICATE INN2  OR  LAST-DUP INN2
42        W-ANTMAKKER = W-ANTMAKKER + 1
43        U-REC1  = I1-REC
44        U-OPPL2 = W-SUM
45        PUT UT
46        W-SUM = 0                      . * NULLSTILLER SUMFELT
47      END-IF
48    ELSE
49      IF INN1
50        W-ANT1ALENE = W-ANT1ALENE + 1
51        U-REC1  = I1-REC
52        U-OPPL2 = 0
53        PUT UT
54      ELSE
55        W-ANT2ALENE = W-ANT2ALENE + 1
56        W-SUM = 0                      . * NULLSTILLER SUMFELT
57      END-IF
58    END-IF
59
60  SLUTT. PROC
61    DISPLAY NEWPAGE '- GEPMATCZ - PÅFØRER SUM AV TIMEVERK -------'
62    DISPLAY INN1:RECORD-COUNT 'REC FRA INN1'
63    DISPLAY INN2:RECORD-COUNT 'REC FRA INN2'
64    DISPLAY W-ANTMAKKER       'REC FRA INN1 MED MAKKER'
65    DISPLAY W-ANT1ALENE       'REC KUN FRA INN1'
66    DISPLAY W-ANT2ALENE       'REC KUN FRA INN2'
67    DISPLAY UT:RECORD-COUNT   'REC SKREVET UT'
68    DISPLAY '-----------------------------------------'
69  END-PROC
```

GEPMATCHX henter opplysninger fra seks forskjellige filer som sammen med opplysninger fra filen INN danner en ny fil, UT. INN er "hovedfilen" som vi skal overføre alle recordene fra, uansett om de har makkere eller ikke i noen av de andre filene.

Vi har definert 6 opptellingsfelt (ett som gjentas 6 ganger) (linje 70) for å telle opp antall matcher mellom INN og hver av de andre filene. Et tall i parentes bak feltnavnet sier hvilket av de seks vi vil bruke.

Vi må ha alle filene med i JOB INPUT (med likt definert matchebegrep). Vi legger "hovedfilen" INN til slutt, dermed får vi også koplet opplysninger til eventuelle dubletter på denne.

Vi starter her med å sjekke om recorden fins på INN (linje 82). Gjør den ikke det, betyr det at en record med lavere matchebegrep er funnet på en av de andre filene, så videre testing er uinteressant og vi hopper til JOB INPUT for ny runde.
Finnes den derimot på INN, klargjør vi ut-recorden: blanker den ut og flytter INN-recorden dit. Så sjekker vi om det er match mellom INN-recorden og en etter en av de andre inn-filene. Blir det match, henter vi opplysningene fra den aktuelle inn-filen og bygger slik opp den nye UT-recorden. Der INN ikke får match, har vi jo allerede satt inn blanke. Når vi har sjekket match mot alle filene, skriver vi UT-recorden.

```
 1 //K415JSJX JOB 4353,'  J O H N  ',MSGLEVEL=(1,1),
 2 //         MSGCLASS=X,CLASS=A,NOTIFY=K415JSJ,REGION=4096K
 3 //STEP1    EXEC PGM=EZTPA00,REGION=4096K
 4 //STEPLIB  DD DSN=SSB2.EASYPL60.LOAD,DISP=SHR
 5 //EZTVFM   DD UNIT=WORK,SPACE=(TRK,(25,50))
 6 //AR88     DD DSN=PX213.S4353.REDIG88,DISP=SHR
 7 //AR87     DD DSN=PX213.S4353.REDIG87,DISP=SHR
 8 //AR86     DD DSN=PX213.S4353.REDIG86,DISP=SHR
 9 //AR85     DD DSN=PX213.S4353.REDIG85,DISP=SHR
10 //AR84     DD DSN=PX213.S4353.REDIG84,DISP=SHR
11 //AR83     DD DSN=PX213.S4353.REDIG83,DISP=SHR
12 //INN      DD DSN=PX213.S4353.UTVALG88,DISP=OLD
13 //UT       DD DSN=PX213.S4353.NAVFKOBL,DISP=OLD
14 //SYSPRINT  DD  SYSOUT=*
15 //SYSOUT    DD  SYSOUT=*
16 //SYSIN     DD  *
17 ************************* EASYTRIEVE PLUS ******************************
18 *  PROD.NR.: 000
19 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
20 ************************************************************************
21 *  PROGRAM : GEPMATCX
22 *  LAGET   : 22/03-93 AV JOHN ERIK SJØRBOTTEN
23 ************************************************************************
24 *  BESKRIV.: MATCHER EN UTVALGSFIL (INN) MOT 6 ÅRGANGSFILER.
25 *            UTVALGSFILEN ER UTGANGSPUNKTET, VI SJEKKER OM DEN HAR
26 *            MAKKER I EN ETTER EN AV DE ANDRE FILENE OG OVERFØRER
27 *            DATA FRA DE FILENE DEN HAR MAKKERE I.
28 *  REF.    :
29 ************************************************************************
30
31  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
32  FILE AR88
33    I88-FNR      1  11 A
34    I88-DAT     12  14 A
35
36  FILE AR87
37    I87-FNR      1  11 A
38    I87-DAT     12  14 A
39
40  FILE AR86
41    I86-FNR      1  11 A
42    I86-DAT     12  13 A
43                        fortsetter...
```

```
44  FILE AR85
45    I85-FNR       1  11 A
46    I85-DAT      12  13 A
47
48  FILE AR84
49    I84-FNR       1  11 A
50    I84-DAT      12  13 A
51
52  FILE AR83
53    I83-FNR       1  11 A
54    I83-DAT      12  13 A
55
56  FILE INN
57    I-REC         1  17 A
58    I-FNR         1  11 A
59
60  FILE UT
61    U-REC         1  97 A
62    U-REC1        1  17 A
63    U-88DAT      18  14 A
64    U-87DAT      32  14 A
65    U-86DAT      46  13 A
66    U-85DAT      59  13 A
67    U-84DAT      72  13 A
68    U-83DAT      85  13 A
69
70  WSUM            W   6 N 0 OCCURS 6 . * 6 sumfelt (ett for hver årg.)
71
72
73  JOB INPUT (AR88 KEY (I88-FNR) +
74             AR87 KEY (I87-FNR) +
75             AR86 KEY (I86-FNR) +
76             AR85 KEY (I85-FNR) +
77             AR84 KEY (I84-FNR) +
78             AR83 KEY (I83-FNR) +
79             INN  KEY (I-FNR))  +
80           FINISH SLUTT
81
82   IF INN
83    U-REC  = ' '
84    U-REC1 = I-REC
85    IF MATCHED INN AR88. U-88DAT = I88-DAT. WSUM(1) = WSUM(1) + 1. END-IF
86    IF MATCHED INN AR87. U-87DAT = I87-DAT. WSUM(2) = WSUM(2) + 1. END-IF
87    IF MATCHED INN AR86. U-86DAT = I86-DAT. WSUM(3) = WSUM(3) + 1. END-IF
88    IF MATCHED INN AR85. U-85DAT = I85-DAT. WSUM(4) = WSUM(4) + 1. END-IF
89    IF MATCHED INN AR84. U-84DAT = I84-DAT. WSUM(5) = WSUM(5) + 1. END-IF
90    IF MATCHED INN AR83. U-83DAT = I83-DAT. WSUM(6) = WSUM(6) + 1. END-IF
91    PUT UT
92   END-IF
93
94  SLUTT. PROC
95    DISPLAY NEWPAGE '------PROGRAM GEPMATCX------'
96    DISPLAY 'ANT LEST FRA INN:' INN:RECORD-COUNT
97    DISPLAY 'ANT SKREVET UT:  ' UT:RECORD-COUNT
98    DISPLAY 'ANT MATCH MOT 88:' WSUM(1)
99    DISPLAY 'ANT MATCH MOT 87:' WSUM(2)
100   DISPLAY 'ANT MATCH MOT 86:' WSUM(3)
101   DISPLAY 'ANT MATCH MOT 85:' WSUM(4)
102   DISPLAY 'ANT MATCH MOT 84:' WSUM(5)
103   DISPLAY 'ANT MATCH MOT 83:' WSUM(6)
104   DISPLAY '----------------------------'
105 END-PROC
```

GEPTABLE henter data fra to filer og skriver resultatat ut på en tredje fil.
Her bruker vi *ikke* den tradisjonelle matche-metoden. I mange tilfeller kan det være hensiktsmessig å definere den ene inn-filen som en tabell (TABLE). Kravene som stilles til denne filen er at den er sortert stigende og at det *ikke finnes dubletter* i søkebegrepet. Hvis tabell-filen har mer enn 256 records, må størrelsen angis i parentes (ref. linje 29). Tallet som angis her kan være større enn antall records, men ikke mindre. Det fine med denne metoden er at det ikke stilles noen krav til den andre filen når det gjelder sortering, dubletter osv..

Søke-argumentet på tabell-filen beskrives med ARG (kommune-nummer), dette må defineres likt på tabell-filen og INN-filen (linje 22 og 30) dvs. begge numeriske eller begge alfanumeriske. Vi leser her record for record fra INN. For å få overført data fra KAT benytter vi SEARCH. Vi søker med kommune-nummer (KOMM) som nøkkel. Data fra KAT overføres til UTV-KODE på UT, dvs. innholdet av DESC (posisjon 40) (linje 31). Hvis vi ikke får makker (IF NOT KAT) setter vi UTV-KODE til blank. Til slutt skrives den ferdige recorden ut.

Det finnes imidlertid begrensninger ved denne metoden. Vi kan kun ha én ARG og én DESC i TABLE-filen og vi kan ikke overføre mer enn 254 posisjoner. En måte å få overført flere posisjoner på er å definere samme inn-fil flere ganger. Hvis vi definerer samme inn-fil som to tabeller må vi gjøre to SEARCH'er (på feks. KAT1 og KAT2). Totalt kan vi da overføre 254 x 2 posisjoner. Hvis vi ønsker å omredigere, benytter vi work-felter som mellomlagring før vi overfører dataene til UT (ref. neste eks. GEPTABLE2).

