NN31545.1139 DTA 1139 I

september 1979

BIBLIOTHEEK
STARINGGEBOUW

BESCHRIJVING VAN DE WERKWIJZE EN DE COMPUTERPROGRAMMA's VOOR HET UITVOEREN VAN EEN TOEDELINGSONDERZOEK

R. Kik

BIBLIOTHEEK DE HAAF
Droevendaalsesteeg 3a
Postbus 241
6700 AE Wageningen

Nota's van het Instituut zijn in principe interne communicatiemiddelen, dus geen officiële publikaties.
Hun inhoud varieert sterk en kan zowel betrekking hebben op een eenvoudige weergave van cijferreeksen, als op een concluderende discussie van onderzoeksresultaten. In de meeste gevallen zullen de conclusies echter van voorlopige aard zijn omdat het onderzoek nog niet is afgesloten.
Bepaalde nota's komen niet voor verspreiding buiten het Instituut in aanmerking

1791344

1 3 FEB. 1998

CENTRALE LANDBOUWCATALOGUS
0000 0941 2160

I N H O U D

	blz.
1. INLEIDING	1
2. WERKWIJZE TOEDELINGSONDERZOEK	2
3. BESCHRIJVING COMPUTERPROGRAMMA's	6
3.1. Magneettape CIN op permanent file brengen	8
3.2. Program BEKAGEG	9
3.3. Program BEDTYPE	11
3.4. Ponsbanden op permanent files brengen	13
3.5. EOR in permanent files ponsbanden	14
3.6. Program TRAFOR	15
3.7. Program TRWEG	17
3.8. Program MUTATWE	18
3.9. Program CONWEG	20
3.10.Program TEKWEG	21
3.11.Program TRBEDR	22
3.12.Program TRVAK	24
3.13.Program MUBEDR	25
3.14.Program MUVAK	27
3.15.Program MIDDEL	27
3.16.Program TOEDEL	29
3.17.Program VAKAFS	39
3.18.Program INSTEP	41
3.19.Program STSTONE	43
3.20.Program PRTOED	44
3.21.Program TOEDKEN	45

blz.

4. AFWIJKENDE VORM GEGEVENS 46

4.1. CIN zonder toedelingsgegevens 47

4.1.1. Aanvullende gegevens op permanent file brengen 48

4.1.2. Program CONTROL 49

4.1.3. Program HUISKAV 50

4.1.4. Program TRKAV 50

4.1.5. Program MUTATKA 52

4.1.6. Program INKAF 54

4.1.7. Program KAVAF 54

4.1.8. Program TOEDEL 55

4.2. Geen CIN beschikbaar 58

5. SAMENVATTING 59

6. LITERATUUR 60

BIJLAGEN 61

1. INLEIDING

Voor het uitvoeren van een toedelingsonderzoek in een vroeg stadium van de voorbereidingsfase van een ruilverkaveling bestaat een methode waarmee de gegevens, die nodig zijn voor het samenstellen van een voorlopig toedelingsplan van de gebruikstoestand, kunnen worden berekend (KIK, 1979).

Aangezien voor de meeste ruilverkavelingen een cultuurtechnische inventarisatie wordt uitgevoerd, is ervan uitgegaan dat de voor de inventarisatie verzamelde gegevens van de gebruikstoestand als basisgegevens voor het toepassen van de methode kunnen dienen. In de CIN zijn bovendien per kavel een tweetal extra gegevens opgenomen die voor het toepassen van de methode nodig zijn.

De bewerkingen van de gegevens om tot de invoer voor de optimaliseringsberekening te komen, waarmee de toedeling wordt berekend, zijn te tijdrovend om met de hand uit te voeren. Daarom zijn computerprogramma's ontwikkeld waarmee het gehele proces langs automatische weg kan worden uitgevoerd. Bovendien wordt een digitiser ingeschakeld voor het vastleggen in coördinaten van het wegennet, de plaats van de bedrijven en de plaats van de vakken, zodat de afstanden van de bedrijven naar de vakken via de kortste route langs het wegennet kunnen worden berekend. Ook voor de verwerking van de gedigitaliseerde gegevens zijn computerprogramma's ontwikkeld.

Het is mogelijk dat de methode moet worden toegepast op een gebied waarvoor een CIN beschikbaar is zonder de extra gegevens die voor de toepassing nodig zijn. De werkwijze die in dit geval moet worden gevolgd zal in een apart hoofdstuk worden beschreven.

Een derde mogelijkheid is dat geen CIN bestaat maar dat de gegevens van de gebruikstoestand op een andere wijze beschikbaar zijn.

Ook voor deze mogelijkheid zal de werkwijze apart worden beschreven.

Deze nota is in hoofdzaak bedoeld als een handleiding voor de automatische verwerking van de toedelingsberekening. De benodigde computerprogramma's worden besproken met hun wijze van verwerking in de computer. De programmateksten zijn als bijlagen bijgevoegd.

De computerprogramma's zijn ontwikkeld om te worden gebruikt op de Cyber 175 van het ENR te Petten. Bij gebruik van een andere computer zullen aanpassingen van de programma's noodzakelijk zijn.

Enkele programma's zijn ontwikkeld door het Mathematisch Centrum te Amsterdam en een paar programma's zijn gemaakt door ing. B.P. Akkers van het TWR te Utrecht. Bij de bespreking van de betreffende programma's zal dit worden vermeld.

2. WERKWIJZE TOEDELINGSONDERZOEK

Wanneer voor een ruilverkaveling een toedelingsonderzoek moet plaatsvinden, moeten allereerst enkele voorbereidende werkzaamheden worden verricht. Deze bestaan uit:

- Het aangeven op een gebruikerskaart van de eventuele ruilklassen. Het is wenselijk hiervoor een gebruikerskaart te nemen zodat de ruilklassegrenzen zoveel mogelijk kunnen samenvallen met kavelgrenzen.
- Op dezelfde kaart wordt de vakindeling aangebracht. De grootte van de vakken is sterk afhankelijk van de schaal van het gebied. Hoe grootschaliger het gebied, hoe groter de vakken. In principe omvat een vak de grond die door één weggedeelte wordt ontsloten, maar in de praktijk is dit niet altijd te realiseren. Ook de vakgrenzen moeten zoveel mogelijk samenvallen met kavelgrenzen, terwijl per vak slechts één ruilklasse mag voorkomen.
- Ten slotte moet worden aangegeven welke veldkavels als niet uitruilbaar worden beschouwd. Meestal kunnen deze kavels worden ontleend aan de cultuurkaart bijvoorbeeld kavels waarop kassen, boomgaarden en dergelijke voorkomen.

De bij de voorbereiding verzamelde gegevens gaan naar de Heidemij te Arnhem waar een volgende bewerking plaats vindt, gelijktijdig

met de verwerking van de cultuurtechnische inventarisatie. Hier worden de kavelgegevens van de input van de CIN aangevuld met per kavel het vaknummer waarin de kavel is gelegen en een code voor niet uitruilbare of te splitsen kavels. Splitsing van kavels kan nodig zijn indien een kavel slechts gedeeltelijk uitruilbaar is of door een vakgrens wordt doorsneden.

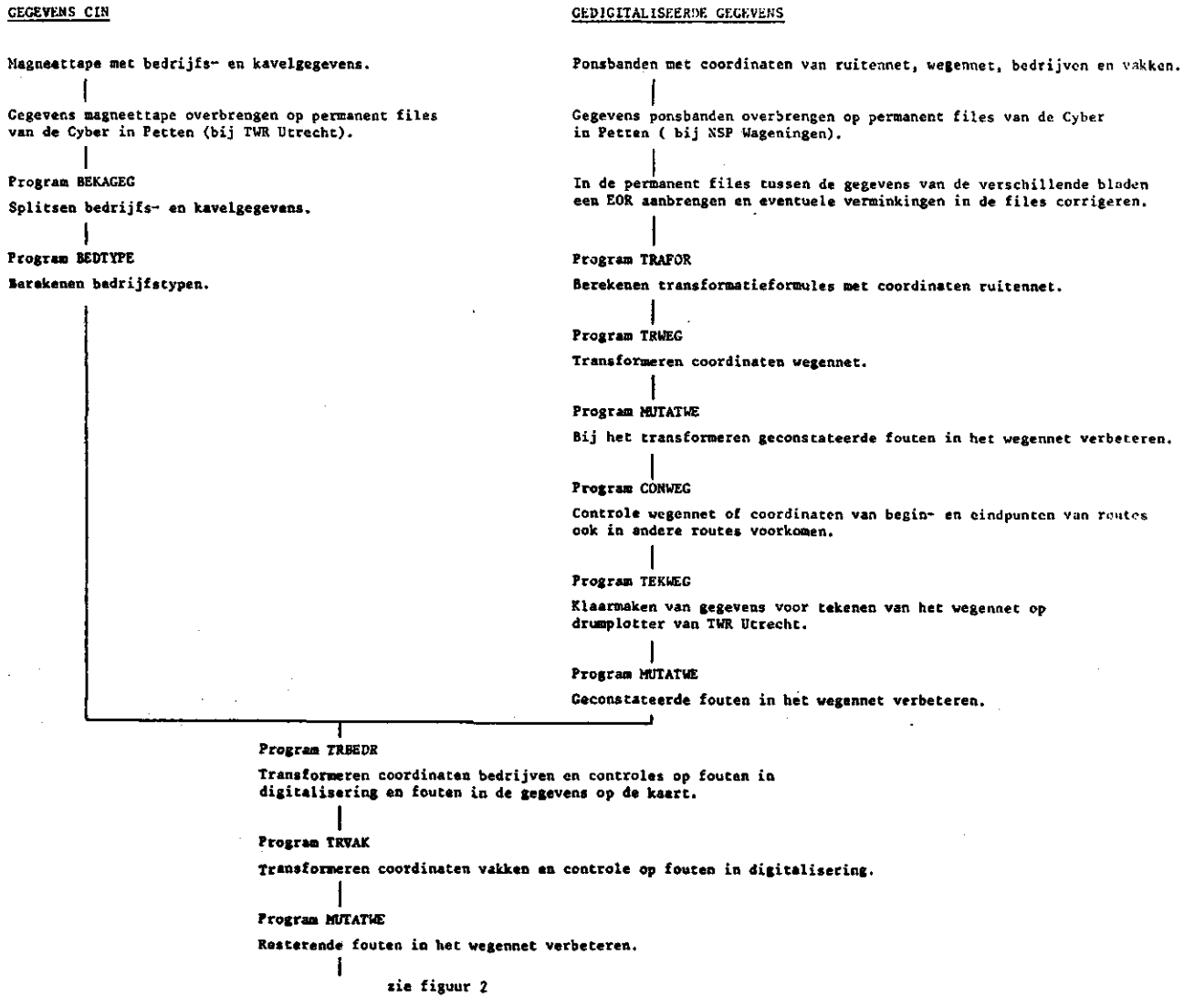
Tevens wordt door de Heidemij een lijst met vaknummers verstrekt waarop de ruilklasse staat aangegeven waarin de vakken zijn gelegen en een lijst met te splitsen kavels. Op deze laatste lijst is per kaveldeel de oppervlakte, het vaknummer en de uitruilbaarheid vermeld. Ook wordt een kaart geleverd waarop het wegennet, de plaats en het nummer van de bedrijven en de plaats en het nummer van de vakken is aangegeven. Van deze kaart wordt door de Heidemij te Roermond het ruitennet, het wegennet, de plaats van de bedrijven en de plaats van de vakken gedigitaliseerd. De instructies voor het gereed maken van de te digitaliseren kaart en voor de digitalisering zijn opgenomen in de bijlagen 27.1 tot en met 27.5.

Als gegevens voor de computerverwerking zijn nu beschikbaar de bedrijfs- en kavelgegevens van de input van de CIN (magneetband) en de gedigitaliseerde gegevens van het ruitennet, het wegennet, de plaats van de bedrijven en de plaats van de vakken (ponsband). Bij grotere gebieden zullen over het algemeen 2 à 3 kaarten voor het digitaliseren nodig zijn, waarvoor dan per kaart een set gedigitaliseerde gegevens beschikbaar is.

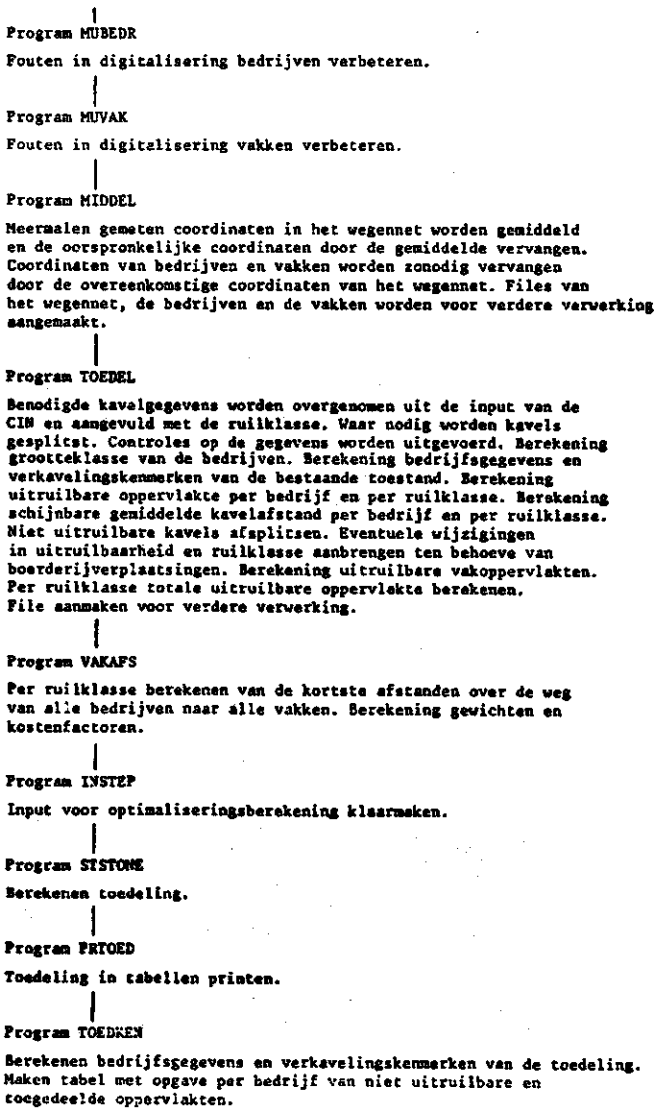
De verwerking met de computer geschiedt met een serie programma's die na elkaar worden gebruikt. Het merendeel van deze programma's is opgenomen in een LIBRARY. Een paar programma's, waarbij het mogelijk moet zijn ingrepen toe te passen, zijn in UPDATE beschikbaar.

In de fig. 1 en 2 is een overzicht gegeven van de benodigde programma's, met daarbij een beknopte omschrijving van de bewerkingen die met de programma's worden verricht. De programma's moeten worden verwerkt in de volgorde zoals ze in het schema zijn opgenomen. Aanvankelijk zijn er twee aparte stromen waarbij in de ene stroom de gegevens van de CIN worden verwerkt en in de andere de gedigitaliseerde gegevens. De beide stromen komen samen bij een programma waarvoor beide soorten gegevens nodig zijn.

Figuur 1. Schema met computerprogramma's voor het uitvoeren van een toedelingsonderzoek



Figuur 2. vervolg van figuur 1



De resultaten van de computerbewerking worden in een aantal tabellen gepresenteerd. Een overzicht van wat in deze tabellen wordt opgenomen is gegeven in fig. 3. In de tabellen 1 en 2 worden bedrijfsgegevens en verkavelingskenmerken gegeven, zowel voor de bestaande toestand als voor de toedeling. Wat de bestaande toestand betreft zijn deze gegevens conform de opgave van de CIN. Voor de toedeling vormen deze gegevens slechts een benadering omdat de juiste gegevens pas beschikbaar komen nadat de toedeling is getekend. Voor het samenstellen van de tabellen wordt elke berekende toedeling als één kavel gerekend. Ook alle niet uitruilbare kavels worden als aparte kavels beschouwd. Alleen toedelingen kleiner dan 0,5 ha worden bij een andere toedeling van hetzelfde bedrijf gevoegd (indien deze voorkomt), terwijl toedelingen in een vak waar van hetzelfde bedrijf een niet uitruilbare kavel voorkomt samen met die niet uitruilbare kavel als één kavel worden gerekend. Voor het bepalen van de kavelafstanden is gebruik gemaakt van de vakafstanden.

Deze benaderingswijze houdt in dat vooral de tabel met bedrijfsgegevens met enige voorzichtigheid moet worden gehanteerd. In de tabel met verkavelingskenmerken wordt gewerkt met gemiddelden waardoor de afwijkingen in deze gegevens gering zullen zijn.

Het uiteindelijk tekenen van het toedelingsplan is momenteel nog handwerk waarvoor gebruik kan worden gemaakt van de gegevens in de tabellen B4, T1.n en T2.

3. BESCHRIJVING COMPUTERPROGRAMMA"S

In de inleiding is reeds aangegeven dat deze nota is bedoeld als een handleiding voor de automatische verwerking van de toedelingsberekening. Er zal daarom worden getracht een zo volledig mogelijke beschrijving te geven van de wijze waarop de computerprogramma's kunnen worden gedraaid en welke output kan worden verwacht. Daarnaast zal een korte beschrijving worden gegeven van de inhoud van de programma's, waarbij extra aandacht zal worden besteed aan die plaatsen waar eventueel ingrepen in de programma's kunnen of moeten worden verricht. De programma's worden behandeld in de volgorde waar-

Figuur 3 Resultaten van de computerverwerking

BESTAANDE TOESTAND

Tabel 1 Bedrijfsgegevens

Opgave per bedrijf van:
aantal kavels
gemiddelde kavelafstand
gemiddelde schijnbare kavelafstand
bedrijfsoppervlakte
oppervlakte huiskavel
percentage van de bedrijfsoppervlakte ingenomen door de huiskavel
percentage van de bedrijfsoppervlakte ingenomen door de grootste veldkavel
bedrijfstype
aanduiding of bedrijfsgebouw buiten het ruilverkavelingsblok ligt

Tabel 2 Verkavelingskenmerken

Per bedrijfstype en per bedrijfsgrootteklasse opgave van:
aantal bedrijven
aantal kavels
oppervlakte cultuurgrond
aantal kavels per bedrijf
gemiddelde bedrijfsoppervlakte
gemiddelde kaveloppervlakte
oppervlakte ingenomen door huiskavels
perc. van de oppervlakte ingenomen door huiskavels
gemiddelde kavelafstand
gemiddelde schijnbare kavelafstand
aantal bedrijven met alle grond bij huis
- perc. van het totale aantal bedrijven
- perc. van de totale oppervlakte
aantal bedrijven met meer dan 80% van de grond bij huis
- perc. van het totale aantal bedrijven
- perc. van de totale oppervlakte
aantal bedrijven met meer dan 60% van de grond bij huis
- perc. van het totale aantal bedrijven
- perc. van de totale oppervlakte

Deze gegevens worden zowel voor de gehele ruilverkaveling verstrekt als voor elk dorpsbehoren afzonderlijk.

Tabel B1 Van alle in de ruilverkaveling voorkomende kavels wordt opgegeven:
bedrijfsnummer
kavelnummer
oppervlakte
kavelafstand, gesplitst in wegsoorten
ruilklasse
vaknummer
het wel of niet uitruilbaar zijn

Tabel B2 Opgave uitruilbare opp. per bedrijf, per ruilklasse en totaal.

Tabel B3 Opgave niet uitruilbare kavels, gesorteerd volgens bedrijfsnummer.

Tabel B4 Opgave per vak van niet uitruilbare oppervlakte.

Tabel B5 Opgave per vak van de uitruilbare oppervlakte.

TOEDELING

Tabel 1 Bedrijfsgegevens

De tabel is identiek aan tabel 1 bestaande toestand behoudens het ontbreken van de gemiddelde kavelafstand, alleen de gemiddelde schijnbare kavelafstand is gegeven.

Tabel 2 Verkavelingskenmerken

De tabel is identiek aan tabel 2 bestaande toestand behoudens het ontbreken van de gemiddelde kavelafstand, alleen de gemiddelde schijnbare kavelafstand is gegeven.

Tabel T1.n Opgave per vak van de toegedeelde oppervlakte in ruilklasse n.

Tabel T2 Opgave per bedrijf van de niet uitruilbare en de toegedeelde oppervlakten met de vaknummers waarin deze oppervlakten voorkomen.

in ze moeten worden verwerkt. Deze volgorde komt overeen met die in de fig. 1 en 2.

Bij de behandeling van de benodigde control cards voor de verwerking van de programma's zal LAXXX als jobnaam worden gehanteerd. De letters LA zijn bindend indien via het TWR te Utrecht op de Cyber in Petten wordt gewerkt, de letters XXX kunnen vrij worden gekozen. Wordt via het NSP te Wageningen in Petten gewerkt, dan moet als jobnaam NSEX1 worden gebruikt. Bij de opgave van de accountkaart zijn de beide check digits vervangen door twee punten, terwijl ook het per ruilverkaveling variërende viercijferige projectnummer, dat na de letters CTD komt, door vier punten is vervangen.

Voor de benamingen van permanent files wordt de standaardnaam opgegeven. Om onderscheid te krijgen tussen de permanent files van verschillende ruilverkavelingen is het wenselijk aan de standaardnaam één of twee letters toe te voegen bijvoorbeeld de eerste letter van de naam van de ruilverkaveling. Voor de indentificatie van de permanent files geldt hetzelfde als voor de jobnaam, de letters LA zijn bindend en de andere drie letters kunnen vrij worden gekozen. Alleen bij de bestaande permanent files waarop de LIBRARY en de UPDATE programma's voorkomen moet de opgegeven ID worden gebruikt.

3.1. Magneettape CIN op permanent file brengen

De inputgegevens van de CIN worden door RAET te Arnhem geleverd op een magneettape met als specificaties:

nine track
unlabelled
recordlengte 80
blocklengte 960
EBCDIC code
formatted

Het overbrengen van de gegevens op de magneettape naar een permanent file van de Cyber te Petten geschiedt bij het TWR te Utrecht. Daarvoor moeten de gegevens van de magneettape eerst op een file op schijf in Utrecht worden gebracht. Dit wordt gedaan door de tape op de tape unit te zetten en op een terminal de volgende commando's te geven.

RUN $ XMT (ret)
HULP.DAT=MM:/RC:80. (ret)
CIN.DAT=HULP.DAT/AS (ret)

De gegevens staan nu op de file CIN.DAT. Deze file moet worden aangevuld met control cards voor het overzenden naar Petten en het op permanent file brengen van de gegevens. De aanvullingen van de file worden op de terminal aangebracht met behulp van de editor EDT, zodat de complete file er als volgt gaat uitzien.

```
LAXXX.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,HULP,*PF.
COPYBF,INPUT,HULP.
REWIND,HULP.
CATALOG,HULP,LATAPECI,ID=LAXXX.
COPYSBF,HULP,OUTPUT.
**EOR**
    gegevens CIN
**EOF**
```

De file CIN.DAT kan nu in MUX worden overgezonden naar Petten door op de terminal de volgende commando's te geven:

I [UIC]CIN.DAT (esc)
R (ret)

Als UIC (user identification code) moet het nummer worden gebruikt waaronder op dat moment op de terminal wordt gewerkt, bijvoorbeeld het algemene nummer van het ICW [50,50].

De output geeft een listing van de op de magneettape voorkomende gegevens, terwijl deze gegevens zijn vastgelegd op de permanent file LATAPECI.

3.2. P r o g r a m BEKAGEG

Voor het overzenden van de magneettape is het noodzakelijk dat de recordlengte niet groter is dan 80 characters. De bij RAET gebruikte recordlengte voor de input van de CIN is echter 146 characters. Op de magneettape worden deze records daarom in tweeën gesplitst waarbij het eerste record 80 characters heeft en het tweede 66 characters. Voor de verdere verwerking is het eenvoudiger de beide records weer tot één samen te voegen. Daarnaast is het wense-

lijk de door elkaar staande bedrijfs- en kavelgegevens te splitsen in een file met bedrijfsgegevens en één met kavelgegevens. Beide bewerkingen, het samenvoegen van de records en het splitsen van de bedrijfs- en kavelgegevens, worden uitgevoerd met het program BEKAGEG. Het programma is opgenomen in de Library TORU; de tekst van het programma is gegeven in bijlage 1. De benodigde control cards voor het uitvoeren van het programma zijn:

```
LAXXX,IO5,T20,CL60000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE2,*PF.
REQUEST,TAPE3,*PF.
ATTACH,TAPE1,LATAPECI,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
BEKAGEG.
CATALOG,TAPE2,LACISLOT,ID=LAXXX.
CATALOG,TAPE3,LABEDRCI,ID=LAXXX.
EOF
```

De output van het programma bestaat alleen uit een dayfile. Aangemaakt zijn de permanet files LABEDRCI en LACISLOT waarop respectievelijk de bedrijfsgegevens en de kavelgegevens voorkomen. Beide permanet files hebben een recordlengte van 146 characters.

Indien men een listing wil hebben van de inhoud van de permanent files kan dit eventueel gedaan worden door het toevoegen van twee COPYSBF kaarten. Door de grote recordlengte kunnen dan echter verminkingen optreden. Om een goede listing te krijgen is het daarom beter een apart programma te maken waarbij een ingelezen lang record als twee korte records wordt geprint. Alleen in bijzondere gevallen kan een dergelijke listing gewenst zijn, normaal bestaat er geen behoefte aan omdat bij het op permanent file brengen van de magneettape reeds een volledige listing van de input van de CIN is verkregen.

3.3. P r o g r a m BEDTYPE

Door het LEI worden voor het samenstellen van de zogenaamde LEI-schets voor ruilverkavelingen diverse bedrijfstypen gehanteerd. Per ruilverkaveling worden de bedrijftypen, en de daarbij behorende criteria, aangepast aan de in het gebied voorkomende bedrijven. In totaal worden per gebied niet meer dan 7 à 8 bedrijfstypen gebruikt.

Om aansluiting bij de LEI-schets te krijgen worden bij het toedelingsonderzoek de door het LEI gebruikte criteria ingevoerd voor het bepalen van de bedrijfstypen. Bij de verdere verwerking kan met maximaal 8 bedrijfstypen worden gewerkt, inclusief de nevenberoepers (bedrijven die als zodanig in de CIN zijn aangegeven). Dit betekent dat indien door het LEI in een gebied meer dan 7 bedrijfstypen worden gebruikt een paar van de onbelangrijkste bedrijfstypen moeten worden samengevoegd voor het toedelingsonderzoek.

Door de variatie per ruilverkaveling in het aantal bedrijfstypen en in de gehanteerde criteria is het nodig dat in het computerprogramma waarmee de bedrijfstypen worden bepaald wijzigingen kunnen worden aangebracht. Vandaar dat program BEDTYPE in UPDATE is vastgelegd. De tekst van het programma is opgenomen in de bijlagen 2.1 en 2.2.

Als voorbeeld zullen de wijzigingen in het programma en de verwerking worden aangegeven indien de volgende bedrijfstypen moeten worden berekend.

1. akkerbouwbedrijven (≥ 60% SBE akkerbouw)
2. veeteeltbedrijven (≥ 60% SBE veeteelt)
3. tuinbouwbedrijven (≥ 60% SBE extensieve en intensieve tuinbouw)
4. veredelingsbedrijven (≥ 60% SBE veredeling)
5. gemengde bedrijven (niet tot één van de andere bedrijfstypen behorend)
6. nevenberoepsbedrijven(hoofdberoep code 4 in CIN)

In het programma wordt van de file met bedrijfsgegevens (tape 1) op regel 6 (UPDATE nummering aan einde regel) achtereenvolgens ingelezen: dorpsbehorennummer, bedrijfsnummer, code hoofdberoep, totaal aantal SBE's en respectievelijk het aantal SBE's akkerbouw, veeteelt, extensieve tuinbouw, intensieve tuinbouw, kalvermesterij,

varkensmesterij en gevogelte. Om de gevraagde bedrijfstypen te berekenen moeten nu de volgende wijzigingen in het programma worden aangebracht.
Regel 10 is alleen voor een bijzonder geval nodig geweest, zodat deze regel kan vervallen.
Regel 12 moet vervangen worden door:

```
PA=FLOAT(SGA)/FLOAT(STOT) $ PV=FLOAT(SGRV)/FLOAT(STOT)
```

waarmee de percentages SBE voor akkerbouw (PA) en veeteelt (PV) worden berekend. De percentages SBE voor tuinbouw (PT) en veredeling (PS) worden reeds berekend op de regels 13 en 14 zodat deze regels kunnen worden gehandhaafd.

Voor het bepalen van het bedrijfstype en het bijbehorende nummer moeten de volgende wijzigingen worden aangebracht:
Regel 15 vervangen door: IF(PA.GE.0.6) GOTO 22 $ IF(PV.GE.0.6) GOTO 23
Regel 16 vervangen door: IF(PT.GE.0.6) GOTO 24 $ IF(PS.GE.0.6) GOTO 25
Regel 17 vervangen door: BD=5 $ GOTO 26
Regel 18 vervangen door: 21 BD=6 $ GOTO 26
Regel 21 vervangen door: 24 BD=3 $ GOTO 26
Na regel 21 toevoegen : 25 BD=4

Op tape 2 wordt nu per bedrijf het dorpsbehorennummer, het bedrijfsnummer en het bedrijfstypenummer geschreven, terwijl ten slotte deze tape op bedrijfsnummer wordt gesorteerd met de subroutine INDCOL. Deze subroutine is overgenomen van het Mathematisch Centrum te Amsterdam.

De voor de verwerking van het programma en het aanbrengen van de wijzigingen benodigde control cards en UPDATE opdrachten zijn:

```
LAXXX.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE2,*PF.
ATTACH,TAPE1,LABEDRCI,ID=LAXXX.
ATTACH,OLDPL,LABEDRTYP,ID=LAKIK.
UPDATE,W,P.
FTN,I=COMPILE.
LGO.
CATALOG,TAPE2,LABEDTYP,ID=LAXXX.
COPYSBF,TAPE2,OUTPUT.
```

```
EOR
*IDENT NEW
*COMPILE BEDRTYP
*DELETE BEDRTYP.10
*DELETE BEDRTYP.12
      PA=FLOAT(SGA)/FLOAT(STOT)  $  PV=FLOAT(SGRV)/FLOAT(STOT)
*DELETE BEDRTYP.15,BEDRTYP.18
      IF(PA.GE.0.6) GOTO 22  $  IF(PV.GE.0.6) GOTO 23
      IF(PT.GE.0.6) GOTO 24  $  IF(PS.GE.0.6) GOTO 25
      BD=5  $  GOTO 26
   21 BD=6  $  GOTO 26
*DELETE BEDRTYP.21
   24 BD=3  $  GOTO 26
   25 BD=4
EOF
```

Met het programma wordt de permanent file LABEDTYP aangemaakt waarop alle in de ruilverkaveling voorkomende bedrijfsnummers zijn vermeld (in numerieke volgorde) met het dorpsbehoren waarin zij zijn gelegen en het bedrijfstypenummer. Als output wordt een listing van deze file gegeven.

3.4. Ponsbanden op permanent files brengen

De gedigitaliseerde gegevens van het ruitennet, het wegennet, de plaats van de bedrijven en de plaats van de vakken worden door de Heidemij te Roermond geleverd op ponsbanden. Per kaartblad wordt voor elk van de vier onderdelen een aparte ponsband gemaakt. Er wordt echter voor elk van de onderdelen één permanent file gemaakt waarop de gegevens van de ponsbanden in volgorde van de kaartbladen worden opgenomen.

Het overbrengen van de gegevens van de ponsbanden naar permanent files van de Cyber te Petten gebeurt bij het NSP te Wageningen, omdat daar een ponsbandlezer beschikbaar is.

De benodigde control cards zijn voor alle vier onderdelen hetzelfde, behalve de naam van de permanent file die wordt aangemaakt.

Er zal daarom alleen een opgave worden gedaan van de control cards voor het op permanent file brengen van de gegevens van het ruitennet. Bij de CATALOG card zullen tussen haakjes de namen worden vermeld van de permanent files als respectievelijk het wegennet, de bedrijven en de vakken worden ingelezen. De eerste en de laatste kaart die worden aangegeven zijn nodig bij elke job die via het NSP wordt verwerkt. Deze kaarten zijn daar bij de balie voorradig. De control cards zijn:

```
CREATE RI:¢20,20!INPUT.DAT/NV/BL
NSEX1.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,HULP,*PF.
COPYBF,INPUT,HULP.
REWIND,HULP.
CATALOG,HULP,LACORUIT,ID=LAXXX.      (LACOWEG,LACOBEDR,LACOVAK)
COPYSBF,HULP,OUTPUT.
EOR
EOR     PRO:PAPER.DAT/VI:XXX         (voor 1e ponsband, bij meerdere
EOR                                   ponsbanden
EOF                                   voor elke volgende band zo'n
EOF PDP                               kaart toevoegen)
```

De output bestaat uit de op de ponsbanden voorkomende gegevens. Deze gegevens zijn daarnaast vastgelegd op de permanent file LACORUIT (LACOWEG,LACOBEDR,LACOVAK).

3.5. E O R i n p e r m a n e n t f i l e s p o n s b a n d e n

De permanent files waarop de gedigitaliseerde gegevens zijn overgebracht zijn alleen geschikt voor verdere verwerking indien voor de digitalisering één kaartblad is gebruikt. Heeft de digitalisering op meerdere kaartbladen plaatsgevonden dan moet op de files tussen de per kaartblad gedigitaliseerde gegevens een EOR worden aangebracht. Een vast programma kan hier niet voor worden gegeven, omdat er per file en per gebied te veel variaties in voorkomen. Als voorbeeld hoe

een dergelijk programma er uit kan zien is in bijlage 3 het program HULP gegeven. In dit programma is in een file van 950 records en een recordlengte van 22 characters na record 503 een EOR aangebracht. Aangezien per file zowel het aantal records als de recordlengte en het aantal malen dat een EOR moet worden aangebracht kan variëren, is het eenvoudiger telkens een nieuw programma te maken dan wijzigingen aan te brengen in een standaardprogramma. Bovendien gebeurt het wel eens dat bij het inlezen van de ponsband verminkingen optreden. Deze verminkingen kunnen gemakkelijk in hetzelfde programma worden verbeterd door een paar extra kaarten met de wijzigingen toe te voegen.

De voor dit programma benodigde control cards zijn:

```
LAXXX.
ACCOUNT,016300..,90,900,CYD.....
REQUEST,TAPE2,*PF.
ATTACH,TAPE1,LACOBUIT,ID=XXXX.          (LACOWEG,LACOBEDR,LACOVAK)
FTN.
LGO.
CATALOG,TAPE2,LACOBUIT,ID=XXXX.         (LACOWEG,LACOBEDR,LACOVAK)
COPYSBF,TAPE2,OUTPUT
EOR
    Programmakaarten
EOF
```

Met het programma is een nieuwepermanent file aangemaakt waarin de EOR's zijn opgenomen en de eventuele verbeteringen zijn aangebracht. De output geeft slechts het gedeelte van de file te zien tot het eerste EOR. Hiermee heeft men een controle dat het EOR inderdaad is aangebracht, maar wil men om de een of andere reden een listing van de gehele file hebben dan zal in het programma een aparte printopdracht moeten worden opgenomen.

3.6. P r o g r a m TRAFOR

De coördinaten die met het digitaliseren zijn verkregen zijn in het stelsel van de gebruikte digitiser. In deze coördinaten komen bovendien onnauwkeurigheden voor door rek of krimp van de kaart, het

tijdens het digitaliseren niet zuiver evenwijdig zijn van de X-as van de kaart met de X-as van de digitiser en een eventuele onzuiverheid in de hoek tussen de assen van de digitiser.