```
1  ************************ EASYTRIEVE PLUS ******************************
2  *  PROD.NR.: 000
3  *  PROSJEKT: EASYTRIEVE-EKSEMPLER
4  ***********************************************************************
5  *  PROGRAM : GEPTABLE
6  *  LAGET   : 17.03.93 AV VIDAR HALVORSEN
7  ***********************************************************************
8  *  BESKRIV.:
9  *            LESER INN - SØKER I TABLE-FILE MED NØKKEL (ARG)
10 *                        VED MAKKER OVERFØRES DATA (DESC) TIL UT.
11 *
12 *            HUSK PÅ:    TABLE-FILEN MÅ VÆRE SORTERT PÅ ARG OG
13 *                        IKKE HA DUBLETTER I NØKKEL-BEGREPET -
14 *                        STØRRELSEN (ANT. RECORDS) ANGIS I PARENTES.
15 *                        (MAX DESC = 254 BYTES)
16 *  REF.    :
17 ***********************************************************************
18
19  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
20
21  FILE INN
22    KOMM        1   4 A
23    INN-REC     1  80 A
24
25  FILE UT
26    UT-REC      1  80 A
27    UTV-KODE   81   1 A
28
29  FILE KAT TABLE (462)
30    ARG         1   4 A
31    DESC       40   1 A
32
33  ANT-UMAKK     W   7 N 0
34
35
36  JOB INPUT INN FINISH SLUTT
37
38    UT-REC = INN-REC
39    SEARCH KAT WITH KOMM GIVING UTV-KODE
40      IF NOT KAT
41        ANT-UMAKK = ANT-UMAKK + 1
42        UTV-KODE = ' '
43      END-IF
44    PUT UT
45
46  SLUTT. PROC
47    DISPLAY '*********************************'
48    DISPLAY '*   PROGRAM-NR.: GEPTABLE        *'
49    DISPLAY '*********************************'
50    DISPLAY ' '
51    DISPLAY 'ANT. LEST FRA INN :' INN:RECORD-COUNT
52    DISPLAY 'ANT. UTEN MAKKER .:' ANT-UMAKK
53    DISPLAY 'ANT. SKREVET .....:' UT:RECORD-COUNT
54    DISPLAY ' '
55  END-PROC
```

GEPTABLE2 henter data fra to filer og skriver resultatat ut på en tredje fil.
Her overfører vi mer enn 254 posisjoner og vi omredigerer noen av feltene. De 200 første posisjonene overføres direkte til UT. Posisjon 201 til 374 overfører vi til work-område med underdefinering. Feltene kan nå legges på UT i de posisjonene vi ønsker. I dette eksemplet skriver vi kun ut records med makker i katalogen, ikke-makkere telles opp.

```
 1 //K415VIHA JOB 4300,' VIDAR H.',MSGLEVEL=(1,1),
 2 //         MSGCLASS=X,CLASS=A,NOTIFY=K415VIH,REGION=4M
 3 /*ROUTE PRINT RMT5
 4 //STEP1     EXEC PGM=EZTPA00
 5 //STEPLIB   DD DSN=SSB2.EASYPL60.LOAD,DISP=SHR
 6 //EZTVFM    DD UNIT=WORK,SPACE=(TRK,(25,50))
 7 //INN       DD DSN=PX215.----------------,DISP=SHR
 8 //KAT1      DD DSN=PX215.S0101.VIH.KATDATA,DISP=SHR
 9 //KAT2      DD DSN=PX215.S0101.VIH.KATDATA,DISP=SHR
10 //UT        DD DSN=PX215.----------------,DISP=OLD
11 //SYSPRINT  DD  SYSOUT=*
12 //SYSOUT    DD  SYSOUT=*
13 //SYSIN     DD  *
14 ************************* EASYTRIEVE PLUS ******************************
15 *  PROD.NR.: 000
16 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
17 ************************************************************************
18 *  PROGRAM : GEPTABL2
19 *  LAGET   : 17.03.93 AV VIDAR HALVORSEN
20 ************************************************************************
21 *  BESKRIV.:
22 *            LESER INN - SØKER I TABLE-FILE MED NØKKEL (ARG)
23 *                        VED MAKKER OVERFØRES DATA (DESC) TIL WORK.
24 *                        VI HAR HER MER ENN 254 KARAKTERER SOM SKAL
25 *                        OVERFØRES - SÅ VI DEFINERER TO KATALOGER FOR
26 *                        SAMME FIL-IDENT OG TAR SEARCH TO GANGER.
27 *                        DEN ENE DELEN SOM OVERFØRES SKAL OMREDIGERES
28 *                        VI UNDERDEFINERER DERFOR WORK-FELTET OG
29 *                        FLYTTER DATAENE DIT VI ØNSKER PÅ UT-FILEN
30 *
31 *            HUSK PÅ:    TABLE-FILEN MÅ VÆRE SORTERT PÅ ARG OG
32 *                        IKKE HA DUBLETTER I NØKKEL-BEGREPET -
33 *                        MAKS. ANTALL RECORDS ANGIS I PARENTES.
34 *  REF.    :
35 ************************************************************************
36
37                          fortsetter...
```

```
38
39  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
40  FILE INN
41    FNR                7  11 A
42    SREGS             26   1 A
43    SREGS-DAT         27   6 A
44    REGDAT-SB         36   6 A
45    TILD-SB           44   3 A
46    KODE-FFLND        47   3 A
47    DATO-FFLND        50   6 A
48
49  FILE UT
50    TABREC1            1 200 A
51    FLT5U            201 170 A
52    FLT4U            371   1 A
53    FLT3U            372   1 A
54    FLT2U            373   1 A
55    FLT1U            374   1 A
56    SREGS            375   1 A
57    SREGS-DAT        376   6 A
58    REGDAT-SB        382   6 A
59    TILD-SB          388   3 A
60    KODE-FFLND       391   3 A
61    DATO-FFLND       394   6 A
62
63  FILE KAT1 TABLE (13750)
64    ARG                7  11 A
65    DESC               1 200 A
66
67  FILE KAT2 TABLE (13750)
68    ARG                7  11 A
69    DESC             201 174 A
70
71  W-TABREC2            W 174 A
72    FLT1  W-TABREC2        1 A
73    FLT2  W-TABREC2 +1     1 A
74    FLT3  W-TABREC2 +2     1 A
75    FLT4  W-TABREC2 +3     1 A
76    FLT5  W-TABREC2 +4   170 A
77
78  W-ANTUMAKK         W     7 N 0
79
80
81  JOB INPUT INN FINISH SLUTT
82
83    SEARCH KAT1 WITH FNR GIVING TABREC1
84    IF KAT1
85      MOVE LIKE INN TO UT
86      SEARCH KAT2 WITH FNR GIVING W-TABREC2
87      FLT5U = FLT5
88      FLT4U = FLT4
89      FLT3U = FLT3
90      FLT2U = FLT2
91      FLT1U = FLT1
92      PUT UT
93    ELSE
94      W-ANTUMAKK = W-ANTUMAKK + 1
95    END-IF
96
97  SLUTT. PROC
98    DISPLAY '*********************************'
99    DISPLAY '*   PROGRAM-NR.: GEPTABLE2      *'
100   DISPLAY '*********************************'
101   DISPLAY ' '
102   DISPLAY 'ANTALL LES FRA INN ...: ' INN:RECORD-COUNT
103   DISPLAY 'ANTALL UTEN FNR-MAKKER: ' W-ANTUMAKK
104   DISPLAY 'ANTALL SKREVET TOT. ..: ' UT:RECORD-COUNT
105   DISPLAY ' '
106 END-PROC
```

GEPDUBL kontrollerer en fil for dubletter, fjerner disse og skriver ut en dublettfri fil. Det som her er viktig, er at filen *må være sortert stigende* på dublett-begrepet. Selve programmet blir veldig enkelt.

Vi trenger bare definere det feltet (eller feltene) på inn-filen (INN) som vi skal teste på. På ut-filen trenger vi ikke definere noen felt i det hele tatt, for når vi bruker PUT UT FROM INN, blir UT-recorden lik INN-recorden.

JOB INPUT skriver vi på samme måte som i en match, med dublettbegrepet som nøkkel. Her definerer vi naturligvis kun *en* fil.

Når en record er lest, tester vi om dette er en dublett (DUPLICATE) og at det samtidig ikke er den første dubletten (NOT FIRST-DUP). Hvis denne testen slår til, skriver vi bare recorden ut på liste (DISPLAY). Hvis testen ikke slår til (ELSE), er recorden en ikke-dublett og vi skriver den til UT-fil.

Husk at med bare DUPLICATE, uten FIRST-DUP, får vi også med første recorden i dublettbegrepet som dublett (se nederst på side 3).