Om aansluiting tussen de coördinaten van de verschillende kaartbladen mogelijk te maken zullen de door digitalisering verkregen coördinaten naar één stelsel moeten worden getransformeerd en de genoemde onnauwkeurigheden zoveel mogelijk moeten worden weggewerkt. Voor het stelsel waar naar getransformeerd wordt is gekozen voor het stelsel van de Rijksdriehoekmeting. Correcties voor een draaiing van de X-as van de kaart ten opzichte van de X-as van de digitiser en de onzuiverheid van de hoek tussen de assen van de digitiser worden uitgevoerd met behulp van regressieberekeningen. Het programma is geschikt om transformaties uit te voeren van digitaliseringen op de kaartschalen 1:10000, 1:5000 en 1:2000, waarvoor tijdens het digitaliseren respectievelijk de codes 1, 2 of 3 worden ingetoest. Het programma is afgeleid van een door ir. A.C. Visser van het ICW vervaardigd programma.

De voor het verwerken van het programma benodigde control cards zijn:

```
LAXXX,IO10,T10,CL60000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE3,*PF.
ATTACH,TAPE1,LACORUIT,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
TRAFOR.
CATALOG,TAPE3,LATRAFOR,ID=LAXXX.
EOF
```

De output van het programma geeft per kaartblad de transformatieformules. Deze formules zijn bovendien voor verdere verwerking vastgelegd op de permanent file LATRAFOR.

Het kan zijn dat de output de foutmelding te zien geeft dat de digitiser niet juist was ingesteld. Deze foutmelding kan betekenen dat inderdaad tijdens het digitaliseren de digitiser op het linksonder gelegen ruitpunt niet op X = 10000 en Y = 10000 is ingesteld, maar het kan ook zijn dat tussen de gegevens van de verschillende

kaartbladen geen EOR voorkomt.

Ter controle is het nuttig met de transformatieformules van elk kaartblad een paar punten met een rekenmachine te transformeren en de daarmee verkregen coördinaten te controleren op de kaart.

De programmatekst is opgenomen in de bijlagen 4.1 en 4.2.

3.7. P r o g r a m TRWEG

Met het programma worden de gedigitaliseerde coördinaten van het wegennet getransformeerd naar het RD stelsel. Gelijktijdig worden enkele controles op de juistheid van de digitalisering uitgevoerd en fouten die tijdens de digitalisering zijn verbeterd verwijderd.

De controles die worden uitgevoerd bestaan uit het nagaan of de coördinaten op de juiste posities in het record staan, of een punt dubbel gedigitaliseerd is en of per route slechts één wegkwaliteitscode voorkomt. Worden op deze punten fouten geconstateerd dan verschijnt in de output na de betreffende coördinaten een vraagteken.

In de bijlage 5.1 en 5.2 is de tekst van het programma vermeld. Het programma wordt met de volgende control cards verwerkt.

```
LAXXX,I010,T30,CL60000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE4,*PF.
ATTACH,TAPE1,LACOWEG,ID=LAXXX.
ATTACH,TAPE3,LATRAFOR,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
TRWEG.
CATALOG,TAPE4,LAWEGNET,ID=LAXXX.
EOF
```

De permanent file LAWEGNET welke met het programma wordt aangemaakt is een random access file, die nodig is om op eenvoudige wijze in de file voorkomende fouten te kunnen verbeteren.

De output van het programma geeft in de eerste kolom een nummering van de punten per route en vervolgens de getransformeerde coördinaten, een code voor begin- of eindpunt van een route, een wegkwaliteitscode, het recordnummer en de oorspronkelijke coördinaten. Op

de permanent file zijn hiervan alleen de getransformeerde coördinaten en de beide codes opgenomen. Indien in de output op een regel een vraagteken voorkomt betekent dit dat op die regel tijdens de transformatie één van de reeds eerdergenoemde fouten is geconstateerd.

3.8 P r o g r a m MUTATWE

Het is onvermijdelijk dat bij het gereed maken van de te digitaliseren kaart en bij het digitaliseren zelf fouten worden gemaakt. Vandaar dat programma's zijn gemaakt om op eenvoudige wijze verbeteringen te kunnen aanbrengen in de files waarop gedigitaliseerde gegevens voorkomen. Met het program MUTATWE kunnen mutaties in de file LAWEGNET worden aangebracht.

De aan te brengen mutaties, welke kunnen bestaan uit het wijzigen, toevoegen of verwijderen van records, worden met behulp van ponskaarten ingevoerd. Voor de kaarten met mutaties wordt een kaart ingevoerd met op de eerste 5 posities het aantal records dat de file bevat en op de volgende 4 posities nullen. Het aantal records kan worden gevonden bij de recordnummering in de output van het program TRWEG. De vier nullen dienen om een aantal variabelen op nul te stellen.

De ponskaarten met mutaties dienen als volgt te worden ingevuld:

posities	1 tot en met 5	: het recordnummer
positie	6	: hier wordt een code aangegeven ter aanduiding van het wijzigen, toevoegen of verwijderen van een record. Een 'blank' betekent dat het aangegeven record moet worden gewijzigd, een plusteken geeft aan dat na het opgegeven recordnummer een record moet worden toegevoegd en een minteken zorgt voor het verwijderen van het opgegeven record
posities	7 tot en met 12	: de X van het punt
posities	13 tot en met 18	: de Y van het punt
positie	19	: indien nodig de code voor het begin- of eindpunt van een route, anders blank
positie	20	: de code voor de wegkwaliteit of een blank als het een eindpunt van een route betreft

Voor het wijzigen of toevoegen van een record moet de ponskaart volledig worden ingevuld. Bij het verwijderen van een record kan worden volstaan met het recordnummer en een minteken.

Indien een volledige route van het wegennet moet worden toegevoegd is het wenselijk dit te doen voor het beginpunt van de laatste route die op de file voorkomt. Wordt namelijk zo'n route aan het einde van de file toegevoegd dan kan er iets mis gaan met de recordnummering, terwijl toevoeging eerder in de file onnodige computertijd kost.

In bijlage 6 is de programmatekst opgenomen.

Bij de voor de verwerking van het programma benodigde control cards komen kaarten met CCOPY voor. Deze kaarten zijn nodig om de inhoud van de permanent file in takt te laten, die anders door het aanbrengen van de wijzigingen zou worden misvormd.

De control cards zijn:

```
LAXXX,I077,T100,CL60000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,HULP,*PF.
ATTACH,UTILITY.
LIBRARY,UTILITY.
ATTACH,A,LAWEGNET,ID=LAXXX.
CCOPY,,A,TAPE3,II.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU,UTILITY.
MUTATWE.
CCOPY,,TAPE3,HULP,II.
CATALOG,HULP,LAWEGNET,ID=LAXXX.
EOR
   kaart met aantal records en vier nullen
   kaart met mutaties
EOF
```

De output van het programma geeft de nieuwe inhoud van de permanent file met aan het einde een opgave van het totale aantal records dat de file na het aanbrengen van de mutaties bevat. De mogelijkheid bestaat dat de output wordt voorafgegaan dooor meldingen van recordnummers met daarachter de tekst 'Opgave fout'. Dit betekent dat op de ingevoerde ponskaart de gegevens niet op de juiste posities staan en dat daardoor de betreffende wijziging in de file niet is uitgevoerd. Om de wijzigingen alsnog uit te voeren zal het programma met

de verbeterde ponskaarten opnieuw moeten worden gedraaid. Daarbij moet er wel aan worden gedacht dat inmiddels de recordnummers door de wel uitgevoerde mutaties kunnen zijn veranderd.

Het program MUTATWE is op dit moment behandeld, hoewel nog niet alle fouten in het wegennet bekend zijn. Het is echter van belang de bij het transformeren van het wegennet geconstateerde fouten reeds nu te verbeteren om foutmeldingen bij het volgende programma en vreemde lijnen bij het tekenen van het wegennet te voorkomen. Het gevolg is dat het programma nog een paar maal moet worden gebruikt, zoals ook in het schema in fig. 1 is aangegeven.

3.9. P r o g r a m CONWEG

Met dit programma wordt gecontroleerd of de coördinaten van begin- en eindpunten van routes ook op andere plaatsen in het wegennet voorkomen. Er wordt dus nagegaan of alle routes aansluiting op elkaar hebben. Een nadeel van het programma is dat ook foutmeldingen gegeven worden voor insteekwegen en dergelijke waarvan één zijde van de route geen aansluiting op het wegennet heeft. Dit nadeel wordt voor lief genomen omdat de fouten die met het programma worden geconstateerd anders helemaal niet of pas in een veel later stadium ontdekt worden. In het laatste geval moeten dan een aantal 'dure' programma's opnieuw worden gedraaid.

In bijlage 7 is de tekst van het programma opgenomen. De voor de verwerking benodigde control cards zijn:

```
LAXXX,IO20,T30,CL70000.
ACCOUNT,016300..,90,900,CTD.....
ATTACH,TAPE1,LAWEGNET,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
CONWEG.
EOR
```

Kaart met aantal records LAWEGNET (posities 1 t/m 5)

```
EOF
```

De output van het programma geeft een lijst met coördinaten van punten die verder niet in het wegennet voorkomen. Zoals reeds is aangegeven zullen de meeste van deze foutmeldingen ten onrechte voorko-

men omdat ze betrekking hebben op het losse einde van insteekwegen. Daarom zullen de vermelde punten op de kaart moeten worden opgezocht om na te gaan wat echte foutmeldingen zijn.

3.10. P r o g r a m TEKWEG

Een volgende controle op de digitalisering van het wegennet is het met behulp van de getransformeerde coördinaten tekenen van het wegennet. Dit tekenen geschiedt op de Calcomp drumplotter van het TWR te Utrecht op kaartschaal 1:10 000. Het programma dient om een file aan te maken met de benodige plotgegevens, zodanig dat de papierbreedte van 75 cm bij het tekenen niet overschreden kan worden. Het wegennet wordt daardoor van zuid naar noord in stroken van 75 cm verdeeld die na elkaar op de rol papier van de plotter worden getekend. De stroken kunnen eventueel aan elkaar worden geplakt om één kaart van het wegennet te krijgen, maar nodig is dit niet omdat ook met de aparte stroken het wegennet goed te controleren is. De controle gebeurt door visuele vergelijking van de tekening met de kaart waarop het wegennet gedigitaliseerd is. Wegen of weggedeelten die bij het digitaliseren vergeten zijn vallen daarbij snel op.

Het programma is vervaardigd door ing. B.P. Akkers; de tekst van het programma is gegeven in de bijlagen 8.1 tot en met 8.3. De voor de verwerking benodigde control cards zijn:

```
LAXXX,IO150,T150,CL220000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE2,*Q.
ATTACH,TAPE3,LAWEGNET,ID=LAXXX.
ATTACH,UTILITY.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU,UTILITY.
TEKWEG.
REWIND,TAPE2.
ROUTE,TAPE2,DC=PR,FC=BA,DEF.
EOR
```

kaart met aantal records LAWEGNET en coördinaten meest ZW ruitpunt

```
EOF
```

Op de toe te voegen kaart wordt het aantal records van LAWEGNET aangegeven in de posities 1 t/m 5, de X van het ruitpunt in de posities 11 t/m 16 en de Y in de posities 17 t/m 22. Het ruitpunt dat wordt opgegeven moet zodanig worden gekozen dat alle coördinaten van het wegennet ten noorden en ten oosten hiervan zijn gelegen.

De aan de control cards toegevoegde ROUTE card zorgt er voor dat de output in de output-queue blijft staan. De output kan nu worden opgevangen op een magneettape. De manier waarop dit gebeurt staat beschreven in het TWR minirapport nr. 5. Indien het programma vanuit het NSP te Wageningen is gedraaid, kan de output naar Utrecht worden gestuurd door aan de ROUTE card TID=LA toe te voegen.

De magneettape met de output wordt op de stuurcomputer van de plotter gezet, waarna de tekening wordt gemaakt.

De fouten die met het programma CONWEG en met de tekening in de digitalisering van het wegennet zijn ontdekt, moeten worden verbeterd door opnieuw het program MUTATWE toe te passen. De verwerking van dit programma staat beschreven in par. 3.8.

3.11. P r o g r a m TRBEDR

Het programma dient voor het transformeren van de gedigitaliseerde coördinaten van de plaats van de bedrijven naar het RD stelsel. Daarbij worden fouten die tijdens de digitalisering zijn verbeterd verwijderd en een aantal controles op de juistheid en volledigheid van de digitalisering uitgevoerd. Gecontroleerd worden:

- de juistheid van de posities van de coördinaten in het record
- het meermalen voorkomen van hetzelfde bedrijfsnummer bij de gedigitaliseerde gegevens
- het overeenkomen van de gedigitaliseerde bedrijfsnummers met de in de CIN voorkomende bedrijfsnummers
- het voorkomen van alle bedrijfsnummers van de CIN bij de digitalisering
- het voorkomen van de coördinaten van de bedrijven in het wegennet

De tekst van het programma is weergegeven in de bijlagen 9.1 t/m 9.3. Bijlage 9.3 geeft de,van het Mathematisch Centrum overgenomen,

subroutine INDCOL die ook al voorkomt in bijlage 2.2. De tekst van deze subroutine is niet zozeer nogmaals opgenomen omdat er kleine verschillen zijn met die in de voorgaande bijlage, maar omdat de hier opgenomen tekst voorkomt in library TORU. Aangezien deze subroutine een sorteerprogramma bevat is het mogelijk dat men ook in andere programma's hiervan gebruik wil maken. Volledigheidshalve zij daarom vermeld dat met het programma de array IK wordt gesorteerd, terwijl in de array's IK1 (integer) en R (real) het resterende gedeelte van de records, waarop niet gesorteerd hoeft te worden, wordt meegenomen. In het hoofdprogramma moet aan ILR het nummer worden gegeven van het record waarmee men het sorteren wil beginnen, en aan IUR het nummer van het record waarmee het sorteren eindigt. Meestal zal ILR daardoor de waarde 1 krijgen en IUR een waarde gelijk aan het totale aantal records omdat over het algemeen een gehele file wordt gesorteerd.

Voor het verwerken van het programma zijn de volgende control cards benodigd:

```
LAXXX,IO10,T150,CL150000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE5,*PF.
ATTACH,TAPE3,LATRAFOR,ID=LAXXX.
ATTACH,TAPE4,LABEDTYP,ID=LAXXX.
ATTACH,TAPE6,LAWEGNET,ID=LAXXX.
ATTACH,TAPE7,LACOBEDR,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
TRBEDR.
CATALOG,TAPE5,LABEDR,ID=LAXXX.
EOR
```

kaart met aantal records LAWEGNET in posities 1 t/m 5

```
EOF
```

De output van het programma is een lijst met achtereenvolgens per regel het recordnummer, het bedrijfsnummer en de getransformeerde coördinaten. Indien tijdens het transformeren een fout is geconstateerd in de posities in het record van de coördinaten, dan is achter het betreffende record een vraagteken geprint. Aan het einde van de

output verschijnen de foutmeldingen ten gevolge van de andere controles die zijn uitgevoerd. De foutmeldingen spreken voor zichzelf, alleen moet nog worden opgemerkt dat vaak één fout twee foutmeldingen tot gevolg heeft. Is bijvoorbeeld bij het digitaliseren een verkeerd bedrijfsnummer ingetoetst dan ontstaat de foutmelding dat dit bedrijfsnummer niet voorkomt bij de bedrijfsnummers. De tweede foutmelding die deze fout tot gevolg heeft is dat het juiste bedrijfsnummer niet gedigitaliseerd is. Op deze wijze zijn meer combinaties van foutmeldingen mogelijk.

3.12. P r o g r a m TRVAK

Ook de gedigitaliseerde coördinaten van de plaats van de vakken moeten naar het RD stelsel worden getransformeerd. Het programma is vrijwel hetzelfde als dat waarmee de coördinaten van de bedrijven worden getransformeerd, alleen bevat het minder controles. Gecontroleerd worden de juistheid van de posities in de records van de coördinaten en het voorkomen van de bij de vakken behorende coördinaten in het wegennet.

De bijlagen 10.1 en 10.2 geven de programmatekst weer. De control cards voor de verwerking van het programma zijn:

```
LAXXX,IO10,T100,CL110000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE4,*PF.
ATTACH,TAPE3,LATRAFOR,ID=LAXXX.
ATTACH,TAPE5,LAWEGNET,ID=LAXXX.
ATTACH,TAPE6,LACOVAK,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
TRVAK.
CATALOG,TAPE4,LAVAKKEN,ID=LAXXX.
EOR
```

kaart met aantal records LAWEGNET in de posities 1 t/m 5

```
EOF
```

De output is een lijst met per regel recordnummer, vaknummer en de getransformeerde coördinaten. Coördinaten die niet in de juiste posities voorkwamen in de records worden met een vraagteken achter

het betreffende record aangegeven. Aan het einde van de output kunnen foutmeldingen voorkomen van coördinaten die niet voorkomen in het wegennet.

In de programma's TRBEDR en TRVAK komen foutmeldingen voor die betrekking hebben op het wegennet. Meestal bestaan deze fouten uit het overslaan van punten bij het digitaliseren van het wegennet, maar het kan ook zijn oorzaak vinden in onnauwkeurig digitaliseren waardoor meer bepalingen van één punt buiten de tolerans vallen om als één punt herkend te worden. Als tolerans wordt aangehouden dat het verschil in de X en in de Y tussen de punten minder dan 9 m moet zijn. In ieder geval moeten de in de beide programma's geconstateerde fouten in het wegennet worden verbeterd, waarvoor opnieuw het program MUTATWE wordt gebruikt (zie 3.8). Als alle in het wegennet gevonden fouten juist zijn verbeterd, is het de laatste maal dat dit programma hoeft te worden gebruikt; het wegennet moet nu foutloos zijn.

3.13. P r o g r a m MUBEDR

Dit programma is bestemd voor het verbeteren van de bij de digitalisering van de bedrijven gemaakte fouten. Deze fouten zijn geconstateerd bij het transformeren van de coördinaten van de bedrijven met het program TRBEDR en kunnen dus in de output van dat programma worden gevonden.

Evenals bij het program MUTATWE kunnen de mutaties, aan te brengen in de file LABEDR, bestaan uit het wijzigen, toevoegen of verwijderen van records. De mutaties worden aangebracht met behulp van ponskaarten, waarbij op de eerste kaart het aantal records van de file moet worden aangegeven (posities 1 t/m 5). Dit aantal is te vinden bij de recordnummering in de output van het program TRBEDR.

Het invullen van de ponskaarten met mutaties geschiedt als volgt:

posities 1 tot en met 5: het recordnummer
positie 6 : een blank, een plusteken of een minteken voor het respectievelijk wijzigen, toevoegen of verwijderen van het record
posities 7 tot en met 14: het bedrijfsnummer
posities 15 tot en met 20: de bij het bedrijf behorende X
posities 21 tot en met 26: de bij het bedrijf behorende Y

De ponskaarten moeten volledig worden ingevuld indien records moeten worden gewijzigd of toegevoegd, bij te verwijderen records is het voldoende dat het recordnummer en een minteken worden opgegeven. Ter besparing van computertijd is het wenselijk records toe te voegen voor het laatste bedrijf dat op de file voorkomt. Toevoeging aan het einde van de file is niet mogelijk.

De programmatekst is weergegeven in bijlage 11. De voor de verwerking van het programma benodigde control cards zijn:

```
LAXXX,IO70,T100,CL60000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,HULP,*PF.
ATTACH,UTILITY.
LIBRARY,UTILITY.
ATTACH,A,LABEDR,ID=LAXXX.
CCOPY,,A,TAPE3,II.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU,UTILITY.
MUBEDR.
CCOPY,,TAPE3,HULP,II.
CATALOG,HULP,LABEDR,ID=LAXXX.
EOR
   kaart met aantal records LABEDR
   kaarten met mutaties
EOF
```

In de output van het programma wordt een listing van de nieuwe inhoud van de permanent file gegeven, met aan het einde een opgave van het aantal records dat de file na het aanbrengen van de mutaties bevat. Mocht op een ponskaart de tekst van een mutatie niet in de juiste posities zijn geplaatst, dan verschijnt in de output een melding van het betreffende recordnummer (indien dit voor de computer nog leesbaar was) met daarachter de mededeling 'Opgave fout'. Na verbetering van de ponskaart moet zo'n mutatie dan alsnog worden aangebracht.

3.14. P r o g r a m MUVAK

Dit programma is identiek aan het program MUBEDR, alleen worden met dit programma de fouten hersteld die bij het digitaliseren van de vakken zijn gemaakt. Voor de beschrijving van het programma wordt daarom verwezen naar het programma MUBEDR (par. 3.13). Een paar opmerkingen kunnen daarbij nog worden gemaakt.

De te gebruiken control cards zijn exact hetzelfde als bij het programma MUBEDR, alleen moet voor de daar gebruikte permanent file naam LABEDR nu de naam LAVAKKEN worden gebruikt.

Bij het transformeren van de coördinaten van de vakken is geen controle uitgeoefend op de juistheid van de vaknummers. De vaknummers worden echter in numerieke volgorde gedigitaliseerd zodat het eenvoudig is om even na te gaan of alle vaknummers goed zijn overgekomen.

Hoewel het vaknummer hoogstens uit drie cijfers kan bestaan zijn op de ponskaart voor de mutaties hiervoor acht posities gereserveerd (7 t/m 14). Het zal duidelijk zijn dat het vaknummer zover mogelijk rechts in deze posities moet worden aangebracht.

Volledigheidshalve is de tekst van het programma opgenomen in bijlage 12.

3.15. P r o g r a m MIDDEL

Alle tijdens de digitalisering gemaakte fouten zijn nu verbeterd en de coördinaten naar het RD stelsel getransformeerd. Veel punten zijn meermalen gedigitaliseerd maar de bij de verschillende bepalingen verkregen coördinaten van een punt zullen niet allemaal exact hetzelfde zijn. Voor het verdere gebruik is het echter nodig dat overal waar eenzelfde punt voorkomt de coördinaten precies gelijk zijn. Dit wordt bewerkstelligd door het program MIDDEL. In dit programma worden de coördinaten van het wegennet eerst gesorteerd op de X-waarden en per gelijke waarde van de X op de Y-waarden. Vervolgens wordt nagegaan of de verschillen tussen de X-waarden en tussen de Y-waarden van twee opeenvolgende punten beide kleiner zijn dan 9 meter. Indien dit het geval is wordt aangenomen dat deze coördinaten betrekking hebben op één punt. De op deze manier gevonden verschillende coördinaten van één punt worden gemiddeld, waarna in de file de oorspronkelijke

coördinaten van het punt worden vervangen door de gemiddelde. Achtereenvolgens worden nu in de files met de plaatsen van de bedrijven en de vakken de coördinaten vervangen door de overeenkomstige coördinaten van het wegennet, waarbij eveneens een tolerans van 9 meter wordt gehanteerd.

Tenslotte worden in het programma voor het wegennet, de plaats van de bedrijven en de plaats van de vakken nieuwe files aangemaakt, omdat de oorspronkelijke random access files niet meer nodig zijn nu alle mutaties zijn aangebracht. Bovendien moeten voor verder gebruik op de file van het wegennet per record vier punten worden opgenomen in plaats van één punt per record zoals dit op de random access file voorkomt.

In de bijlagen 13.1 t/m 13.4 is de programmatekst opgenomen, inclusief de tekst van de subroutine SORRY. De benodigde controlcards waarmee het programma kan worden verwerkt zijn:

```
LAXXX,IO150,T100,CL140000.
ACCOUNT,016300..,90,900,CTD.....
ATTACH,UTILITY.
LIBRARY,UTILITY.
ATTACH,B,LAWEGNET,ID=LAXXX.
ATTACH,L,LABEDR,ID=LAXXX.
ATTACH,M,LAVAKKEN,ID=LAXXX.
CCOPY,,B,TAPE3,II.
CCOPY,,L,TAPE4,II.
CCOPY,,M,TAPE5,II.
REQUEST,HULP,*PF.
REQUEST,TAPE6,*PF.
REQUEST,TAPE7,*PF.
REQUEST,TAPE8,*PF.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU,UTILITY.
MIDDEL.
CCOPY,,TAPE3,HULP,II.
CATALOG,HULP,LAWEGNET,ID=LAXXX.
CATALOG,TAPE6,LAWEGCO,ID=LAXXX.
```

```
CATALOG,TAPE7,LABEDRCO,ID=LAXXX.
CATALOG,TAPE8,LAVAKCO,ID=LAXXX.
COPYSBF,TAPE6,OUTPUT.
COPYSBF,TAPE7,OUTPUT.
COPYSBF,TAPE8,OUTPUT.
EOR
   kaart met aantal records LAWEGNET (posities 1 t/m 5), aantal
   records LABEDR (posities 6 t/m 10) en aantal records LAVAKKEN
   (posities 11 t/m 15)
EOF
```

Er dient op gelet te worden dat de, op de ponskaart aan te geven, aantallen records van de diverse files worden overgenomen uit de output van de laatste versies van de mutatieprogramma's.

De output van het programma bevat achtereenvolgens de inhoud van de permanent files LAWEGCO, LABEDRCO en LAVAKCO. Hoewel het voorkomen van de coördinaten van de bedrijven en de vakken in het wegennet reeds bij de transformaties is gecontroleerd, bestaat de mogelijkheid dat door het middelen van coördinaten van het wegennet het verschil tussen coördinaten van bedrijven of vakken en de overeenkomstige coördinaten van het wegennet buiten de tolerans komt te vallen. Mocht dit zich voordoen dan begint de output met de mededeling van welke bedrijven en/of vakken de coördinaten niet in het wegennet voorkomen. Aangezien dit erg weinig zal voorkomen kan in dat geval het beste een klein hulpprogramma worden geschreven waarmee in de files LABEDRCO en/of LAVAKCO de coördinaten van de betreffende bedrijven en/of vakken door de juiste coördinaten van het wegennet worden vervangen.

Het program MIDDEL is het laatste programma dat nodig is om de op ponsbanden geleverde gedigitaliseerde gegevens geschikt te maken voor het berekenen van de afstanden over de weg van de bedrijven naar de vakken.

3.16. P r o g r a m TOEDEL

Dit vrij grote programma is in hoofdzaak bestemd om de gegevens van de CIN voor verdere bewerking geschikt te maken. Het programma

is in UPDATE gezet omdat ingrepen mogelijk moeten zijn onder andere voor het opvangen van fouten in de CIN gegevens en het uitvoeren van alternatieve berekeningen. Het is daarom nodig de tekst, die gegeven is in de bijlagen 14.1 t/m 14.18, wat uitvoeriger te behandelen. Voor het aanduiden van de regels zal daarbij gebruik worden gemaakt van de achteraan de regels voorkomende UPDATE nummering.

Na het aangeven van de te gebruiken tapenummers en de voor de diverse arrays benodigde dimensions worden de op ponskaarten gezette gegevens ingelezen. Voordat wordt vermeld waar deze gegevens worden ingelezen, zal eerst worden uiteengezet hoe de ponskaarteninvoer er precies uitziet.

Op de eerste ponskaart staan achtereenvolgens aangegeven: het nummer van het eerste dorpsbehoren met buitenblok bedrijven (posities 1 en 2), de naam van de ruilverkaveling (posities 3 t/m 32) en de grootste afstand in m waarover het nog nodig is de afstand van een bedrijf naar een vak te berekenen (posities 33 t/m 37). Deze afstand wordt opgegeven omdat toedelingen op zeer grote afstand toch niet voorkomen, zodat het onnodige computertijd zou kosten vakafstanden van bijvoorbeeld meer dan 15 km te berekenen. Per gebied moet worden nagegaan welke afstand wordt opgegeven.

De volgende ponskaarten bevatten de vaknummers en de ruilklasse waarin de vakken zijn gelegen. Per kaart wordt in de posities 1 t/m 5 een vaknummer vermeld en in de posities 6 t/m 8 de bijbehorende ruilklasse. De kaarten moeten in numerieke volgorde van de vaknummers voorkomen en worden afgesloten met een EOR kaart.

Hierna komen ponskaarten waarop de gegevens van de te splitsen kavels zijn vermeld. Voor elk kavelgedeelte wordt een ponskaart gebruikt, met daarop het bedrijfsnummer (posities 1 t/m 8), het kavelnummer (posities 9 t/m 11), de oppervlakte van het kavelgedeelte in ha (posities 12 t/m 20, met het decimaalteken in positie 16), het vaknummer (posities 21 t/m 23) en de code voor de uitruilbaarheid (positie 24). Ook deze serie kaarten wordt afgesloten met een EOR kaart. Mochten in een gebied geen te splitsen kavels voorkomen, dan moet een blanco kaart worden opgenomen aangezien anders het programma niet goed werkt.

De nu volgende serie ponskaarten bevat de namen van de bedrijfstypen. Voor elke benaming wordt een ponskaart gebruikt, waarvoor de posities 1 t/m 30 beschikbaar zijn. Elke naam moet op positie 1 beginnen. De eerste kaart vermeldt ALLE BEDRIJVEN, de tweede BINNEN BLOK BEDRIJVEN en de derde HOOFDBEROEPS BEDR BINNEN BLOK. Op de volgende kaarten worden de in het gebied voorkomende bedrijfstypen vermeld bijvoorbeeld VEETEELT BEDR BINNEN BLOK. In deze serie moet aan alle benamingen van de bedrijfstypen BINNEN BLOK worden toegevoegd. De serie wordt afgesloten met een EOR kaart.

Opnieuw volgt nu een serie kaarten met benamingen van bedrijfstypen. De wijze van vermelding op de kaart is dezelfde als bij de voorgaande serie, alleen komt nu op de eerste kaart te staan HOOFDBEROEPS BEDR en op de volgende kaarten dezelfde typebenamingen als bij de voorgaande serie maar zonder de toevoeging BINNEN BLOK. Het aantal kaarten van deze serie is dus twee minder dan van de vorige serie. Direct hierna komt nogmaals een serie kaarten met benamingen van bedrijfstypen. Op deze kaarten zijn voor de benaming 10 posities beschikbaar, waarbij de naam zoveel mogelijk rechts moet worden vermeld als minder dan 10 posities nodig zijn. Hier worden alleen de namen (of afkortingen) van de bedrijfstypen gegeven bijvoorbeeld AKKERBOUW, VEETEELT enzovoort. Het aantal kaarten is weer één minder dan bij de vorige serie.

Met de volgende kaarten worden de gegevens verstrekt die nodig zijn voor het wijzigen van uitruilbaarheid of ruilklasse ten behoeve van boerderijverplaatsing. Voor het reserveren van grond op de plaats waar het nieuwe bedrijf moet komen wordt op een kaart opgegeven: het bedrijfsnummer (posities 1 t/m 8), het kavelnummer (posities 9 t/m 11, dit nummer is 1 omdat het een huiskavel betreft), de oppervlakte die de kavel moet krijgen (posities 12 t/m 20, met het decimaal teken op positie 16), de ruilklasse (posities 21 t/m 23), het vaknummer waar het bedrijf komt (posities 24 t/m 27) en een 2 als code voor de uitruilbaarheid op positie 29.

De mogelijkheid bestaat dat de grond van het te verplaatsen bedrijf in meerdere ruilklassen ligt, terwijl door de verplaatsing alle grond in één ruilklasse moet worden toegedeeld. Teneinde per ruilklasse een juiste grondbalans te houden zal bij één of meerdere

andere bedrijven een verschuiving van de grond van de ene naar de andere ruilklasse moeten worden toegepast. Dit gebeurt door het toevoegen van een kaart waarop staat aangegeven: het bedrijfsnummer (posities 1 t/m 8), het kavelnummer (posities 9 t/m 11), de oppervlakte die naar een andere ruilklasse moet worden overgebracht, wat de gehele kaveloppervlakte of een gedeelte ervan kan zijn (posities 12 t/m 20, met het decimaalteken op positie 16), de ruilklasse waar de oppervlakte naar toe moet (posities 21 t/m 23), het nummer 999 wat betekent dat het een wijziging van de ruilklasse betreft (posities 24 t/m 27) en een 1 als code voor de uitruilbaarheid op positie 29.

De kaarten moeten in volgorde van de bedrijfs- en kavelnummers worden verstrekt. Indien geen boerderijverplaatsing voorkomt wordt in plaats van deze kaarten een blanco kaart toegevoegd. De kaarten worden afgesloten met een EOR kaart.

De volledige invoer van het programma bestaat uit de bovengenoemde ponskaarten en twee permanent files waarop de gegevens van de CIN en de lijst met bedrijfstypen voorkomen.

De op de ponskaarten vermelde gegevens worden door het programma ingelezen met de regels 18 t/m 61. De eerste ponskaart met onder andere de naam van de ruilverkaveling wordt ingelezen op regel 18. De volgende kaarten met de vaknummers en de bijbehorende ruilklasse worden ingelezen op regel 22, waarbij gelijktijdig het vaknummer en de ruilklasse worden samengevoegd tot één getal en weggezet in de array IRX. Bovendien wordt het totale aantal vakken vastgelegd in de variabele MIRT en het hoogste vaknummer in de variabele IVM. Hierna volgt op regel 29 het inlezen van de gegevens voor de te splitsen kavels. Deze gegevens worden in array's gezet, zodat ze met de subroutine INDCOL kunnen worden gesorteerd op bedrijfsnummer en kavelnummer. De gesorteerde gegevens worden daarna weggeschreven op tape 4. Vervolgens worden de drie series benamingen van de bedrijfstypen in array's ingelezen op de regels 42, 47 en 50, waarna de array IAAA wordt gevuld met streepjes (regel 53) ten behoeve van het printen van tabellen. Ten slotte worden de voor boerderijverplaatsing benodigde gegevens ingelezen op regel 56 en weggeschreven naar tape 10 op regel 59.

Het programma gaat nu verder met het inlezen van de kavelgegevens van de CIN (regel 66). Niet alle in de CIN voorkomende gegevens zijn nodig voor het berekenen van de toedeling. Daarom worden per kavel ingelezen: dorpsbehorennummer, bedrijfsnummer, kavelnummer, oppervlakte, bijzonderheid 8 (nummer dorpsbehoren waar de kavel ligt), bijzonderheid 10 (kan een aanduiding bevatten over een bijzondere ligging van de kavel), kavelafstand verdeeld over de diverse soorten verharding, vaknummer en code uitruilbaarheid. Kavels die in de CIN zijn opgenomen maar buiten de ruilverkaveling liggen worden buiten het bestand gehouden (regel 69). Na deze regel kunnen statements aan het programma worden toegevoegd voor het verbeteren van fouten in de CIN of het aanbrengen van wijzigingen in de gegevens ten behoeve van het berekenen van alternatieven. Op het verbeteren van fouten wordt op de plaatsen waar foutmeldingen zich voordoen nader ingegaan.

De ingelezen gegevens van de CIN worden samengevoegd in een drietal array's, gesorteerd op bedrijfs- en kavelnummer, weer uit de array's teruggehaald en weggeschreven op tape 3.

Met de statements op de regels 89 t/m 106 wordt controle uitgeoefend op de kavelnummers die zijn opgegeven bij de te splitsen kavels. Opgegeven kavelnummers die niet voorkomen bij de gegevens van de CIN worden als foutmelding geprint. Indien na het beëindigen van de controle fouten zijn geconstateerd wordt het programma gestopt, waarna eerst de fouten moeten worden verbeterd op de ingevoerde ponskaarten met te splitsen kavels.

Het volgende onderdeel dat door het programma wordt uitgevoerd is het waar nodig splitsen van kavels, het aanbrengen van de juiste uitruilbaarheidscode en het wegschrijven van de complete gegevens naar tape 11 (regels 108 t/m 166). Het programma is er op berekend dat een kavel maximaal in vier delen kan worden gesplitst. Bij de splitsing behoudt elk deel het oorspronkelijke kavelnummer zodat bij een latere telling van het aantal kavels een gesplitste kavel als één kavel telt. Het wijzigen van de in de CIN opgegeven uitruilbaarheidscode is nodig omdat in het programma van een andere code wordt gebruik gemaakt. In de CIN heeft een te splitsen kavel één van de codes 2 t/m 5, een niet uitruilbare kavel code 1 en

een uitruilbare kavel een blanco code. Bij de verdere verwerking in het programma wordt alleen gebruik gemaakt van de codes 1 en 2 voor respectievelijk uitruilbare en niet uitruilbare kavels.

Indien in de CIN een fout is gemaakt met de uitruilbaarheids-code van te splitsen kavels kan het programma de gegevens van de kavel en van de kaveldelen niet bij elkaar zoeken. Dit kan zowel voorkomen indien een code 2 of groter is vergeten of dat zo'n code ten onrechte bij een niet te splitsen kavel is vermeld. In de output verschijnt dan de foutmelding dat de als te splitsen opgegeven kavel niet voorkomt bij de kavelgegevens. Doordat het programma de gegevens van de te splitsen kavel blijft vasthouden worden ten onrechte ook de volgende te splitsen kavels van een foutmelding voorzien. In een dergelijk lijstje met foutmeldingen hoeft dus de eerste foutmelding alleen maar serieus te worden genomen. Komt een dergelijke fout voor, dan wordt het programma gestopt en moet na regel 69 een statement aan het programma worden toegevoegd om de fout op te heffen. Hoewel het niet vaak voorkomt bestaat de mogelijkheid dat het programma meermalen wordt gestopt omdat bij meerdere kavels een fout in de uitruilbaarheids-code is gemaakt. Na elke stop zal dan een extra statement aan het programma moeten worden toegevoegd ter verbetering van de geconstateerde fout, waarna het programma opnieuw moet worden gedraaid. De al eerder toegevoegde statements moeten blijven gehandhaafd omdat deze niet in het programma worden vastgelegd.

In het volgende deel van het programma (regels 168 t/m 210) wordt eerst voor de bedrijven berekend in welke bedrijfsgrootteklasse ze vallen, waarna aan de lijst waarop per bedrijf het dorpsbehorennummer en het bedrijfstypenummer staat vermeld een nummer voor de bedrijfsgrootteklasse wordt toegevoegd. De nieuwe lijst wordt weggeschreven naar tape 20.

Aan de hand van de array IRX, waarin de vaknummers met de bijbehorende ruilklasse zijn opgenomen, wordt voor elke kavel de ruilklasse waarin hij voorkomt bepaald. Alle voor de verdere verwerking benodigde kavelgegevens zijn nu bekend en worden weggeschreven naar tape 5.

In dit deel van het programma zijn foutmeldingen opgenomen voor het ontbreken van een bedrijf op de lijst met bedrijfsnummers en het niet voorkomen van een vaknummer in de array met vaknummers. Ook hier wordt het programma gestopt indien zich fouten voordoen. Een fout in de lijst met bedrijfsnummers kan worden hersteld door toevoeging van statements na regel 183 van het programma; fouten in de vaknummers kunnen worden verbeterd door na regel 69 een statement toe te voegen of door het herstellen van een fout in de ponskaarten met vaknummers.

Alle fouten die eventueel in de gegevens voorkwamen moeten nu zijn verbeterd. In het resterende gedeelte van het programma komen dan ook geen foutmeldingen meer voor. Dit resterende gedeelte van het programma zal daarom minder uitvoerig worden besproken dan het voorafgaande.

Achtereenvolgens berekent het programma bedrijfsgegevens en verkavelingskenmerken, gevolgd door een serie printopdrachten om de berekende gegevens in tabellen te presenteren. Indien boerderijverplaatsing voorkomt worden de daaruit voortvloeiende wijzigingen in uitruilbaarheid en ruilklasse aangebracht, waarna per bedrijf de uitruilbare oppervlakte en de schijnbare gemiddelde kavelafstand per ruilklasse wordt berekend. De niet uitruilbare kavels worden uit het bestand gelicht en gesorteerd op vaknummer. De vakoppervlakten worden berekend door sommatie van de oppervlakten van de binnen een vak gelegen uitruilbare kavels. Hierna worden per ruilklasse files aangemaakt waarop de bedrijven voorkomen waaraan in de betreffende ruilklasse grond moet worden toegedeeld en de vakken die in die ruilklasse zijn gelegen. Bij de bedrijven wordt de in de ruilklasse toe te delen oppervlakte en de schijnbare kavelafstand vermeld, bij de vakken alleen de oppervlakte. Gelijktijdig wordt per ruilklasse de totale uitruilbare oppervlakte berekend, zowel door sommatie van de bedrijfsoppervlakten als van de vakoppervlakten. Beide uitkomsten moeten hetzelfde zijn maar door afrondingen kunnen kleine verschillen optreden. Deze kleine verschillen worden gecorrigeerd door het aanbrengen van correcties in de vakoppervlakten. Ten slotte worden de per ruilklasse aangemaakte files met bedrijven en vakken in numerieke volgorde van de ruilklassen op één random access file geplaatst.

Het programma wordt verwerkt met de volgende control cards:

```
LAXXX,IO30,T500,CL270000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE5,*PF.
REQUEST,TAPE16,*PF.
REQUEST,TAPE18,*PF.
REQUEST,TAPE20,*PF.
ATTACH,TAPE17,LACISLOT,ID=LAXXX.
ATTACH,TAPE19,LABEDTYP,ID=LAXXX.
ATTACH,OLDPL,LATOEDNCI,ID=LAKIK.
UPDATE,W,P.
FTN,I=COMPILE,L=0,PL=100000.
LGO.
CATALOG,TAPE5,LAAFMMAT,ID=LAXXX.
CATALOG,TAPE16,LABEDRVAK,ID=LAXXX.
CATALOG,TAPE18,LAVASTKAV,ID=LAXXX.
CATALOG,TAPE20,LABEDTYGR,ID=LAXXX.
COPYSBF,TAPE5,OUTPUT.
COPYSBF,TAPE6,OUTPUT.
COPYSBF,TAPE7,OUTPUT.
COPYSBF,TAPE8,OUTPUT.
COPYSBF,TAPE18,OUTPUT.
COPYSBF,TAPE20,OUTPUT.
EOR
*IDENT NEW
*COMPILE TOEDNCI
*INSERT TOEDNCI.69
      IF(I.EQ.15000586.AND.K.EQ.4) IU=2
EOR
```

kaart met nummer 1e dorpsbehoren met buiten blok bedrijven, naam ruilverkaveling en maximale afstand voor het berekenen van vak-afstanden.

kaarten met vaknummers en het nummer van de ruilklasse waarin de vakken zijn gelegen

EOR

kaarten met gegevens voor te splitsen kavels

EOR

kaarten met namen bedrijfstypen voor tabellen met binnen het blok gelegen bedrijven

EOR

kaarten met namen bedrijfstypen voor tabellen met buiten het blok gelegen bedrijven

kaarten met namen bedrijfstypen voor tabel met bedrijfsgegevens

kaarten met gegevens voor boerderijverplaatsing of, indien geen boerderijverplaatsing voorkomt, een blanco kaart

EOF

De hier aangegeven control cards behoeven enige nadere toelichting omdat per gebied kleine verschillen kunnen optreden. Zo is het aantal COPYSBF kaarten afhankelijk van het aantal ruilklassen waarmee in de ruilverkaveling wordt gewerkt. De kaarten die zorgen voor het printen van de inhoud van de tapes 5, 18 en 20 moeten altijd voorkomen. Met de kaart voor tape 6 worden de gegevens van de bedrijven en vakken in ruilklasse 1 geprint, de kaart voor tape 7 geeft hetzelfde van ruilklasse 2 enzovoort. Bij de in het voorbeeld gegeven control cards is er dus van uitgegaan dat er drie ruilklassen voorkomen. Komt in een gebied slechts één ruilklasse voor dan is een COPYSBF kaart voor tape 6 voldoende maar zouden in een gebied bijvoorbeeld vier ruilklassen voorkomen dan moeten kaarten voor de tapes 6 t/m 9 worden toegevoegd.

Het zal duidelijk zijn dat de voor het aanbrengen van wijzigingen in het programma benodigde UPDATE kaarten volledig afhankelijk zijn van de betreffende wijzigingen. In het voorbeeld is er van uitgegaan dat na regel 69 van het programma een statement moet worden toegevoegd waarmee de uitruilbaarheidscode van een kavel wordt gewijzigd. Indien geen wijzigingen hoeven te worden aangebracht moet alleen de kaart *COMPILE TOEDNCL worden gebruikt.

De output van het programma geeft eerst enige UPDATE informatie, onder andere de aan het programma toegevoegde statements. Daarna worden eventueel de foutmeldingen geprint die hiervoor reeds zijn beschreven. Als er foutmeldingen zijn volgt in de output alleen nog de day-file. Eerst moeten de voor herstellen van de fouten benodigde ingrepen worden verricht, waarna het programma opnieuw kan worden gedraaid.

Als controle op de eerste ingevoerde ponskaart geeft de output vervolgens de naam van de ruilverkaveling en het nummer van het eerste dorpsbehoren met buiten blok bedrijven, waarna tabel 1 wordt geprint. In deze tabel worden in volgorde van de bedrijfsnummers per bedrijf een aantal bedrijfsgegevens verstrekt, zoals die in de bestaande toestand voorkomen. In de output komen nu de tabellen 2 (bestaande toestand) en B1 t/m B5. Voor de inhoud van deze tabellen kan worden verwezen naar de omschrijving in fig. 3.

Na de tabellen wordt een opgave gedaan van het aantal bedrijven en vakken dat per ruilklasse voorkomt. Deze opgave is van belang omdat hiermee de afmetingen van de diverse matrices bij de toedelingsberekening bekend zijn. In een volgend programma moeten de afmetingen worden gebruikt voor het vaststellen van een aantal dimensions. Ter controle wordt ook de totale oppervlakte per ruilklasse gegeven, berekend uit de bedrijfsoppervlakten en uit de vakoppervlakten. Per ruilklasse moeten beide berekende oppervlakten exact gelijk zijn. Het is hoogst onwaarschijnlijk dat een verschil tussen beide oppervlakten zal worden geconstateerd, maar als dit geval zich voordoet zal in het programma en de gegevens moeten worden nagezocht wat de oorzaak van de fout is.

In het resterende gedeelte van de output wordt de inhoud van de files gegeven die in de COPYSBF kaarten zijn vermeld. Deze output kan eventueel worden gebruikt voor controle op de juistheid van de uitgevoerde berekeningen.

Het kan voorkomen dat op verzoek van gebruikers de in de output voorkomende tabellen moeten worden gepresenteerd in dezelfde boekvorm als de CIN. In dat geval is het nodig dat de output op blanco papier wordt geprint, waarvoor na de laatste COPYSBF kaart de control card ROUTE,OUTPUT,DC=PR,FC=BL,DEF. moet worden toegevoegd.



Door deze kaart blijft de output in de output-queue staan. Nadat op de printer blanco papier is aangebracht kan de output dan worden opgeroepen. Dit kan op meerdere manieren worden gedaan. Het beste is hiervoor de assistentie van de operator in te roepen.

3.17. P r o g r a m VAKAFS

Dit programma dient voor het berekenen van de kortste afstanden over de weg, rekening houdend met de wegkwaliteit, van de bedrijven naar de diverse vakken. Bovendien worden met het programma de voor de toedelingsberekening benodigde kostenmatrices berekend. Het programma is afgeleid van een door het Mathematisch Centrum te Amsterdam vervaardigd programma waarmee de afstanden van het bedrijfsgebouw naar de bij het bedrijf behorende percelen kunnen worden berekend.

Het programma begint met het inlezen van de gegevens van het wegennet, welke gegevens gelijktijdig met behulp van een paar subroutines worden opgeborgen in een aantal array's. Vervolgens worden de gegevens van de in ruilklasse 1 voorkomende bedrijven en vakken ingelezen. De bij de bedrijven en vakken behorende coördinaten worden opgezocht, waarna de bedrijven en de vakken met hun coördinaten worden vastgezet in een twee dimensionale array. Het gedeelte van de array waarin de bedrijven voorkomen wordt gesorteerd op bedrijfsnummer met de subroutines SORT en PERMMAT.

In een aantal array's worden gegevens vastgelegd waaruit bij het berekenen van de afstanden kan worden bepaald met welke andere punten een bepaald punt van het wegennet directe verbinding heeft.

Voor het berekenen van de vakafstanden worden met de subroutine NUMBER eerst de vakken opgezocht waar naar toe de afstanden moeten worden berekend. Dezelfde subroutine wordt gebruikt voor het opzoeken van de bedrijven waar vandaan de afstandsberekening moet plaatsvinden. Het berekenen van de afstanden wordt gedaan met de subroutine SYMDIST, waarna de berekende afstanden met de subroutine RESULT in een array worden gezet. De subroutine KOSMAT berekent ten slotte de gewichten en de kostenmatrix. Indien in het gebied meerdere ruilklassen voorkomen worden de vanaf het inlezen van de gegevens van de bedrijven en vakken

vermelde handelingen voor elke ruilklasse herhaald, zodat voor elke ruilklasse een kostenmatrix beschikbaar komt. Het laatste gedeelte van het programma zorgt voor het printen van de kostenmatrices.

De tekst van het programma en de bijbehorende subroutines is opgenomen in de bijlagen 15.1 t/m 15.8. Met de volgende control cards kan het programma worden verwerkt:

```
LAXXX,IO20,T400,CL230000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE7,*PF.
REQUEST,TAPE8,*PF.
ATTACH,TAPE1,LAWEGCO,ID=LAXXX.
ATTACH,TAPE2,LABEDRCO,ID=LAXXX.
ATTACH,TAPE4,LAVAKCO,ID=LAXXX.
ATTACH,TAPE5,LAAFMMAT,ID=LAXXX.
ATTACH,TAPE6,LABEDRVAK,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
VAKAFS.
CATALOG,TAPE7,LAKOSMAT,ID=LAXXX.
CATALOG,TAPE8,LAVAKAFS,ID=LAXXX.
EOF
```

Het is mogelijk dat in de output van het programma een foutmelding voorkomt. Met behulp van de tekst van de foutmelding zal dan in het programma of in de gegevens naar de fout moeten worden gezocht. Alle foutmeldingen worden gedaan door de subroutine ERROR. Door na te gaan waar de betreffende foutmelding in het programma of de andere subroutines wordt opgeroepen, kan de plaats waar de fout is geconstateerd worden gelocaliseerd.

De output van het programma bevat de kostenmatrices van de verschillende ruilklassen. Daarbij zijn per bedrijf de kostenfactoren naar alle vakken (16 per regel) geprint. Het is wenselijk de output door te kijken omdat het kan voorkomen dat bij een bedrijf alle kostenfactoren 99999 bedragen. Dit zou betekenen dat voor het betreffende bedrijf alle vakken op een grotere afstand liggen dan de opgegeven maximale afstand tot waar afstandsberekening plaatsvindt.

Het kan zijn dat de maximale afstand te laag is gekozen maar waarschijnlijker is het dat het een bedrijf betreft dat aan een insteekweg is gelegen waarvan de coördinaten geen aansluiting hebben op de rest van het wegennet. Als dit laatste het geval is zal in de file LAWEGCO in de betreffende routes van het wegennet correcties moeten worden aangebracht zodat de insteekweg wel aansluiting verkrijgt met het wegennet.

3.18. P r o g r a m INSTEP

Dit program dient voor het gereed maken van de input voor de toedelingsberekening. Er wordt een file aangemaakt waarop voor de diverse ruilklassen na elkaar per ruilklasse achtereenvolgens de afmetingen van de matrix, het nummer van de ruilklasse, de vakoppervlakten, de vaknummers, de bedrijfsoppervlakten, de bedrijfsnummers en de kostenmatrix komen te staan. Het programma voorziet in de mogelijkheid dat in een bepaalde ruilklasse slechts één vak voorkomt. De gegevens van een dergelijke ruilklasse worden niet op de file geplaatst, maar de bedrijfsnummers met de toe te delen oppervlakten worden geprint.

Het programma is vervaardigd door ing. B.P. Akkers; de tekst van het programma is weergegeven in de bijlagen 16.1 t/m 16.3. De voor de verwerking van het programma benodigde control cards zijn:

```
LAXXX,IO10,T100,CL370000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE4,*PF.
ATTACH,TAPE1,LABEDRVAK,ID=LAXXX.
ATTACH,TAPE2,LAAFMMAT,ID=LAXXX.
ATTACH,TAPE3,LAKOSMAT,ID=LAXXX.
UPDATE,W,P,C.
FTN,I=COMPILE,L=0.
LGO.
CATALOG,TAPE4,LAKOSTENMATRIX,ID=LAXXX.
EOR
*IDENT NEW
```

```
*COMPILE INSTEP
*DELETE INSTEP.6,INSTEP.7
      DIMENSION BDR(3),NAAM(3),VAK(2),MA(10),NA(10),INDEX(4000),KOST(2,3
     10000),KOS(4000),IBDR(2,500),IOPPB(2,500),IVAK(2,60),IOPPV(2.60)
EOF
```

Uit de control cards blijkt al dat het programma in UPDATE is gezet. Dit is niet gedaan omdat ingrepen in het programma mogelijk moeten zijn, maar voor het aanpassen van de dimensions aan het gebied waarin wordt gewerkt. Het voor de variabelen KOST,IBDR,IOPPB, IVAK en IOPPV benodigde dimension is afhankelijk van het aantal ruilklassen en het grootste aantal bedrijven en vakken dat in een ruilklasse voorkomt. Voor de genoemde variabelen geldt dat de eerste index gelijk moet zijn aan het aantal ruilklassen. De tweede index van KOST moet voldoende zijn om het product van het aantal bedrijven en vakken in de grootste ruilklasse te kunnen bevatten. Voor IBDR en IOPPB moet de tweede index voldoende zijn voor het grootste aantal bedrijven dat in een ruilklasse voorkomt en voor IVAK en IOPPV moet het grootste aantal vakken dat in een ruilklasse voorkomt binnen de tweede index passen. Bij de control cards zijn als voorbeeld de UPDATE statements opgenomen voor een gebied met twee ruilklassen. Het product van het aantal bedrijven en vakken in de grootste ruilklasse is niet groter dan 30000, het grootste aantal bedrijven dat in een ruilklasse voorkomt is niet meer dan 500 en het grootste aantal vakken in een ruilklasse is kleiner dan 60. Het programma vraagt zeer veel geheugenruimte van de computer, waardoor het nodig is de dimensions zo krap mogelijk te houden. Voor de meeste gebieden zal het programma op deze manier gebruikt kunnen worden, maar in een gebied met veel ruilklassen en een groot aantal bedrijven en vakken kan het voorkomen dat de geheugenruimte van de computer tekort schiet. In dat geval zal het programma zodanig moeten worden gewijzigd dat slechts voor één ruilklasse een file wordt aangemaakt, waarna voor die ruilklasse de toedelings-berekening kan worden uitgevoerd. Voor elke volgende ruilklasse moeten de beide programma's dan opnieuw worden gedraaid.

Als output geeft het programma alleen een dayfile, tenzij een ruilklasse voorkomt met één vak. Zoals reeds is vermeld wordt in dat geval een listing gegeven van de in die ruilklasse voorkomende bedrijven met de toe te delen oppervlakten.

3.19. P r o g r a m STSTONE

De toedelingsberekening volgens het stepping stone algorithme wordt met het programma uitgevoerd. Het is het enige programma dat in Algol 60 is geschreven. Dit is niet zo vreemd omdat het programma is vervaardigd door het Mathematisch Centrum, waar die programmeertaal veel wordt gebruikt. De aanpassing van het programma om het geschikt te maken voor het berekenen van de toedeling is uitgevoerd door ing. B.P. Akkers.

De tekst van het programma komt voor in de bijlagen 17.1 t/m 17.10. De benodigde control cards zijn:

```
LAXXX,IO10,T400,CL370000.
ACCOUNT,016300..,90,900,CTD.....
RFL,370000.
ATTACH,LOCAL,LAKOSTENMATRIX,ID=LAXXX.
REWIND,LOCAL.
REQUEST,HULP,*PF.
ATTACH,OLDPL,LAOPTBER,ID=LAKIK.
UPDATE,W,P,C.
ALGOL,N,I=COMPILE,C=3,L=0.
LGO,C,S.
REWIND,HULP.
CATALOG,HULP,LATOEDEL,ID=LAXXX.
EOR
*COMPILE STSTONE
EOR
CHANNEL,10=LOCAL,P128
CHANNEL,20=PRIMAL,P80
CHANNEL,11=RESULT,P80
CHANNEL,12=HULP,P80
EOF
```

De output van het programma geeft, na de gebruikelijke UPDATE mededelingen, de resultaten van de toedelingsberekening. Per ruilklasse wordt daarbij per vak aangegeven welke bedrijven in dat vak een bepaalde oppervlakte krijgen toegedeeld. Voor de uniformiteit is het gewenst dat deze output in dezelfde tabelvorm wordt geprint als tabel B4 waarin per vak de niet uitruilbare oppervlakte staat aangegeven. Door gebrek aan ervaring met Algol was het niet mogelijk de output in de gewenste vorm te gieten, zodat een apart Fortran programma is gemaakt om de gewenste tabellen te verkrijgen.

3.20. P r o g r a m PRTOED

Hiervoor is reeds vermeld dat dit programma alleen dient om de uitkomsten van de toedelingsberekening in de juiste tabelvorm te presenteren.

Bij de behandeling van program INSTEP is opgemerkt dat indien in een ruilklasse slechts één vak voorkomt, de toedeling van dat vak in de output van INSTEP wordt gegeven. Om ook deze toedeling in de tabellen opgenomen te krijgen is het noodzakelijk met een hulpprogramma deze toedeling op de juiste plaats toe te voegen aan de permanent file LATOEDEL voordat het program PRTOED wordt gedraaid.

In bijlage 18 is de tekst van het programma en de bijbehorende subroutine opgenomen, terwijl voor de verwerking de volgende control cards moeten worden gebruikt:

```
LAXXX,I020,T20,CL60000.
ACCOUNT,016300..,90,900,CTD.....
ATTACH,TAPE1,LATOEDEL,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
PRTOED.
EOR
```

kaart met in volgorde van de ruilklassen het laagste vaknummer dat in elke ruilklasse voorkomt. Voor elk vaknummer moeten vier posities worden gebruikt.

```
EOF
```

De output van het programma bestaat uit de tabellen T1.n. Het getal n geeft het nummer van de ruilklasse aan waarop de tabel betrekking heeft. Indien de output op blanco papier moet worden geprint, kan voor de te volgen handelwijze worden verwezen naar de laatste alinea van par. 3.16 (blz.38).

3.21. P r o g r a m TOEDKEN

In hoofdstuk 2 is al aangegeven dat ook voor de toedeling tabellen met bedrijfsgegevens en verkavelingskenmerken worden gemaakt. De beperkingen van deze tabellen zijn daarbij eveneens vermeld. Het programma zorgt voor de berekening van de gegevens voor deze tabellen en tabel T2. Ook het printen van de tabellen wordt door het programma verzorgd.

De input op ponskaarten van het programma bestaat uit dezelfde drie series met benamingen van bedrijfstypen die ook voor het program TOEDEL worden gebruikt (par. 3.16). Dit gedeelte van de bij dat programma gebruikte ponskaarten kan dus hier opnieuw worden toegevoegd aan de control cards.

Een beschrijving van de werking van het programma kan achterwege blijven aangezien de aan de programmatekst toegevoegde comment cards voldoende informatie geven over de handelingen die met het programma worden verricht.

De tekst van het programma en de subroutines is opgenomen in de bijlagen 19.1 t/m 19.8. Voor de tekst van de gebruikte subroutine INDCOL wordt verwezen naar bijlage 9.3. Met de volgende control cards kan het programma worden verwerkt.

```
LAXXX,IO500,T1200,CL270000.
ACCOUNT,016300..,90,900,CTD.....
ATTACH,TAPE2,LAVAKAFS,ID=LAXXX.
ATTACH,TAPE3,LATOEDEL,ID=LAXXX.
ATTACH,TAPE4,LABEDRVAK,ID=LAXXX.
ATTACH,TAPE5,LAVASTKAV,ID=LAXXX.
ATTACH,TAPE6,LAAFMMAT,ID=LAXXX.
ATTACH,TAPE8,LABEDTYGR,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
```

```
LIBRARY,TORU.
TOEDKEN,PL=100000.
EOR
   kaarten met namen bedrijfstypen voor tabellen met binnen het blok
   gelegen bedrijven
EOR
   kaarten met namen bedrijfstypen voor tabellen met buiten het blok
   gelegen bedrijven
   kaarten met namen bedrijfstypen voor tabel met bedrijfsgegevens
EOF
```

In de output komt eerst tabel T2, waarin per bedrijf een opgave wordt gedaan van de niet uitruilbare en de toegedeelde oppervlakten. Daarna volgen de tabellen 1, met bedrijfsgegevens na toedeling, en 2, met de verkavelingskenmerken na toedeling. Ook hier kan voor de handelwijze wanneer op blanco papier moet worden geprint worden verwezen naar de laatste alinea van par. 3.16 (blz. 38).

Het kan voorkomen dat op verzoek meer output moet worden geleverd dan waarin de programma's voorzien. Zeker wanneer zo'n verzoek betrekking heeft op gegevens over de toedeling kunnen de op tape 7 gezette gegevens, betreffende de toegedeelde kavels met hun oppervlakte en kavelafstand, van pas komen. In een dergelijke situatie zal tape 7 op permanent file moeten worden gezet, zodat de daarop voorkomende gegevens in een extra programma kunnen worden gebruikt. In normale gevallen worden van dit programma geen permanent files aangemaakt, omdat dit het laatste programma is in de reeks waarmee het toedelingsonderzoek wordt uitgevoerd.

4. AFWIJKENDE VORM GEGEVENS

In het voorgaande is de procedure omschreven zoals deze meestal zal voorkomen. Een enkele maal kan zich echter het geval voordoen dat de gegevens waarmee moet worden gewerkt in een afwijkende vorm beschikbaar zijn. Dit kan zijn dat alleen een CIN voorhanden is waarin geen extra gegevens voor de toedelingsberekening zijn opgenomen, maar er kan ook in een gebied worden gewerkt waarvoor helemaal

geen CIN aanwezig is of een te sterk verouderde, en daardoor onbruikbare, CIN. In het laatste geval zullen de gegevens van de gebruikstoestand op een andere wijze beschikbaar moeten zijn. Voor beide gevallen zal worden aangegeven hoe de computerverwerking verloopt en welke gegevens worden gebruikt, althans voor zover zich afwijkingen met de normale procedure voordoen.

4.1. CIN zonder toedelingsgegevens

Indien bij het vervaardigen van de CIN geen rekening is gehouden met een toedelingsberekening, betekent dit dat bij de gegevens per kavel het vaknummer en de uitruilbaarheidscode ontbreken en dat bij de kavels van de buiten het blok gelegen bedrijven de kavelafstand niet is opgenomen. Er zullen dus aanvullende gegevens moeten worden verzameld om de aan de CIN te ontlenen gegevens te completeren. Deze aanvullende gegevens bestaan per kavel uit het bedrijfsnummer, het kavelnummer, het nummer van de ruilklasse waarin de kavel ligt, het vaknummer en een code voor de uitruilbaarheid. Bij te splitsen kavels moeten deze gegevens voor elk gedeelte worden opgegeven en bovendien de oppervlakten van de gedeelten. Voor kavels van buiten het blok gelegen bedrijven moet ook de kavelafstand, verdeeld over de diverse soorten verharding, worden opgegeven. Dit laatste geldt alleen indien dit aantal kavels niet te groot is, zodat de afstanden met de hand worden gemeten. Gaat het om een groter aantal kavels, dan zullen de afstanden over het algemeen na digitalisering, met de computer worden berekend zodat opgave van de kavelafstanden in dit geval niet nodig is. De aanvullende gegevens worden op een ponsdocument vermeld met de volgende indeling:

posities 1 t/m 8 :het bedrijfsnummer
posities 9 t/m 11 :het kavelnummer
posities 12 t/m 14: het nummer van de ruilklasse
posities 15 t/m 18: het vaknummer
posities 19 en 20 : code uitruilbaarheid, een 1 betekent uitruilbaar en een 2 niet uitruilbaar
posities 21 t/m 29: de oppervlakte in ha, met in positie 25 het decimaalteken

posities 30 t/m 33: afstand over verharde weg in hm
posities 34 t/m 36: afstand over semi-verharde weg in hm
posities 37 t/m 39: afstand over onverharde weg in hm
posities 40 t/m 42: afstand over water in hm
posities 43 t/m 45: afstand over land in hm

Het opmaken van het ponsdocument kan het eenvoudigst geschieden door vanaf de gebruikerskaart in volgorde van de vaknummers per vak achtereenvolgens alle daarin gelegen kavels te vermelden.

Voor de verwerking met de computer is een aantal extra programma's nodig ten opzichte van de reeds besproken procedure. Bij de behandeling van deze programma's zal worden aangegeven waar zij in het in de figuren 1 en 2 gegeven schema moeten worden gevoegd. Ook de voor het berekenen van de kavelafstanden benodigde programma's zullen worden besproken, hoewel deze uiteraard kunnen vervallen indien de afstanden van de kavels van de buiten blok bedrijven via het ponsdocument zijn opgegeven.

4.1.1. Aanvullende gegevens op permanent file brengen

Nadat de aanvullende gegevens van de ponsdocumenten op ponskaarten zijn overgebracht, kunnen zij op permanent file worden gezet. In het in fig. 2 gegeven schema kan deze handeling het beste worden ingepast na program BEDTYPE. Voor het aanmaken van de permanent file zijn de volgende control cards nodig:

```
LAXXX,IO20,T30,CL60000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,HULP,*PF
COPYBF,INPUT,HULP.
REWIND,HULP.
CATALOG,HULP,LAAANGEG,ID=LAXXX.
COPYSBF,HULP,OUTPUT.
EOR
   ponskaarten met aanvullende gegevens
EOF
```

De output bestaat uit een listing van de inhoud van de permanent file.

4.1.2. P r o g r a m CONTROL

Met dit programma wordt controle uitgeoefend op de juistheid en volledigheid van de aanvullende gegevens. Door vergelijking van de opgegeven kavelnummers met de kavelnummers die in de input van de CIN voorkomen wordt nagegaan of alle opgegeven kavelnummers juist zijn en of alle in de input van de CIN voorkomende kavelnummers bij de aanvullende gegevens zijn opgegeven. Het verwerken van het programma volgt direct na het aanmaken van de permanent file met aanvullende gegevens.

De tekst van het programma is opgenomen in bijlage 20, terwijl de voor de verwerking van het programma benodigde control cards zijn:

```
LAXXX,IO50,T150,CL150000.
ACCOUNT,016300..,90,900,CTD.....
ATTACH,TAPE1,LACISLOT,ID=LAXXX.
ATTACH,TAPE2,LAAANGEG,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
CONTROL.
EOR
   kaart met het nummer van het eerste dorpsbehoren met buiten blok
   bedrijven; voor het nummer zijn de posities 1 en 2 beschikbaar
EOF
```

De output van het programma geeft na elkaar twee series foutmeldingen. De eerste serie bevat de kavelnummers die niet voorkomen bij de aanvullende gegevens, in de tweede serie worden de kavelnummers vermeld die bij de aanvullende gegevens zijn opgegeven maar niet in de CIN voorkomen. Veel fouten zullen in beide series voorkomen; bijvoorbeeld heeft een foutief opgegeven kavelnummer tot gevolg dat dit nummer wordt gemeld als niet voorkomend in de CIN, terwijl het goede kavelnummer als niet voorkomend bij de aanvullende gegevens wordt opgegeven.

Voor het verbeteren van de fouten in de aanvullende gegevens bestaat geen standaard programma. De benodigde correcties in de permanent file kunnen met een te schrijven hulpprogramma worden aangebracht. Het is echter ook mogelijk de fouten te verbeteren in

de ponskaarten waarop de aanvullende gegevens staan en een nieuwe permanent file aan te maken op de manier zoals in de vorige paragraaf staat beschreven.

4.1.3. Program HUISKAV

Dit programma wordt alleen gebruikt indien de kavelafstanden met de computer moeten worden berekend. Het dient voor het aanmaken van een file met bedrijfsnummers en een aanduiding voor het binnen of buiten blok gelegen zijn van de betreffende bedrijfsgebouwen. In het programma waarmee de kavelafstanden worden berekend, wordt daarmee vastgesteld of een huiskavel wel of niet moet worden opgenomen.

Het programma wordt verwerkt nadat de fouten in de aanvullende gegevens zijn verbeterd. Bijlage 21 geeft de tekst van het programma; de voor de verwerking benodigde control cards zijn:

```
LAXXX,IO30,T50,CL50000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE2,*PF.
ATTACH,TAPE1,LAAANGEG,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU;
HUISKAV.
CATALOG,TAPE2,LABEDRNRS,ID=LAXXX.
COPYSBF,TAPE2,OUTPUT.
EOF
```

De output van het programma geeft een listing van de aangemaakte permanent file waarop de bedrijfsnummers van de in de ruilverkaveling voorkomende bedrijven staan. Het bedrijfsnummer wordt gevolgd door een 1 indien het betreffende bedrijfsgebouw buiten het blok is gelegen.

4.1.4. Program TRKAV

Evenals de volgende programma's die nodig zijn voor het verwerken van de digitalisering van de plaats van de kavels en het berekenen van de kavelafstanden, wordt ook dit programma alleen gebruikt indien kavelafstanden met de computer moeten worden berekend. De verwerking van het programma wordt in het in fig. 1 gegeven schema ingepast na het

program TRVAK. Met het programma worden de gedigitaliseerde coördinaten van de plaats van de kavels getransformeerd naar het RD stelsel. Gelijktijdig worden de tijdens de digitalisering verbeterde fouten verwijderd en een aantal controles uitgevoerd. De controles bestaan uit:

- het nagaan of de gedigitaliseerde coördinaten in de juiste posities in het record staan
- zijn de bij de digitalisering ingetoetste kavelnummers juist
- zijn alle in aanmerking komende kavels gedigitaliseerd
- komen de coördinaten van de kavels voor in het wegennet

In de bijlagen 22.1 en 22.2 is de tekst van het programma opgenomen. De voor de verwerking benodigde control cards zijn:

```
LAXXX,IO70,T100,CL220000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE4,*PF.
ATTACH,TAPE3,LATRAFOR,ID=LAXXX.
ATTACH,TAPE5,LACISLOT,ID=LAXXX.
ATTACH,TAPE6,LAWEGNET,ID=LAXXX.
ATTACH,TAPE7,LACOKAV,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
TRKAV,PL=100000.
CATALOG,TAPE4,LAKAVEL,ID=LAXXX.
EOR
```

kaart met in posities 1 t/m 5 het aantal records van LAWEGNET, in posities 6 en 7 het nummer van het eerste dorpsbehoren met buiten blok bedrijven en in positie 8 een 1 indien alleen de afstanden van de kavels van de buiten blok bedrijven moeten worden berekend

```
EOF
```

In het programma wordt gebruik gemaakt van de nog niet eerder ter sprake gebrachte permanent file LACOKAV. Deze file bevat de gedigitaliseerde gegevens van de kavels en wordt gelijktijdig aangemaakt met de andere files met gedigitaliseerde gegevens zoals dit in de paragrafen 3.4 en 3.5 is aangegeven.

De output van het programma geeft eerst een lijst met de getransformeerde gegevens. Deze bestaan per kavel uit een aantal records, waarbij de recordnummering van de random access file LAKAVEL is aangegeven. Het eerste record van een kavel bevat het kavelnummer en de volgende records de coördinaten van de route van het ontsluitingspunt van de kavel naar het punt waar de weg wordt bereikt, met de wegkwaliteitscoëfficient van de route. Als bij de transformatie een fout wordt geconstateerd in de posities van de coördinaten in het record, wordt na de betreffende coördinaten een vraagteken geprint. De output vervolgt met een listing van de inhoud van de aangemaakte permanent file, waarna de bij de controles geconstateerde fouten worden vermeld.

4.1.5. Program MUTATKA

De tijdens de transformatie geconstateerde fouten in de gedigitaliseerde kavelgegevens kunnem met dit programma worden verbeterd. Aangezien met het programma tevens de coördinaten van de punten waar vanaf de kavels het wegennet wordt bereikt worden vervangen door de gemiddelde coördinaten van het wegennet, moet het programma in het in fig. 2 gegeven schema worden ingevoegd direct na het program MIDDEL.