```
1  //K415JSJ1 JOB 4354,' J O H N ',MSGLEVEL=(1,1),
2  //         MSGCLASS=X,CLASS=A,NOTIFY=K415JSJ,REGION=4096K
3  /*ROUTE PRINT RMT11
4  //STEP1    EXEC PGM=EZTPA00,REGION=4096K
5  //STEPLIB  DD DSN=SSB2.EASYPL60.LOAD,DISP=SHR
6  //EZTVFM   DD UNIT=WORK,SPACE=(TRK,(25,50))
7  //INN         DD DSN=PX213.------------,DISP=SHR
8  //UT          DD DSN=PX213.------------,DISP=OLD
9  //SYSPRINT  DD  SYSOUT=*
10 //SYSOUT    DD  SYSOUT=*
11 //SYSIN     DD  *
12 ************************* EASYTRIEVE PLUS ******************************
13 *  PROD.NR.: 000
14 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
15 ************************************************************************
16 *  PROGRAM : GEPDUBL
17 *  LAGET   : 26.03.93 AV JOHN ERIK SJØRBOTTEN
18 ************************************************************************
19 *  BESKRIV.: DUBLETTKONTROLL.
20 *            FJERNER EVENTUELLE DUBLETTER
21 *            FILEN MÅ VÆRE SORTERT PÅ IDENTEN VI TESTER PÅ.
22 *  REF.    :
23 ************************************************************************
24
25  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
26
27  FILE INN
28    I-FNR      20  11 A
29
30  FILE UT
31
32  W-ANTDUBL     W  10 N 0
33
34
35  JOB INPUT (INN KEY (I-FNR))  FINISH SLUTT
36
37    IF DUPLICATE INN  AND  NOT FIRST-DUP INN
38      W-ANTDUBL = W-ANTDUBL + 1
39      DISPLAY 'DUBLETT (FJERNET): ' I-FNR +
40              ',  REC.NR PÅ FIL:' INN:RECORD-COUNT
41    ELSE
42      PUT UT FROM INN
43    END-IF
44
45  SLUTT. PROC
46    DISPLAY NEWPAGE +
47            '- GEPDUBL - FJERNING AV DUBLETTER ---'
48    DISPLAY INN:RECORD-COUNT 'REC LEST'
49    DISPLAY W-ANTDUBL        'DUBLETTER (FJERNET)'
50    DISPLAY UT:RECORD-COUNT  'REC SKREVET UT'
51    DISPLAY '-------------------------------------'
52  END-PROC
```

GEPOCCUR produserer en tabell-fil v.h.a. intern aggregering i programmet.
Vi begynner med å nullstille alle aggregerings-feltene for denne tabellen. Selve nullstillingen skjer i en egen prosedyre, NULLSTILL PROC. Her nullstiller vi først alle kolonnene i første raden (XR=1 og XK=1 til 9), så nullstiller vi neste rad (XR=2 og XK=1 til 9) osv. til hele tabellen er nullstillt (XR=7 og XK=1 til 9). Dette *må gjøres* før vi begynner aggregeringen ellers vil vi få feilmelding (aggregerings-feltene inneholder unumeriske verdier). For å få til dette benytter vi oss av START i JOB-setningen, proc'n du legger inn her (NULLSTILL) blir utført før alt annet.

Når dette er gjort bestemmer vi indeks-verdiene. Vi reserverer indeks-verdien 1 for totaler. Først bestemmes kolonne-indeksen (XK) i BYGGE-ÅR PROC, så rad-indeksen (XR) i LANDS-DEL PROC. Vi aggregerer så opp i de aktuelle cellene i tabellen. Her har vi valgt å aggregere opp med en oppblåsingsfaktor (brukes ved utvalg), hvis vi derimot har en record pr. enhet skal det aggregeres opp med faktor = +1. Her aggregerer vi total for totalraden (1 1), total for kolonnene (1 XK), total for radene (XR 1) og til slutt cellen vi bestemte i BYGGE-ÅR og LANDS-DEL PROC (XR XK).

Da alle recordene er behandlet, overfører vi dataene fra intern-tabellen til UTFIL (tilsv. måte som ved NULLSTILL) og skriver. I dette eksemplet har vi brukt desimaler, foretar derfor en avrunding (ROUNDED).

```
 1 ************************* EASYTRIEVE PLUS ******************************
 2 *  PROD.NR.: 000
 3 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
 4 ************************************************************************
 5 *  PROGRAM : GEPOCCUR
 6 *  LAGET   : 17.03.93 AV VIDAR HALVORSEN
 7 ************************************************************************
 8 *  BESKRIV.:
 9 *             PRODUSERER EN TABELL PÅ FIL V.H.A. INTERN AGGREGERING.
10 *             LANDSDELER I FORSPALTEN (INDEKS = XR)
11 *             OG HUSETS BYGGEÅR I HODET (INDEKS = XK)
12 *  REF.    :
13 ************************************************************************
14
15 PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
16
17  FILE INNFIL
18    I-KOM           1   4 A
19    I-FLK           1   2 A
20    I-BÅR          12   1 A
21    I-BOLVEKT      13   4 N 2
22
23  FILE UTFIL
24    U-REC           1  72 A
25    U-TALLBL        1   8 A   OCCURS 9 INDEX XK.    * TALL + BLANK
26    U-TALL  U-TALLBL    7 N 0
27 * --------------------
28 *    INTERN-TABELL
29 * --------------------
30  W-RAD             W  90 A   OCCURS 7 INDEX XR
31  W-KOL    W-RAD       10 N 2 OCCURS 9 INDEX XK
32
33  JOB INPUT  INNFIL +
34             START  NULLSTILL +
35             FINISH LEGG-INN-SKRIV
36
37    PERFORM BYGGE-ÅR
38    PERFORM LANDS-DEL
39 *                                                 I-BOLVEKT ADDERES I:
40    W-KOL(1  1)   = W-KOL(1  1)   + I-BOLVEKT . * TOTAL FOR TOTALRAD
41    W-KOL(1  XK)  = W-KOL(1  XK)  + I-BOLVEKT . * TOTAL FOR KOLONNEN
42    W-KOL(XR 1)   = W-KOL(XR 1)   + I-BOLVEKT . * TOTAL FOR RADEN
43    W-KOL(XR XK)  = W-KOL(XR XK)  + I-BOLVEKT . * SPES. KOL/RAD
44                          fortsetter...
```

```
45  BYGGE-ÅR. PROC
46 *  ---------  FINNER RETTE KOLONNEN SOM DET SKAL TELLES OPP I
47     IF I-BÅR = '1'. XK = 2. ELSE
48       IF I-BÅR = '2'. XK = 3. ELSE
49         IF I-BÅR = '3'. XK = 4. ELSE
50           IF I-BÅR = '4'. XK = 5. ELSE
51             IF I-BÅR = '5'. XK = 6. ELSE
52               IF I-BÅR = '6'. XK = 7. ELSE
53                 IF I-BÅR = '7'. XK = 8. ELSE
54                   IF I-BÅR = '8'. XK = 9
55                   END-IF
56                 END-IF
57               END-IF
58             END-IF
59           END-IF
60         END-IF
61       END-IF
62     END-IF
63  END-PROC
64
65  LANDS-DEL. PROC
66 *  ---------  FINNER RETTE RADEN SOM DET SKAL TELLES OPP I
67     IF FYLKE = '02' '03'.            XR = 2. ELSE
68       IF FYLKE = '01' '04' THRU '08'. XR = 3. ELSE
69         IF FYLKE = '09' THRU '11'.      XR = 4. ELSE
70           IF FYLKE = '12' THRU '15'.      XR = 5. ELSE
71             IF FYLKE = '16' '17'.           XR = 6. ELSE
72               IF FYLKE = '18' THRU '20'.      XR = 7
73               END-IF
74             END-IF
75           END-IF
76         END-IF
77       END-IF
78     END-IF
79  END-PROC
80
81  NULLSTILL. PROC
82     XR = 1
83     DO WHILE XR LE 7
84       XK = 1
85       DO WHILE XK LE 9
86         W-KOL(XR XK) = 0
87         XK = XK + 1
88       END-DO
89      XR = XR + 1
90      END-DO
91  END-PROC
92
93  LEGG-INN-SKRIV. PROC
94     XR = 1
95     DO WHILE XR LE 7
96       U-REC = ' '
97       XK = 1
98       DO WHILE XK LE 9
99         U-TALL(XK) ROUNDED = W-KOL(XR XK)
100        XK = XK + 1
101      END-DO
102      PUT UTFIL
103      XR = XR + 1
104    END-DO
105    DISPLAY ' '
106    DISPLAY ' ------------------------'
107    DISPLAY '  PROGRAM-ID.: GEPOCCUR'
108    DISPLAY ' ------------------------'
109    DISPLAY ' '
110    DISPLAY ' ANT. LEST FRA INNFIL :' INNFIL:RECORD-COUNT
111    DISPLAY ' ANT. PÅ UTFIL .......:' UTFIL:RECORD-COUNT
112    DISPLAY ' '
113  END-PROC
```

GEPVIRT teller opp antall forekomster pr. tettsted. Inn-filen er ikke sortert på tettsted, men dette gjøres enkelt med en egen SORT-aktivitet (linje 33). Vi sorterer med andre ord inn-filen over på en midlertidig fil (VIRTUAL). Deretter leses denne filen i JOB INPUT og vi kan nå teste på brudd i tettsted. For at vi ikke skal få brudd ved første les, tester vi på om det er den første recorden og setter W-TETTSTED-FORR = V-TETTSTED (linje 39). W-ANTALL adderes opp med 1 for hver dublett. Da vi får brudd, skriver vi ut det ferdig behandlede tettstedet (W-TETTSTED-FORR) og tilhørende antall. Slik jobber programmet helt til alle recordene er lest.

Husk på at vi hele tiden ligger en record foran den vi skriver ut. Dette fører til at vi til slutt har et tettsted igjen som vi ikke har skrevet. For å få skrevet ut dette benytter vi oss av FINISH i JOB-setningen. Proc'n du legger inn her (SLUTT) blir utført etter at alt annet er ferdig. Da vi har utskriften i en egen proc er det nok å skrive PERFORM SKRIV i SLUTT PROC.