De aan te brengen mutaties worden in het programma ingevoerd met behulp van ponskaarten. Aan deze ponskaarten gaat een kaart vooraf waarop in de posities 1 t/m 5 het aantal records van de file LAKAVEL wordt aangegeven, een 0 komt in de posities 6, 7 en 8 en het aantal records van het wegennet (LAWEGNET) wordt in de posities 9 t/m 13 vermeld. Het invullen van de ponskaarten met mutaties geschiedt als volgt:

posities 1 t/m 5: het recordnummer

positie 6: een code voor het wijzigen, toevoegen of verwijderen van een record. Een 'blank' geeft aan dat het opgegeven record moet worden gewijzigd, een plusteken betekent dat na het opgegeven recordnummer een record moet worden toegevoegd en een minteken houdt in dat het opgegeven record moet worden verwijderd.

posities 7 t/m 18: afhankelijk van het te muteren record wordt hier het kavelnummer opgegeven of de wegkwaliteitscoëfficient

posities 19 t/m 26: indien het een record met een kavelnummer betreft wordt een nul opgegeven, anders de X van het punt

posities 27 t/m 34: voor een record met een kavelnummer een nul opgeven, anders de Y van het punt

Voor een te verwijderen record hoeft alleen het recordnummer en een minteken te worden opgegeven, in de andere gevallen moet het record volledig worden ingevuld. Indien de volledige gegevens van een kavel moeten worden toegevoegd, moeten de daarvoor benodigde records worden aangebracht voor de laatste kavel die in de file voorkomt. Hiermee wordt een besparing van computertijd verkregen.

De tekst van het programma is opgenomen in de bijlagen 23.1 en 23.2; de voor de verwerking benodigde control cards zijn:

```
LAXXX,IO300,T300,CL130000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,HULP,*PF.
ATTACH,UTILITY.
LIBRARY,UTILITY.
ATTACH,TAPE4,LAWEGNET,ID=LAXXX.
ATTACH,A,LAKAVEL,ID=LAXXX.
CCOPY,,A,TAPE3,II.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU,UTILITY.
MUTATKA,PL=100000.
CCOPY,,TAPE3,HULP,II.
CATALOG,HULP,LAKAVEL,ID=LAXXX.
EOR
   kaart met aantal records LAKAVEL, drie nullen en aantal records
   LAWEGNET
   kaarten met mutaties
EOF
```

De output van het programma geeft de nieuwe inhoud van de file met de nummering van de records, gevolgd door het totale aantal records. Het is mogelijk dat in de output een foutmelding voorkomt ten gevolge van een niet in de juiste positie geponste kaart.
Bij het verbeteren van de betreffende kaart of kaarten moet rekening worden gehouden met de wijziging in de recordnummering door de wel uitgevoerde mutaties. Met de verbeterde kaarten kan het programma opnieuw worden gedraaid.

Door het middelen van de coördinaten van meermalen gemeten punten in het wegennet bestaat er een kleine kans dat in de output een foutmelding verschijnt dat de coördinaten van een kavel niet voorkomen in het wegennet. Deze fout kan het eenvoudigst verholpen worden door met het programma de coördinaten van de kavel aan te passen aan de overeenkomstige coördinaten van het wegennet.

4.1.6. Program INKAF

Direct na het program MUTATKA moet dit programma worden verwerkt. Het programma zet de gedigitaliseerde gegevens van de kavels op een file die als input dient voor het programma waarmee de kavelafstanden worden berekend. De tekst van het programma komt voor in bijlage 24, terwijl het programma wordt verwerkt met de volgende control cards:

```
LAXXX,I060,T100,CL70000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE2,*PF.
ATTACH,TAPE1,LAKAVEL,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
INKAF,PL=100000.
CATALOG,TAPE2,LAKAVGEG,ID=LAXXX.
COPYSBF,TAPE2,OUTPUT.
EOR
```

kaart met in de posities 1 t/m 5 het aantal records van LAKAVEL

```
EOF
```

De output van het programma geeft een listing van de aangemaakte file.

4.1.7. Program KAVAF

Het programma moet direct na het program INKAF worden verwerkt en dient voor het berekenen van de afstanden van het bedrijfsgebouw naar de bij het bedrijf behorende kavels. Evenals het program VAKAFS is het programma afgeleid van een door het Mathematisch Centrum te Amsterdam vervaardigd programma (zie par. 3.17). Het logische gevolg is dat het programma grote overeenkomst vertoont met het

program VAKAFS, alleen bij het inlezen van de bedrijfsgegevens en bij de subroutine RESTAT (RESULT bij VAKAFS) treden grote verschillen op.

In de bijlagen 25.1 t/m 25.8 is de tekst van het programma opgenomen. Voor de verwerking van het programma zijn de volgende control cards benodigd:

```
LAXXX,IO60,T400,CL220000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE5,*PF.
ATTACH,TAPE1,LAWEGCO,ID=LAXXX.
ATTACH,TAPE2,LAKAVGEG,ID=LAXXX.
ATTACH,TAPE4,LABEDRCO,ID=LAXXX.
ATTACH,TAPE6,LABEDRNRS,ID=LAXXX.
ATTACH,TORU,ID=LAKIK.
LIBRARY,TORU.
KAVAF,PL=100000.
CATALOG,TAPE5,LAKAVAFS,ID=LAXXX.
COPYSBF,TAPE5,OUTPUT.
EOF
```

De output van het programma bestaat uit een lijst met de berekende kavelafstanden, verdeeld over de diverse soorten verharding.

4.1.8. Program TOEDEL

De naam van dit programma komt reeds voor in par. 3.16. Het gaat in feite ook om hetzelfde programma, maar door de andere vorm waarin de input beschikbaar is wijkt het eerste gedeelte van het programma af van het programma dat al eerder besproken is. Ook dit programma is in UPDATE gezet om ingrepen mogelijk te maken.

In de bijlagen 26.1 t/m 26.4 wordt de tekst van het eerste gedeelte van het programma gegeven t/m regel 220. De rest van de tekst van het programma is identiek aan de in de bijlagen 14.1 t/m 14.18 gegeven tekst vanaf regel 212.

De invoer van het programma bestaat gedeeltelijk uit ponskaarten. Er kan hier worden volstaan met aan te geven welke ponskaarten moeten worden ingevoerd, de indeling van de ponskaarten is reeds uitvoerig vermeld in par. 3.16.

De eerste ponskaart geeft het nummer van het eerste dorpsbehoren met buiten blok bedrijven, de naam van de ruilverkaveling en de grootste afstand in m waarover de afstand van de bedrijven naar de vakken moet worden berekend. Hierna volgt de eerste serie benamingen van bedrijfstypen, gevolgd door een EOR kaart. Nu komen de beide volgende series met benamingen van bedrijfstypen, waarna tenslotte de kaarten komen die dienen voor het wijzigen van uitruilbaarheid of ruilklasse ten behoeve van boerderijverplaatsing.

De rest van de invoer van het programma bestaat uit een viertal permanent files waarop de gegevens van de CIN, de aanvullende gegevens, de berekende kavelafstanden en de bij de bedrijven behorende bedrijfstypenummers voorkomen. De file met kavelafstanden behoeft nog enige nadere toelichting omdat het in een gebied kan voorkomen dat de kavelafstanden niet hoeven te worden berekend maar op een andere manierbeschikbaar zijn. In dat geval moet wel een file voor de kavelafstanden worden aangemaakt waarop slechts één record voorkomt. Volgens het op regel 89 gegeven Format 260 moet op dit record een fictief bedrijfsnummer, het kavelnummer 1 en nullen voor de diverse afstanden worden gezet.

Het programma verloopt ongeveer op dezelfde manier als het reeds eerder besproken programma. Eerst worden de gegevens van de ponskaarten ingelezen, waarna de kavelgegevens van de CIN worden ingelezen, gesorteerd en op tape gezet. Vervolgens worden aanvullende gegevens ingelezen, in array's gezet en gesorteerd. Indien nodig worden de berekende kavelafstanden aan de aanvullende gegevens toegevoegd en de volledige aanvullende gegevens naar tape weggeschreven. Na een controle op de juistheid van de aanvullende gegevens worden deze gegevens met de kavelgegevens van de CIN samengevoegd en de bedrijfsgrootteklassen berekend. De rest van het programma is gelijk aan het in par. 3.16 besprokene.

De verwerking van hetprogramma is gelijk aan die van het al eerder besproken programma, zodat ook de daar gemaakte opmerkingen over de verwerking hier van kracht zijn. Alleen de plaats waar fouten in de CIN kunnen worden opgevangen is nu niet na regel 69 maar na regel 52, terwijl de UPDATE deckname nu TOEDOCI is.

De voor de verwerking benodigde control cards wijken iets af van die welke in par. 3.16 zijn vermeld, zodat ze voor de duidelijkheid volledig worden weergegeven. Het toevoegen van een UPDATE statement is hier achterwege gelaten.

```
LAXXX,IO30,T500,CL270000.
ACCOUNT,016300..,90,900,CTD.....
REQUEST,TAPE5,*PF.
REQUEST,TAPE16,*PF.
REQUEST,TAPE18,*PF.
REQUEST,TAPE20,*PF.
ATTACH,TAPE17,LACISLOT,ID=LAXXX.
ATTACH,TAPE19,LAKAVAFS,ID=LAXXX.
ATTACH,TAPE21,LAAANGEG,ID=LAXXX.
ATTACH,TAPE22,LABEDTYP,ID=LAXXX.
ATTACH,OLDPL,LATOEDOCI,ID=LAKIK.
UPDATE,W,P.
FTN,I=COMPILE,L=0,PL=100000.
LGO.
CATALOG,TAPE5,LAAFMMAT,ID=LAXXX.
CATALOG,TAPE16,LABEDRVAK,ID=LAXXX.
CATALOG,TAPE18,LAVASTKAV,ID=LAXXX.
CATALOG,TAPE20,LABEDTYGR,ID=LAXXX.
COPYSBF,TAPE5,OUTPUT.
COPYSBF,TAPE6,OUTPUT.
COPYSBF,TAPE7,OUTPUT
COPYSBF,TAPE8,OUTPUT.
COPYSBF,TAPE18,OUTPUT.
COPYSBF,TAPE20,OUTPUT.
EOR
*COMPILE TOEDOCI
```

kaart met nummer eerste dorpsbehoren met buiten blok bedrijven, naam ruilverkaveling en maximale afstand voor het berekenen van de vakafstanden

kaarten met namen bedrijfstypen voor tabellen met binnen blok gelegen bedrijven

EOR

kaarten met namen bedrijfstypen voor tabellen met buiten blok gelegen bedrijven

kaarten met namen bedrijfstypen voor tabel met bedrijfsgegevens

kaarten met gegevens voor boerderijverplaatsing of, indien geen boerderijverplaatsing voorkomt, een blanco kaart

EOF

De output van het programma is volledig gelijk aan de output die in par. 3.16 is beschreven.

Dit is het laatste programma dat moet worden tussengevoegd of gewijzigd als met een CIN wordt gewerkt waarin geen gegevens voor de toedelingsberekening zijn opgenomen. De resterende programma's verlopen op dezelfde wijze als in hoofdstuk 3 is beschreven.

4.2. Geen CIN beschikbaar

Het zal slechts zelden voorkomen dat een toedelingsonderzoek moet worden uitgevoerd in een gebied waarvoor helemaal geen CIN beschikbaar is. Wanneer dit echter het geval is zullen de gegevens van de gebruikstoestand die normaal uit de CIN, eventueel met aanvullende gegevens, worden verkregen, op een andere manier voorhanden moeten zijn. De wijze waarop de benodigde gegevens beschikbaar komen kan per gebied verschillend zijn, zodat de te volgen procedure bij het uitvoeren van de berekening niet exact is te behandelen. Er moet daarom worden volstaan met het globaal aangeven van de werkwijze die in dit geval moet worden gevolgd.

Voor het uitvoeren van de berekening zal gebruik moeten worden gemaakt van één van de beide procedures die reeds zijn behandeld. Dit zal meestal de procedure zijn die wordt gevolgd indien in de CIN geen gegevens voor de toedeling zijn opgenomen, omdat over het algemeen de kavelafstanden door de computer moeten worden berekend. De voor het verwerken van de gedigitaliseerde gegevens benodigde programma's moeten alle op de reeds aangegeven wijze en volgorde worden gedraaid. Uiteraard vervallen de programma's waarmee de gegevens van de CIN worden verwerkt. In plaats daarvan zullen enkele hulpprogramma's nodig zijn om de gegevens van de gebruiks-

toestand zodanig op permanent files te zetten dat ze zo eenvoudig mogelijk in het program TOEDEL kunnen worden gebruikt. In het begin van dit programma zal het nodig zijn wijzigingen aan te brengen om de kavelgegevens op de juiste wijze op een file (tape 5) te krijgen, zodat er in de rest van het programma mee kan worden gewerkt. Naarmate met de hulpprogramma's de files beter in overeenstemming zijn gebracht met de vorm waarin ze normaal voorkomen, zullen minder wijzigingen in het program TOEDEL nodig zijn. De resterende programma's kunnen op de gebruikelijke wijze worden afgewerkt.

5. SAMENVATTING

De methode waarmee in een vroeg stadium van de voorbereidingsfase van een ruilverkaveling een toedelingsonderzoek kan worden uitgevoerd (KIK,1979), is er op gebaseerd dat de benodigde bewerkingen zoveel mogelijk automatisch kunnen worden verricht. Voor deze automatische verwerking is een serie computerprogramma's vervaardigd die na elkaar moeten worden gedraaid. Deze nota is hoofdzakelijk bedoeld als een handleiding voor het werken met de programma's.

Na een kort overzicht van de werkwijze die bij het toepassen van de methode moet worden gevolgd, worden de te gebruiken computerprogramma's uitvoerig behandeld. Een schema van de programma's, met de volgorde waarin ze moeten worden verwerkt, is gegeven in de fig. 1 en 2. De resultaten die met de uitgevoerde berekening worden verkregen zijn opgenomen in fig. 3. In de diverse bijlagen zijn de volledige teksten van alle programma's weergegeven.

Aangezien de voor de toepassing van de methode benodigde gegevens van de gebruikstoestand meestal kunnen worden ontleend aan een CIN waarin speciaal voor de toedelingsberekening een paar extra gegevens zijn opgenomen, is de voor dit geval te volgen werkwijze beschreven. Soms kan het echter gebeuren dat met een CIN moet worden gewerkt zonder extra toedelingsgegevens of dat helemaal geen CIN beschikbaar is. De werkwijze die in deze beide gevallen moet worden gevolgd is in een apart hoofdstuk beschreven.

Enkele van de beschreven programma's zijn ontwikkeld door het

Mathematisch Centrum te Amsterdam en door ing. B.P. Akkers van het TWR te Utrecht.Dit is vermeld bij de beschrijving van de betreffende programma's.