```
 1 ************************* EASYTRIEVE PLUS ******************************
 2 *  PROD.NR.: 000
 3 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
 4 ************************************************************************
 5 *  PROGRAM : GEPVIRT
 6 *  LAGET   : 17.03.93 AV VIDAR HALVORSEN
 7 ************************************************************************
 8 *  BESKRIV.:
 9 *            SORTERER INPUT OVER PÅ EN VIRTUELL FIL (INNV),
10 *            LESER DENNE OG TESTER PÅ BRUDD I TETTSTED,
11 *            TELLER OPP ANTALL FOREKOMSTER FOR HVERT TETTSTED,
12 *            SKRIVER UT TETTSTED MED TILHØRENDE ANTALL
13 *  REF.    :
14 ************************************************************************
15
16  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
17
18  FILE INN
19    I-TETTSTED   2   4 A
20
21  FILE INNV VIRTUAL F(5)
22    V-TETTSTED   2   4 A
23
24  FILE UT
25    U-TETTSTED   1   4 A
26    U-BLANK      5   1 A
27    U-ANTALL     6   7 N
28
29  W-ANTALL       W   7 N 0
30  W-TETTST-FORR  W   4 A
31
32
33  SORT INN TO INNV USING I-TETTSTED
34
35
36  JOB INPUT INNV  FINISH SLUTT
37
38    IF INNV:RECORD-COUNT = 1
39      W-TETTST-FORR = V-TETTSTED.    * For å unngå brudd på 1. record
40    END-IF
41
42    IF V-TETTSTED NE W-TETTST-FORR
43      PERFORM SKRIV
44    END-IF
45    W-ANTALL = W-ANTALL + 1
46
47  SKRIV. PROC
48    U-TETTSTED     = W-TETTST-FORR
49    U-BLANK        = ' '
50    U-ANTALL       = W-ANTALL
51    PUT UT
52    W-ANTALL       = 0
53    W-TETTST-FORR = V-TETTSTED
54  END-PROC
55
56  SLUTT. PROC
57    PERFORM SKRIV
58    DISPLAY ' '
59    DISPLAY '**********************'
60    DISPLAY '*  PROGRAM: GEPVIRT  *'
61    DISPLAY '**********************'
62    DISPLAY ' '
63    DISPLAY ' ANT. LEST (INNV) SORT.:' INNV:RECORD-COUNT
64    DISPLAY ' ANT. SKREVET (UT) ....:'   UT:RECORD-COUNT
65    DISPLAY ' '
66  END-PROC
```

GEPVARIO har en inn-fil med et varierende antall meldinger (maks. 42 dubletter) pr. løpenr. Vi skal produsere en ut-fil som inneholder kun en record pr. løpenr, men med et varierende antall meldinger (U-MELDING OCCURS 42 INDEX XU). XU brukes for å lokalisere en ønsket melding. På ut-recorden har vi lagt inn et felt som sier hvor mange meldinger recorden har. Da slipper vi ved senere bruk å teste på dette.
UTFIL blir altså en fil med variabel recordlengde. Slike recorder har record-lengden lagret i 4 posisjoner foran de vanlige datafeltene. Datablokkene som inneholder slike recorder har også et tilsvarende 4 posisjoners felt liggende først i hver blokk. Dette trenger vi ikke tenke på når vi bruker feltene i Easytrieve, men når vi lager slike recorder må vi regne ut recordlengden og legge det i et felt som heter RECORD-LENGTH for filen før vi skriver recorden (linje 57).
Når vi definerer recordlengden i JCL'en, tar vi maksimal recordlengde (her 264 posisjoner) og plusser på 4 posisjoner (feltet for record-lengde). Blokklengden regnes ut tilsvarende: JCL-recordlengden multipisert med antall recorder vi vil ha i blokka pluss 4 posisjoner (feltet for blokklengde).

Vi har lagt inn en test som stopper programmet med en feilmelding hvis det leses flere enn 42 meldinger inn på et løpenr. Hvis alt er ok, legger vi inn meldingen ut i U-MELDING (index XU) i ut-recorden. Vi sjekker videre på om recorden er den siste på løpenummeret ("single" eller LAST-DUP). Er den det, gjør vi klar til utskriving. Vi flytter de faste feltene fra INNFIL- til UTFIL-recorden, flytter indexen XU til antall-meldinger-feltet og regner ut record-lengden. Vi har 12 faste posisjoner, XU antall meldinger på 6 posisjoner hver. Recordlengden blir da: 12 + (6 * XW).

Vi må så huske på å nullstille indexen XU, så den er klar til å bli brukt ved neste løpenr. Vi trenger ikke å blanke ut-recorden, for vi overskriver de gamle meldingene med nye og skriver den ut med ny utregnet recordlengde for hver gang.

Eksempel.
Data inn til programmet:

```
          1    1                        1    1
1...5....0....5...          1...5....0....5...
520001 7004 700117          520004 8012 880818
520001 7004 760123          520004 8012 890203
520001 7004 810719          520004 8012 891104
520001 7004 830804          520004 8012 900403
520002 7006 681117          520004 8012 900519
520002 7006 700020          520005 6506 650217
520002 7006 720923          520005 6506 710020
520002 7006 721218          520005 6506 720203
520002 7006 730203          520005 6506 720804
520003 7502 700717          520005 6506 731223
520003 7502 880226          520005 6506 770819
520004 8012 810917          520005 6506 771204
520004 8012 860720          520007 8011 800117
520004 8012 871123          520007 8011 830620
520004 8012 880104 ↗        520007 8011 850723
```

Data ut fra programmet (variabel recordlengde):
(l=løpenr, f=fødselsår-mnd, a=antall, m1=melding, m2=melding2, osv)

```
l.....f...a.m1....m2....m3....m4....m5....m6....m7....m8....m9.... → m42...
5200017004047001177601238107198308040
5200027006056811177000207209237212187302030
52000375020270071788022600
5200048012098109178607208711238801048808188902038911049004039005190
5200056506076502177100207202037208047312237708197712040
52000780110380011783062085072300
```

```
1 ----
2 //*              rec.lengde:       264 + 4 = 268
3 //*              blokklengde: 80 * 268 + 4 = 21444
4 //*
5 //UTFIL    DD DSN=PX214.S0159.KNY.-------,
6 //             DISP=(NEW,CATLG,DELETE),UNIT=SSB,
7 //             DCB=(RECFM=VB,LRECL=268,BLKSIZE=21444),
8 //             SPACE=(21444,(6000,100),RLSE)
9 //SYSIN    DD *
10
11 ************************* EASYTRIEVE PLUS *****************************
12 *  PROD.NR.: 000
13 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
14 ***********************************************************************
15 *  PROGRAM : GEPVARIO
16 *  LAGET   : 26.03.93 AV KÅRE NYGÅRD
17 ***********************************************************************
18 *  BESKRIV.: PRODUSERER EN FIL MED VARIABEL REC.LENGDE:
19 *            DANNER EN RECORD PR. LØPENR UT FRA EN FIL MED VARIERENDE
20 *            ANTALL RECORDS (1 --> 42) PR LØPENR.
21 *            FILEN MÅ VÆRE SORTERT PÅ DETTE LØPENR.
22 *  REF.    :
23 ***********************************************************************
24
25  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
26
27 *                                        Barnefil-melding
28   FILE INNFIL
29     I-LØPENR       1   6 A
30     I-FØDTDATO     8   4 A
31     I-MELDING     13   6 A
32
33 *                                        Barnefil-variabel
34   FILE UTFIL
35     U-RECORD       1 264 A
36     U-LØPENR       1   6 A
37     U-FØDTDATO     7   4 A
38     U-ANTALL      11   2 N
39     U-MELDING     13   6 A   OCCURS 42 INDEX XU
40
41
42   JOB INPUT (INNFIL KEY (I-LØPENR))  FINISH SLUTT
43     IF XU < 42             . * plasserer meldingen i ut-reccord
44       XU = XU + 1
45       U-MELDING(XU) = I-MELDING
46     ELSE
47       DISPLAY 'flere enn 42 rec på løpenr:' I-LØPENR
48       DISPLAY ' Kjøringen BRYTES !!!'
49       STOP EXECUTE
50     END-IF
51
52 *                                        skriver en rec pr.løpenr til UTFIL
53     IF (NOT DUPLICATE INNFIL) OR (LAST-DUP INNFIL)
54       U-LØPENR   = I-LØPENR
55       U-FØDTDATO = I-FØDTDATO
56       U-ANTALL   = XU
57       UTFIL:RECORD-LENGTH = 12 + (6 * XU)
58       PUT UTFIL
59       XU = 0
60     END-IF
61
62   SLUTT. PROC
63     DISPLAY NEWPAGE 'PROG: GEPVARIO  KJØRT ' SYSDATE '  KL.' SYSTIME
64     DISPLAY '-------------------------------------------'
65     DISPLAY 'ANTALL FRA INNFIL: ' INNFIL:RECORD-COUNT
66     DISPLAY 'ANTALL TIL UTFIL.: ' UTFIL:RECORD-COUNT
67     DISPLAY '-------------------------------------------'
68   END-PROC
```

GEPRAPP velger først ut recorder med en bestemt dato som vi styrer med en parameter. Deretter produseres det to rapporter av dette utvalget.

Parameteren er egentlig en egen datafil som vi her legger inn i JCL'en (linje 8 og 9). Ved start kaller vi opp en prosedyre (LES-PRDATO. PROC). I denne leser vi parameteret ved hjelp av GET (les uten automatikk) og sjekker om vi har gitt lovlig dato.

Her er vi ikke ikke fornøyd med å bruk feltnavnene som hode til kolonnene i rapportene, derfor bruker vi HEADING ved definisjonen av de forskjellige feltene. Vi vil ha pr.datoen skrevet i tittel-linja på rapportene på formen dd.mm.åå og bruker derfor MASK på denne.

Hver gang vi leser en record med rett dato, skriver vi den til begge rapportene. Da hele inn-filen er ferdig lest, sorteres rapport-recordene (SEQUENCE). Når vi bruker CONTROL (sjekk på brudd), må vi ha rapportene sortert.

I rapporten KOM-RAPP (linje 65) lister vi ut alle recordene vi sender til rapporten. Vi hopper til ny side når vi starter på nytt fylke, uten å skrive ut fylkessummene. Ville vi fjerne totalsummen også, måtte vi skrive: CONTROL FINAL NOPRINT I-FYLKE (NEW-PAGE NOPRINT). Alle numeriske felt (NB: heltall må være definert med 0 desimaler) som er i LINE og samtidig ikke er med i CONTROL blir automatisk summert når vi bruker CONTROL.

I rapporten FYLKE-RAPP (linje 73) vil vi bare ha ut fylkessummer uten de enkelte kommunelinjene. Vi bruker da SUMMARY. Vi teller her opp antall kommunerecorder i hvert fylke ved hjelp av system-variabelen TALLY. Denne teller automatisk opp antall recorder som er skrevet til rapporten. Da TALLY er en system-variabel, legger vi inn HEADING på den her i rapporten, ellers vil hodet til denne kolonna bli TALLY.

For å få regnet ut prosenter, bruker vi en rapport-prosedyre, BEFORE-BREAK (linje 81). Denne blir utført ved brudd, rett før sumlinja skrives ut. Feltene vi bruker her må være med i LINE.

Et eksempel ligger bak program-lista.