6. LITERATUUR

KIK, R., 1979. Toedelingsonderzoek in de voorbereidingsfase van een ruilverkaveling. Cult. Techn. Tijdschrift, 18.4

```
      PROGRAM BEKAGEG (TAPE1,TAPE2,TAPE3)
      DIMENSION K(7),L(7)
      IB=IB1=0
    1 READ(1,2) KA,KI,(K(M),M=1,7)
      IF(EOF(1)) 999,3
    2 FORMAT(I2,I8,7A10)
    3 READ(1,5) (L(M),M=1,7)
    5 FORMAT(6A10,A6)
      IF(KA.EQ.IB.AND.KI.EQ.IB1) 4,6
    6 IB=KA
      IB1=KI
      WRITE(3,10) KA,KI,(K(M),M=1,7),(L(M),M=1,7)
   10 FORMAT(I2,I8,13A10,A6)
      GOTO 1
    4 WRITE(2,10) KA,KI,(K(M),M=1,7),(L(M),M=1,7)
      GOTO 1
  999 REWIND2  $  REWIND3
      STOP
      END
```

```
      PROGRAM BEDTYPE (TAPE1,TAPE2,OUTPUT,TAPE3=OUTPUT)                 BEDRTYP     2
      IMPLICIT INTEGER (A-Z)                                            BEDRTYP     3
      REAL PA,PV,PT,PS,R                                                BEDRTYP     4
      COMMON /INV/ IK(2500),IK1(2500),R(2500) /SORT/ ILR,IUR            BEDRTYP     5
    5 READ(1,10) ND,I,H,STOT,SGA,SGRV,SGTE,SNGTI,SNGK,SNGV,SNGS         BEDRTYP     6
      IF(EOF(1)) 30,20                                                  BEDRTYP     7
   10 FORMAT(I2,I8,I1,33X,I5,4X,2I3,I4,6X,4I3)                          BEDRTYP     8
   20 IF(H.EQ.4) GOTO 21                                                BEDRTYP     9
      IF(I.LT.10**7) GOTO 5                                             BEDRTYP    10
      IF(STOT.EQ.0) GOTO 28                                             BEDRTYP    11
                                        PV=FLOAT(SGRV)/FLOAT(STOT)      BEDRTYP    12
      PT=FLOAT(SGTE+SNGTI)/FLOAT(STOT)                                  BEDRTYP    13
      PS=FLOAT(SNGK+SNGV+SNGS)/FLOAT(STOT)                              BEDRTYP    14
                                     IF(PV.GE.0.6) GOTO 22              BEDRTYP    15
      IF(PT.GE.0.4) GOTO 24  $  IF(PS.GE.0.6) GOTO 23                   BEDRTYP    16
      BD=4  $  GOTO 26                                                  BEDRTYP    17
   21 BD=5  $  GOTO 26                                                  BEDRTYP    18
   22 BD=1  $  GOTO 26                                                  BEDRTYP    19
   23 BD=2  $  GOTO 26                                                  BEDRTYP    20
   24 BD=3                                                              BEDRTYP    21
   26 WRITE(2,27) ND,I,BD                                               BEDRTYP    22
   27 FORMAT(I2,I10,I3)                                                 BEDRTYP    23
      GOTO 5                                                            BEDRTYP    24
   28 WRITE(3,29) I                                                     BEDRTYP    25
   29 FORMAT(1H ,"BEDRIJF",I10,"HEEFT GEEN OPGAVE SBE")                 BEDRTYP    26
      GOTO 5                                                            BEDRTYP    27
   30 L=0  $  REWIND 2                                                  BEDRTYP    28
   40 L=L+1                                                             BEDRTYP    29
      READ(2,27) ND,I,BD                                                BEDRTYP    30
      IF(EOF(2)) 50,45                                                  BEDRTYP    31
   45 IK(L)=I  $  IK1(L)=ND*10+BD                                       BEDRTYP    32
      GOTO 40                                                           BEDRTYP    33
   50 IUR=L-1  $  ILR=1  $  REWIND 2                                    BEDRTYP    34
      CALL INDCOL                                                       BEDRTYP    35
      DO 60 L = 1,IUR                                                   BEDRTYP    36
      I=IK(L)  $  ND=IK1(L)/10  $  BD=IK1(L)-ND*10                      BEDRTYP    37
   60 WRITE(2,27) ND,I,BD                                               BEDRTYP    38
      REWIND 2                                                          BEDRTYP    39
      STOP                                                              BEDRTYP    40
      END                                                               BEDRTYP    41
```

```
      SUBROUTINE INDCOL                                                 BEDRTYP    42
      IMPLICIT INTEGER (A-Z)                                            BEDRTYP    43
      REAL R,COPY                                                       BEDRTYP    44
      DIMENSION LST(25),IND(2500)                                       BEDRTYP    45
      COMMON /INV/ IK(2500),IK1(2500),R(2500) /SORT/ ILR,IUR            BEDRTYP    46
      DO 10 I = ILR,IUR                                                 BEDRTYP    47
   10 IND(I) = I                                                        BEDRTYP    48
      IS=0  $  LL=ILR  $  UL=IUR                                        BEDRTYP    49
  100 IP=LL $ M=IND(IP) $ X=IK(M)                                       BEDRTYP    50
      IQ=UL $ N=IND(IQ) $ Z=IK(N)                                       BEDRTYP    51
         IF (X.LE.Z) GOTO 110                                           BEDRTYP    52
         K=M  $  Y=X                                                    BEDRTYP    53
         IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y                    BEDRTYP    54
  110    IF ((UL-LL).LE.1) GOTO 600                                     BEDRTYP    55
            XX=X  $  IX=IP  $  ZZ=Z  $  IZ=IQ                           BEDRTYP    56
  200       IP=IP+1  $  IF (IP.GE.IQ) GOTO 210                          BEDRTYP    57
      M=IND(IP) $ X=IK(M) $ IF(X-XX) 200,300,300                        BEDRTYP    58
  210       IP=IQ-1  $  GOTO 500                                        BEDRTYP    59
  300       IQ=IQ-1  $  IF (IQ.LE.IP) GOTO 310                          BEDRTYP    60
      N=IND(IQ) $ Z=IK(N) $ IF(ZZ-Z) 300,400,400                        BEDRTYP    61
  310 IP=IP-1 $ Z=X $ N=M $ M=IND(IP) $ X=IK(M)                         BEDRTYP    62
  400       IF (X.LE.Z) GOTO 410                                        BEDRTYP    63
            K=M  $  Y=X                                                 BEDRTYP    64
            IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y                 BEDRTYP    65
  410       IF (X.LE.XX) GOTO 420                                       BEDRTYP    66
            XX=X  $  IX=IP                                              BEDRTYP    67
  420       IF (Z.GE.ZZ) GOTO 430                                       BEDRTYP    68
            ZZ=Z  $  IZ=IQ                                              BEDRTYP    69
  430       GOTO 200                                                    BEDRTYP    70
  500       IF (IP.LE.IX .OR. X.GE.XX) GOTO 510                         BEDRTYP    71
            IND(IP)=IND(IX)  $  IND(IX)=M                               BEDRTYP    72
  510       IF (IQ.GE.IZ .OR. Z.LE.ZZ) GOTO 520                         BEDRTYP    73
            IND(IQ)=IND(IZ)  $  IND(IZ)=N                               BEDRTYP    74
  520       IF ((UL-IQ).GT.(IP-LL)) GOTO 530                            BEDRTYP    75
            IX=LL  $  IZ=IP-1  $  LL=IQ+1  $  GOTO 540                  BEDRTYP    76
  530       IX=IQ+1  $  IZ=UL  $  UL=IP-1                               BEDRTYP    77
  540       IF (IX.GE.IZ) GOTO 550                                      BEDRTYP    78
            IS=IS+2  $  LST(IS-1)=IX  $  LST(IS)=IZ                     BEDRTYP    79
  550       IF (UL.GT.LL) GOTO 100                                      BEDRTYP    80
  600    IF (IS.EQ.0) GOTO 700                                          BEDRTYP    81
         LL=LST(IS-1)  $  UL=LST(IS)  $  IS=IS-2  $  GOTO 100           BEDRTYP    82
  700 CONTINUE                                                          BEDRTYP    83
      DO 120  I= ILR, IUR                                               BEDRTYP    84
         IF (IND(I)) 120,150,125                                        BEDRTYP    85
  125    K=I                                                           BEDRTYP    86
      CO=IK(K) $ COP=IK1(K) $ COPY=R(K)                                 BEDRTYP    87
  130    J=IND(K)  $  IF (J - 1) 150,140,225                           BEDRTYP    88
  225 IK(K)=IK(J) $ IK1(K)=IK1(J) $  R(K)= R(J)                         BEDRTYP    89
         IND(K)= -1 $ K=J                                               BEDRTYP    90
      GOTO 130                                                          BEDRTYP    91
  140 IK(K)=CO $ IK1(K)=COP $  R(K)=COPY                                BEDRTYP    92
         IND(K)= -1                                                     BEDRTYP    93
  120 CONTINUE                                                          BEDRTYP    94
      RETURN                                                            BEDRTYP    95
  150    STOP "PERMERROR"                                              BEDRTYP    96
      END                                                               BEDRTYP    97
```

```
      PROGRAM HULP (TAPE1,TAPE2,OUTPUT)
      DIMENSION K(3)
      DO 10 I = 1,950
      READ(1,20)(K(L),L=1,3)
   20 FORMAT(2A10,A2)
      IF(I.NE.504) GOTO 30
      ENDFILE 2
   30 WRITE(2,20)(K(L),L=1,3)
   10 CONTINUE
      ENDFILE2  S  REWIND2
      STOP
      END
```

```
      C  PER BLAD WORDEN DE PARAMETERS VOOR DE TRANSFORMATIEFORMULES
      C  BEPAALD UIT DE GEDIGITALISEERDE COORDINATEN VAN HET RUITENNET.
      C  CORRECTIES WORDEN AANGEBRACHT VOOR:
      C    1 REK OF KRIMP VAN DE KAART
      C    2 AFWIJKING T.O.V. 100GR VAN DE HOEK TUSSEN X-AS EN Y-AS VAN DE
      C      DIGITISER
      C    3 HET NIET ZUIVER EVENWIJDIG ZIJN VAN HET RUITENNET MET DE X-AS
      C      VAN DE DIGITISER.
      C  DE PARAMETERS WORDEN OP DE PERMANENT FILE "TRAFOR" GEZET.

            PROGRAM TRAFOR(INPUT,OUTPUT,TAPE1,TAPE2=OUTPUT,TAPE3)
            DIMENSION X(20,20),Y(20,20),TX(20,20),TY(20,20),R(8)
            REWIND 3
          5 READ(1,100) NRB
        100 FORMAT (I3)
            NRT=NRB/100
            WRITE(2,115)
        115 FORMAT(1H1)
            WRITE(2,104) NRB
        104 FORMAT (1H0,13H BLADNUMMER: ,I3)
            READ(1,105) NS
        105 FORMAT (I1)
            IF(NS.EQ.1) S=1. $ IF(NS.EQ.2) S=.5 $ IF(NS.EQ.3) S=.2
            READ (1,106) IXO,IYO
        106 FORMAT (2I6)
            WRITE(2,108) IXO,IYO
        108 FORMAT(1H0,"COORDINATEN ZW RUITPUNT: ",2I7)
            DO 210 I = 1,20
            DO 210 J = 1,20
            X(I,J)=22. $ TX(I,J)=22. $ Y(I,J)=22.
        210 TY(I,J)=22.
         10 READ(1,110) XPI,YPI
            IF(EOF(1)) 20,12
        110 FORMAT(2F7.0)
         12 XP=XPI/10000.  $  YP=YPI/10000.
            IX=XP+.2  $  IY=YP+.2
            IF(IX.LT.1.OR.IX.GT.20) GOTO 262
            IF(IY.LT.1.OR.IY.GT.20) GOTO 262
            X(IY,IX)=XP  $  TX(IY,IX)=XP
            Y(IY,IX)=YP $ TY(IY,IX)=YP $ GOTO 10
         20 DO 220 K = 1,8
        220 R(K)=0.
            N=0
            DO 230 IY = 1,20
            DO 230 IX = 1,20
            IF(X(IY,IX).GT.21.) GOTO 230
            N=N+1 $ XP=X(IY,IX) $ YP=Y(IY,IX) $ XF=FLOAT(IX) $ YF=FLOAT(IY)
            R(1)=R(1)+XP $ R(2)=R(2)+XF $ R(3)=R(3)+YP $ R(4)=R(4)+YF
            R(5)=R(5)+XP*XF  $  R(6)=R(6)+YP*YF
            R(7)=R(7)+XP**2  $  R(8)=R(8)+YP**2
        230 CONTINUE
            DO 240 K = 1,8
        240 R(K)=R(K)/N
            AX=(R(5)-R(1)*R(2))/(R(7)-R(1)**2) $ BX=R(2)-AX*R(1)
            AY=(R(6)-R(3)*R(4))/(R(8)-R(3)**2) $ BY=R(4)-AY*R(3)
            DO 244 K = 1,8
        244 R(K)=0.
```

```
      DO 246 IY = 1,20
      DO 246 IX = 1,20
60    IF(Y(IY,IX).GT.21.) GOTO 246
      X(IY,IX)=X(IY,IX)*AX+BX $ Y(IY,IX)=Y(IY,IX)*AY+BY
      XH=FLOAT(IX) $ YH=FLOAT(IY)-Y(IY,IX) $ R(1)=R(1)+XH
      R(2)=R(2)+YH $ R(3)=R(3)+XH*YH $ R(4)=R(4)+XH**2
  246 CONTINUE
65    DO 248 K = 1,4
  248 R(K)=R(K)/N
      AR=(R(3)-R(1)*R(2))/(R(4)-R(1)**2)
      DO 250 IY = 1,20
      DO 250 IX = 1,20
70    IF(X(IY,IX).GT.21.) GOTO 250
      XH=FLOAT(IY) $ YH=FLOAT(IX)-X(IY,IX)-AR*IY
      R(5)=R(5)+XH $ R(6)=R(6)+YH $ R(7)=R(7)+XH*YH $ R(8)=R(8)+XH**2
  250 CONTINUE
      DO 255 K = 5,8
75    255 R(K)=R(K)/N
      BR=(R(7)-R(5)*R(6))/(R(8)-R(5)**2) $ BR=BR+AR $ DX=0. $ DY=0.
      DO 260 IY = 1,20
      DO 260 IX = 1,20
      IF(TX(IY,IX).GT.21.) GOTO 260
80    DX=DX+FLOAT(IX)-AX*TX(IY,IX)-BR*IY
      DY=DY+FLOAT(IY)-AY*TY(IY,IX)-AR*IX
  260 CONTINUE
      DX=IXO-1000.*S*(1.-DX/N) $ DY=IYO-1000.*S*(1.-DY/N)
      AR=AR*S*.1 $ BR=BR*S*.1 $ AX=AX*S*.1 $ AY=AY*S*.1
85    WRITE(2,120) DX,AX,BR,DY,AY,AR
  120 FORMAT(1H0,"DE TRANSFORMATIEFORMULES ZIJN: "//5H X"= ,F9.1,
     1 3H + ,F9.6,4H*X +,F9.6,3H *Y/5H Y"= ,F9.1,3H + ,F9.6,4H*Y +,
     2 F9.6,3H *X)
      WRITE(3,130)DX,AX,BR,DY,AY,AR
90    130 FORMAT(F9.1,F9.6,F9.6,F9.1,2F9.6)
      NRTT=NRB-NRT*100
      IF(NRTT.LT.NRT) GOTO 5
      ENDFILE 3  $  REWIND 3
      GOTO 264
95    262 WRITE(2,140)
      140 FORMAT("  DIGITISER IS NIET INGESTELD OP X=10000 Y=10000")
      264 STOP
      END
```

```
      C  DE MACHINECOORDINATEN VAN HET WEGENNET WORDEN GETRANSFORMEERD IN
      C  RD COORDINATEN.
      C  FOUTIEVE PONSINGEN TIJDENS HET DIGITALISEREN WORDEN VERWIJDERD.
      C  DE GETRANSFORMEERDE COORDINATEN WORDEN OP DE PERMANENT FILE
      C  "WEGNET" GEZET.
         PROGRAM TRWEG (TAPE1,OUTPUT,TAPE2=OUTPUT,TAPE3,TAPE4)
         IMPLICIT INTEGER (A-Z)
         REAL DX,AXI,BR,DY,AYI,AR
         DIMENSION BUF(3),TAB(100),IND(2500),COOR(4),CODE(100,2),X(100),
        1 Y(100),HULP(27)
         DATA B/"B"/,E/"E"/,STER/"*"/,BLANC/" "/,VRAG/"?"/
         CALL OPENMS (4,IND,2500,0)
         IREC=0  $  WRITE(2,100)
       100 FORMAT(1H1)
         5 READ(3,110) DX,AXI,BR,DY,AYI,AR
       110 FORMAT (2(F9.1,2F9.6))
         6 READ(1,112) NRB
         IF(EOF(1)) 6,7
       112 FORMAT (I3)
         7 NRT=NRB/100
         CODE(1,1)=B  $  NE=0
        10 READ(1,101)(BUF(I),I=1,3)
       101 FORMAT (A1,2A7)
         DECODE(27,104,BUF)(HULP(I),I=1,27)
       104 FORMAT(27A1)
         IF(HULP(1).NE.E) GOTO 20
         CODE(NE,1)=E  $  CODE(NE,2)=BLANC  $ IF(NE.EQ.1) GOTO 16
         DO 14 I = 1,NE
         IREC=IREC+1
         IF(I.EQ.1) GOTO 11
         IF(IABS(X(I)-X(I-1)).LE.90.AND.IABS(Y(I)-Y(I-1)).LE.90) TAB(I)=VRA
        1G
        11 COOR(1)=IFIX(DX+AXI*X(I)+BR*Y(I)+.5)
         COOR(2)=IFIX(DY+AYI*Y(I)+AR*X(I)+.5)
         COOR(3)=CODE(I,1)  $  COOR(4)=CODE(I,2)
         IF(I.EQ.1.OR.I.EQ.NE) GOTO 12
         IF(CODE(I,2).NE.CODE(I-1,2)) TAB(I)=VRAG
        12 WRITE(2,102) I,COOR(1),COOR(2),COOR(3),COOR(4),IREC,X(I),Y(I),
        1 TAB(I)
       102 FORMAT(1H ,I5,2I8,1X,2A1,2X,I5,3X,2I8,1X,A1)
         CALL WRITMS (4,COOR,4,IREC)
        14 CONTINUE
        16 NE=0
        20 DO 22 IPL = 1,27
         IF(HULP(IPL).NE.STER) GOTO 22
         NE=0  $  GOTO 10
        22 CONTINUE
         IF(BUF(3).NE.BLANC) GOTO 24
         GOTO 30
        24 NE=NE+1
         TAB(NE)=BLANC
         IF(HULP(11).NE.BLANC) GOTO 29
         IF(NE.EQ.1) GOTO 28
         CODE(NE,1)=BLANC  $  CODE(NE-1,2)=HULP(1)
        28 DECODE(27,103,BUF)(COOR(I),I=1,3)
       103 FORMAT(A10,I10,I7)
         X(NE)=COOR(2)/1000  $  Y(NE)=COOR(3)
```

```
      GOTO 10
   29 IF(NE.EQ.1) GOTO 31
      CODE(NE,1)=BLANC  $  CODE(NE-1,2)=HOLP(1)
   31 TAB(NE)=VRAG  $  X(NE)=Y(NE)=0
      GOTO 10
   30 NRTT=NRB-NRT*100  $  IF(NRTT.LT.NR1) GOTO 5  $  REWIND3
      STOP
      END
```

```
      C  RESTERENDE FOUTEN IN DE PERMANENT FILE "WEGNET" WORDEN HERSTELD
      C  DOOR WIJZIGEN, TOEVOEGEN OF VERWIJDEREN VAN RECORDS
         PROGRAM MUTATWE(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT,TAPE3)
         IMPLICIT INTEGER (A-Z)
         DIMENSION COOR(4),IND(2500),HULP(4),PUR(4)
         READ(1,100) IT,(PUR(I),I=1,4)
       100 FORMAT(I5,2I1,2A1)
         CALL OPENMS (3,IND,2500,0)
         WRITE(2,101)
       101 FORMAT (1H1)
         M=ML=0
         2 READ(1,102) NR,T,(COOR(I),I=1,4)
         IF(EOF(1)) 40,5
       102 FORMAT(I5,A1,2I6,2A1)
         5 NR=NR+(M-ML)
         IF(T.EQ.1H ) GOTO 10
         IF(T.EQ.1H+) GOTO 20
         IF(T.EQ.1H-) GOTO 30
         WRITE(2,105) NR
       105 FORMAT(1H ,I5,3X,"OPGAVE FOUT")
         GOTO 2
        10 CALL WRITMS (3,COOR,4,NR)
         GOTO 2
        20 J=NR+1
         DO 201 I = J,IT
         K=(IT+J)-1
         CALL READMS (3,HULP,4,K)
         L=K+1
         CALL WRITMS (3,HULP,4,L)
       201 CONTINUE
         CALL WRITMS (3,COOR,4,J)
         M=M+1
         IT=IT+1
         GOTO 2
        30 J=NR+1
         DO 202 I = J,IT
         CALL READMS (3,HULP,4,I)
         L=I-1
         CALL WRITMS (3,HULP,4,L)
       202 CONTINUE
         CALL WRITMS (3,PUR,4,IT)
         ML=ML+1
         IT=IT-1
         GOTO 2
        40 DO 203 I = 1,IT
         CALL READMS (3,COOR,4,I)
       203 WRITE(2,104) COOR
       104 FORMAT (2I8,2X,2A1)
         WRITE(2,110) IT
       110 FORMAT (I5)
         STOP
         END
```

```
      PROGRAM CONWEG(TAPE1,OUTPUT,TAPE2=OUTPUT,INPUT,TAPE3=INPUT)
      IMPLICIT INTEGER (A-Z)
      DIMENSION INDEX(2500),COOR(4),X(2500),Y(2500),C(2500)
      CALL OPENMS (1,INDEX,2500,0)
      READ(3,10) LW
   10 FORMAT(I5)
      WRITE(2,15)
   15 FORMAT(1H1)
      DO 20 I = 1,LW
      CALL READMS (1,COOR,4,1)
      X(I)=COOR(1)  $  Y(I)=COOR(2)  $  C(I)=COOR(3)
   20 CONTINUE
      DO 35 I = 1,LW
      IF(C(I).EQ.1H ) GOTO 35
      DO 25 L = 1,LW
      IF(I.EQ.L) GOTO 25
      IF(IABS(X(I)-X(L)).LT.9.AND.IABS(Y(I)-Y(L)).LT.9) GOTO 35
   25 CONTINUE
      WRITE(2,30) X(I),Y(I),C(I)
   30 FORMAT(1H ,"PUNT MET COORDINATEN :",2I8,2X,"EN CODE :",2X,A1,2X,
     1"HEEFT GEEN AANSLUITING OP HET WEGENNET")
   35 CONTINUE
      STOP
      END
```

```
      PROGRAM TEKWEG(INPUT,OUTPUT,TAPE1=INPUT,TAPE2,TAPE3,TAPE4=OUTPUT)
Cs    DEBUG
Cs    ARRAYS
      DIMENSION IND(2500),COOR(4),IHUL(300),AH(300,75,2)
      INTEGER TL,COOR
      CALL OPENMS(3,IND,2500,0)
      DO 20 I=1,300
   20 IHUL(I)=0
      READ(1,1000)M,ISC,IX,IY
 1000 FORMAT(2I5,2I6)
      IK=0
      TL=IS=0
      IS=IS+1
      IF(ISC.LE.1)ASC=100.
      IF(ISC.EQ.2)ASC=50.
      IF(ISC.EQ.3)ASC=20.
      DO 10 I=1,M
      TL=TL+1
      CALL READMS(3,COOR,4,TL)
      IF(IK.EQ.1)1,200
  200 IF(COOR(3).EQ.1HE)201,202
  201 IK=1
      GOTO 2
  202 IK=0
      GOTO 2
    1 IS=IS+1
      IK=0
    2 IHUL(IS)=IHUL(IS)+1
      IP=IHUL(IS)
      IX1=COOR(1)
      IY1=COOR(2)
      XX=(IX1-IX)/ASC
      YY=(IY1-IY)/ASC
      AH(IS,IP,1)=XX
      AH(IS,IP,2)=YY
   10 CONTINUE
      AY=AX=0.
      DO 220 I=1,IS
      IP=IHUL(I)
      DO 220 J=1,IP
      IF(AH(I,J,2).GT.AY)AY=AH(I,J,2)
      IF(AH(I,J,1).GT.AX)AX=AH(I,J,1)
  220 CONTINUE
      AX=AX+5.
      IT=(AY+70.00001)/70.
      PRINT 141,AX,AY,IT
  141 FORMAT(1H0,2F10.2,I5)
      AKO=ASC*70.
      CALL NAMCTD(5HTAPE2)
      CALL PLOTX(0.,-75.,3)
      CALL PLOTX(5.,-73.,-3)
      DO 21 I=1,IS
      AJS=0.
      IP=IHUL(I)
      DO 21 J=1,IP
      IF(J.EQ.1)22,23
   22 DO 24 IJ=1,10
```

```
      AJS=AJS+70.
      IF(AH(I,J,2).GT.AJS)24,25
   24 CONTINUE
   25 ILS=IJ-1
      AXX=AH(I,J,1)+AX*ILS
      AXY=AH(I,J,2)-70.*ILS
      CALL PLOTX(AXX,AXY,3)
      GOTO 21
   23 AJS=0.
      DO 26 IJ=1,10
      AJS=AJS+70.
      IF(AH(I,J,2).GT.AJS)26,27
   26 CONTINUE
   27 ILS1=IJ-1
      IF(ILS-1LS1)100,101,102
  101 AXX=AH(I,J,1)+AX*ILS
      AXY=AH(I,J,2)-70.*ILS
      CALL PLOTX(AXX,AXY,2)
      GOTO 21
  100 IKJ=J-1
      AXY=70.
      YY1=AH(I,J,2)-70.*ILS1
      YY2=(70.-(AH(I,IKJ,2)-70.*ILS))
      XX1=AH(I,J,1)-AH(I,IKJ,1)
      IF(XX1)30,31,32
   31 AXX=AH(I,J,1)+AX*ILS
      AXY1=0.
      AXX2=AXX1=AH(I,J,1)+AX*ILS1
      AXY2=AH(I,J,2)-70.*ILS1
      GOTO 40
   32 XHUL=(YY2*XX1/(YY1+YY2))
      AXX=AH(I,IKJ,1)+XHUL+AX*ILS
      AXY1=0.
      AXX1=AXX+AX
      AXX2=AH(I,J,1)+AX*ILS1
      AXY2=AH(I,J,2)-70.*ILS1
      GOTO 40
   30 XX1=-1.*XX1
      XHUL=(YY2*XX1/(YY1+YY2))
      AXX=AH(I,IKJ,1)-XHUL+AX*ILS
      AXY1=0.
      AXX1=AXX+AX
      AXX2=AH(I,J,1)+AX*ILS1
      AXY2=AH(I,J,2)-70.*ILS1
      GOTO 40
  102 IKJ=J-1
      AXY=0.
      YY1=(70.-(AH(I,J,2)-70.*ILS1))
      YY2=AH(I,IKJ,2)-70.*ILS
      XX1=AH(I,J,1)-AH(I,IKJ,1)
      IF(XX1)50,51,52
   51 AXX=AH(I,IKJ,1)+AX*ILS
      AXY1=70.
      AXX2=AXX1=AH(I,IKJ,1)+AX*ILS1
      AXY2=AH(I,J,2)-70.*ILS1
      GOTO 40
   52 XHUL=(YY2*XX1/(YY1+YY2))
```

```
      AXX=AH(I,IKJ,1)+XHUL+AX*ILS
      AXY1=70.
      AXX1=AXX-AX
      AXX2=AH(I,J,1)+AX*ILS1
      AXY2=AH(I,J,2)-70.*ILS1
      GOTO 40
   50 XX1=-1.*XX1
      XHUL=(YY2*XX1/(YY1+YY2))
      AXX=AH(I,IKJ,1)-XHUL+AX*ILS
      AXY1=70.
      AXX1=AXX-AX
      AXX2=AH(I,J,1)+AX*ILS1
      AXY2=AH(I,J,2)-70.*ILS1
   40 CALL PLOTX(AXX,AXY,2)
      CALL PLOTX(AXX1,AXY1,3)
      CALL PLOTX(AXX2,AXY2,2)
      ILS=ILS1
   21 CONTINUE
      AT=IT*AX
      CALL PLOTX(AT+5.,0.,3)
      CALL ENDPLT
      END
```

```
      C  DE TRANSFORMATIE VAN DE BEDRIJVEN VERLOOPT OP IDENTIEKE WIJZE
      C  ALS BIJ HET WEGENNET. PERMANENT FILE "BEDR" WORDT GEVORMD.
            PROGRAM TRBEDR (INPUT,TAPE1=INPUT,OUTPUT,TAPE2=OUTPUT,TAPE3,TAPE4,
           1 TAPE5,TAPE6,TAPE7)
            IMPLICIT INTEGER (A-Z)
            REAL DX,AXI,BR,DY,AYI,AR,R
            DIMENSION BUF(3),COOR(3),IND(1500),HULP(27),BDR(1000)
            DIMENSION IB(1000),IX(1000),IY(1000),WEG(3000),COWE(4),LX(3000),LY
           1(3000)
            COMMON /INV/ IK(4000),IK1(4000),R(4000) /SORT/ ILR,IUR
            DATA STER/"F"/,VRAG/"?"/,BLANC/" "/
            CALL OPENMS (5,IND,1500,0)
            READ(1,90) N
         90 FORMAT (I5)
            IREC=0  $  WRITE (2,104)
        104 FORMAT(1H1)
          5 READ(3,100) DX,AXI,BR,DY,AYI,AR
        100 FORMAT (2(F9.1,2F9.6))
            READ(7,102) NRB
        102 FORMAT (I3)
            NRT=NRB/100
         10 READ(7,106)(BUF(J),J=1,3)
            IF(EOF(7)) 40,12
        106 FORMAT (A8,2A7)
         12 FOUT=BLANC
            IREC=IREC+1
            DECODE(27,108,BUF)(HULP(I),I=1,27)
        108 FORMAT(27A1)
            IF(HULP(1).EQ.STER) GOTO 20
            DO 200 I = 2,27
            IF(HULP(I).EQ.STER) GOTO 14
        200 CONTINUE
            IF(HULP(11).NE.BLANC) GOTO 13
            DECODE(27,110,BUF)(COOR(I),I=1,3)
        110 FORMAT(2I10,I7)
            GOTO 15
         13 FOUT=VRAG  $  COOR(1)=X=Y=0
            GOTO 16
         15 COOR(1)=COOR(1)/100  $  X=COOR(2)/1000  $  Y=COOR(3)
         16 COOR(2)=IFIX(DX+AXI*X+BR*Y+.5) $ COOR(3)=IFIX(DY+AYI*Y+AR*X+.5)
            WRITE(2,112) IREC,COOR,FOUT
        112 FORMAT (1H ,I5,I10,2I8,2X,A1)
            CALL WRITMS (5,COOR,3,IREC)
            GOTO 10
         14 IREC=IREC-1  $  GOTO 10
         20 IREC=IREC-2  $  GOTO 10
         40 NRTT=NRB-NRT*100  $  IF(NRTT.LT.NRT) GOTO 5
            REWIND 3
            IUR=0  $  ILR=1
            DO 205 I = 1,IREC
            CALL READMS (5,COOR,3,I)
            IK(I)=IB(I)=COOR(1) $ IK1(I)=IX(I)=COOR(2) $ IY(I)=COOR(3)
            R(I)=FLOAT(COOR(3))  $ IUR=IUR+1
        205 CONTINUE
            CALL INDCOL
            DO 250 L = 2,IREC
            Z=0
```

```
                IF(IK(L).EQ.IK(L-1)) Z=1
                IF(Z.EQ.0) GO10 250
60              WRITE(2,270) IK(L),IK1(L-1),K(L-1),IK1(L),K(L)
              270 FORMAT(1H ,"BEDRIJF ",I8," KOMT VOOR OP COORDINATEN",2X,I8,F8.0,"
                 1 EN",2X,I8,F8.0)
              250 CONTINUE
                K=0
65             42 K=K+1
                READ(4,116) BDR(K)
                IF(EOF(4)) 50,42
              116 FORMAT(2X,I10)
               50 KT=K-1  $  REWIND4
70              DO 206 I = 1,IREC
                DO 210 K = 1,KT
                IF(IB(I).EQ.BDR(K)) GOTO 206
              210 CONTINUE
                WRITE(2,118) IB(I),IX(I),IY(I)
75            118 FORMAT(1H ,I8,"  KOMT NIET VOOR BIJ BEDRIJFSNUMMERS, COORDINATEN :
                1",2X,2I8)
              206 CONTINUE
                DO 212 K = 1,KT
                DO 214 I = 1,IREC
80              IF(BDR(K).EQ.IB(I))  GOTO 212
              214 CONTINUE
                WRITE(2,120) BDR(K)
              120 FORMAT(1H ,I8,"  NIET GEDIGITALISEERD")
              212 CONTINUE
85              CALL OPENMS (6,WEG,3000,0)
                DO 220 L = 1,N
                CALL READMS (6,COWE,4,L)
                LX(L)=COWE(1)  $  LY(L)=COWE(2)
              220 CONTINUE
90              DO 230 J = 1,IREC
                DO 240 M = 1,N
                IF(IABS(IX(J)-LX(M)).LT.9.AND.IABS(IY(J)-LY(M)).LT.9) GOTO 230
              240 CONTINUE
                WRITE(2,130) IX(J),IY(J),IB(J)
95            130 FORMAT(1H ,"COORDINATEN :",2I8,"  VAN BEDRIJF",I10,"  KOMEN NIET V
                1OOR IN WEGENNET")
              230 CONTINUE
                STOP
                END
```

```
      SUBROUTINE INDCOL
      IMPLICIT INTEGER (A-Z)
      REAL R,COPY
      DIMENSION LST(25),IND(4000)
      COMMON /INV/ IK(4000),IK1(4000),R(4000) /SORT/ ILR,IUR
      DO 10 I = ILR,IUR
   10 IND(I)=I
      IS=0  $  LL=ILR  $  UL=IUR
  100 IP=LL  $  M=IND(IP)  $  X=IK(M)
      IQ=UL  $  N=IND(IQ)  $  Z=IK(N)
      IF(X.LE.Z) GOTO 110
      K=M  $  Y=X
      IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y
  110 IF((UL-LL).LE.1) GOTO 600
      XX=X  $  IX=IP  $  ZZ=Z  $  IZ=IQ
  200 IP=IP+1  $  IF(IP.GE.IQ) GOTO 210
      M=IND(IP)  $  X=IK(M)  $  IF(X-XX) 200,300,300
  210 IP=IQ-1  $  GOTO 500
  300 IQ=IQ-1  $  IF(IQ.LE.IP) GOTO 310
      N=IND(IQ)  $  Z=IK(N)  $  IF(ZZ-Z) 300,400,400
  310 IP=IP-1  $  Z=X  $  N=M  $  M=IND(IP)  $  X=IK(M)
  400 IF(X.LE.Z) GOTO 410
      K=M  $  Y=X
      IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y
  410 IF(X.LE.XX) GOTO 420
      XX=X  $  IX=IP
  420 IF(Z.GE.ZZ) GOTO 430
      ZZ=Z  $  IZ=IQ
  430 GOTO 200
  500 IF(IP.LE.IX.OR.X.GE.XX) GOTO 510
      IND(IP)=IND(IX)  $  IND(IX)=M
  510 IF(IQ.GE.IZ.OR.Z.LE.ZZ) GOTO 520
      IND(IQ)=IND(IZ)  $  IND(IZ)=N
  520 IF((UL-IQ).GT.(IP-LL)) GOTO 530
      IX=LL  $  IZ=IP-1  $  LL=IQ+1  $  GOTO 540
  530 IX=IQ+1  $  IZ=UL  $  UL=IP-1
  540 IF(IX.GE.IZ) GOTO 550
      IS=IS+2  $  LST(IS-1)=IX  $  LST(IS)=IZ
  550 IF(UL.GT.LL) GOTO 100
  600 IF(IS.EQ.0) GOTO 700
      LL=LST(IS-1)  $  UL=LST(IS)  $  IS=IS-2  $  GOTO 100
  700 DO 120 I = ILR,IUR
      IF(IND(I)) 120,150,125
  125 K=I
      CO=IK(K)  $  COP=IK1(K)  $  COPY=R(K)
  130 J=IND(K)  $  IF(J-I) 150,140,225
  225 IK(K)=IK(J)  $  IK1(K)=IK1(J)  $  R(K)=R(J)
      IND(K)=-1  $  K=J
      GOTO 130
  140 IK(K)=CO  $  IK1(K)=COP  $  R(K)=COPY
      IND(K)=-1
  120 CONTINUE
      RETURN
  150 STOP "PERMERROR"
      END
```

```
      C  DE TRANSFORMATIE VAN DE VAKKEN VERLOOPT OP IDENTIEKE WIJZE
      C  ALS BIJ HET WEGENNET. PERMANENT FILE "VAKKEN" WORDT GEVORMD.
         PROGRAM TRVAK (INPUT,TAPE1=INPUT,OUTPUT,TAPE2=OUTPUT,TAPE3,TAPE4,
        1 TAPE5,TAPE6)
         IMPLICIT INTEGER (A-Z)
         REAL DX,AXI,BR,DY,AYI,AR
         DIMENSION WEG(3000),VAK(200),COWE(4),IX(3000),IY(3000)
         DIMENSION BUF(3),COOR(3),HULP(27),IND(1500)
         DATA STER/"F"/,VRAG/"?"/,BLANC/" "/
         READ(1,90) N
      90 FORMAT(I5)
         IREC=0
         WRITE(2,100)
     100 FORMAT(1H1)
         CALL OPENMS (4,IND,1500,0)
       5 READ(3,101) DX,AXI,BR,DY,AYI,AR
     101 FORMAT (2(F9.1,2F9.6))
         READ(6,102) NRB
     102 FORMAT (I3)
         NRT=NRB/100
     205 READ(6,106) (BUF(K),K=1,3)
         IF(EOF(6)) 40,8
     106 FORMAT (A3,2A7)
       8 FOUT=BLANC
         IREC=IREC+1
         DECODE(27,103,BUF)(HULP(I),I=1,27)
     103 FORMAT(27A1)
         IF(HULP(1).EQ.STER) GOTO 20
         DO 300 I = 2,27
         IF(HULP(I).EQ.STER) GOTO 10
     300 CONTINUE
         IF(HULP(11).NE.BLANC) GOTO 9
         DECODE(27,107,BUF)(COOR(I),I=1,3)
     107 FORMAT(2I10,I7)
         GOTO 11
       9 FOUT=VRAG  $  COOR(1)=X=Y=0
         GOTO 12
      11 COOR(1)=COOR(1)/10**7  $  X=COOR(2)/1000  $  Y=COOR(3)
      12 COOR(2)=IFIX(DX+AXI*X+BR*Y+.5)
         COOR(3)=IFIX(DY+AYI*Y+AR*X+.5)
         WRITE(2,105) IREC,COOR,FOUT
     105 FORMAT (1H ,2I5,2I8,2X,A1)
         CALL WRITMS (4,COOR,3,IREC)
         GOTO 205
      10 IREC=IREC-1
         GOTO 205
      20 IREC=IREC-2
         GOTO 205
      40 NRTT=NRB-NRT*100
         IF(NRTT.LT.NRT) GOTO 5
         REWIND 3
         CALL OPENMS (5,WEG,3000,0)
         DO 50 I = 1,N
         CALL READMS(5,COWE,4,I)
         IX(I)=COWE(1)  $  IY(I)=COWE(2)
      50 CONTINUE
         DO 60 L = 1,IREC
```

```
      CALL READMS (4,COOR,3,L)
      DO 70 K = 1,N
60    IF(IABS(COOR(2)-IX(K)).LE.9.AND.IABS(COOR(3)-IY(K)).LE.9) GOTO 60
   70 CONTINUE
      WRITE(2,80) COOR(2),COOR(3),COOR(1)
   80 FORMAT(1H ,"COORDINATEN :",2I8,"  VAN VAK",I5,"  KOMEN NIET VOOR I
     1N WEGENNET")
65    60 CONTINUE
      STOP
      END
```

```
      C  RESTERENDE FOUTEN IN DE PERMANENT FILE "BEDR" WORDEN HERSTELD
      C  DOOR WIJZIGEN, TOEVOEGEN OF VERWIJDEREN VAN RECORDS
            PROGRAM MUBEDR (INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT,TAPE3)
            IMPLICIT INTEGER (A-Z)
            DIMENSION COOR(3),IND(1500),HULP(3),PUR(3)
            READ(1,100) IT,(PUR(I),I=1,3)
        100 FORMAT(I5,3I1)
            CALL OPENMS (3,IND,1500,0)
            WRITE(2,101)
        101 FORMAT (1H1)
            M=ML=0
          2 READ(1,102) NR,T,(COOR(I),I=1,3)
            IF(EOF(1)) 40,5
        102 FORMAT(I5,A1,I8,2I6)
          5 NR=NR+(M-ML)
            IF(T.EQ.1H ) GOTO 10
            IF(T.EQ.1H+) GOTO 20
            IF(T.EQ.1H-) GOTO 30
            WRITE(2,105) NR
        105 FORMAT(1H ,I5,3X,"OPGAVE FOUT")
            GOTO 2
         10 CALL WRITMS (3,COOR,3,NR)
            GOTO 2
         20 J=NR+1
            DO 201 I = J,IT
            K=(IT+J)-I
            CALL READMS (3,HULP,3,K)
            L=K+1
            CALL WRITMS (3,HULP,3,L)
        201 CONTINUE
            CALL WRITMS (3,COOR,3,J)
            M=M+1
            IT=IT+1
            GOTO 2
         30 J=NR+1
            DO 202 I = J,IT
            CALL READMS (3,HULP,3,I)
            L=I-1
            CALL WRITMS (3,HULP,3,L)
        202 CONTINUE
            CALL WRITMS (3,PUR,3,IT)
            ML=ML+1
            IT=IT-1
            GOTO 2
         40 DO 203 I = 1,IT
            CALL READMS (3,COOR,3,I)
        203 WRITE(2,104) COOR
        104 FORMAT(I10,2I8)
            WRITE(2,110) IT
        110 FORMAT (I5)
            STOP
             END
```

```
      C  RESTERENDE FOUTEN IN DE PERMANENT FILE "VAKKEN" WORDEN HERSTELD
      C  DOOR WIJZIGEN, TOEVOEGEN OF VERWIJDEREN VAN RECORDS
         PROGRAM MUVAK  (INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT,TAPE3)
         IMPLICIT INTEGER (A-Z)
         DIMENSION COOR(3),IND(1500),HULP(3),PUR(3)
         READ(1,100) IT,(PUR(I),I=1,3)
       100 FORMAT(I5,3I1)
         CALL OPENMS (3,IND,1500,0)
         WRITE(2,101)
       101 FORMAT (1H1)
         M=ML=0
         2 READ(1,102) NR,T,(COOR(I),I=1,3)
         IF(EOF(1)) 40,5
       102 FORMAT(I5,A1,I8,2I6)
         5 NR=NR+(M-ML)
         IF(T.EQ.1H ) GOTO 10
         IF(T.EQ.1H+) GOTO 20
         IF(T.EQ.1H-) GOTO 30
         WRITE(2,105) NR
       105 FORMAT(1H ,I5,3X,"OPGAVE FOUT")
         GOTO 2
        10 CALL WRITMS (3,COOR,3,NR)
         GOTO 2
        20 J=NR+1
         DO 201 I = J,IT
         K=(IT+J)-I
         CALL READMS (3,HULP,3,K)
         L=K+1
         CALL WRITMS (3,HULP,3,L)
       201 CONTINUE
         CALL WRITMS (3,COOR,3,J)
         M=M+1
         IT=IT+1
         GOTO 2
        30 J=NR+1
         DO 202 I = J,IT
         CALL READMS (3,HULP,3,I)
         L=I-1
         CALL WRITMS (3,HULP,3,L)
       202 CONTINUE
         CALL WRITMS (3,PUR,3,IT)
         ML=ML+1
         IT=IT-1
         GOTO 2
        40 DO 203 I = 1,IT
         CALL READMS (3,COOR,3,I)
       203 WRITE(2,104) COOR
       104 FORMAT(I10,2I8)
         WRITE(2,110) IT
       110 FORMAT (I5)
         STOP
          END
```

```
      C DE PUNTEN VAN HET WEGENNET WORDEN GESORTEERD OP X-COORD. EN PER
      C  GELIJKE X-COORD. OP Y-COORD.
      C  DE COORDINATEN VAN MEERMALEN GEMETEN PUNTEN WORDEN GEMIDDELD,
      C  WAARNA DE OORSPRONKELIJKE COORDINATEN WORDEN VERVANGEN DOOR DE
      C  GEMIDDELDE COORDINATEN.
      C  DE COORD. VAN DE BEDRIJVEN EN DE VAKKEN WORDEN VERGELEKEN MET DE
      C  COORDINATEN VAN HET WEGENNET EN BIJ GERINGE AFWIJKING VERVANGEN
      C  DOOR DEZE COORDINATEN. DOOR DEZE HANDELWIJZE KOMEN DE COORDINATEN
      C  VAN DE BEDRIJVEN EN DE VAKKEN ALTIJD OVEREEN MET EEN PUNT VAN HET
      C  WEGENNET.
      C  TENSLOTTE WORDEN DE COORDINATEN VAN HET WEGENNET, DE BEDRIJVEN
      C  EN DE VAKKEN VAN DE RA-FILES OVERGEBRACHT NAAR NORMALE FILES EN
      C  PERMANENT GEMAAKT ALS "WEGCO", "BEDRCO" EN "VAKCO".

         PROGRAM MIDDEL(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT,TAPE3,TAPE4,
        1 TAPE5,TAPE6,TAPE7,TAPE8)
         IMPLICIT INTEGER (A-Z)
         REAL GX,GY
         DIMENSION IC(2500),IN(2500),IL(2500),IND(2500),COOR(4),
        1 BEDRN(1500),VAKS(1500),BDR(3),VAK(3),PUNT(30,3),X(4),Y(4),BE(4),
        2 W(4),A(3)
         DIMENSION AWEG(6000)
         CALL OPENMS (3,IND,2500,0)
         CALL OPENMS(4,BEDRN,1500,0)
         CALL OPENMS (5,VAKS,1500,0)
         READ(1,5) NR,NS,NT
       5 FORMAT(3I5)
         N=0
         DO 10 I = 1,NR
         CALL READMS (3,COOR,4,I)
         N=N+1
         IC(N)=COOR(1)  $  IN(N)=COOR(2)  $  IL(N)=N
      10 CONTINUE
         ILR=1  $  IUR=N
         CALL SORRY (IC(1),IN(1),IL(1),ILR,IUR)
         IX=1
      12 JX=IX+1  $  IF(JX.GT.N) GOTO 20
         DO 14 IX = JX,N
         IF((IC(IX)-IC(IX-1)).GT.9) GOTO 16
      14 CONTINUE
         IX=N+1
      16 IF(IX.EQ.JX) GOTO 12
         ILR=JX-1  $  IUR=IX-1
         CALL SORRY (IN(1),IC(1),IL(1),ILR,IUR)
         GOTO 12
      20 IY=1
      32 IF(IY.GT.N) GOTO 50
         JY=IY+1  $  NP=1
         DO 34 IY = JY,N
         IF(IABS(IN(IY)-IN(IY-1)).GT.9.OR.IABS(IC(IY)-IC(IY-1)).GT.9)
        1 GOTO 35
      34 NP=NP+1
      35 IF(NP.EQ.1) GOTO 32
         NPP=0  $  L=IY-NP  $  K=IY-1
         DO 44 I = L,K
         NPP=NPP+1
         PUNT(NPP,1)=IC(I) $ PUNT(NPP,2)=IN(I) $ PUNT(NPP,3)=IL(I)
```

```
      44 CONTINUE
         GX=GY=0.
         DO 41 I = 1,NPP
         GX=GX+PUNT(I,1)
      41 GY=GY+PUNT(I,2)
         GX=GX/NPP $  GY=GY/NPP
         DO 42 I = 1,NPP
         PUNT(I,1)=IFIX(GX+.5)
      42 PUNT(I,2)=IFIX(GY+.5)
         DO 46 I = 1,NPP
         J=PUNT(I,3)
         CALL READMS (3,COOR,4,J)
         COOR(1)=PUNT(I,1) $ COOR(2)=PUNT(I,2)
      46 CALL WRITMS(3,COOR,4,J)
         GOTO 32
      50 IJ=0
         DO 52 IJK=1,N
         CALL READMS(3,COOR,4,IJK)
         DO 752 IKJ=1,4
         IJ=IJ+1
     752 AWEG(IJ)=COOR(IKJ)
      52 CONTINUE
         DO 60 I = 1,NS
         CALL READMS (4,BDR,3,I)
         IJ=0
         DO 56 J = 1,N
         DO 861 IJK=1,4
         IJ=IJ+1
     861 COOR(IJK)=AWEG(IJ)
         IF(IABS(COOR(1)-BDR(2)).GT.9.OR.IABS(COOR(2)-BDR(3)).GT.9) GOTO56
         BDR(2)=COOR(1) $ BDR(3)= COOR(2)
         CALL WRITMS (4,BDR,3,I)
         GOTO 60
      56 CONTINUE
         WRITE(2,104) BDR(1)
     104 FORMAT(1H ,I10,3X,"KOMT NIET VOOR IN WEGENNET")
      60 CONTINUE
         DO 70 K = 1,NT
         CALL READMS(5,VAK,3,K)
         IJ=0
         DO 64 L = 1,N
         DO 862 IJK=1,4
         IJ=IJ+1
     862 COOR(IJK)=AWEG(IJ)
         IF(IABS(COOR(1)-VAK(2)).GT.9.OR.IABS(COOR(2)-VAK(3)).GT.9) GOTO 64
         VAK(2)=COOR(1) $ VAK(3)=COOR(2)
         CALL WRITMS(5,VAK,3,K)
         GOTO 70
      64 CONTINUE
         WRITE(2,104) VAK(1)
      70 CONTINUE
         RNR2=0  $  NX=0
         NA=N+1
      76 WNR2=0
      78 NB=0
      80 NX=NX+1 $ NB=NB+1 $ IF(NX.EQ.NA) GOTO 84
         CALL READMS (3,COOR,4,NX)
```

```
115               X(NB)=COOR(1) $ Y(NB)=COOR(2) $ BE(NB)=COOR(3)
                  W(NB)=COOR(4)  $  IF(NB.EQ.4) GOTO 82
                  IF(COOR(3).NE.1HE) GOTO 80
                82 RNR2=RNR2+1  $  WNR2=WNR2+1
                  AA=10H(1(2I8,2X, $ A(2)=10HA1,A1),T83 $ A(3)=7H,I4,I2)
120               AB=10H(2(2I8,2X, $ AC=10H(3(2I8,2X, $ AD=10H(4(2I8,2X,
                  IF(NB.EQ.1) A(1)=AA $ IF(NB.EQ.2) A(1)=AB
                  IF(NB.EQ.3) A(1)=AC $ IF(NB.EQ.4) A(1)=AD
                  WRITE(6,A)(X(I),Y(I),BE(I),W(I),I=1,NB),RNR2,WNR2
                  IF(COOR(3).EQ.1HE) GOTO 76
125               GOTO 78
                84 ENDFILE6  $  REWIND6
                  DO 96 I = 1,NS
                  CALL READMS (4,BDR,3,I)
                96 WRITE(7,122) BDR
130            122 FORMAT (3I8)
                  DO 98 I = 1,NT
                  CALL READMS (5,VAK,3,I)
                98 WRITE(8,122) VAK
                  ENDFILE7 $ ENDFILE8 $ REWIND7 $ REWIND8
135               CALL EXIT
                  END
```

```
      SUBROUTINE SORRY (IK,IK1,IL,ILR,IUR)
      IMPLICIT INTEGER (A-Z)
      DIMENSION LST(25),IND(2500),IK(2500),IK1(2500),IL(2500)
      DO 10 I = ILR,IUR
   10 IND(I) = I
      IS=0  $  LL=ILR  $  UL=IUR
  100 IP=LL $ M=IND(IP) $ X=IK(M)
      IQ=UL $ N=IND(IQ) $ Z=IK(N)
         IF (X.LE.Z) GOTO 110
         K=M  $  Y=X
         IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y
  110    IF ((UL-LL).LE.1) GOTO 600
            XX=X  $  IX=IP  $  ZZ=Z  $  IZ=IQ
  200       IP=IP+1  $  IF (IP.GE.IQ) GOTO 210
      M=IND(IP) $ X=IK(M) $ IF(X-XX) 200,300,300
  210       IP=IQ-1  $  GOTO 500
  300       IQ=IQ-1  $  IF (IQ.LE.IP) GOTO 310
      N=IND(IQ) $ Z=IK(N) $ IF(ZZ-Z) 300,400,400
  310 IP=IP-1 $ Z=X $ N=M $ M=IND(IP) $ X=IK(M)
  400       IF (X.LE.Z) GOTO 410
            K=M  $  Y=X
            IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y
  410       IF (X.LE.XX) GOTO 420
            XX=X  $  IX=IP
  420       IF (Z.GE.ZZ) GOTO 430
            ZZ=Z  $  IZ=IQ
  430       GOTO 200
  500       IF (IP.LE.IX .OR. X.GE.XX) GOTO 510
            IND(IP)=IND(IX)  $  IND(IX)=M
  510       IF (IQ.GE.IZ .OR. Z.LE.ZZ) GOTO 520
            IND(IQ)=IND(IZ)  $  IND(IZ)=N
  520       IF ((UL-IQ).GT.(IP-LL)) GOTO 530
            IX=LL  $  IZ=IP-1  $  LL=IQ+1  $  GOTO 540
  530       IX=IQ+1  $  IZ=UL  $  UL=IP-1
  540       IF (IX.GE.IZ) GOTO 550
            IS=IS+2  $  LST(IS-1)=IX  $  LST(IS)=IZ
  550       IF (UL.GT.LL) GOTO 100
  600    IF (IS.EQ.0) GOTO 700
         LL=LST(IS-1)  $  UL=LST(IS)  $  IS=IS-2  $  GOTO 100
  700 CONTINUE
      DO 120  I= ILR, IUR
         IF (IND(I)) 120,150,125
  125    K=I
      CO=IK(K) $ COP=IK1(K) $ COPY=IL(K)
  130    J=IND(K)  $  IF (J - I) 150,140,225
  225 IK(K)=IK(J) $ IK1(K)=IK1(J) $ IL(K)=IL(J)
         IND(K)= -1 $ K=J
      GOTO 130
  140 IK(K)=CO $ IK1(K)=COP $ IL(K)=COPY
         IND(K)= -1
  120 CONTINUE
      RETURN
  150    STOP "PERMERROR"
      END
```

```
      PROGRAM TOEDEL (INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT,TAPE3,TAPE4,
     1TAPE5,TAPE6,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12,TAPE13,TAPE14,
     2TAPE15,TAPE16,TAPE17,TAPE18,TAPE19,TAPE20)
      INTEGER FOUT
      DIMENSION INDEX(4000),MB(50),MK(50),OTO(50),OT(50),MA(50),MR(50),
     1 MV(50),MU(50),IRX(500),IAA(11),IBB(11),ICC(11),IAAA(10),
     2 STO(11,9),STOS(11,9),TOP(11,9),TOH(11,9),KAK(11,9),KAB(11,9),
     3 OBB1(11,9),KABB1(11,9),OBB2(11,9),KABB2(11,9),OBB(11,9),
     4 KABB(11,9),GKA(11,9),GKAS(11,9),AKB(11,9),GKO(11,9),POH(11,9),
     5 GBO(11,9),POHB(11,9),PAB(11,9),POHB1(11,9),PAB1(11,9),
     6 POHB2(11,9),PAB2(11,9),IAB(11),IBC(11),ICD(11),IKP(4000),
     7 LU(4000),IKN(4000),IBTT(8),LAB(11),LBC(11),LCD(11)
      COMMON /INV/ IK(4000),IK1(4000),R(4000) /SORT/ ILR,IUR
      EQUIVALENCE (INDEX,IK)
C     INLEZEN PONSKAARTEN : NAAM GEBIED, VAKNUMMERS,AANV GEGEVENS,
C     BEDRIJFSTYPEN EN VERPLAATSINGEN
      READ(1,10)  NDA,IA,IB,IC,IZ
   10 FORMAT (I2,3A10,I5)
      M=0
   20 M=M+1
      READ(1,30)  IVN,IRN
      IF(EOF(1)) 50,40
   30 FORMAT(I5,I3)
   40 IRX(M)=IVN*100+IRN  $  GOTO 20
   50 MIRT=M-1  $  IVM=IVN
      N=0
   60 N=N+1
      READ(1,70)  I1,K1,O1,IV,IU
      IF(EOF(1)) 90,80
   70 FORMAT (I8,I3,F9.4,I3,I1)
   80 IK(N)=I1*100+K1  $  IK1(N)=IV*10+IU  $  R(N)=O1  $  GOTO 60
   90 NT=N-1  $  ILR=1  $  IUR=NT
      CALL INDCOL
      DO 100 N = 1,NT
      IV=IK1(N)/10  $  IU=IK1(N)-IV*10  $  O1=R(N)
  100 WRITE(4,110) IK(N),IV,IU,O1
  110 FORMAT(I11,I4,I2,F9.4)
      ENDFILE 4  $  REWIND 4
      M=0
  120 M=M+1
      READ(1,130) IAA(M),IBB(M),ICC(M)
      IF(EOF(1)) 140,120
  130 FORMAT (3A10)
  140 IS=M-1
      DO 150 M = 3,IS
  150 READ(1,130) IAB(M),IBC(M),ICD(M)
      IST=IS-3
      DO 160 M = 1,IST
  160 READ (1,170) IBTT(M)
  170 FORMAT (A10)
      DO 180 I = 1,10
  180 IAAA(I)=10H----------
      J=0
  190 J=J+1
      READ(1,200) IKH,IKK,OG,IRH,IVH,IUH
      IF(EOF(1)) 220,210
  200 FORMAT(I8,I3,F9.4,I3,I4,I2)
```

Bijlage 14.1

```
      210 WRITE(10,200) IKH,IKK,OG,IRH,IVH,IUH                                  TOEDNCI    59
          GOTO 190                                                              TOEDNCI    60
60    220 JA=J-1  $  REWIND 10                                                  TOEDNCI    61
     C  OVERNEMEN INPUT C1, SORTEREN OP BEDRIJFS- EN KAVELNUMMER, KAVELS        TOEDNCI    62
     C  BUITEN HET BLOK VERWIJDEREN                                            TOEDNCI    63
          M=0  $  NDG=1                                                         TOEDNCI    64
      230 M=M+1                                                                 TOEDNCI    65
65        READ(17,240) ND,I,K,IO,IVC,IGG,LV,LS,LO,LW,LL,IV,IU                   TOEDNCI    66
          IF(EOF(17)) 270,250                                                   TOEDNCI    67
      240 FORMAT(I2,I8,2X,I2,8X,I7,47X,I2,1X,I1,19X,I3,4I2,25X,I3,I1)           TOEDNCI    68
      250 IF(IVC.GE.NDA.OR.IV.EQ.900) GOTO 260                                  TOEDNCI    69
          IF(ND.GT.NDG) NDG=ND                                                  TOEDNCI    70
70        O=IO*1./10000.                                                        TOEDNCI    71
          IK(M)=I*100+K  $  IK1(M)=LV*10**8+LS*10**6+LO*10**4+LW*100+LL         TOEDNCI    72
          R(M)=ND*10**7+IV*10**4+IU*1000+O  $  GOTO 230                         TOEDNCI    73
      260 M=M-1  $  GOTO 230                                                    TOEDNCI    74
      270 MT=M-1  $  ILR=1  $  IUR=MT                                           TOEDNCI    75
75        CALL INDCOL                                                           TOEDNCI    76
          DO 280 N = 1,MT                                                       TOEDNCI    77
          JL=IK1(N)  $  LV=JL/10**8  $  JL=JL-LV*10**8  $  LS=JL/10**6         TOEDNCI    78
          JL=JL-LS*10**6  $  LO=JL/10**4  $  JL=JL-LO*10**4  $  LW=JL/100       TOEDNCI    79
          LL=JL-LW*100  $  BY=R(N)  $  ND=BY/10**7  $  BY=BY-ND*10**7          TOEDNCI    80
80        IV=BY/10**4  $  BY=BY-IV*10**4  $  IU=BY/1000  $  O=BY-IU*1000       TOEDNCI    81
      280 WRITE(3,290) ND,IK(N),O,LV,LS,LO,LW,LL,IV,IU                          TOEDNCI    82
      290 FORMAT(I3,I10,F9.4,I4,4I3,I4,I2)                                      TOEDNCI    83
          ENDFILE 3  $  REWIND 3                                                TOEDNCI    84
          WRITE(2,300)                                                          TOEDNCI    85
85    300 FORMAT(1H1,"KAVELS KOMEN NIET VOOR BIJ INPUT CI"//3X,"BEDRNR KAV V TOEDNCI    86
         1AK"/)                                                                 TOEDNCI    87
     C  FOUTIEVE KAVELNUMMERS BIJ AANV GEGEVENS OPZOEKEN                        TOEDNCI    88
          DO 310 L = 1,NT                                                       TOEDNCI    89
      310 READ(4,320)  IK1(L)                                                   TOEDNCI    90
90    320 FORMAT(I11)                                                           TOEDNCI    91
          REWIND 4                                                              TOEDNCI    92
          NTT=0                                                                 TOEDNCI    93
          DO 360 N = 1,NT                                                       TOEDNCI    94
          READ(4,330) IV                                                        TOEDNCI    95
95    330 FORMAT(11X,I4)                                                        TOEDNCI    96
          DO 340 M = 1,MT                                                       TOEDNCI    97
          IF(IK1(N).EQ.IK(M)) GOTO 360                                          TOEDNCI    98
      340 CONTINUE                                                              TOEDNCI    99
          JM=IK1(N)  $  I1=JM/100  $  K1=JM-I1*100                              TOEDNCI   100
100       WRITE(2,350) I1,K1,IV                                                 TOEDNCI   101
      350 FORMAT(1H ,I8,2I4)                                                    TOEDNCI   102
          NTT=1                                                                 TOEDNCI   103
      360 CONTINUE                                                              TOEDNCI   104
          M=N=IVF=0  $  REWIND 4                                                TOEDNCI   105
105       IF(NTT.EQ.1) GOTO 2380                                                TOEDNCI   106
     C  INPUT CI EN AANV GEGEVENS SAMENVOEGEN                                   TOEDNCI   107
          DO 370 L = 1,NT                                                       TOEDNCI   108
      370 READ(4,320)  IK(L)                                                    TOEDNCI   109
          REWIND 4                                                              TOEDNCI   110
110   380 N=N+1                                                                 TOEDNCI   111
          READ(4,110) IK(N),IV1,IU1,O1                                          TOEDNCI   112
      390 M=M+1                                                                 TOEDNCI   113
          IF(M.GT.MT) GOTO 540                                                  TOEDNCI   114
          READ(3,290) ND,IK1(M),O,LV,LS,LO,LW,LL,IV,IU                          TOEDNCI   115
```

```
115             IF(IU.LT.2) GOTO 400                                                TOEDNCI  116
                IF(IK(N).NE.IK1(M)) GOTO 510                                        TOEDNCI  117
                IF(IK(N).EQ.IK(N+3)) GOTO 420                                       TOEDNCI  118
                IF(IK(N).EQ.IK(N+2)) GOTO 450                                       TOEDNCI  119
                IF(IK(N).EQ.IK(N+1)) GOTO 480                                       TOEDNCI  120
120         400 I=IK1(M)/100  $  K=IK1(M)-I*100                                     TOEDNCI  121
                IF(IU.EQ.1)  IU=2  $  IF(IU.EQ.0) IU=1  $  IF(K.EQ.1) IU=2          TOEDNCI  122
                WRITE(11,410) ND,I,K,O,LV,LS,LO,LW,LL,IV,IU                         TOEDNCI  123
            410 FORMAT(I2,I8,I3,F9.4,I4,4I3,I4,I2)                                  TOEDNCI  124
                GOTO 390                                                            TOEDNCI  125
125         420 LE=N  $  LK=N+3                                                     TOEDNCI  126
                DO 440 L = LE,LK                                                    TOEDNCI  127
                IF(IV1.EQ.900) GOTO 430                                             TOEDNCI  128
                I1=IK(N)/100  $  K1=IK(N)-I1*100                                    TOEDNCI  129
                IF(IU1.EQ.1) IU1=2  $  IF(IU1.EQ.0) IU1=1                           TOEDNCI  130
130             WRITE(11,410) ND,I1,K1,O1,LV,LS,LO,LW,LL,IV1,IU1                    TOEDNCI  131
            430 IF(N.EQ.LK) GOTO 440                                                TOEDNCI  132
                N=N+1                                                               TOEDNCI  133
                READ(4,110) IK(N),IV1,IU1,O1                                        TOEDNCI  134
            440 CONTINUE                                                            TOEDNCI  135
135             GOTO 530                                                            TOEDNCI  136
            450 LE=N  $  LK=N+2                                                     TOEDNCI  137
                DO 470 L = LE,LK                                                    TOEDNCI  138
                IF(IV1.EQ.900) GOTO 460                                             TOEDNCI  139
                I1=IK(N)/100  $  K1=IK(N)-I1*100                                    TOEDNCI  140
140             IF(IU1.EQ.1) IU1=2  $  IF(IU1.EQ.0) IU1=1                           TOEDNCI  141
                WRITE(11,410) ND,I1,K1,O1,LV,LS,LO,LW,LL,IV1,IU1                    TOEDNCI  142
            460 IF(N.EQ.LK) GOTO 470                                                TOEDNCI  143
                N=N+1                                                               TOEDNCI  144
                READ(4,110) IK(N),IV1,IU1,O1                                        TOEDNCI  145
145         470 CONTINUE                                                            TOEDNCI  146
                GOTO 530                                                            TOEDNCI  147
            480 LE=N  $  LK=N+1                                                     TOEDNCI  148
                DO 500 L = LE,LK                                                    TOEDNCI  149
                IF(IV1.EQ.900) GOTO 490                                             TOEDNCI  150
150             I1=IK(N)/100  $  K1=IK(N)-I1*100                                    TOEDNCI  151
                IF(IU1.EQ.1) IU1=2  $  IF(IU1.EQ.0) IU1=1                           TOEDNCI  152
                WRITE(11,410) ND,I1,K1,O1,LV,LS,LO,LW,LL,IV1,IU1                    TOEDNCI  153
            490 IF(N.EQ.LK) GOTO 500                                                TOEDNCI  154
                N=N+1                                                               TOEDNCI  155
155             READ(4,110) IK(N),IV1,IU1,O1                                        TOEDNCI  156
            500 CONTINUE                                                            TOEDNCI  157
                GOTO 530                                                            TOEDNCI  158
            510 IVF=1                                                               TOEDNCI  159
                WRITE(2,520) IK(N),IV1                                              TOEDNCI  160
160         520 FORMAT(1H ,"KAVEL",I12," IN VAK",I5," KOMT NIET VOOR BIJ KAVELGEGE  TOEDNCI  161
               1VENS")                                                              TOEDNCI  162
            530 IF(N,LT,NT) GOTO 380                                                TOEDNCI  163
                GOTO 390                                                            TOEDNCI  164
            540 ENDFILE 11  $  REWIND 3  $  REWIND 4  $  REWIND 11                  TOEDNCI  165
165             IF(IVF.EQ.1) GOTO 2380                                              TOEDNCI  166
          C  RUILKLASSE PER KAVEL EN GROOTTEKLASSE PER BEDRIJF BEREKENEN            TOEDNCI  167
                M=FOUT=OP=LBD=MAB=MTT=0                                             TOEDNCI  168
            550 M=M+1                                                               TOEDNCI  169
                READ(11,410) ND,IK(M),K,O,LV,LS,LO,LW,LL,IV,IU                      TOEDNCI  170
170             IF(EOF(11)) 580,560                                                 TOEDNCI  171
            560 IF(M.EQ.1) GOTO 570                                                 TOEDNCI  172
```

```
      IF(IK(M).NE.IK(M-1)) GOTO 590                                     TOEDNCI  173
    570  OP= OP+0                                                       TOEDNCI  174
      GOTO 640                                                          TOEDNCI  175
175 580 LBD=1                                                           TOEDNCI  176
    590 IF( OP.LT.5.) IBO=1  $ IF( OP.GE.5..AND. OP.LT.10.) IBO=2       TOEDNCI  177
      IF( OP.GE.10..AND. OP.LT.15.) IBO=3                               TOEDNCI  178
      IF( OP.GE.15..AND. OP.LT.20.) IBO=4                               TOEDNCI  179
      IF( OP.GE.20..AND. OP.LT.30.) IBO=5                               TOEDNCI  180
180   IF( OP.GE.30..AND. OP.LT.40.) IBO=6                               TOEDNCI  181
      IF( OP.GE.40..AND. OP.LT.50.) IBO=7  $  IF( OP.GE.50.) IBO=8      TOEDNCI  182
      READ(19,600) NE,IKB,IBT                                           TOEDNCI  183
    600 FORMAT(I2,I10,I3)                                               TOEDNCI  184
      IF(IK(M-1).NE.IKB) GOTO 620                                       TOEDNCI  185
185   MAB=MAB+1                                                         TOEDNCI  186
      WRITE(20,610) NE,IKB,IBT,IBO                                      TOEDNCI  187
    610 FORMAT(I2,I10,2I3)                                              TOEDNCI  188
       OP=0  $   OP= OP+0  $  GOTO 640                                  TOEDNCI  189
    620 WRITE(2,630) IK(M-1)                                            TOEDNCI  190
190 630 FORMAT(1H ,"BEDRIJFSNUMMER",I10," KOMT NIET VOOR OP LIJST MET BEDR TOEDNCI  191
     1IJFSNUMMERS")                                                     TOEDNCI  192
       OP=0  $   OP= OP+0  $  FOUT=1                                    TOEDNCI  193
    640 IF(LBD.EQ.1) GOTO 690                                           TOEDNCI  194
      DO 650 L = 1,MIRT                                                 TOEDNCI  195
195   IVN=IRX(L)/100  $  IRN=MOD(IRX(L),100)                            TOEDNCI  196
      IF(IV.EQ.IVN) GOTO 670                                            TOEDNCI  197
    650 CONTINUE                                                        TOEDNCI  198
      WRITE(2,660) IV,IK(M),K                                           TOEDNCI  199
    660 FORMAT(1H ,"VAKNUMMER",I5," VAN KAVEL",I10,I3," KOMT NIET VOOR OP  TOEDNCI  200
200  1LIJST MET VAKNUMMERS")                                            TOEDNCI  201
      FOUT=1                                                            TOEDNCI  202
      IF(LBD.EQ.1) GOTO 690                                             TOEDNCI  203
      GOTO 550                                                          TOEDNCI  204
    670 IR=IRN                                                          TOEDNCI  205
205   WRITE(5,680) IK(M),K,O,LV,LS,LO,LW,LL,IR,IV,IU                    TOEDNCI  206
    680 FORMAT(I8,I3,F9.4,I4,5I3,I4,I2)                                 TOEDNCI  207
      MTT=MTT+1                                                         TOEDNCI  208
      GOTO 550                                                          TOEDNCI  209
    690 IF(FOUT.EQ.1) GOTO 2380                                         TOEDNCI  210
210   ENDFILE 5  $  ENDFILE 20  $  REWIND 5  $  REWIND 20               TOEDNCI  211
      WRITE(2,700)  IA,IB,IC,NDA                                        TOEDNCI  212
    700 FORMAT(1H1,"RUILVERKAVELING :",3A10//" EERSTE DORPSBEHOREN MET BUI TOEDNCI  213
     1TEN BLOK BEDRIJVEN:",I5)                                          TOEDNCI  214
      DO 710  LI = 1,MTT                                                TOEDNCI  215
215   READ(5,680) IKP(LI),IKN(LI),R(LI),LV,LS,LO,LW,LL,IR,IV,IU         TOEDNCI  216
    710 LU(LI)=LV*10**8+LS*10**6+LO*10**4+LW*100+LL                     TOEDNCI  217
      REWIND 5                                                          TOEDNCI  218
C   BEREKENEN VERKAVELINGSKENMERKEN                                     TOEDNCI  219
      ITEL=NPB=0  $  CALL PRINT1(NPB,ITEL,IA,IB,IC)                     TOEDNCI  220
220   NDGA=NDG+1  $  NDAT=NDA+1                                         TOEDNCI  221
      DO 1170 I = 1,NDGA                                                TOEDNCI  222
      DO 720 J = 1,IS                                                   TOEDNCI  223
      DO 720 N = 1,9                                                    TOEDNCI  224
      STO(J,N)=STOS(J,N)=TOP(J,N)=KAK(J,N)=KAB(J,N)=TOH(J,N)=OBB(J,N)=  TOEDNCI  225
225  1KABB(J,N)=OBB1(J,N)=OBB2(J,N)=KABB1(J,N)=KABB2(J,N)=0             TOEDNCI  226
      AKB(J,N)=GBO(J,N)=GKO(J,N)=POH(J,N)=GKA(J,N)=GKAS(J,N)=PAB1(J,N)=  TOEDNCI  227
     1POHB1(J,N)=PAB2(J,N)=POHB2(J,N)=PAB(J,N)=POHB(J,N)=0              TOEDNCI  228
    720 CONTINUE                                                        TOEDNCI  229
```

```
                SBO=SBOS=KAKB=TOPB=TOHB=OMAX=0                                  TOEDNCI   230
230             DO 900 K= 1,IS                                                  TOEDNCI   231
                LA=0                                                            TOEDNCI   232
                DO 890 M = 1,MAB                                                TOEDNCI   233
                LC=0                                                            TOEDNCI   234
                READ(20,610) ND,I1,IBT,IBO                                      TOEDNCI   235
235             LBB=1H                                                          TOEDNCI   236
                IF(I.EQ.1) GOTO 730                                             TOEDNCI   237
                IF(ND.NE.(I-1)) GOTO 890                                        TOEDNCI   238
            730 IF(K.EQ.1) GOTO 750                                             TOEDNCI   239
                IF(I.GE.2) GOTO 740                                             TOEDNCI   240
240             IF(ND.GE.NDA) GOTO 890                                          TOEDNCI   241
            740 IF(K.EQ.2) GOTO 750                                             TOEDNCI   242
                IF(K.EQ.3.AND.IBT.EQ.IST) GOTO 890                              TOEDNCI   243
                IF(K.EQ.3) GOTO 750                                             TOEDNCI   244
                IF(IBT.NE.(K-3)) GOTO 890                                       TOEDNCI   245
245         750 DO 880 L = 1,MTT                                                TOEDNCI   246
                LA=LA+1                                                         TOEDNCI   247
                IK(LA)=IKP(L)                                                   TOEDNCI   248
                IF(IK(LA).NE.I1) GOTO 860                                       TOEDNCI   249
                IK1(LA)=IKN(L)  $  O=R(L)  $  ITL=LO(L)                         TOEDNCI   250
250             LC=1                                                            TOEDNCI   251
                                        LV=ITL/10**8 $ ITL=ITL-LV*10**8         TOEDNCI   252
                LS=ITL/10**6 $ ITL=ITL-LS*10**6 $ LO=ITL/10**4 $ ITL=ITL-LO*10**4 TOEDNCI 253
                LW=ITL/100 $ LL=ITL-LW*100                                      TOEDNCI   254
                V=FLOAT(LV) $ S=FLOAT(LS) $ OV=FLOAT(LO) $ W=FLOAT(LW)          TOEDNCI   255
255             A=FLOAT(LL)                                                     TOEDNCI   256
                STO(K,9)=STO(K,9)+O*(V+S+OV+W+A)                                TOEDNCI   257
                STO(K,IBO)=STO(K,IBO)+O*(V+S+OV+W+A)                            TOEDNCI   258
                STOS(K,9)=STOS(K,9)+O*(V+S*1.5+OV*2.2+W*4.+A*3.)                TOEDNCI   259
                STOS(K,IBO)=STOS(K,IBO)+O*(V+S*1.5+OV*2.2+W*4.+A*3.)            TOEDNCI   260
260             TOP(K,9)=TOP(K,9)+O $ TOP(K,IBO)=TOP(K,IBO)+O                   TOEDNCI   261
                IF(IK1(LA).EQ.1) TOH(K,9)=TOH(K,9)+O                            TOEDNCI   262
                IF(IK1(LA).EQ.1) TOH(K,IBO)=TOH(K,IBO)+O                        TOEDNCI   263
                IF(LA.EQ.1) GOTO 760                                            TOEDNCI   264
                IF(IK1(LA).EQ.IK1(LA-1).AND.IK(LA).EQ.IK(LA-1)) GOTO 765        TOEDNCI   265
265         760 KAKB=KAKB+1                                                     TOEDNCI   266
                KAK(K,9)=KAK(K,9)+1  $  KAK(K,IBO)=KAK(K,IBO)+1                 TOEDNCI   267
                GOTO 767                                                        TOEDNCI   268
            765 OE=OE+O  $  GOTO 770                                            TOEDNCI   269
            767 OE=O                                                            TOEDNCI   270
270         770 SBO=SBO+O*(V+S+OV+W+A)                                          TOEDNCI   271
                SBOS=SBOS+O*(V+S*1.5+OV*2.2+W*4.+A*3.)                          TOEDNCI   272
                TOPB=TOPB+O  $  IF(IK1(LA).EQ.1) GOTO 780                       TOEDNCI   273
                IF(OE.GT.OMAX) OMAX=OE  $  GOTO 790                             TOEDNCI   274
            780 TOHB=TOHB+O                                                     TOEDNCI   275
275         790 IF(L.EQ.MTT) GOTO 800                                           TOEDNCI   276
                GOTO 880                                                        TOEDNCI   277
            800 LA=LA-1                                                         TOEDNCI   278
                KAB(K,9)=KAB(K,9)+1  $  KAB(K,IBO)=KAB(K,IBO)+1                 TOEDNCI   279
                KAG=SBO/TOPB+.5  $  KAGS=SBOS/TOPB+.5                           TOEDNCI   280
280             PH=TOHB/TOPB*100.  $  PV=OMAX/TOPB*100.                         TOEDNCI   281
                IF(TOHB/TOPB.LT.1) GOTO 810                                     TOEDNCI   282
                OBB1(K,9)=OBB1(K,9)+TOPB  $  OBB1(K,IBO)=OBB1(K,IBO)+TOPB       TOEDNCI   283
                KABB1(K,9)=KABB1(K,9)+1  $  KABB1(K,IBO)=KABB1(K,IBO)+1         TOEDNCI   284
            810 IF(TOHB/TOPB.LT..8) GOTO 820                                    TOEDNCI   285
285             OBB2(K,9)=OBB2(K,9)+TOPB  $  OBB2(K,IBO)=OBB2(K,IBO)+TOPB       TOEDNCI   286
```

```
      KABB2(K,9)=KABB2(K,9)+1  $  KABB2(K,IBO)=KABB2(K,IBO)+1          TOEDNCI  287
  820 IF(TOHB/TOPB.LT..6) GOTO 830                                     TOEDNCI  288
      OBB(K,9)=OBB(K,9)+TOPB  $  OBB(K,IBO)=OBB(K,IBO)+TOPB            TOEDNCI  289
      KABB(K,9)=KABB(K,9)+1  $  KABB(K,IBO)=KABB(K,IBO)+1              TOEDNCI  290
  830 IF(I.GT.1.OR.K.GT.1) GOTO 850                                    TOEDNCI  291
      IF(ND.GE.NDA) LBB=1HX                                            TOEDNCI  292
      NPB=NPB+1  $  CALL PRINT1(NPB,ITEL,IA,IB,IC)                     TOEDNCI  293
      WRITE(2,840) I1,KAKB,KAG,KAGS,TOPB,TOHB,PH,PV,IBTT(IBT),LBB      TOEDNCI  294
  840 FORMAT(1H ,15X,I8,I6,2I9,F12.4,F8.2,2F9.1,4X,A10,4X,A1)          TOEDNCI  295
  850 KAKB=SBO=SBOS=TOHB=TOPB=OMAX=0                                   TOEDNCI  296
      IF(L.EQ.MTT) GOTO 880                                            TOEDNCI  297
      GOTO 890                                                         TOEDNCI  298
  860 IF(LC.NE.1) GOTO 870                                             TOEDNCI  299
      GOTO 800                                                         TOEDNCI  300
  870 LA=LA-1                                                          TOEDNCI  301
  880 CONTINUE                                                         TOEDNCI  302
  890 CONTINUE                                                         TOEDNCI  303
      REWIND 20                                                        TOEDNCI  304
  900 CONTINUE                                                         TOEDNCI  305
      DO 930 KK = 1,IS                                                 TOEDNCI  306
      DO 930 LL = 1,9                                                  TOEDNCI  307
      IF(TOP(KK,LL).EQ.0) GOTO 910                                     TOEDNCI  308
      GKA(KK,LL)=STO(KK,LL)/TOP(KK,LL)                                 TOEDNCI  309
      GKAS(KK,LL)=STOS(KK,LL)/TOP(KK,LL)                               TOEDNCI  310
      POH(KK,LL)=(TOH(KK,LL)/TOP(KK,LL))*100.                          TOEDNCI  311
      POHB(KK,LL)=(OBB(KK,LL)/TOP(KK,LL))*100.                         TOEDNCI  312
      POHB1(KK,LL)=(OBB1(KK,LL)/TOP(KK,LL))*100.                       TOEDNCI  313
      POHB2(KK,LL)=(OBB2(KK,LL)/TOP(KK,LL))*100.                       TOEDNCI  314
  910 IF(KAB(KK,LL).EQ.0) GOTO 920                                     TOEDNCI  315
      PAB2(KK,LL)=(FLOAT(KABB2(KK,LL))/FLOAT(KAB(KK,LL)))*100.        TOEDNCI  316
      PAB1(KK,LL)=(FLOAT(KABB1(KK,LL))/FLOAT(KAB(KK,LL)))*100.        TOEDNCI  317
      PAB(KK,LL)=(FLOAT(KABB(KK,LL))/FLOAT(KAB(KK,LL)))*100.          TOEDNCI  318
      GBO(KK,LL)=TOP(KK,LL)/KAB(KK,LL)                                 TOEDNCI  319
      AKB(KK,LL)=FLOAT(KAK(KK,LL))/FLOAT(KAB(KK,LL))                   TOEDNCI  320
  920 IF(KAK(KK,LL).EQ.0) GOTO 930                                     TOEDNCI  321
      GKO(KK,LL)=TOP(KK,LL)/KAK(KK,LL)                                 TOEDNCI  322
  930 CONTINUE                                                         TOEDNCI  323
C  PRINTEN VERKAVELINGSKENMERKEN                                       TOEDNCI  324
      DO 1160 N = 1,IS                                                 TOEDNCI  325
      IF(I.GT.1.AND.N.EQ.2) GOTO 1160                                  TOEDNCI  326
      IF(TOP(N,9).EQ.0) GOTO 1160                                      TOEDNCI  327
      III=I+1                                                          TOEDNCI  328
      IF(I.LT.NDAT.OR.N.EQ.1) GOTO 940                                 TOEDNCI  329
      LAB(N)=IAB(N)  $  LBC(N)=IBC(N)  $  LCD(N)=ICD(N)                TOEDNCI  330
      GOTO 950                                                         TOEDNCI  331
  940 LAB(N)=IAA(N) $ LBC(N)=IBB(N) $ LCD(N)=ICC(N)                    TOEDNCI  332
  950 IF(I.GT.1) GOTO 970                                              TOEDNCI  333
      WRITE(2,960) III,N,IA,IB,IC,LAB(N),LBC(N),LCD(N)                 TOEDNCI  334
  960 FORMAT(1H1//16X,"TABEL",I3,".",12,2X,"GEHELE RUILVERKAVELING"//16X TOEDNCI  335
     1,"RUILVERKAVELING :",3A10//16X,"VERKAVELINGSKENMERKEN BESTAANDE TO TOEDNCI  336
     2ESTAND"//16X,3A10)                                               TOEDNCI  337
      GOTO 990                                                         TOEDNCI  338
  970 IIA=I-1                                                          TOEDNCI  339
      WRITE(2,980) III,N,IIA,IA,IB,IC,LAB(N),LBC(N),LCD(N)             TOEDNCI  340
  980 FORMAT(1H1//16X,"TABEL",I3,".",I2,2X,"DORPSBEHOREN",I3//16X,"RUILV TOEDNCI  341
     1ERKAVELING :",3A10//16X,"VERKAVELINGSKENMERKEN BESTAANDE TOESTAND" TOEDNCI  342
     2//16X,3A10)                                                      TOEDNCI  343
```

```
      WRITE(2,1000) IAAA,IAAA                                                  TOEDNCI   344
 1000 FORMAT(1HO,15X,10A10/16X,"BEDRIJFSOPPERVLAKTE IN HA     < 5     5- TOEDNCI   345
     110   10-15   15-20   20-30   30-40   40-50   > 50    TOTAAL "/16X, TOEDNCI   346
     210A10/)                                                                  TOEDNCI   347
      WRITE(2,1010)(KAB(N,L),L=1,9)                                            TOEDNCI   348
 1010 FORMAT(1H ,15X,"AANTAL BEDRIJVEN",11X,9(I6,2X)/)                         TOEDNCI   349
      WRITE(2,1020)(KAK(N,L),L=1,9)                                            TOEDNCI   350
 1020 FORMAT(1H ,15X,"AANTAL KAVELS",14X,9(I6,2X)/)                            TOEDNCI   351
      WRITE(2,1030)(TOP(N,L),L=1,9)                                            TOEDNCI   352
 1030 FORMAT(1H ,15X,"OPP CULTUURGROND IN HA",5X,9(F8.1)/)                     TOEDNCI   353
      WRITE(2,1040)(AKB(N,L),L=1,9)                                            TOEDNCI   354
 1040 FORMAT(1H ,15X,"AANTAL KAVELS PER BEDRIJF",2X,9(F8.1)/)                  TOEDNCI   355
      WRITE(2,1050)(GBO(N,L),L=1,9)                                            TOEDNCI   356
 1050 FORMAT(1H ,15X,"GEM BEDRIJFSOPP IN HA",6X,9(F8.1)/)                      TOEDNCI   357
      WRITE(2,1060)(GKO(N,L),L=1,9)                                            TOEDNCI   358
 1060 FORMAT(1H ,15X,"GEM KAVELOPP IN HA",9X,9(F8.1)/)                         TOEDNCI   359
      WRITE(2,1070)(TOH(N,L),L=1,9)                                            TOEDNCI   360
 1070 FORMAT(1H ,15X,"OPP INGENOMEN DOOR HUISKAV",1X,9(F8.1)/)                 TOEDNCI   361
      WRITE(2,1080)(POH(N,L),L=1,9)                                            TOEDNCI   362
 1080 FORMAT(1H ,15X,"PERC VAN DE OPP INGENOMEN"/16X,"DOOR HUISKAVELS",        TOEDNCI   363
     112X,9(F8.1)/)                                                            TOEDNCI   364
      WRITE(2,1090)(GKA(N,L),L=1,9)                                            TOEDNCI   365
 1090 FORMAT(1H ,15X,"GEM KAVELAFSTAND IN HM",5X,9(F8.1)/)                     TOEDNCI   366
      WRITE(2,1100)(GKAS(N,L),L=1,9)                                           TOEDNCI   367
 1100 FORMAT(1H ,15X,"GEM SCHIJNB KAVAFST IN HM",2X,9(F8.1)/)                  TOEDNCI   368
      IF(I.GE.NDAT) GOTO 1160                                                  TOEDNCI   369
      WRITE(2,1110)(KABB1(N,L),L=1,9)                                          TOEDNCI   370
 1110 FORMAT(1HO,15X,"BEDRIJVEN WAARVAN DE HUISKAVEL 100 PROC VAN DE BED TOEDNCI   371
     1ROPP UITMAAKT :"//16X,"AANTAL BEDRIJVEN",11X,9(I6,2X)/)                  TOEDNCI   372
      WRITE(2,1120)(PAB1(N,L),L=1,9)                                           TOEDNCI   373
 1120 FORMAT(1H ,15X,"PERC VAN TOT AANT BEDR",5X,9(F8.1)/)                     TOEDNCI   374
      WRITE(2,1130)(POHB1(N,L),L=1,9)                                          TOEDNCI   375
 1130 FORMAT(1H ,15X,"PERC VAN DE TOT OPP",8X,9(F8.1)/)                        TOEDNCI   376
      WRITE(2,1140)(KABB2(N,L),L=1,9)                                          TOEDNCI   377
 1140 FORMAT(1HO,15X,"BEDRIJVEN WAARVAN DE HUISKAVEL 80 PROC OF MEER VAN TOEDNCI   378
     1 DE BEDROPP UITMAAKT :"//16X,"AANTAL BEDRIJVEN",11X,9(I6,2X)/)           TOEDNCI   379
      WRITE(2,1120)(PAB2(N,L),L=1,9)                                           TOEDNCI   380
      WRITE(2,1130)(POHB2(N,L),L=1,9)                                          TOEDNCI   381
      WRITE(2,1150)(KABB(N,L),L=1,9)                                           TOEDNCI   382
 1150 FORMAT(1HO,15X,"BEDRIJVEN WAARVAN DE HUISKAVEL 60 PROC OF MEER VAN TOEDNCI   383
     1 DE BEDROPP UITMAAKT :"//16X,"AANTAL BEDRIJVEN",11X,9(I6,2X)/)           TOEDNCI   384
      WRITE(2,1120)(PAB(N,L),L=1,9)                                            TOEDNCI   385
      WRITE(2,1130)(POHB(N,L),L=1,9)                                           TOEDNCI   386
 1160 CONTINUE                                                                 TOEDNCI   387
 1170 CONTINUE                                                                 TOEDNCI   388
C  PRINTEN TABEL B1                                                           TOEDNCI   389
      NPB=ITEL=0  $  CALL PRINTB1 (NPB,ITEL)                                   TOEDNCI   390
      DO 1180 N = 1,MTT                                                        TOEDNCI   391
      READ(5,680)IK(N),K,O,LV,LS,LO,LW,LL,IR,IV,IU                             TOEDNCI   392
      NPB=NPB+1  $ CALL PRINTB1 (NPB,ITEL)                                     TOEDNCI   393
 1180 WRITE(2,1190)IK(N),K,O,LV,LS,LO,LW,LL,IR,IV,IU                           TOEDNCI   394
 1190 FORMAT(1H ,15X,I8,I4,F9.4,I4,5I3,I4,I3)                                  TOEDNCI   395
      REWIND 5                                                                 TOEDNCI   396
C BEPALEN UITRUILBARE OPPERVLAKTE PER BEDRIJF PER RUILKLASSE ( TAPE 3 )       TOEDNCI   397
C BEPALEN SCHIJNBARE GEM KAVELAFST PER BEDRIJF PER RUILKLASSE (TAPE 5 )       TOEDNCI   398
C NIET UITRUILBARE KAVELS UIT BESTAND HALEN EN PRINTEN OP VOLGORDE            TOEDNCI   399
C VAN DE BEDRIJFSNUMMERS                                                      TOEDNCI   400
```

```
      IK(MTT+1)=100  $  N=NJ=NK=JZ=0                                    TOEDNCI  401
      NPB=ITEL=0  $  CALL PRINTB2 (NPB,ITEL)                            TOEDNCI  402
 1200 READ(10,200) IKH,IKK,OG,IRH,IVH,IUH                               TOEDNCI  403
      JZ=JZ+1                                                           TOEDNCI  404
 1210 KA=KHG=J=L=JCN=OB=KC=KD=0                                         TOEDNCI  405
 1220 OP1=OP2=OP3=OP4=OP5=OP6=OP7=OP8=OP9=OP10=OPT=0                    TOEDNCI  406
      A1=A2=A3=A4=A5=A6=A7=A8=A9=A10=0                                  TOEDNCI  407
      IAG1=IAG2=IAG3=IAG4=IAG5=IAG6=IAG7=IAG8=IAG9=IAG10=0              TOEDNCI  408
 1230 N=N+1                                                             TOEDNCI  409
      READ(5,680)IK(N),K,O,LV,LS,LO,LW,LL,IR,IV,IU                      TOEDNCI  410
C  WIJZIGINGEN  IN UITRUILBAARHEID EN RUILKLASSE AANBRENGEN VOOR        TOEDNCI  411
C  BOERDERIJVERPLAATSINGEN                                             TOEDNCI  412
      IF(IK(N).NE.IKH) GOTO 1480                                        TOEDNCI  413
      IF(K.NE.1.AND.KHG.EQ.0.AND.IVH.NE.999) GOTO 1250                  TOEDNCI  414
 1240 L=L+1  $  MB(L)=IK(N)  $  MK(L)=K  $  OTO(L)=0  $  OT(L)=0        TOEDNCI  415
      MA(L)=LV*10**8+LS*10**6+LO*10**4+LW*100+LL                        TOEDNCI  416
      MR(L)=IR  $  MU(L)=IU  $  MV(L)=IV  $  KHG=1  $  GOTO 1260       TOEDNCI  417
 1250 L=L+1  $  KA=1  $  KHG=1  $  GOTO 1240                            TOEDNCI  418
 1260 IF(IK(N+1).EQ.IKH) GOTO 1230                                      TOEDNCI  419
      IF(IVH.NE.999) GOTO 1330                                          TOEDNCI  420
      GOTO 1280                                                         TOEDNCI  421
 1270 JCN=1                                                             TOEDNCI  422
 1280 OT(IKK)=OT(IKK)-OG                                                TOEDNCI  423
      DO 1290 I = 1,L                                                   TOEDNCI  424
      IF(MR(I).NE.IRH.OR.MU(I).EQ.2) GOTO 1290                          TOEDNCI  425
      OT(I)=OT(I)+OG  $  GOTO 1310                                     TOEDNCI  426
 1290 CONTINUE                                                          TOEDNCI  427
      DO 1300 I = 1,L                                                   TOEDNCI  428
      IF(MU(I).EQ.2) GOTO 1300                                          TOEDNCI  429
      IF(OT(I).NE.0) GOTO 1300                                          TOEDNCI  430
      OT(I)=OG  $  MR(I)=IRH  $  MU(I)=IUH  $  GOTO 1310               TOEDNCI  431
 1300 CONTINUE                                                          TOEDNCI  432
      OB=OG  $  KB=IRH                                                  TOEDNCI  433
 1310 IF(JZ.EQ.JA) GOTO 1320                                            TOEDNCI  434
      READ(10,200) IKH,IKK,OG,IRH,IVH,IUH                               TOEDNCI  435
      JZ=JZ+1                                                           TOEDNCI  436
      IF(IKH.EQ.MB(1)) GOTO 1270                                        TOEDNCI  437
 1320 JCN=1                                                             TOEDNCI  438
      GOTO 1400                                                         TOEDNCI  439
 1330 DO 1370 I = 1,L                                                   TOEDNCI  440
      IF(I.NE.1) GOTO 1350                                              TOEDNCI  441
      IF(KA.EQ.1) GOTO 1340                                             TOEDNCI  442
      IF(MR(I).NE.IRH) KD=MR(I)                                         TOEDNCI  443
      OR=OG-OT(I)  $  OT(I)=OG  $  MR(I)=IRH  $  MV(I)=IVH  $  MU(I)=IUH TOEDNCI  444
      GOTO 1370                                                         TOEDNCI  445
 1340 OR=OT(I)=OG  $  OTO(I)=0  $  MB(I)=IKH  $  MK(I)=IKK  $  MA(I)=0  TOEDNCI  446
      MR(I)=IRH  $  MV(I)=IVH  $  MU(I)=IUH  $  GOTO 1370               TOEDNCI  447
 1350 IF(MR(I).NE.MR(1).OR.MU(I).EQ.2) GOTO 1370                        TOEDNCI  448
      IF(OR.GT.OT(I)) GOTO 1360                                         TOEDNCI  449
      OT(I)=OT(I)-OR  $  GOTO 1400                                     TOEDNCI  450
 1360 OR=OR-OT(I)  $  OT(I)=0                                           TOEDNCI  451
 1370 CONTINUE                                                          TOEDNCI  452
      DO 1390 I = 2,L                                                   TOEDNCI  453
      IF(MR(I).EQ.MR(1).OR.MU(I).EQ.2) GOTO 1390                        TOEDNCI  454
      IF(OR.GT.OT(I)) GOTO 1380                                         TOEDNCI  455
      OT(I)=OT(I)-OR  $  GOTO 1400                                     TOEDNCI  456
 1380 OR=OR-OT(I)  $  OT(I)=0                                           TOEDNCI  457
```

```
      1390 CONTINUE                                                                  TOEDNCI   458
      1400 IF(KA.EQ.1) GOTO 1440                                                     TOEDNCI   459
           IF(KD.EQ.0) GOTO 1430                                                     TOEDNCI   460
460        OG=OG=OTO(1)  $  KG=KD                                                   TOEDNCI   461
           DO 1420 I = 2,L                                                          TOEDNCI   462
           IF(MR(I).NE.MR(1)) GOTO 1420                                             TOEDNCI   463
           IF(OG.GT.OT(I)) GOTO 1410                                                TOEDNCI   464
           OT(I)=OT(I)-OG  $  GOTO 1430                                             TOEDNCI   465
465   1410 OG=OG-OT(I)  $  OT(I)=0                                                  TOEDNCI   466
      1420 CONTINUE                                                                 TOEDNCI   467
      1430 N=N-L  $  GOTO 1450                                                      TOEDNCI   468
      1440 N=N-(L-1)  $  J=J+1                                                      TOEDNCI   469
           IK(N)=IKH  $  K=IKK  $  O=OG  $  IU=IUH  $  IV=IVH  $  GOTO 1780         TOEDNCI   470
470   1450 J=J+1  $  N=N+1                                                          TOEDNCI   471
           IF(OT(J).NE.OTO(J).AND.MK(J).NE.1) GOTO 1460                             TOEDNCI   472
           O=OT(J)  $  GOTO 1470                                                    TOEDNCI   473
      1460 O=OTO(J)                                                                 TOEDNCI   474
      1470 IK(N)=MB(J)  $  K=MK(J)  $  JN=MA(J)  $  LV=JN/10**8                     TOEDNCI   475
475        JN=JN-LV*10**8  $  LS=JN/10**6  $  JN=JN-LS*10**6  $  LO=JN/10**4       TOEDNCI   476
           JN=JN-LO*10**4  $  LW=JN/100  $  LL=JN-LW*100  $  IR=MR(J)               TOEDNCI   477
           IV=MV(J)  $  IU=MU(J)                                                    TOEDNCI   478
      1480 IF(IU.EQ.2) GOTO 1780                                                    TOEDNCI   479
           IF(IR.EQ.1)GOTO 1490 $ IF(IR.EQ.2)GOTO 1500 $ IF(IR.EQ.3)GOTO 1510       TOEDNCI   480
480        IF(IR.EQ.4)GOTO 1520 $ IF(IR.EQ.5)GOTO 1530 $ IF(IR.EQ.6)GOTO 1540       TOEDNCI   481
           IF(IR.EQ.7)GOTO 1550 $ IF(IR.EQ.8)GOTO 1560 $ IF(IR.EQ.9)GOTO 1570       TOEDNCI   482
           IF(IR.EQ.10) GOTO 1580                                                   TOEDNCI   483
      1490 OP1=OP1+O $ A1=A1+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   484
      1500 OP2=OP2+O $ A2=A2+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   485
485   1510 OP3=OP3+O $ A3=A3+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   486
      1520 OP4=OP4+O $ A4=A4+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   487
      1530 OP5=OP5+O $ A5=A5+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   488
      1540 OP6=OP6+O $ A6=A6+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   489
      1550 OP7=OP7+O $ A7=A7+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   490
490   1560 OP8=OP8+O $ A8=A8+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   491
      1570 OP9=OP9+O $ A9=A9+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0) $ GOTO 1590         TOEDNCI   492
      1580 OP10=OP10+O $ A10=A10+O*(LV+LS*1.5+LO*2.2+LW*4.0+LL*3.0)                 TOEDNCI   493
      1590 IF(KHG.EQ.1.AND.J.LT.L) GOTO 1450                                        TOEDNCI   494
           IF(IK(N).EQ.IK(N+1)) GOTO 1230                                           TOEDNCI   495
495   1600 OPT=OP1+OP2+OP3+OP4+OP5+OP6+OP7+OP8+OP9+OP10                             TOEDNCI   496
           IF(OP1.EQ.0) GOTO 1610 $ IAG1=IFIX(A1/OP1*100+.5)                        TOEDNCI   497
      1610 IF(OP2.EQ.0) GOTO 1620 $ IAG2=IFIX(A2/OP2*100+.5)                        TOEDNCI   498
      1620 IF(OP3.EQ.0) GOTO 1630 $ IAG3=IFIX(A3/OP3*100+.5)                        TOEDNCI   499
      1630 IF(OP4.EQ.0) GOTO 1640 $ IAG4=IFIX(A4/OP4*100+.5)                        TOEDNCI   500
500   1640 IF(OP5.EQ.0) GOTO 1650 $ IAG5=IFIX(A5/OP5*100+.5)                        TOEDNCI   501
      1650 IF(OP6.EQ.0) GOTO 1660 $ IAG6=IFIX(A6/OP6*100+.5)                        TOEDNCI   502
      1660 IF(OP7.EQ.0) GOTO 1670 $ IAG7=IFIX(A7/OP7*100+.5)                        TOEDNCI   503
      1670 IF(OP8.EQ.0) GOTO 1680 $ IAG8=IFIX(A8/OP8*100+.5)                        TOEDNCI   504
      1680 IF(OP9.EQ.0) GOTO 1690 $ IAG9=IFIX(A9/OP9*100+.5)                        TOEDNCI   505
505   1690 IF(OP10.EQ.0) GOTO 1700 $ IAG10=IFIX(A10/OP10*100+.5)                    TOEDNCI   506
      1700 IF(L.EQ.0)  GOTO 1740                                                    TOEDNCI   507
           IF(OB.NE.0) KC=L+1                                                       TOEDNCI   508
           DO 1730 J = 1,L                                                          TOEDNCI   509
           IF(OT(J).EQ.OTO(J).OR.MU(J).EQ.2) GOTO 1720                              TOEDNCI   510
510   1710 IF(MR(J).EQ.1) OP1=OP1+OT(J)-OTO(J)                                      TOEDNCI   511
           IF(MR(J).EQ.2) OP2=OP2+OT(J)-OTO(J)                                      TOEDNCI   512
           IF(MR(J).EQ.3) OP3=OP3+OT(J)-OTO(J)                                      TOEDNCI   513
           IF(MR(J).EQ.4) OP4=OP4+OT(J)-OTO(J)                                      TOEDNCI   514
```

```
      IF(MR(J).EQ.5) OP5=OP5+OT(J)-OTO(J)                                TOEDNCI  515
515   IF(MR(J).EQ.6) OP6=OP6+OT(J)-OTO(J)                                TOEDNCI  516
      IF(MR(J).EQ.7) OP7=OP7+OT(J)-OTO(J)                                TOEDNCI  517
      IF(MR(J).EQ.8) OP8=OP8+OT(J)-OTO(J)                                TOEDNCI  518
      IF(MR(J).EQ.9) OP9=OP9+OT(J)-OTO(J)                                TOEDNCI  519
      IF(MR(J).EQ.10) OP10=OP10+OT(J)-OTO(J)                             TOEDNCI  520
520  1720 IF((J+1).NE.KC) GOTO 1730                                      TOEDNCI  521
      KC=0  $  MR(J)=KB  $  OT(J)=OB  $  OTO(J)=0  $  GOTO 1710         TOEDNCI  522
     1730 CONTINUE                                                       TOEDNCI  523
      OPT=OP1+OP2+OP3+OP4+OP5+OP6+OP7+OP8+OP9+OP10                       TOEDNCI  524
     1740 WRITE(3,1750)IK(N),OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP8,OP9,OP10,OPT TOEDNCI  525
525   NPB=NPB+2  $ CALL PRINTB2 (NPB,ITEL)                               TOEDNCI  526
      WRITE(2,1760)IK(N),OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP8,OP9,OP10,OPT   TOEDNCI  527
     1750 FORMAT(I8,10F8.2,F10.2)                                        TOEDNCI  528
     1760 FORMAT(1H0,15X,I8,10F8.2,F10.2)                                TOEDNCI  529
      WRITE(4,1770)IK(N),IAG1,IAG2,IAG3,IAG4,IAG5,IAG6,IAG7,IAG8,IAG9,  TOEDNCI  530
530  1IAG10                                                              TOEDNCI  531
     1770 FORMAT(I8,10I6)                                                TOEDNCI  532
     C NK = AANTAL BEDRIJVEN                                             TOEDNCI  533
      NK=NK+1                                                            TOEDNCI  534
      IF(JCN.EQ.1.AND.N.LT.MTT) GOTO 1210                                TOEDNCI  535
535   IF(KHG.EQ.1.AND.JZ.LT.JA) GOTO 1200                                TOEDNCI  536
      IF(N.LT.MTT) GOTO 1220                                             TOEDNCI  537
      GOTO 1800                                                          TOEDNCI  538
     1780 LT=LV+IFIX(FLOAT(LS)*1.5+FLOAT(LO)*2.2+.5)+LW*4+LL*3           TOEDNCI  539
      WRITE(6,1790)IK(N),K,O,IV,LT                                       TOEDNCI  540
540   1790 FORMAT(I8,I3,F8.2,2I4)                                        TOEDNCI  541
     C NJ = AANTAL NIET UITRUILBARE KAVELS                               TOEDNCI  542
      NJ=NJ+1                                                            TOEDNCI  543
      IF(KHG.EQ.1.AND.J.LT.L) GOTO 1450                                  TOEDNCI  544
      IF(IK(N).EQ.IK(N+1)) GOTO 1230                                     TOEDNCI  545
545   GOTO 1600                                                          TOEDNCI  546
     1800 ENDFILE3 $ ENDFILE4 $ ENDFILE6                                 TOEDNCI  547
      REWIND3  $  REWIND4  $  REWIND5  $  REWIND6  $  REWIND10           TOEDNCI  548
      NPB=ITEL=0  $  CALL PRINTB3 (NPB,ITEL)                             TOEDNCI  549
      DO 1820 J=1,NJ                                                     TOEDNCI  550
550   READ(6,1790)I,IK1(J),R(J),IV,LT                                    TOEDNCI  551
      NPB=NPB+1  $ CALL PRINTB3 (NPB,ITEL)                               TOEDNCI  552
      WRITE(2,1810) I,IK1(J),R(J),IV                                     TOEDNCI  553
     1810 FORMAT(1H ,15X,I8,I4,F8.2,I4)                                  TOEDNCI  554
      WRITE(18,1790) I,IK1(J),R(J),IV,LT                                 TOEDNCI  555
555   IK(J)=IV*10**8+I                                                   TOEDNCI  556
     1820 CONTINUE                                                       TOEDNCI  557
      ENDFILE18  $  REWIND18                                             TOEDNCI  558
     C NIET UITRUILBARE KAVELS SORTEREN OP VAKNUMMER EN PRINTEN PER VAK  TOEDNCI  559
      ILR=1 $ IUR=NJ                                                     TOEDNCI  560
560   CALL INDCOL                                                        TOEDNCI  561
      REWIND6                                                            TOEDNCI  562
      DO 1830 J = 1,NJ                                                   TOEDNCI  563
      JK=IK(J) $ IK(J)=JK/10**8 $ IK1(J)=JK-IK(J)*10**8                  TOEDNCI  564
     1830 CONTINUE                                                       TOEDNCI  565
565   NPB=ITEL=0  $  CALL PRINTB4 (NPB,ITEL)                             TOEDNCI  566
      IK (NJ+1)=100 $ J=0                                                TOEDNCI  567
     1840 J=J+1  $  NPB=NPB+2  $ CALL PRINTB4 (NPB,ITEL)                 TOEDNCI  568
      WRITE(2,1850)IK(J),IK1(J),R(J)                                     TOEDNCI  569
     1850 FORMAT(1H0,15X,I5,I12,F8.2)                                    TOEDNCI  570
570   1860 IF(IK (J).EQ.IK (J+1)) GOTO 1870                              TOEDNCI  571
```

```
                IF(J.LT.NJ) GOTO 1840                                   TOEDNCI  572
                GOTO 1890                                               TOEDNCI  573
             1870 J=J+1  $  NPB=NPB+1  $ CALL PRINTB4 (NPB,ITEL)        TOEDNCI  574
                WRITE(2,1880)IK1(J),R(J)                                TOEDNCI  575
575          1880 FORMAT(21X,I12,F8.2)                                  TOEDNCI  576
                GOTO 1860                                               TOEDNCI  577
             C BEPALEN VAN DE VAKOPPERVLAKTEN                           TOEDNCI  578
             1890 DO 1900 N =1,IVM                                      TOEDNCI  579
             1900 R(N)=0                                                TOEDNCI  580
580             KV=KK=0                                                 TOEDNCI  581
                DO 1960 N = 1,MTT                                       TOEDNCI  582
                READ(5,1910) I,K,O,IR,IV,IU                             TOEDNCI  583
             1910 FORMAT(I8,I3,F9.4,16X,I3,I4,I2)                       TOEDNCI  584
                IF(KK.EQ.1.OR.KV.EQ.JA) GOTO 1930                       TOEDNCI  585
585          1920 READ(10,200) IKH,IKK,OG,IRH,IVH,IUH                   TOEDNCI  586
                KV=KV+1  $  KK=1                                        TOEDNCI  587
                IF(IKK.NE.1.AND.KV.LT.JA) GOTO 1920                     TOEDNCI  588
             1930 IF(I.EQ.IKH.AND.IVH.NE.999) GOTO 1940                 TOEDNCI  589
                IF(IU.EQ.2) GOTO 1960                                   TOEDNCI  590
590             IK(N)=IV*100+IR  $  R(IV)=R(IV)+O                       TOEDNCI  591
                GOTO 1960                                               TOEDNCI  592
             1940 IF(K.NE.1) GOTO 1950                                  TOEDNCI  593
                R(IVH)=R(IVH)-OG  $  KK=0                               TOEDNCI  594
                R(IV)=R(IV)+O                                           TOEDNCI  595
595             GOTO 1960                                               TOEDNCI  596
             1950 R(IVH)=R(IVH)-OG  $  KK=0                             TOEDNCI  597
                IK(N)=IV*100+IR  $  R(IV)=R(IV)+O                       TOEDNCI  598
             1960 CONTINUE                                              TOEDNCI  599
                REWIND5                                                 TOEDNCI  600
600             NPB=ITEL=0  $  CALL PRINTB5 (NPB,ITEL)                  TOEDNCI  601
                DO 2000 M =1,IVM                                        TOEDNCI  602
                IF(R(M).EQ.0) GOTO 2000                                 TOEDNCI  603
                N=0                                                     TOEDNCI  604
             1970 N=N+1                                                 TOEDNCI  605
605             IL=IK(N) $ IV=IL/100 $ IR=IL-IV*100 $ IF(IV.NE.M) GOTO 1970 TOEDNCI  606
                WRITE(5,1980) IV,R(M),IR                                TOEDNCI  607
                NPB=NPB+1  $ CALL PRINTB5 (NPB,ITEL)                    TOEDNCI  608
                WRITE(2,1990) IV,R(M),IR                                TOEDNCI  609
             1980 FORMAT(I9,F10.2,I8)                                   TOEDNCI  610
610          1990 FORMAT(1H ,15X,I8,F10.2,I8)                           TOEDNCI  611
             2000 CONTINUE                                              TOEDNCI  612
                ENDFILE5 $ REWIND5                                      TOEDNCI  613
             C BEREKENEN OPP PER RUILKLASSE TOTAAL UIT BEDRUPP          TOEDNCI  614
             C OP TAPE ZETTEN PER RUILKLASSE: BEDRNR, TOE TE DELEN OPP EN TOEDNCI  615
615          C GEM KAVELAFSTAND                                         TOEDNCI  616
             C ALLE BEDRIJFS- EN VAKOPERVLAKTEN WORDEN INTEGER GEMAAKT DOOR TOEDNCI  617
             C VERMENIGVULDIGING MET 100                                TOEDNCI  618
                DO 2010 I = 7,15                                        TOEDNCI  619
             2010 REWIND I                                              TOEDNCI  620
620             J1=J2=J3=J4=J5=J6=J7=J8=J9=J10=0                        TOEDNCI  621
                NOP1T=NOP2T=NOP3T=NOP4T=NOP5T=NOP6T=NOP7T=NOP8T=NOP9T=NOP10T=0 TOEDNCI  622
             2020 NOP1=NOP2=NOP3=NOP4=NOP5=NOP6=NOP7=NOP8=NOP9=NOP10=0  TOEDNCI  623
                READ(3,2030)J,OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP8,OP9,OP10  TOEDNCI  624
             2030 FORMAT(I8,10F8.2)                                     TOEDNCI  625
625             IF(EOF(3)) 2160,2040                                    TOEDNCI  626
             2040 READ(4,1770)I,IAG1,IAG2,IAG3,IAG4,IAG5,IAG6,IAG7,IAG8,IAG9,IAG10 TOEDNCI  627
                IF(J.NE.I) GOTO 2380                                    TOEDNCI  628
```

```
      IF(OP1.LE.0) GOTO 2060                                                    TOEDNCI  629
      NOP1=OP1*100+.5 $ NOP1T=NOP1T+NOP1                                        TOEDNCI  630
630   WRITE(6,2050)J,NOP1,IAG1                                                  TOEDNCI  631
2050  FORMAT(I9,I5,I6)                                                          TOEDNCI  632
      J1=J1+1                                                                   TOEDNCI  633
2060  IF(OP2.LE.0) GOTO 2070                                                    TOEDNCI  634
      NOP2=OP2*100+.5 $ NOP2T=NOP2T+NOP2                                        TOEDNCI  635
635   WRITE(7,2050)J,NOP2,IAG2                                                  TOEDNCI  636
      J2=J2+1                                                                   TOEDNCI  637
2070  IF(OP3.LE.0) GOTO 2080                                                    TOEDNCI  638
      NOP3=OP3*100+.5 $ NOP3T=NOP3T+NOP3                                        TOEDNCI  639
      WRITE(8,2050)J,NOP3,IAG3                                                  TOEDNCI  640
640   J3=J3+1                                                                   TOEDNCI  641
2080  IF(OP4.LE.0) GOTO 2090                                                    TOEDNCI  642
      NOP4=OP4*100+.5 $ NOP4T=NOP4T+NOP4                                        TOEDNCI  643
      WRITE(9,2050)J,NOP4,IAG4                                                  TOEDNCI  644
      J4=J4+1                                                                   TOEDNCI  645
645   2090 IF(OP5.LE.0) GOTO 2100                                               TOEDNCI  646
      NOP5=OP5*100+.5 $ NOP5T=NOP5T+NOP5                                        TOEDNCI  647
      WRITE(10,2050)J,NOP5,IAG5                                                 TOEDNCI  648
      J5=J5+1                                                                   TOEDNCI  649
2100  IF(OP6.LE.0) GOTO 2110                                                    TOEDNCI  650
650   NOP6=OP6*100+.5 $ NOP6T=NOP6T+NOP6                                        TOEDNCI  651
      WRITE(11,2050)J,NOP6,IAG6                                                 TOEDNCI  652
      J6=J6+1                                                                   TOEDNCI  653
2110  IF(OP7.LE.0) GOTO 2120                                                    TOEDNCI  654
      NOP7=OP7*100+.5 $ NOP7T=NOP7T+NOP7                                        TOEDNCI  655
655   WRITE(12,2050)J,NOP7,IAG7                                                 TOEDNCI  656
      J7=J7+1                                                                   TOEDNCI  657
2120  IF(OP8.LE.0) GOTO 2130                                                    TOEDNCI  658
      NOP8=OP8*100+.5 $ NOP8T=NOP8T+NOP8                                        TOEDNCI  659
      WRITE(13,2050)J,NOP8,IAG8                                                 TOEDNCI  660
660   J8=J8+1                                                                   TOEDNCI  661
2130  IF(OP9.LE.0) GOTO 2140                                                    TOEDNCI  662
      NOP9=OP9*100+.5 $ NOP9T=NOP9T+NOP9                                        TOEDNCI  663
      WRITE(14,2050)J,NOP9,IAG9                                                 TOEDNCI  664
      J9=J9+1                                                                   TOEDNCI  665
665   2140 IF(OP10.LE.0) GOTO 2150                                              TOEDNCI  666
       NOP10=OP10*100+.5 $ NOP10T=NOP10T+NOP10                                  TOEDNCI  667
      WRITE(15,2050)J,NOP10,IAG10                                               TOEDNCI  668
      J10=J10+1                                                                 TOEDNCI  669
2150  GOTO 2020                                                                 TOEDNCI  670
670   C BEREKENEN OPP PER RUILKLASSE TOTAAL UIT VAKOPP                          TOEDNCI  671
      C PER RUILKLASSE OP TAPE ZETTEN VAN VAKNRS MET OPP                        TOEDNCI  672
2160  IVO1T=IVO2T=IVO3T=IVO4T=IVO5T=IVO6T=IVO7T=IVO8T=IVO9T=IVO10T=0            TOEDNCI  673
      I1=I2=I3=I4=I5=I6=I7=I8=I9=I10=0                                          TOEDNCI  674
2170  IVO1=IVO2=IVO3=IVO4=IVO5=IVO6=IVO7=IVO8=IVO9=IVO10=0                      TOEDNCI  675
675   READ(5,2180) IV,OIV,IR                                                    TOEDNCI  676
      IF(EOF(5)) 2310,2190                                                      TOEDNCI  677
2180  FORMAT(I9,F10.2,I8)                                                       TOEDNCI  678
2190  IF(IR.EQ.1)GOTO 2200 $ IF(IR.EQ.2)GOTO 2220 $ IF(IR.EQ.3)GOTO 2230        TOEDNCI  679
      IF(IR.EQ.4)GOTO 2240 $ IF(IR.EQ.5)GOTO 2250 $ IF(IR.EQ.6)GOTO 2260        TOEDNCI  680
680   IF(IR.EQ.7)GOTO 2270 $ IF(IR.EQ.8)GOTO 2280 $ IF(IR.EQ.9)GOTO 2290        TOEDNCI  681
      IF(IR.EQ.10) GOTO 2300                                                    TOEDNCI  682
2200  IVO1=OIV*100+.5 $ IVO1T=IVO1T+IVO1                                        TOEDNCI  683
       WRITE(6,2210)IV,IVO1                                                     TOEDNCI  684
2210  FORMAT(2I9)                                                               TOEDNCI  685
```

```
                I1=I1+1 $ GOTO 2170                                                TOEDNCI   686
             2220 IVO2=OIV*100+.5 $ IVO2T=IVO2T+IVO2                               TOEDNCI   687
                WRITE(7,2210)IV,IVO2                                               TOEDNCI   688
                I2=I2+1 $ GOTO 2170                                                TOEDNCI   689
             2230 IVO3=OIV*100+.5 $ IVO3T=IVO3T+IVO3                               TOEDNCI   690
                WRITE(8,2210)IV,IVO3                                               TOEDNCI   691
                I3=I3+1 $ GOTO 2170                                                TOEDNCI   692
             2240 IVO4=OIV*100+.5 $ IVO4T=IVO4T+IVO4                               TOEDNCI   693
                WRITE(9,2210)IV,IVO4                                               TOEDNCI   694
                I4=I4+1 $ GOTO 2170                                                TOEDNCI   695
             2250 IVO5=OIV*100+.5 $ IVO5T=IVO5T+IVO5                               TOEDNCI   696
                WRITE(10,2210)IV,IVO5                                              TOEDNCI   697
                I5=I5+1 $ GOTO 2170                                                TOEDNCI   698
             2260 IVO6=OIV*100+.5 $ IVO6T=IVO6T+IVO6                               TOEDNCI   699
                WRITE(11,2210)IV,IVO6                                              TOEDNCI   700
                I6=I6+1 $ GOTO 2170                                                TOEDNCI   701
             2270 IVO7=OIV*100+.5 $ IVO7T=IVO7T+IVO7                               TOEDNCI   702
                WRITE(12,2210)IV,IVO7                                              TOEDNCI   703
                I7=I7+1 $ GOTO 2170                                                TOEDNCI   704
             2280 IVO8=OIV*100+.5 $ IVO8T=IVO8T+IVO8                               TOEDNCI   705
                WRITE(13,2210)IV,IVO8                                              TOEDNCI   706
                I8=I8+1 $ GOTO 2170                                                TOEDNCI   707
             2290 IVO9=OIV*100+.5 $ IVO9T=IVO9T+IVO9                               TOEDNCI   708
                WRITE(14,2210)IV,IVO9                                              TOEDNCI   709
                I9=I9+1 $ GOTO 2170                                                TOEDNCI   710
             2300 IVO10=OIV*100+.5 $ IVO10T=IVO10T+IVO10                           TOEDNCI   711
                WRITE(15,2210)IV,IVO10                                             TOEDNCI   712
                I10=I10+1 $ GOTO 2170                                              TOEDNCI   713
            C PER RUILKLASSE TOTALE VAKOPP GELIJK MAKEN AAN TOTALE BEDROPP         TOEDNCI   714
             2310 L=6                                                              TOEDNCI   715
                CALL VEROPP (NOP1T,IVO1T,L)    $   IF(J2.EQ.0) GOTO 2320           TOEDNCI   716
                CALL VEROPP (NOP2T,IVO2T,L)    $   IF(J3.EQ.0) GOTO 2320           TOEDNCI   717
                CALL VEROPP (NOP3T,IVO3T,L)    $   IF(J4.EQ.0) GOTO 2320           TOEDNCI   718
                CALL VEROPP (NOP4T,IVO4T,L)    $   IF(J5.EQ.0) GOTO 2320           TOEDNCI   719
                CALL VEROPP (NOP5T,IVO5T,L)    $   IF(J6.EQ.0) GOTO 2320           TOEDNCI   720
                CALL VEROPP (NOP6T,IVO6T,L)    $   IF(J7.EQ.0) GOTO 2320           TOEDNCI   721
                CALL VEROPP (NOP7T,IVO7T,L)    $   IF(J8.EQ.0) GOTO 2320           TOEDNCI   722
                CALL VEROPP (NOP8T,IVO8T,L)    $   IF(J9.EQ.0) GOTO 2320           TOEDNCI   723
                CALL VEROPP (NOP9T,IVO9T,L)    $   IF(J10.EQ.0) GOTO 2320          TOEDNCI   724
                CALL VEROPP (NOP10T,IVO10T,L)                                      TOEDNCI   725
             2320 DO 2330 I = 6,15                                                 TOEDNCI   726
                ENDFILE I                                                          TOEDNCI   727
             2330 REWIND I                                                         TOEDNCI   728
                REWIND3 $ REWIND4 $ REWIND5                                        TOEDNCI   729
                WRITE(2,2360)I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,J1,J2,J3,J4,J5,J6,J7,  TOEDNCI   730
               1J8,J9,J10                                                          TOEDNCI   731
                WRITE(5,2340) IA,IB,IC,IZ,NDA,NDG,MAB                              TOEDNCI   732
             2340 FORMAT(3A10,I5,2I2,I5)                                           TOEDNCI   733
                WRITE(5,2350)I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,J1,J2,J3,J4,J5,J6,     TOEDNCI   734
               1 J7,J8,J9,J10                                                      TOEDNCI   735
             2350 FORMAT(20I5)                                                     TOEDNCI   736
                REWIND5                                                            TOEDNCI   737
                WRITE(2,2370)NOP1T,NOP2T,NOP3T,NOP4T,NOP5T,NOP6T,NOP7T,NOP8T,      TOEDNCI   738
               1NOP9T,NOP10T,IVO1T,IVO2T,IVO3T,IVO4T,IVO5T,IVO6T,IVO7T,IVO8T,      TOEDNCI   739
               2IVO9T,IVO10T                                                       TOEDNCI   740
             2360 FORMAT(1H1,24X,"RUILKLASSE"//9X"1","    2","    3    4    5    6", TOEDNCI   741
               1"    7    8    9   10"//" M=",I7,9I5/" N=",I7,9I5)                 TOEDNCI   742
```