```
1  //K415JSJM JOB 4354,' J O H N ',MSGLEVEL=(1,1),
2  //        MSGCLASS=X,CLASS=A,NOTIFY=K415JSJ,REGION=4096K
3  /*ROUTE PRINT RMT11
4  //STEP1    EXEC PGM=EZTPA00,TIME=20,REGION=4096K
5  //STEPLIB  DD DSN=SSB2.EASYPL60.LOAD,DISP=SHR
6  //EZTVFM   DD UNIT=WORK,SPACE=(TRK,(25,50))
7  //INN      DD DSN=PX213.-----------,DISP=SHR
8  //PARAM    DD *
9  311291
10 //SYSPRINT  DD  SYSOUT=*
11 //SYSOUT    DD  SYSOUT=*
12 //SYSIN     DD  *
13 ************************* EASYTRIEVE PLUS ******************************
14 *  PROD.NR.: 000
15 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
16 ************************************************************************
17 *  PROGRAM : GEPRAPP
18 *  LAGET   : 23.03.93 AV JOHN ERIK SJØRBOTTEN
19 ************************************************************************
20 *  BESKRIV.: SKRIVER UT TO RAPPORTER FRA EN KOMMUNEFIL.
21 *            EN REN UTLISTING, OG EN AGGREGERT LISTE PÅ FYLKESNIVÅ.
22 *            VHA PARAMETER HENTES RETTE PR.DATO UT.
23 *  REF.    :
24 ************************************************************************
25
26  PARM DEBUG (NOCLIST NODMAP FLOW NOXREF) LIST (NOPARM) SORT (MSG NO)
27
28  FILE PARAM                    . * PARAMETER FOR Å HENTE RIKTIG PR.DATO
29    P-REC          1   80  A
30    P-PRDATO       1    6  N
31     P-DAGMND      1    4  A
32     P-AAR         5    2  A
33
34  FILE INN                      . * KOMMUNEFIL MED FLERE PR.DATOER
35    I-PRDATO       1    6  A
36    I-KOMM         8    4  A   HEADING 'KOMMUNE'
37     I-FYLKE       8    2  A   HEADING 'FYLKE'
38    I-BEFOLKN     13    6  N 0 HEADING 'BEFOLKNING'
39    I-MENN        20    6  N 0 HEADING 'MENN'
40    I-KVIN        27    6  N 0 HEADING 'KVINNER'
41      .
42    W-PRDATO       W    6  N   MASK (A '99.99.99')
43    W-MENNPROS     W    5  N 2 HEADING ('PROSENT' 'MENN')
44    W-KVINPROS     W    5  N 2 HEADING ('PROSENT' 'KVINNER')
45
46  JOB INPUT INN  START LES-PRDATO
47    IF I-PRDATO NUMERIC
48      IF I-PRDATO = W-PRDATO
49        PRINT KOMM-RAPP
50        PRINT FYLKE-RAPP
51      END-IF
52    END-IF
53
54  LES-PRDATO. PROC
55    GET PARAM
56    IF P-DAGMND = '3006' '3112' AND P-AAR = '91' THRU '99'
57      W-PRDATO = P-PRDATO
58    ELSE
59      DISPLAY 'FEIL I PARAMETER:' P-REC
60      DISPLAY 'DATO SKAL VÆRE 3006 EL 3112 I ÅR FRA 91 TIL 99'
61      STOP EXECUTE
62    END-IF
63  END-PROC
64
```

fortsetter...

```
65  REPORT KOMM-RAPP PAGESIZE 40
66    SEQUENCE I-KOMM
67    CONTROL  I-FYLKE (NEWPAGE NOPRINT)    . * NY SIDE VED BRUDD I FYLKE.
68    TITLE 1 'BEFOLKNINGEN PÅ KOMMUNENIVÅ.'
69    TITLE 2 'PR.DATO ER' W-PRDATO
70    LINE 1  I-KOMM I-BEFOLKN I-MENN I-KVIN
71
72
73  REPORT FYLKE-RAPP SUMMARY SPACE 2
74    SEQUENCE I-FYLKE
75    CONTROL  I-FYLKE
76    TITLE 1 'BEFOLKNINGEN PÅ FYLKESNIVÅ MED PROSENT MENN/KVINNER.'
77    TITLE 2 'PR.DATO ER' W-PRDATO
78    HEADING TALLY ('ANT.' 'KOMMUNER')
79    LINE 1  I-FYLKE TALLY I-BEFOLKN I-MENN I-KVIN W-MENNPROS W-KVINPROS
80
81  BEFORE-BREAK. PROC
82    IF I-BEFOLKN > 0
83      W-MENNPROS ROUNDED = 100 * I-MENN / I-BEFOLKN
84      W-KVINPROS ROUNDED = 100 * I-KVIN / I-BEFOLKN
85    END-IF
86  END-PROC
```

Eksempel på data inn og ut i program GEPRAPP.

Data inn:

File PARAM (parameteret):

```
311291  pr.dato
```

File INN:

```
         1    1    2    2    3
1...5....0....5....0....5....0..
311291 0425 008500 004444 004056
311291 0216 013204 006521 006683
300691 0401 016350 007999 008351
311291 0401 016351 008000 008351
300691 0402 017469 009000 008469
311291 0402 017475 009004 008471
150392 0402 017480 009005 008475
300692 0402 017488 009007 008481
311291 0219 090500 045032 045468
```

Data ut:

Rapporten KOM-RAPP:

29/04/93 BEFOLKNINGEN PÅ KOMMUNENIVÅ. PAGE 1
PR.DATO ER 31.12.91

KOMMUNE	BEFOLKNING	MENN	KVINNER
0216	13.204	6.521	6.683
0219	90.500	45.032	45.468

29/04/93 BEFOLKNINGEN PÅ KOMMUNENIVÅ. PAGE 2
PR.DATO ER 31.12.91

KOMMUNE	BEFOLKNING	MENN	KVINNER
0401	16.351	8.000	8.351
0402	17.475	9.004	8.471
0425	8.500	4.444	4.056
	146.030	73.001	73.029

Rapporten FYLKE-RAPP:

29/04/93 BEFOLKNINGEN PÅ FYLKESNIVÅ MED PROSENT MENN/KVINNER. PAGE 1
PR.DATO ER 31.12.91

FYLKE	ANT. KOMMUNER	BEFOLKNING	MENN	KVINNER	PROSENT MENN	PROSENT KVINNER
02	2	103.704	51.553	52.151	49,71	50,29
04	3	42.326	21.448	20.878	50,67	49,33
	5	146.030	73.001	73.029	49,99	50,01

GEPRAPPF produserer en rapport på en permanent fil. Vi benytter oss av automatiske opptellinger ut på den virtuelle filen UTVIRT (SUMMARY SUMFILE UTVIRT). For at tallene skal starte i pos. 1 og uten mellomrom, må vi også ha med NOADJUST og SPACE 0. Rapporten sorteres på antall personer x hustype x kommune (SEQUENCE) med brudd på de samme tre feltene (CONTROL). Opptelling på antall rom (W-KOL1....W-KOL9) skjer ved at verdien til den aktuelle telleren settes = 1, etter at alle tellere er nullstilt.

Ved SUMMARY SUMFILE får vi ut en del ting vi ikke alltid ønsker. Feltene blir definert med 10 pos. pakket og vi får et ialt-felt, TALLY (her kalt VTALLY). Redigeringen ordner vi i siste JOB-trinn. Her har vi brukt MOVE LIKE, som flytter felter med likt navn fra UTVIRT til UTFIL og konverterer fra pakket (P) til numerisk (N) format. Vi ønsker å få med ialt-feltet, og definerer derfor VTALLY på UTFIL.

```
 1 ************************* EASYTRIEVE PLUS ******************************
 2 *  PROD.NR.: 000
 3 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
 4 ************************************************************************
 5 *  PROGRAM : GEPRAPPF
 6 *  LAGET   : 17.03.93 AV VIDAR HALVORSEN
 7 ************************************************************************
 8 *  BESKRIV.: PRODUSERER RAPPORT PÅ DISK-FIL VED HJELP AV REPORT.
 9 *            OPPTELLINGS-FELTENE BLIR AUTOMATISK DEFINERT PÅ 10 POS.
10 *            PAKKET.
11 *            VI LEGGER DERFOR RESULTATET PÅ EN VIRTUELL FIL FØRST.
12 *            DENNE FILEN LESES I ET NYTT JOBSTEP HVOR VI PAKKER UT
13 *            FELTENE OG SKRIVER PÅ EN PERMANENT DISK-FIL
14 *  REF.    :
15 ************************************************************************
16
17  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
18
19  FILE INNFIL
20    I-ANTROM    1  1 A
21    I-HTYPE     3  1 A
22    I-KOMM      5  4 A
23    I-ANT-P    10  2 A
24
25  FILE UTVIRT VIRTUAL RETAIN F(117)
26    VANT-P      1  2 A
27    VHTYPE      *  1 A
28    VKOMM       *  4 A
29    VTALLY      * 10 P . * IALT
30    VKOL1       * 10 P
31    VKOL2       * 10 P
32    VKOL3       * 10 P
33    VKOL4       * 10 P
34    VKOL5       * 10 P
35    VKOL6       * 10 P
36    VKOL7       * 10 P
37    VKOL8       * 10 P
38    VKOL9       * 10 P
39
```

fortsetter...

```
 40  FILE UTFIL
 41    VANT-P      1  2 A
 42    VHTYPE      *  1 A
 43    VKOMM       *  4 A
 44    VTALLY      *  3 N 0   . * IALT
 45    VKOL1       *  3 N 0
 46    VKOL2       *  3 N 0
 47    VKOL3       *  3 N 0
 48    VKOL4       *  3 N 0
 49    VKOL5       *  3 N 0
 50    VKOL6       *  3 N 0
 51    VKOL7       *  3 N 0
 52    VKOL8       *  3 N 0
 53    VKOL9       *  3 N 0
 54
 55  W-KOL1        W  4 N 0
 56  W-KOL2        W  4 N 0
 57  W-KOL3        W  4 N 0
 58  W-KOL4        W  4 N 0
 59  W-KOL5        W  4 N 0
 60  W-KOL6        W  4 N 0
 61  W-KOL7        W  4 N 0
 62  W-KOL8        W  4 N 0
 63  W-KOL9        W  4 N 0
 64
 65
 66  JOB INPUT INNFIL
 67
 68    W-KOL1 = 0
 69    W-KOL2 = 0
 70    W-KOL3 = 0
 71    W-KOL4 = 0
 72    W-KOL5 = 0
 73    W-KOL6 = 0
 74    W-KOL7 = 0
 75    W-KOL8 = 0
 76    W-KOL9 = 0
 77
 78    IF I-ANTROM = '1'. W-KOL1 = 1. ELSE
 79      IF I-ANTROM = '2'. W-KOL2 = 1. ELSE
 80        IF I-ANTROM = '3'. W-KOL3 = 1. ELSE
 81          IF I-ANTROM = '4'. W-KOL4 = 1. ELSE
 82            IF I-ANTROM = '5'. W-KOL5 = 1. ELSE
 83              IF I-ANTROM = '6'. W-KOL6 = 1. ELSE
 84                IF I-ANTROM = '7'. W-KOL7 = 1. ELSE
 85                  IF I-ANTROM = '8'. W-KOL8 = 1. ELSE
 86                                     W-KOL9 = 1
 87                  END-IF
 88                END-IF
 89              END-IF
 90            END-IF
 91          END-IF
 92        END-IF
 93      END-IF
 94    END-IF
 95
 96    PRINT RAPPORT1
 97
 98  REPORT RAPPORT1 SUMMARY SUMFILE UTVIRT NOADJUST SPACE 0
 99    SEQUENCE I-ANT-P I-HTYPE I-KOMM
100    CONTROL  I-ANT-P I-HTYPE I-KOMM
101    LINE 01  I-ANT-P I-HTYPE I-KOMM
102    LINE 02  W-KOL1 W-KOL2 W-KOL3 W-KOL4 W-KOL5 W-KOL6
103    LINE 03  W-KOL7 W-KOL8 W-KOL9
104
105
106  JOB INPUT UTVIRT
107    MOVE LIKE UTVIRT TO UTFIL
108    PUT UTFIL
```

GEPRAPPS produserer en ferdig rapport, med hode og styrekarakter for side- og linje-skift, direkte på disk-fil (linje 4). Programmet benytter tilsvarende metode som forrige eksempel GEPRAPPF. Her må vi imidlertid aggregere totalen selv. Dette gjøres ved at vi ikke nullstiller telleren W-IALT, men lar den ha en fast verdi = 1 (VALUE 1, linje 29) for hver record. Summen blir antall records.

Pga. at vi vil ha en ferdig rapport som senere kan skrives ut på en skriver, gir vi kolonnene en passende tekst. Dette ordnes enkelt med HEADING og ønsket tekst bak. Hvis vi ikke vil ha med ledende nuller, bruker vi MASK. Det er nok å definere selve masken for det første feltet hvis feltene er like lange. Vi skriver f.eks. MASK (M 'ZZZ9)'). M refererer da til ZZZ9, som betyr at de tre første nullene ikke skal skrives ut. På de resterende feltene skriver vi kun MASK M.

NOADJUST betyr at tabellen ikke skal midtstilles på sida, NODATE betyr at vi ikke vil ha med dato og PAGESIZE 42 vil si at vi får ut 42 linjer pr. side.

Blanke linjer og sideskift styres av printeren vha. et tegn som programmet legger ut i første posisjon på filen. Vi må derfor huske på å definere recordlengden for skriveren +1 (132 + 1) dvs. 133.

```
1 ----
2 //UTFIL    DD DSN=PX215.---------------,
3 //             UNIT=SSB,
4 //             DCB=(LRECL=133,BLKSIZE=23408,RECFM=FBA),
5 //             SPACE=(23408,(5,5),RLSE),DISP=(,CATLG,DELETE)
6 //SYSIN     DD  *
7 ************************* EASYTRIEVE PLUS ******************************
8 *  PROD.NR.: 000
9 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
10 ************************************************************************
11 *  PROGRAM : GEPRAPPS
12 *  LAGET   : 17.03.93 AV VIDAR HALVORSEN
13 ************************************************************************
14 *  BESKRIV.:
15 *            PRODUSERER EN REPORT-TABELL DIREKTE PÅ DISK-FIL
16 *            DENNE INNEHOLDER DA OGSÅ STYREKARAKTERER TIL PRINTER
17 *            HUSK PÅ AT UTFIL DEFINERS MED RECORD-LENGDE = 133
18 *  REF.    :
19 ************************************************************************
20
21  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM)
22  FILE INNFIL
23    I-POSTNR    1   4 A HEADING ('POSTNR.')
24    I-KJØNN     5   1 A HEADING ('KJØNN')
25    I-ALDER     6   3 A
26
27  FILE UTFIL PRINTER
28
29  W-IALT        W   4 N 0 VALUE 1 +
30                            MASK (M 'ZZZ9') +
31                                  HEADING ('   IALT')
32  W0-14         W   4 N 0 MASK  M HEADING ('   0-14')
33  W15           W   4 N 0 MASK  M HEADING ('     15')
34  W16           W   4 N 0 MASK  M HEADING ('     16')
35  W17           W   4 N 0 MASK  M HEADING ('     17')
36  W18           W   4 N 0 MASK  M HEADING ('     18')
37  W19           W   4 N 0 MASK  M HEADING ('     19')
38  W20-39        W   4 N 0 MASK  M HEADING ('  20-39')
39  W40-MM        W   4 N 0 MASK  M HEADING ('    40+')
40
41  JOB INPUT INNFIL
42    W0-14  = 0. W15 = 0. W16 = 0. W17 = 0. W18 = 0. W19 = 0
43    W20-39 = 0. W40-MM = 0
44
45    IF I-ALDER < '015'. W0-14 = 1. ELSE
46      IF I-ALDER = '015'. W15 = 1. ELSE
47        IF I-ALDER = '016'. W16 = 1. ELSE
48          IF I-ALDER = '017'. W17 = 1. ELSE
49            IF I-ALDER = '018'. W18 = 1. ELSE
50              IF I-ALDER = '019'. W19 = 1. ELSE
51                IF I-ALDER = '020' THRU '039'. W20-39 = 1. ELSE
52                  IF I-ALDER > '039'. W40-MM = 1
53                  END-IF
54                END-IF
55              END-IF
56            END-IF
57          END-IF
58        END-IF
59      END-IF
60    END-IF
61   PRINT RAPPORT1
62
63   REPORT RAPPORT1 SUMMARY PRINTER UTFIL  NOADJUST NODATE PAGESIZE 42
64     SEQUENCE I-POSTNR I-KJØNN
65     CONTROL I-POSTNR NOPRINT I-KJØNN
66     TITLE 1 'TABELL1 - GJELDER POST-NR FOR AUST- OG VESTAGDER +
67              FORDELT ETTER KJØNN OG ALDERS-GRUPPER'
68     LINE 1 I-POSTNR I-KJØNN W-IALT +
69            W0-14 W15 W16 W17 W18 W19 W20-39 W40-MM
```

GEPSLIPP skriver ut adresseslipper (etiketter).

Når vi skriver på spesialformular, må vi ta bort så mye unødvendig utskrift som mulig. Vi må også avtale med operatør hvilken kø vi skal legge rapporten i.
I JOB-setningen i JCL'n (linje 1) har vi satt MSGLEVEL=(0,0) og fjerner dermed så mye som mulig av JCL-utskriften. For å ignorere sideskift har vi med /*JOBPARM LINECT=0 (linje 3).
Når vi har kompilert programmet og fjernet alle syntaks-feil, skriver vi inn LIST OFF som første linje i programmet (linje 12). Det fjerner selve utskriften av programmet. Vi må også huske på PARM-parameteret:
PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM) SORT (MSG NO)
Alt dette gjør at vi får ut minimalt med utskrift.

For at operatøren skal få justert inn skriveren, bør vi la programmet skrive ut noen prøveslipper først. Dette gjøres lettvint med en oppstart-prosedyre. Hvis vi har sortering (SEQUENCE) i rapporten er det viktig at prøveslippene har så lav verdi i sorteringsbegrepet at vi er sikre på de blir skrevet ut før de andre.

I rapportdelen (REPORT) angir vi hvilket slippformular vi skal bruke. I GEPSLIPP bruker vi formular med kun en slipp i bredden, og denne har 6 linjer fra 1. linje i den ene slippen til 1. linje i neste. Den har en bredde på 40 posisjoner.
SPACE 1 betyr at vi vil ha kun en blank pos. mellom feltene vi skriver ut (postnr og postnavn), standard er 3 pos..
Her skrives det på linje 1, 2 og 4. Linje 3, 5 og 6 blir blanke.