Bijlage 14.13

```
      2370 FORMAT(1H0,48X,"RUILKLASSE"//21X,"1       2       3       4",7X,   TOEDNCI   743
          1"5       6       7       8       9      10"//3X,"OPP BEDR:",I10,   TOEDNCI   744
          2918/" OPP VAKKEN:",I10,9I8)                                        TOEDNCI   745
745   C  ALLE RUILKLASSEN NA ELKAAR OP RA-FILE ZETTEN                         TOEDNCI   746
           CALL OPENMS (16,INDEX,4000,0)                                      TOEDNCI   747
           IT=6 $ NL=0                                                        TOEDNCI   748
           CALL OVER (J1,I1,IT,NL) $ IF(J2.EQ.0) GOTO 2380                    TOEDNCI   749
           CALL OVER (J2,I2,IT,NL) $ IF(J3.EQ.0) GOTO 2380                    TOEDNCI   750
750        CALL OVER (J3,I3,IT,NL) $ IF(J4.EQ.0) GOTO 2380                    TOEDNCI   751
           CALL OVER (J4,I4,IT,NL) $ IF(J5.EQ.0) GOTO 2380                    TOEDNCI   752
           CALL OVER (J5,I5,IT,NL) $ IF(J6.EQ.0) GOTO 2380                    TOEDNCI   753
           CALL OVER (J6,I6,IT,NL) $ IF(J7.EQ.0) GOTO 2380                    TOEDNCI   754
           CALL OVER (J7,I7,IT,NL) $ IF(J8.EQ.0) GOTO 2380                    TOEDNCI   755
755        CALL OVER (J8,I8,IT,NL) $ IF(J9.EQ.0) GOTO 2380                    TOEDNCI   756
           CALL OVER (J9,I9,IT,NL) $ IF(J10.EQ.0) GOTO 2380                   TOEDNCI   757
           CALL OVER (J10,I10,IT,NL)                                          TOEDNCI   758
      2380 CALL EXIT                                                          TOEDNCI   759
            END                                                               TOEDNCI   760
```



```
 1          SUBROUTINE VEROPP (N,I,L)                                        TOEDNCI   761
           J=N-I                                                            TOEDNCI   762
           BACKSPACE L                                                      TOEDNCI   763
           READ(L,100) IV,IVO                                               TOEDNCI   764
 5       100 FORMAT(2I9)                                                    TOEDNCI   765
           IVO=IVO+J  $  I=I+J                                              TOEDNCI   766
           BACKSPACE L                                                      TOEDNCI   767
           WRITE(L,100) IV,IVO                                              TOEDNCI   768
           L=L+1                                                            TOEDNCI   769
10         RETURN                                                           TOEDNCI   770
           END                                                              TOEDNCI   771
```

```
SUBROUTINE OVER       74/175  OPT=1                    FTN 4.6+452       28/08/79  11.26.49       PAGE      1

 1                SUBROUTINE OVER (K,L,IT,NL)                                    TOEDNCI   772
                  DIMENSION IBEDR(3),IVAK(2)                                     TOEDNCI   773
                  NB=NL+1 $ NL=NL+K                                              TOEDNCI   774
                  DO 10 I = NB,NL                                                TOEDNCI   775
 5                READ(IT,100) IBEDR                                             TOEDNCI   776
               100 FORMAT (19,I5,I6)                                              TOEDNCI   777
                10 CALL WRITMS (16,IBEDR,3,I)                                     TOEDNCI   778
                  NB=NL+1 $ NL=NL+L                                              TOEDNCI   779
                  DO 20 I = NB,NL                                                TOEDNCI   780
10                READ(IT,110) IVAK                                              TOEDNCI   781
               110 FORMAT (2I9)                                                   TOEDNCI   782
                20 CALL WRITMS (16,IVAK,2,I)                                      TOEDNCI   783
                  REWIND IT                                                      TOEDNCI   784
                  IT=IT+1                                                        TOEDNCI   785
15                RETURN                                                         TOEDNCI   786
                  END                                                            TOEDNCI   787


SUBROUTINE PRINT1     74/175  OPT=1                    FTN 4.6+452       28/08/79  11.26.49       PAGE      1

 1                SUBROUTINE PRINT1 (NPB,ITEL,IA,IB,IC)                          TOEDNCI   788
                  IF(NPB.EQ.0) GOTO 5                                            TOEDNCI   789
                  IF(ITEL.GT.1) GOTO 4                                           TOEDNCI   790
                  IF(NPB.LT.50) RETURN                                           TOEDNCI   791
 5                GOTO 5                                                         TOEDNCI   792
                4 IF(NPB.LT.49) RETURN                                           TOEDNCI   793
                5 ITEL=ITEL+1                                                    TOEDNCI   794
                  WRITE(2,10) ITEL,IA,IB,IC                                      TOEDNCI   795
               10 FORMAT(1H1//16X,"TABEL 1",83X,"PAG",I3//16X,"RUILVERKAVELING :",3A TOEDNCI   796
10              110//16X,"BEDRIJFSGEGEVENS BESTAANDE TOESTAND"//43X,"GEM",57X,"X="/ TOEDNCI   797
                226X,"AANTAL  GEM       SCHIJNB",12X,"OPP       PERC      PERC",9X,  TOEDNCI   798
                3"BEDR      BEDR"/18X,"BEDRNR  KAVELS  KAVAFST  KAVAFST   BEDROPP  H TOEDNCI   799
                4UISKAV  HUISKAV  GROOTSTE     TYPE      BUITEN"/34X,"IN HM     IN HM TOEDNCI   800
                5",32X,"VELDKAV",15X,"BLOK"/)                                    TOEDNCI   801
15              NPB=0                                                            TOEDNCI   802
                RETURN                                                           TOEDNCI   803
                END                                                              TOEDNCI   804
```