```
1  //K415JSJS JOB 4351,' *** JOHN ERIK *** ',MSGLEVEL=(0,0),
2  //        MSGCLASS=X,CLASS=A,NOTIFY=K415JSJ,REGION=4096K
3  /*JOBPARM LINECT=0
4  //*        >>>  S L I P P T Y P E   1  <<<
5  //STEP1    EXEC PGM=EZTPA00,REGION=4096K
6  //STEPLIB  DD DSN=SSB2.EASYPL60.LOAD,DISP=SHR
7  //EZTVFM   DD UNIT=WORK,SPACE=(TRK,(5,5))
8  //INN      DD DSN=PX213.S4351.----------,DISP=SHR
9  //SYSOUT   DD SYSOUT=*
10 //SYSPRINT DD SYSOUT=*
11 //SYSIN    DD  *
12  LIST OFF
13 ************************* EASYTRIEVE PLUS ******************************
14 *  PROD.NR.: 000
15 *  PROSJEKT: EASYTRIEVE-EKSEMPLER
16 ************************************************************************
17 *  PROGRAM : GEPSLIPP
18 *  LAGET   : 16.03.93 AV JOHN ERIK SJØRBOTTEN
19 ************************************************************************
20 *  BESKRIV.: UTSKRIVING AV ADRESSE-SLIPPER (SLIPPTYPE 1)
21 *            MED 10 TESTSLIPPER FØRST
22 *  REF.    :
23 ************************************************************************
24
25  PARM DEBUG (NOCLIST NODMAP NOXREF STATE) LIST (NOPARM) SORT (MSG NO)
26  FILE INN
27    I-NAVN          1   35 A
28    I-ADRESSE      36   20 A
29    I-POSTNR       60    4 A
30    I-POSTNAVN     64   14 A
31    I-RECTYPE      80    1 A
32
33  W-TELLER         W     3 N 0
34  W-NAVN           W    35 A
35  W-ADRESSE        W    20 A
36  W-POSTNR         W     4 A
37  W-POSTNAVN       W    14 A
38
39
40  JOB INPUT INN  START TESTSLIPP
41    IF I-RECTYPE = '1' '2'
42      W-NAVN     = I-NAVN
43      W-ADRESSE  = I-ADRESSE
44      W-POSTNR   = I-POSTNR
45      W-POSTNAVN = I-POSTNAVN
46      PRINT SLIPP1
47    END-IF
48
49  TESTSLIPP. PROC
50 * ------ 10 TESTSLIPPER -------
51    W-NAVN     = 'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA'
52    W-ADRESSE  = 'AAAAAAAAAAAAAAAAAAAA'
53    W-POSTNR   = '0000'
54    W-POSTNAVN = 'AAAAAAAAAAAAAA'
55    DO WHILE W-TELLER < 10
56      PRINT SLIPP1
57      W-TELLER = W-TELLER + 1
58    END-DO
59  END-PROC
60
61 REPORT SLIPP1 LABELS (ACROSS 1 DOWN 6 SIZE 40) SPACE 1
62 SEQUENCE W-POSTNR W-NAVN
63  LINE 1 W-NAVN
64  LINE 2 W-ADRESSE
65  LINE 4 W-POSTNR W-POSTNAVN
66
```

Eksempel.

Data inn til GEPSLIPP:

```
          1    1    2    2    3    3    4    4    5    5    6    6    7    7    8
1...5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0
PETTER SAKSHAUG                   SOLEIESVINGEN 43         2200KONGSVINGER     1
OLE JACOB LIAKLEIV                KLØVERBAKKEN 7C          2200KONGSVINGER     2
HEIDI JACOBSEN DAHL                                        2266ARNEBERG        1
SIGNE LOVISE FURUBERG             HAKKESPETTVEIEN 89       2200KONGSVINGER     1
TORVALD EDWARD SIGFRIDSSTAD       SLALOMSTIEN 16           2600LILLEHAMMER     1
PETTER SAKSHAUG                   STORGATA 43              2200KONGSVINGER     1
FINN STARTVEIDT                   KRØTTERFARET 8           2200LILLEHAMMER     3
```

Etiketter ut fra GEPSLIPP:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA

0000 AAAAAAAAAAAAA


AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA

0000 AAAAAAAAAAAAA

   .
   .
   .
   .

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA

0000 AAAAAAAAAAAAA


AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA

0000 AAAAAAAAAAAAA


OLE JACOB LIAKLEIV
KLØVERBAKKEN 7C

2200 KONGSVINGER


PETTER SAKSHAUG
SOLEIESVINGEN 43

2200 KONGSVINGER


PETTER SAKSHAUG
STORGATA 43

2200 KONGSVINGER


SIGNE LOVISE FURUBERG
HAKKESPETTVEIEN 89

2200 KONGSVINGER


HEIDI JACOBSEN DAHL


2266 ARNEBERG


TORVALD EDWARD SIGFRIDSSTAD
SLALOMSTIEN 16

2600 LILLEHAMMER
```

GEPLOTUS er skrevet i CA-Easytrieve for PC. Det produserer en ferdig regneark-fil fra er en vanlig ascii-fil.

Numeriske felt på ascii-filer må vi definere med F (fixed point ascii), ikke med N. Her har vi valgt å definere fil-navnene inne i programmet (linje 18 og 26). Dette betyr at vi må kompilere programmet for hver ny fil vi skal kjøre.
For å få programmet til å produsere en regneark-fil, skriver vi ganske enkelt ACCESS LOTUS for vedkommende fil.
I CA-Easytrieve kan vi også slippe å bruke mange END-IF etter hverandre, vi kan isteden bruke ELSE-IF og da kun med én END-IF (linje 40 til 49).

```
 1 * PC-Easytrieve
 2
 3 ************************** CA-EASYTRIEVE ******************************
 4 *  PROD.NR.: 000
 5 *  PROSJEKT: Easytrieve-eksempler
 6 ***********************************************************************
 7 *  PROGRAM : GEPLOTUS
 8 *  LAGET   : 19.10.93 av John Erik Sjørbotten
 9 ***********************************************************************
10 *  BESKRIV.: Produserer en fil i regneark-format (kan tas rett inn i
11 *            QUATTRO.  (access lotus))
12 *            Fil-identene skrives inn i selve programmet som må
13 *            kompileres hver gang en endrer disse.
14 *            Numeriske ASCII-fil-felt defineres med 'F' istedenfor 'N'
15 *  REF.:
16 ***********************************************************************
17
18  FILE INN V(56)  SYSTEM (PC PATH 'D:\GRSKOLE\ELEVTALL.DAT')
19
20    KOMMUNE       1  4 A
21    SKOLETYPE     5  2 A
22    KLASSER       7  2 F  . * ('F' = fixed point ASCII)
23    GUTTER        9  3 F
24    JENTER       12  3 F
25
26  FILE UT F(21)    SYSTEM (PC, PATH 'D:\REGN\ELEVTALL.WK1' ACCESS LOTUS)
27    KOMMUNE       *  4 A
28    SKOLETYPE     *  2 A
29    STØRRELSE     *  7 A
30    KLASSER       *  2 F
31    GUTTER        *  3 F
32    JENTER        *  3 F
33
34    WELEVER       W  3 N
35
36
37  JOB INPUT INN
38    MOVE LIKE INN TO UT
39    WELEVER = INN:GUTTER + INN:JENTER
40    IF      WELEVER < 19 .  STØRRELSE = '  1- 19'
41    ELSE-IF WELEVER < 50 .  STØRRELSE = ' 20- 49'
42    ELSE-IF WELEVER < 100.  STØRRELSE = ' 50- 99'
43    ELSE-IF WELEVER < 200.  STØRRELSE = '100-199'
44    ELSE-IF WELEVER < 300.  STØRRELSE = '200-299'
45    ELSE-IF WELEVER < 400.  STØRRELSE = '300-399'
46    ELSE-IF WELEVER < 500.  STØRRELSE = '400-499'
47    ELSE
48                            STØRRELSE = '500 +  '
49    END-IF
50
51    PUT UT
```

GEPGRAF er skrevet i CA-Easytrieve for PC. Programmet produserer forskjellig grafikk eller en vanlig tabell. Vi gjør her bruk av et skjermbilde hvor vi taster inn fil-navn og type utskrift. På denne måten kan vi da kjøre den ferdige, eksekverbare program-modulen (GEPGRAF.EXE) med forskjellige datafiler og med forskjellig typer utskrift uten å måtte kompilere programmet for hver gang.
Et hjelpevindu er også lagt inn for skjermbildet.

Oppbyggingen av program som bruker skjermbilder blir annerledes enn vi er vant til fra før. Nye aktiviteter, *PROGRAM* og *SCREEN* brukes. PROGRAM-aktiviteten består her av en loop som går helt til avslutt-tasten F12 brukes. Kontrollen returnerer hele tiden til skjermbildet hvor vi aktiviserer de forskjellige programdelene ved hjelp av funksjonstaster og kommandoer.

Hvor rapportene skal styres bestemmes i Easytrieves *'option-table'* som hver enkelt kan endre. Det gjøres ved å kjøre systemprogrammet **ezoptbl.exe** i Easytrieve. Brukerprogrammet må kompileres og linkes på nytt for å få inn slike valg, da disse blir lagt inn i den ferdige .exe-modulen.
Rapporten LISTE blir i dette programmet lagt på filen 'D:\EZTLISTE.RAP' som ved slutt hentes opp og vises på skjermen ved help av LIST-programmet. Dette gjøres med LINK-kommandoen: LINK 'LIST' USING 'D:\EZLISTE.RAP'. (Ville vi i tillegg ha muligheten til å editere rapporten, kunne vi bare skrevet 'EDIT' istedet for 'LIST'.)

Eksempel på skjermbilde og grafikk ligger bak programlista.

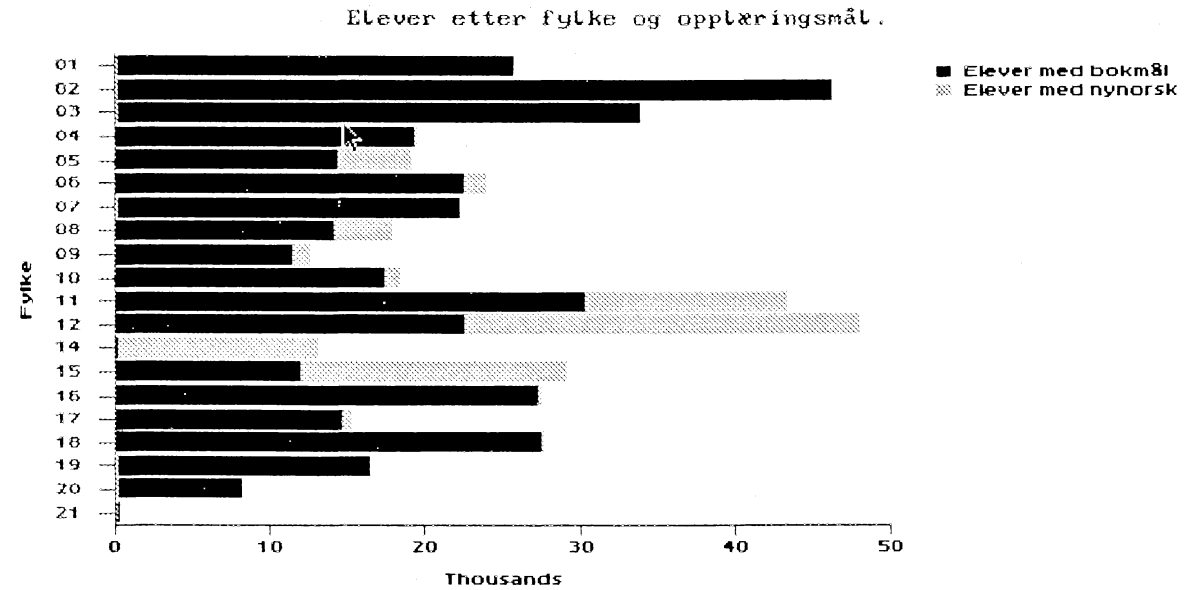
```
1  * PC-Easytrieve
2
3  ************************** CA-EASYTRIEVE ******************************
4  *  PROD.NR.: 000
5  *  PROSJEKT: Easytrieve-eksempler
6  ***********************************************************************
7  *  PROGRAM : GEPGRAF
8  *  LAGET   : 19.10.93 av John Erik Sjørbotten
9  ***********************************************************************
10 *  BESKRIV.: Produserer grafikk eller en tabell over antall elever
11 *            fordelt på fylke og opplæringsmål.
12 *            Her kan den ferdige .exe-modulen kjøres mot forskjellige
13 *            datafiler. Ident og type grafikk eller liste tastes inn
14 *            i et skjermbilde. Hjelpevindu er også lagt inn.
15    *  REF.:
16
17 ***********************************************************************
18
19  FILE INP V(7)  SYSNAME WFILNAVN
20    IFYLKE       1  2 A   HEADING 'Fylke'
21    ISPRÅK       3  1 A
22    IELEVER      4  4 F 0
23
24    WFILNAVN  W   35 A   VALUE 'D:\GRSKOLE\ELEVER.FLK'
25    WGRAFTYPE W    8 A
26    WMODETYPE W    4 A   VALUE 'HØY '
27    WSPRÅK    W    7 A   HEADING 'Språk'
28    WELEVER-N W    5 N 0 HEADING ('Elever' 'med' 'nynorsk')
29    WELEVER-B W    5 N 0 HEADING ('Elever' 'med' 'bokmål')
30
31 *    ----------- Hovedrutine: -------------
32
33   PROGRAM  NAME HOVEDRUTINE
34     DO WHILE KEY-PRESSED NE F12
35       EXECUTE SKJERM-INPUT
36       IF KEY-PRESSED =  F1. EXECUTE HJELP.  END-IF
37       IF KEY-PRESSED = F10. EXECUTE TABELL
38         IF WGRAFTYPE = 'LISTE   '
39           LINK 'LIST' USING 'D:\EZTLISTE.RAP'
40         END-IF
41       END-IF
42       WGRAFTYPE = ' '
43     END-DO
44
45 * --- Skjermbilde som vi kan taste inn ident og grafikktype ---
46
47   SCREEN NAME SKJERM-INPUT  BORDER (DOUBLE ATTR  RED) UPPERCASE
48
49     KEY F1    NAME 'Hjelp '  IMMEDIATE EXIT
50     KEY F10   NAME 'Kjør  '  EXIT
51     KEY F12   NAME 'Avbryt'  IMMEDIATE EXIT
52
53     TITLE  1 SYSDATE '          - Program GEPGRAF -          ' SYSTIME
54     TITLE  3 'Elever fordelt på fylke / opplæringsmål'
55
56       ROW  8 'Input-ident er.....:' WFILNAVN
57
58       ROW 12 'Grafikk-type, liste:' WGRAFTYPE VALUE +
59                 ('PIE' 'VBAR' 'SVBAR' 'HBAR' 'SHBAR' 'LINE' 'LISTE') +
60                ERROR 'Skriv PIE,VBAR,SVBAR,HBAR,SHBAR,LINE el. LISTE'
61
62       ROW 15 'Oppløsning.....:' WMODETYPE +
63                VALUE ('HØY' 'LAV' 'HIGH' 'LOW') +
64                ERROR 'Skriv HØY eller LAV'
65
66     BEFORE-SCREEN. PROC
67        CURSOR AT WGRAFTYPE
68     END-PROC
69                        fortsetter...
```

```
70     AFTER-SCREEN. PROC
71        IF      WMODETYPE = 'LAV '. WMODETYPE = 'LOW '
72        ELSE-IF WMODETYPE = 'HØY '. WMODETYPE = 'HIGH'
73        END-IF
74     END-PROC
75
76 *     ----- Hjelpevindu (F1) ------
77    SCREEN NAME HJELP +
78           BORDER (WIDE  ATTR GREEN) +
79           LINESIZE 26  ROWCOUNT 11 +
80           ROW 14  COL 46
81      KEY F3 NAME 'Tilbake  ' EXIT
82      TITLE 1 'H J E L P .'
83        ROW 3 'Grafikktype:          '
84        ROW 4 '  PIE,VBAR,SVBAR,HBAR '
85        ROW 5 '  SHBAR,LINE EL. LISTE'
86        ROW 7 'Oppløsning:           '
87        ROW 8 '  HØY eller LAV       '
88
89
90   JOB INPUT INP   NAME TABELL
91     IF ISPRÅK = '2' 'N' 'n'     . * Hvis nynorsk
92       WELEVER-N = IELEVER
93       WELEVER-B = 0
94     ELSE                        . * ellers bokmål
95       WELEVER-N = 0
96       WELEVER-B = IELEVER
97     END-IF
98
99     IF WGRAFTYPE = 'LISTE   '
100       PRINT LISTE               . * produserer vanlig rapport
101    ELSE
102       DRAW GRAFIKK              . * produserer grafikk (flere valg)
103    END-IF
104
105    REPORT LISTE SUMMARY LINESIZE 80
106      SEQUENCE IFYLKE
107      CONTROL  IFYLKE
108      TITLE 1 'Elever etter fylke og opplæringsmål.'
109      LINE 1  IFYLKE WELEVER-B WELEVER-N
110
111    GRAPH GRAFIKK  STYLE(WGRAFTYPE)  MODE WMODETYPE  SUMMARY
112      SEQUENCE IFYLKE
113      TITLE 1 'Elever etter fylke og opplæringsmål.'
114      VALUE  IFYLKE WELEVER-B WELEVER-N
```

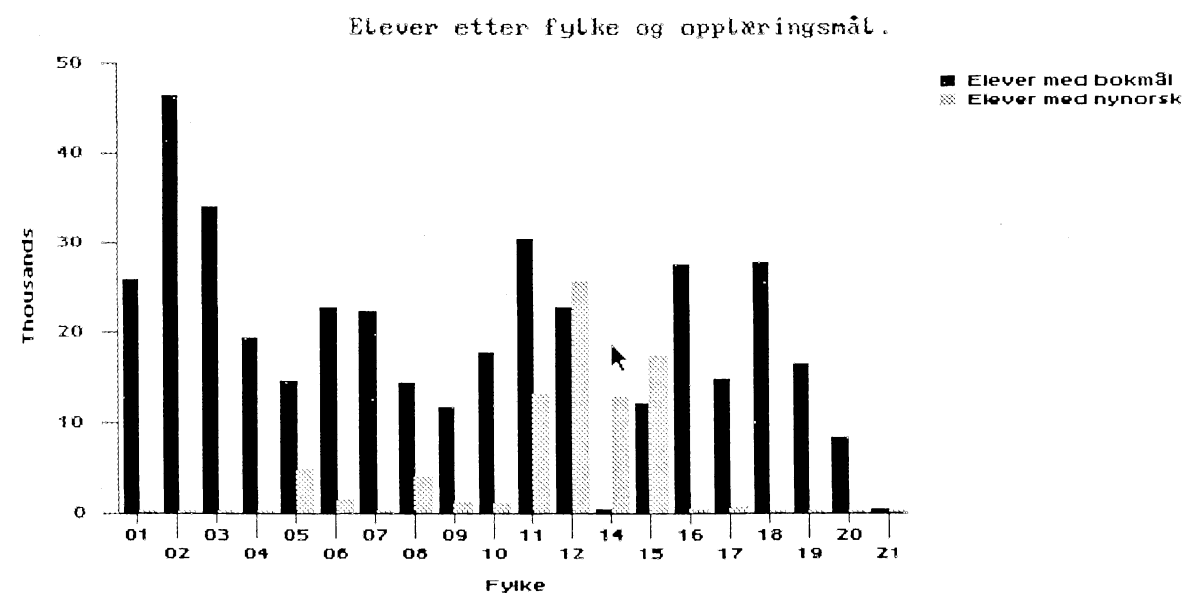
Skjermbilde som programmet gir:

```
+-------------------------------------------------------------------------------+
|          20/10/93           - Program GEPGRAF -          13:49:52            |
|                                                                               |
|                 Elever fordelt på fylke / opplæringsmål                       |
|                                                                               |
|                                                                               |
|                                                                               |
| Input-ident er.....: D:\GRSKOLE\ELEVER.FLK                                    |
|                                                                               |
|                                                                               |
| Grafikk-type, liste: SVBAR                                                    |
|                                                                               |
|                                          _____________________                |
| Oppløsning.........: HØY                 _      H J E L P .    _              |
|                                          _ Grafikktype:        _              |
|                                          _   PIE,VBAR,SVBAR,HBAR,_            |
|                                          _   SHBAR,LINE EL. LISTE_            |
|                                          _ Oppløsning:         _              |
|                                          _   HØY eller LAV     _              |
| F1=Hjelp   F10=Kjør    F12=Avbryt        _ F3=Tilbake          _              |
+-------------------------------------------_____________________-------+
```

Eksempel på grafikk fra GEPGRAF:



F1=Help F2=Keys F3=Exit F4=Shell F6=Print F12=Cancel



F1=Help F2=Keys F3=Exit F4=Shell F6=Print F12=Cancel

Statistisk sentralbyrå

Oslo
Postboks 8131 Dep.
0033 Oslo

Tlf.: 22 86 45 00
Fax: 22 86 49 73

Kongsvinger
Postboks 1260
2201 Kongsvinger

Tlf.: 62 88 50 00
Fax. 62 88 50 30

Statistisk sentralbyrå
Statistics Norway