Bijlage 14.15

```
 SUBROUTINE PRINTB1    74/175  OPT=1                    FTN 4.6+452        28/08/79  11.20.49        PAGE     1


 1                SUBROUTINE PRINTB1 (NPB,ITEL)                                    TOEDNC1   805
                  IF(NPB.EQ.0) GOTO 5                                              TOEDNC1   806
                  IF(ITEL.GT.1) GOTO 4                                             TOEDNC1   807
                  IF(NPB.LT.55) RETURN                                             TOEDNC1   808
 5                GOTO 5                                                           TOEDNC1   809
                4 IF(NPB.LT.54) RETURN                                             TOEDNC1   810
                5 ITEL=ITEL+1                                                      TOEDNC1   811
                  WRITE(2,10) ITEL                                                 TOEDNC1   812
               10 FORMAT(1H1//16X,"TABEL B 1",41X,"PAG",I3//16X,"LIJST VAN IN DE RUI TOEDNC1   813
 10              1LVERKAVELING VOORKOMENDE KAVELS"//18X,"BEDRNR KAV      OPP  VW SW  TOEDNC1   814
                 2OW  W  L RU VAK UR"/)                                            TOEDNC1   815
                  NPB=0                                                            TOEDNC1   816
                  RETURN                                                           TOEDNC1   817
                  END                                                              TOEDNC1   818


 SUBROUTINE PRINTB2    74/175  OPT=1                    FTN 4.6+452        28/08/79  11.20.49        PAGE     1

 1                SUBROUTINE PRINTB2 (NPB,ITEL)                                    TOEDNC1   819
                  IF(NPB.EQ.0) GOTO 5                                              TOEDNC1   820
                  IF(ITEL.GT.1) GOTO 4                                             TOEDNC1   821
                  IF(NPB.LT.53) RETURN                                             TOEDNC1   822
 5                GOTO 5                                                           TOEDNC1   823
                4 IF(NPB.LT.51) RETURN                                             TOEDNC1   824
                5 ITEL=ITEL+1                                                      TOEDNC1   825
                  WRITE(2,10) ITEL                                                 TOEDNC1   826
               10 FORMAT(1H1//16X,"TABEL B 2",89X,"PAG",I3//16X,"UITRUILBARE OPPERVL TOEDNC1   827
 10              1AKTE PER BEDRIJF, AANGEGEVEN PER RUILKLASSE EN TOTAAL"//52X,"OPPER TOEDNC1   828
                 2VLAKTE IN RUILKLASSE"/18X,"BEDRNR",7X,"1",7X,"2",7X,"3",7X,"4",7X, TOEDNC1   829
                 3"5",7X,"6",7X,"7",7X,"8",7X,"9",6X,"10 BEDR.OPP."/)              TOEDNC1   830
                  NPB=0                                                            TOEDNC1   831
                  RETURN                                                           TOEDNC1   832
 15               END                                                              TOEDNC1   833
```

```
 SUBROUTINE PRINTB3    74/175  OPT=1                    FTN 4.6+452        28/08/79  11.26.49        PAGE      1

 1                SUBROUTINE PRINTB3 (NPB,ITEL)                                    TOEDNCI   834
                  IF(NPB.EQ.0) GOTO 5                                              TOEDNCI   835
                  IF(ITEL.GT.1) GOTO 4                                             TOEDNCI   836
                  IF(NPB.LT.53) RETURN                                             TOEDNCI   837
 5                GOTO 5                                                           TOEDNCI   838
                4 IF(NPB.LT.52) RETURN                                             TOEDNCI   839
                5 ITEL=ITEL+1                                                      TOEDNCI   840
                  WRITE(2,10) ITEL                                                 TOEDNCI   841
               10 FORMAT(1H1//16X,"TABEL B 3",24X,"PAG",I3//16X,"OPGAVE NIET UITRUIL TOEDNCI   842
 10              1BARE KAVELS"//16X,"GESORTEERD VOLGENS BEDRIJFSNUMMER"//18X,"BEDRNR TOEDNCI   843
                 2 KAV     OPP VAK"/)                                             TOEDNCI   844
                  NPB=0                                                            TOEDNCI   845
                  RETURN                                                           TOEDNCI   846
                  END                                                              TOEDNCI   847


 SUBROUTINE PRINTB4    74/175  OPT=1                    FTN 4.6+452        28/08/79  11.26.49        PAGE      1

 1                SUBROUTINE PRINTB4 (NPB,ITEL)                                    TOEDNCI   848
                  IF(NPB.EQ.0) GOTO 5                                              TOEDNCI   849
                  IF(ITEL.GT.1) GOTO 4                                             TOEDNCI   850
                  IF(NPB.LT.55) RETURN                                             TOEDNCI   851
 5                GOTO 5                                                           TOEDNCI   852
                4 IF(NPB.LT.54) RETURN                                             TOEDNCI   853
                5 ITEL=ITEL+1                                                      TOEDNCI   854
                  WRITE(2,10) ITEL                                                 TOEDNCI   855
               10 FORMAT(1H1//16X,"TABEL B 4",19X,"PAG",I3//16X,"NIET UITRUILBARE OP TOEDNCI   856
 10              1P PER VAK"//16X,"VAKNR",6X,"BEDRNR",5X,"OPP"/)                   TOEDNCI   857
                  NPB=0                                                            TOEDNCI   858
                  RETURN                                                           TOEDNCI   859
                  END                                                              TOEDNCI   860


 SUBROUTINE PRINTB5    74/175  OPT=1                    FTN 4.6+452        28/08/79  11.26.49        PAGE      1

 1                SUBROUTINE PRINTB5 (NPB,ITEL)                                    TOEDNCI   861
                  IF(NPB.EQ.0) GOTO 5                                              TOEDNCI   862
                  IF(ITEL.GT.1) GOTO 4                                             TOEDNCI   863
                  IF(NPB.LT.55) RETURN                                             TOEDNCI   864
 5                GOTO 5                                                           TOEDNCI   865
                4 IF(NPB.LT.54) RETURN                                             TOEDNCI   866
                5 ITEL=ITEL+1                                                      TOEDNCI   867
                  WRITE(2,10) ITEL                                                 TOEDNCI   868
               10 FORMAT(1H1//16X,"TABEL B 5",17X,"PAG",I3//16X,"VAKOPPERVLAKTEN"//1 TOEDNCI   869
 10              19X,"VAKNR",7X,"OPP  RUILKL"/)                                   TOEDNCI   870
                  NPB=0                                                            TOEDNCI   871
                  RETURN                                                           TOEDNCI   872
                  END                                                              TOEDNCI   873
```

```
      SUBROUTINE INDCOL                                                 TOEDNCI   874
      IMPLICIT INTEGER (A-Z)                                            TOEDNCI   875
      REAL R,COPY                                                       TOEDNCI   876
      DIMENSION LST(25),IND(4000)                                       TOEDNCI   877
      COMMON /INV/ IK(4000),IK1(4000),R(4000)  /SORT/ ILR,IUR           TOEDNCI   878
      DO 10 I = ILR,IUR                                                 TOEDNCI   879
   10 IND(I) = I                                                        TOEDNCI   880
      IS=0  $  LL=ILR  $  UL=IUR                                        TOEDNCI   881
  100 IP=LL $ M=IND(IP) $ X=IK(M)                                       TOEDNCI   882
      IQ=UL $ N=IND(IQ) $ Z=IK(N)                                       TOEDNCI   883
         IF (X.LE.Z) GOTO 110                                           TOEDNCI   884
         K=M  $  Y=X                                                    TOEDNCI   885
         IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y                    TOEDNCI   886
  110    IF ((UL-LL).LE.1) GOTO 600                                     TOEDNCI   887
            XX=X  $  IX=IP  $  ZZ=Z  $  IZ=IQ                           TOEDNCI   888
  200       IP=IP+1  $  IF (IP.GE.IQ) GOTO 210                          TOEDNCI   889
      M=IND(IP) $ X=IK(M) $ IF(X-XX) 200,300,300                        TOEDNCI   890
  210       IP=IQ-1  $  GOTO 500                                        TOEDNCI   891
  300       IQ=IQ-1  $  IF (IQ.LE.IP) GOTO 310                          TOEDNCI   892
      N=IND(IQ) $ Z=IK(N) $ IF(ZZ-Z) 300,400,400                        TOEDNCI   893
  310 IP=IP-1 $ Z=X $ N=M $ M=IND(IP) $ X=IK(M)                         TOEDNCI   894
  400       IF (X.LE.Z) GOTO 410                                        TOEDNCI   895
            K=M  $  Y=X                                                 TOEDNCI   896
            IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y                 TOEDNCI   897
  410       IF (X.LE.XX) GOTO 420                                       TOEDNCI   898
            XX=X  $  IX=IP                                              TOEDNCI   899
  420       IF (Z.GE.ZZ) GOTO 430                                       TOEDNCI   900
            ZZ=Z  $  IZ=IQ                                              TOEDNCI   901
  430       GOTO 200                                                    TOEDNCI   902
  500       IF (IP.LE.IX .OR. X.GE.XX) GOTO 510                         TOEDNCI   903
            IND(IP)=IND(IX)  $  IND(IX)=M                               TOEDNCI   904
  510       IF (IQ.GE.IZ .OR. Z.LE.ZZ) GOTO 520                         TOEDNCI   905
            IND(IQ)=IND(IZ)  $  IND(IZ)=N                               TOEDNCI   906
  520       IF ((UL-IQ).GT.(IP-LL)) GOTO 530                            TOEDNCI   907
            IX=LL  $  IZ=IP-1  $  LL=IQ+1  $  GOTO 540                  TOEDNCI   908
  530       IX=IQ+1  $  IZ=UL  $  UL=IP-1                               TOEDNCI   909
  540       IF (IX.GE.IZ) GOTO 550                                      TOEDNCI   910
            IS=IS+2  $  LST(IS-1)=IX  $  LST(IS)=IZ                     TOEDNCI   911
  550       IF (UL.GT.LL) GOTO 100                                      TOEDNCI   912
  600    IF (IS.EQ.0) GOTO 700                                          TOEDNCI   913
         LL=LST(IS-1)  $  UL=LST(IS)  $  IS=IS-2  $  GOTO 100           TOEDNCI   914
  700 CONTINUE                                                          TOEDNCI   915
      DO 120  I= ILR, IUR                                               TOEDNCI   916
         IF (IND(I)) 120,150,125                                        TOEDNCI   917
  125    K=I                                                           TOEDNCI   918
      CO=IK(K) $ COP=IK1(K) $ COPY=R(K)                                 TOEDNCI   919
  130    J=IND(K)  $  IF (J - 1) 150,140,225                           TOEDNCI   920
  225 IK(K)=IK(J) $ IK1(K)=IK1(J) $  R(K)= R(J)                         TOEDNCI   921
         IND(K)= -1 $ K=J                                               TOEDNCI   922
      GOTO 130                                                          TOEDNCI   923
  140 IK(K)=CO $ IK1(K)=COP $  R(K)=COPY                                TOEDNCI   924
         IND(K)= -1                                                     TOEDNCI   925
  120 CONTINUE                                                          TOEDNCI   926
      RETURN                                                            TOEDNCI   927
  150    STOP "PERMERROR"                                              TOEDNCI   928
      END                                                               TOEDNCI   929
```

```
      PROGRAM VAKAFS (TAPE1,TAPE2,OUTPUT,TAPE3=OUTPUT,TAPE4,TAPE5,
     1 TAPE6,TAPE7,TAPE8)
      IMPLICIT INTEGER (A-Z)
      DIMENSION X(5),Y(5),C(4),BE(4),FIFO(2000),FIBA(2000),D(2000),
     1 R(2000),NEFO(2200),NEBA(2200),BAR(2200),EAR(2200),LAR(2200),
     2 ONTSLPP(2000),LRO(2000),TODO(1450),TREEC(4000),MA(10),NA(10),
     3 INDEX(4000),BDR(3),VAK(2),KFF(16),KOST(4000)
      LOGICAL LIST (2000)
      COMMON /TREE/ TREEK(2000,2) /SORTVAR/ ILR,IUR,ILC,IUC
     2 /AFST/ IA(500)
      DATA TREEC/4000*0/,BAR/2200*0/,EAR/2200*0/,LAR/2200*0/,
     1 ONTSLPP/2000*0/,PTRC,PTRP,PTRK,NRC,RNR2,WNR2/6*0/,
     2 NC/2000/,NP/2200/

C     INLEZEN WEGENNET (VAN DATAW)
  100 WNR1=WNR2
      READ(1,102) X(1),Y(1),BE(1),W,(X(I),Y(I),BE(I),I=2,4),RNR2,WNR2
  102 FORMAT(2I8,2X,A1,I1,3(2I8,2X,A1,1X),2X,I4,I2)
      IF (EOF(1).NE.0) GO TO 200
      DO 110 I=1,4
      C(I)=X(I)*1000000+Y(I)
      IF (BE(1).NE.1HE) GO TO 110
      L=I
      GO TO 120
  110 CONTINUE
      L=4
  120 CALL FILLCTR(C(1),TREEC,NC,NRC,PTRC)
      IF (BE(1).EQ.1HB) GO TO 130
      CALL FILLARR(LASTC,C(1),LASTW,PTRP,TREEC,BAR,EAR,LAR,NP)
      GO TO 140
  130 WNR1=0
  140 IF (WNR1+1 .NE. WNR2) CALL ERROR(1,RNR2*100+WNR2)
      IF (L.EQ.1) GO TO 100
      DO 150 I=2,L
      CALL FILLCTR(C(I),TREEC,NC,NRC,PTRC)
  150 CALL FILLARR(C(I-1),C(I),W,PTRP,TREEC,BAR,EAR,LAR,NP)
      IF (BE(L).EQ.1HE) GO TO 100
      LASTC=C(4) $ LASTW=W
      GO TO 100

C     INLEZEN BEDRIJFSGEGEVENS
  200 PTRK=RU=NL=KL=0
      READ(5,206) AA,AB,AC,Z
  206 FORMAT(3A10,I5)
      READ(5,201)(MA(I),I=1,10),(NA(I),I=1,10)
  201 FORMAT(20I5)
      REWIND5
      AR=4000  $  AG=4000
      CALL OPENMS (6,INDEX,AG,0)
      CALL OPENMS (7,KOST,AR,0)
  202 RU=RU+1 $           IF(MA(RU).EQ.0.OR.RU.EQ.11) GOTO 660
      PRINT 20,RU
   20 FORMAT(1H ,"RUILKLASSE",I3)
      NB=NL+1 $ NL=NL+NA(RU)
  203 READ(2,204) BEDRNR,X1,Y1
  204 FORMAT(3I8)
      IF(EOF(2)) 210,205
```

```
      205 DO 207 I = NB,NL
          CALL READMS (6,BDR,3,I)
          IF(BEDRNR.NE.BDR(1)) GOTO 207
          LAST=X1*1000000+Y1 $ PTRK=PTRK+1
          TREEK(PTRK,1)=BEDRNR $ TREEK(PTRK,2)= LAST
          GOTO203
      207 CONTINUE
          GOTO203
      210 ABDR=PTRK
          REWIND 2
          NB=NL+1 $ NL=NL+MA(RU)
      211 READ(4,204) VAKNR,X1,Y1
          IF(EOF(4)) 230,212
      212 DO 220 I = NB,NL
          CALL READMS (6,VAK,2,I)
          IF(VAKNR.NE.VAK(1)) GOTO 220
          LAST=X1*1000000+Y1 $ PTRK=PTRK+1
          TREEK(PTRK,1)=VAKNR $ TREEK(PTRK,2)=LAST
          GOTO211
      220 CONTINUE
          GOTO211
      230 AVAK=PTRK-ABDR  $  PTRK=0  $  REWIND 4

C     VERWERKING INGELEZEN GEGEVENS
C     INITIALISEREN EN SORTEREN
          DO 501 I=1,NRC
      501 FIFO(I)=FIBA(I)=0
          DO 502 K=1,PTRP
          I=BAR(K) $ NEFO(K)=FIFO(I) $ FIFO(I)=K
          I=EAR(K) $ NEBA(K)=FIBA(I) $ FIBA(I)=K
      502 CONTINUE
          ILR = 1 $ IUR = ABDR $ ILC = 1 $ IUC = 2
          CALL SORT

C     BEREKENING KORTSTE AFSTANDEN
          IND1=1 $ NR=AN=0
          DO 601 I = 1,NRC
      601 LIST(I)=.FALSE.
          VB=ABDR+1 $ VL=ABDR+AVAK
          DO 630 I = VB,VL
          CALL NUMBER(N,TREEK(I,2),TREEC),RETURNS(630) $ LIST(N)=.TRUE.
          NR=NR+1
          ONTSLPP(I)=N
      630 CONTINUE
          AN=NR $  JL=NL-MA(RU)-NA(RU)
      631 NR=AN
          CALL NUMBER(ONTSLPB,TREEK(IND1,2),TREEC),RETURNS(650)
          CALL SYMDIST(Z,NRC,ONTSLPB,NR,LIST,FIFO,FIBA,NEFO,NEBA,D,R,
         C TODO,BAR,EAR,LAK,LRO)
          CALL RESULT (Z,D,LRO,ONTSLPP,VB,VL)
          CALL KOSMAT (AVAK,JL,KL)
      650 IF(IND1.EQ.ABDR) GOTO 202
          IND1=IND1+1  $  GOTO 631
      660 WRITE(3,720) AA,AB,AC
      720 FORMAT(1H1," RUILVERKAVELING :",3A10//"  KOSTENMATRICES"//)
          ENDFILE8  $  REWIND8  $  RU=TLR=0
      725 RU=RU+1  $  TLR=0  $  IF(MA(RU).EQ.0.OR.RU.EQ.11) GOTO 820
```

```
115             WRITE(3,730) RU
              730 FORMAT (1HO," RUILKLASSE :",I4/)
              735 L=-15
              740 L=L+16 $ K=L+15 $ IF(K.LE.MA(RU)) GOTO 750 $ K=MA(RU)
              750 IJ=K-L+1  $  TLR=TLR+1
120           CALL READMS (7,KFF,IJ,TLR)
              WRITE(3,810)(KFF(I),I=1,IJ)
              810 FORMAT(1H ,1616)
              IF(K.LT.MA(RU)) GOTO 740
              TLB=TLB+1  $  IF(TLB.LT.NA(RU)) GOTO 735
125           GOTO 725
              820 CALL EXIT
              END
```



```
1             SUBROUTINE ERROR(NR,VAR),RETURNS(A)
              IMPLICIT INTEGER (A-Z)

              GO TO (10,20,30,40,50) NR
5           10 WRITE (3,11) VAR
            11 FORMAT(* ----- FOUT IN VOLGNUMMER WEGENNET *,I7,/)
              CALL EXIT
            20 WRITE (3,21) VAR
            21 FORMAT(* ----- FOUT IN INTERNE NUMMERING COORDINATEN *,I12,/)
10            CALL EXIT
            30 WRITE(3,31) VAR
            31 FORMAT(* ----- PUNT OP DE WEG BIJ PERCEEL OF BOERDERIJ*,
             F * NIET VERMELD IN WEGENNET *,I12,/)
              RETURN A
15          40 WRITE(3,41) VAR
            41 FORMAT(* ----- DE BOVENGRENS VOOR HET AANTAL PUNTEN IN HET*,
             F * WEGENNET IS TE KLEIN *,I6,/)
              CALL EXIT
            50 WRITE(3,51) VAR
20          51 FORMAT(* ----- DE BOVENGRENS VOOR HET AANTAL PIJLEN IN HET*,
             F * WEGENNET IS TE KLEIN *,I6,/)
              CALL EXIT
              END
```

```
 SUBROUTINE RESULT     74/175  OPT=1                    FTN 4.6+452        28/08/79  13.08.24        PAGE      1

 1                SUBROUTINE RESULT (Z,D,LRO,ONTSLPP,VB,VL)
                  IMPLICIT INTEGER (A-Z)
                  DIMENSION D(1),LRO(1),ONTSLPP(1)
                  COMMON /AFST/ IA(500)
 5
                  VNR=0
                  DO 400 IND = VB,VL
                  VNR=VNR+1
                  J=ONTSLPP(IND)
10                IF(D(J).EQ.Z) GOTO 330
                  IA(VNR)=LRO(J)
                  GOTO 400
                330 IA(VNR)=99999
                400 CONTINUE
15                M=VL+1-VB  $  L=-15
                500 L=L+16  $  K=L+15  $  IF(K.LE.M) GOTO 520  $  K=M
                520 WRITE(8,540) (IA(I),I=L,K)
                540 FORMAT(1615)
                    IF(K.LT.M) GOTO 500
20                  RETURN
                    END
```

```
 SUBROUTINE NUMBER     74/175  OPT=1                    FTN 4.6+452        28/08/79  13.08.24        PAGE      1

 1                SUBROUTINE NUMBER(NR,C,TREEC),RETURNS(A)
                  IMPLICIT INTEGER (A-Z)
                  DIMENSION TREEC(1)

 5                ADDR=1
                100 CTRIAL=TREEC(ADDR)
                    IF (C-CTRIAL) 110,200,120
                110 ADDR=TREEC(ADDR+1)/10000 $ GO TO 130
                120 ADDR=MOD(TREEC(ADDR+1),10000)
10              130 IF (ADDR .EQ. 0) CALL ERROR(3,C),RETURNS(210)
                    GO TO 100
                200 NR=(ADDR+1)/2
                    RETURN
                210 RETURN A
15                  END
```

```
1                SUBROUTINE SORT
                 IMPLICIT INTEGER (A-Z)
                 DIMENSION LST(25)
                 COMMON /TREE/ TREEK(2000,2)
5                COMMON /SORTVAR/ ILR,IUR,ILC,IUC
                 COMMON /INDEX/ IND(2000)

                 DO 10 I = ILR,IUR
              10 IND(I) = I
10               IS=0  $  LL=ILR  $  UL=IUR
             100 IP=LL  $  M=IND(IP)  $  X=TREEK(M,ILC)
                    IQ=UL  $  N=IND(IQ)  $  Z=TREEK(N,ILC)
                    IF (X.LE.Z) GOTO 110
                    K=M  $  Y=X
15                  IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y
             110    IF ((UL-LL).LE.1) GOTO 600
                       XX=X  $  IX=IP  $  ZZ=Z  $  IZ=IQ
             200       IP=IP+1  $  IF (IP.GE.IQ) GOTO 210
                       M=IND(IP)  $  X=TREEK(M,ILC)  $  IF (X-XX)  200, 300, 300
20           210       IP=IQ-1  $  GOTO 500
             300       IQ=IQ-1  $  IF (IQ.LE.IP) GOTO 310
                       N=IND(IQ)  $  Z=TREEK(N,ILC)  $  IF (ZZ-Z)  300, 400, 400
             310       IP=IP-1  $  Z=X  $  N=M  $  M=IND(IP)  $  X=TREEK(M,ILC)
             400       IF (X.LE.Z) GOTO 410
25                     K=M  $  Y=X
                       IND(IP)=M=N  $  X=Z  $  IND(IQ)=N=K  $  Z=Y
             410       IF (X.LE.XX) GOTO 420
                       XX=X  $  IX=IP
             420       IF (Z.GE.ZZ) GOTO 430
30                     ZZ=Z  $  IZ=IQ
             430       GOTO 200
             500       IF (IP.LE.IX .OR. X.GE.XX) GOTO 510
                       IND(IP)=IND(IX)  $  IND(IX)=M
             510       IF (IQ.GE.IZ .OR. Z.LE.ZZ) GOTO 520
35                     IND(IQ)=IND(IZ)  $  IND(IZ)=N
             520       IF ((UL-IQ).GT.(IP-LL)) GOTO 530
                       IX=LL  $  IZ=IP-1  $  LL=IQ+1  $  GOTO 540
             530       IX=IQ+1  $  IZ=UL  $  UL=IP-1
             540       IF (IX.GE.IZ) GOTO 550
40                     IS=IS+2  $  LST(IS-1)=IX  $  LST(IS)=IZ
             550       IF (UL.GT.LL) GOTO 100
             600    IF (IS.EQ.0) GOTO 700
                    LL=LST(IS-1)  $  UL=LST(IS)  $  IS=IS-2  $  GOTO 100
             700 CONTINUE
45               CALL PERMMAT
                 END
```

```
 SUBROUTINE PERMMAT    74/175  OPT=1                    FTN 4.6+452        28/08/79  13.08.24        PAGE      1

 1               SUBROUTINE PERMMAT
                 IMPLICIT INTEGER (A-Z)
                 DIMENSION COPY(3)
                 COMMON /TREE/ TREEK(2000,2)
 5               COMMON /SORTVAR/ ILR,IUR,ILC,IUC
                 COMMON /INDEX/ IND(2000)

                 DO 400  I= ILR, IUR
                    IF (IND(I)) 400,500,125
10           125    K=I
                    DO 150 N= ILC, IUC
                150 COPY(N)=TREEK(K,N)
                200 J=IND(K)  $  IF (J - I) 500,300,225
                225 DO 250 N= ILC, IUC
15              250 TREEK(K,N)=TREEK(J,N)
                    IND(K)= -1 $ K=J
                    GOTO 200
                300 DO 350 N= ILC, IUC
                350 TREEK(K,N)=COPY(N)
20                  IND(K)= -1
                400 CONTINUE
                    RETURN
                500 STOP "PERMERROR"
                    END
```

```
 SUBROUTINE KOSMAT     74/175  OPT=1                    FTN 4.6+452        28/08/79  13.08.24        PAGE      1

 1               SUBROUTINE KOSMAT (M,JL,KL)
                 DIMENSION KF(500),IAG(3),KFF(16)
                 COMMON /AFST/ IA(500)
                 EQUIVALENCE (IA,KF)
 5
                 JL=JL+1
                 CALL READMS (6,IAG,3,JL)
                 AG=FLOAT(IAG(3))
                 IF(AG.EQ.0) AG=.1
10               DO 203 I = 1,M
                 IF(IA(I).EQ.99999) GOTO 105
                 A=FLOAT(IA(I))
                 IF(A.EQ.0) A=.1
                 IC=IFIX(SQRT(AG/A)*100.+.5)
15               KF(I)=IC+IA(I)     $   GOTO 203
                105 KF(I)=99999
                203 CONTINUE
                    L=-15
                  4 L=L+16 $ K=L+15 $ IJ=0 $ IF(K.LE.M) GOTO 5 $ K=M
20                5 DO 204 I = L,K
                    IJ=IJ+1
                204 KFF(IJ)=KF(I)
                    KL=KL+1
                    CALL WRITMS (7,KFF,IJ,KL)
25                  IF(K.LT.M) GOTO 4
                    RETURN
                    END
```

```
      PROGRAM INSTEP(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4)                INSTEP      2
      IMPLICIT INTEGER (A-Z)                                              INSTEP      3
      DIMENSION ITEL(10)                                                  INSTEP      4
      DIMENSION KFF(16)                                                   INSTEP      5
      DIMENSION BDR(3),VAK(2),NAAM(3),MA(10),NA(10),INDEX(4000),KUST(4,2  INSTEP      6
     10000),KOS(4000),IBDR(4,500),IOPPB(4,500),IVAK(4,50),IOPPV(4,50)     INSTEP      7
      READ(2,100)(NAAM(M),M=1,3)                                          INSTEP      8
  100 FORMAT(3A10)                                                        INSTEP      9
      READ(2,101)(MA(M),M=1,10),(NA(M),M=1,10)                            INSTEP     10
  101 FORMAT(2015)                                                        INSTEP     11
      CALL OPENMS(1,INDEX,4000,0)                                         INSTEP     12
      CALL OPENMS(3,KOS,4000,0)                                           INSTEP     13
      DO 1 I=1,10                                                         INSTEP     14
      IF(MA(1).EQ.0)2,1                                                   INSTEP     15
    1 CONTINUE                                                            INSTEP     16
      IAAN=I                                                              INSTEP     17
      GOTO 3                                                              INSTEP     18
    2 IAAN=I-1                                                            INSTEP     19
    3 NS=NL=0                                                             INSTEP     20
      J=0                                                                 INSTEP     21
   10 J=J+1                                                               INSTEP     22
      IF(J.GT.IAAN)GOTO 20                                                INSTEP     23
      NB=NL+1                                                             INSTEP     24
      NL=NL+NA(J)                                                         INSTEP     25
      NS=0                                                                INSTEP     26
      DO 200 I=NB,NL                                                      INSTEP     27
      NS=NS+1                                                             INSTEP     28
      CALL READMS(1,BDR,3,I)                                              INSTEP     29
      IBDR(J,NS)=BDR(1)                                                   INSTEP     30
      IOPPB(J,NS)=BDR(2)                                                  INSTEP     31
  200 CONTINUE                                                            INSTEP     32
      MB=NL+1                                                             INSTEP     33
      NL=NL+MA(J)                                                         INSTEP     34
      NS=0                                                                INSTEP     35
      DO 210 I=MB,NL                                                      INSTEP     36
      NS=NS+1                                                             INSTEP     37
      CALL READMS(1,VAK,2,I)                                              INSTEP     38
      IVAK(J,NS)=VAK(1)                                                   INSTEP     39
      IOPPV(J,NS)=VAK(2)                                                  INSTEP     40
  210 CONTINUE                                                            INSTEP     41
      GOTO 10                                                             INSTEP     42
   20 J=0                                                                 INSTEP     43
      ITLR=0                                                              INSTEP     44
  300 J=J+1                                                               INSTEP     45
      IF(J.GT.IAAN)GOTO 999                                               INSTEP     46
      ITLB=0                                                              INSTEP     47
      NS=0                                                                INSTEP     48
  305 L=-15                                                               INSTEP     49
  310 L=L+16                                                              INSTEP     50
      K=L+15                                                              INSTEP     51
      IF(K.LT.MA(J))GOTO 320                                              INSTEP     52
      K=MA(J)                                                             INSTEP     53
  320 IJ=K-L+1                                                            INSTEP     54
      ITLR=ITLR+1                                                         INSTEP     55
      CALL READMS(3,KFF,IJ,ITLR)                                          INSTEP     56
      DO 321 I=1,IJ                                                       INSTEP     57
      NS=NS+1                                                             INSTEP     58
```

Bijlage 16.1

```
      KOST(J,NS)=KFF(I)                                                 INSTEP     59
    321 CONTINUE                                                        INSTEP     60
60      IF(K.LT.MA(J))GOTO 310                                          INSTEP     61
        ITLB=ITLB+1                                                     INSTEP     62
        IF(ITLB.LT.NA(J))GOTO 305                                       INSTEP     63
        ITEL(J)=NS                                                      INSTEP     64
        GOTO 300                                                        INSTEP     65
65  999 CONTINUE                                                        INSTEP     66
        IOON=IAAN                                                       INSTEP     67
        DO 40 I=1,IAAN                                                  INSTEP     68
        IF(MA(I).EQ.1)41,40                                             INSTEP     69
     41 PRINT 42,I,MA(I),NA(I)                                          INSTEP     70
70   42 FORMAT(1H1,//,1X,*MATRIX VAN RUILKLASSE :*,I5,3X,*IS GEDEGENEREERD INSTEP     71
       1, MATRIXGROOTTE =*,2X,I4,2X,1H*,2X,I4)                          INSTEP     72
        PRINT 43,IVAK(I,1),IOPPV(I,1)                                   INSTEP     73
     43 FORMAT(1H0,*TOEDELING VAN GROND UIT VAKNUMMER :*,I8,*  NAAR DE VOL INSTEP     74
       1GENDE BEDRIJVEN ,TOTALE OPPERVLAKTE =*,I8)                      INSTEP     75
75      PRINT 44                                                        INSTEP     76
     44 FORMAT(1H0,5X,*BEDRIJFSNUMMER*,5X,*OPPERVLAKTE*)                INSTEP     77
        PRINT 45                                                        INSTEP     78
     45 FORMAT(1H ,35(*-*))                                             INSTEP     79
        INA=NA(I)                                                       INSTEP     80
80      DO 46 J=1,INA                                                   INSTEP     81
        PRINT 47,IBDR(I,J),IOPPB(I,J)                                   INSTEP     82
     47 FORMAT(1H ,8X,I8,I8,9X,I8)                                      INSTEP     83
     46 CONTINUE                                                        INSTEP     84
        IOON=IAAN-1                                                     INSTEP     85
85   40 CONTINUE                                                        INSTEP     86
        IF(IOON.EQ.0)GOTO 997                                           INSTEP     87
        WRITE(4,51)IOON                                                 INSTEP     88
     51 FORMAT(I5)                                                      INSTEP     89
        DO 50 I=1,IAAN                                                  INSTEP     90
90      IF(MA(I).EQ.1)GOTO 50                                           INSTEP     91
        ITY=900                                                         INSTEP     92
        IST=0                                                           INSTEP     93
        WRITE(4,52)ITY,IST,MA(I),NA(I)                                  INSTEP     94
     52 FORMAT(I10,I5,45X,I3,I4)                                        INSTEP     95
95      WRITE(4,53)I                                                    INSTEP     96
     53 FORMAT(I5)                                                      INSTEP     97
        K=0                                                             INSTEP     98
        IS=0                                                            INSTEP     99
        DO 54 IJ=1,20                                                   INSTEP    100
100     IF(IS.EQ.1)GOTO 58                                              INSTEP    101
        L=K+1                                                           INSTEP    102
        K=K+16                                                          INSTEP    103
        IF(K.GE.MA(I))55,56                                             INSTEP    104
     55 K=MA(I)                                                         INSTEP    105
105     IS=1                                                            INSTEP    106
     56 WRITE(4,57)(IOPPV(I,NI),NI=L,K)                                 INSTEP    107
     57 FORMAT(16I5)                                                    INSTEP    108
     54 CONTINUE                                                        INSTEP    109
     58 K=0                                                             INSTEP    110
110     IS=0                                                            INSTEP    111
        DO 540 IJ=1,20                                                  INSTEP    112
        IF(IS.EQ.1)GOTO 580                                             INSTEP    113
        L=K+1                                                           INSTEP    114
        K=K+16                                                          INSTEP    115
```

```
115              IF(K.GE.MA(I))550,560                                  INSTEP   116
                550 K=MA(I)                                            INSTEP   117
                    IS=1                                               INSTEP   118
                560 WRITE(4,57)(IVAK(I,NI),NI=L,K)                     INSTEP   119
                540 CONTINUE                                           INSTEP   120
120             580 K=0                                                INSTEP   121
                    IS=0                                               INSTEP   122
                    DO 59 IJ=1,50                                      INSTEP   123
                    IF(IS.EQ.1)GOTO 60                                 INSTEP   124
                    L=K+1                                              INSTEP   125
125                 K=K+16                                             INSTEP   126
                    IF(K.GE.NA(I))61,62                                INSTEP   127
                 61 K=NA(I)                                            INSTEP   128
                    IS=1                                               INSTEP   129
                 62 WRITE(4,57)(IOPPB(I,NI),NI=L,K)                    INSTEP   130
130              59 CONTINUE                                           INSTEP   131
                 60 K=0                                                INSTEP   132
                    IS=0                                               INSTEP   133
                    DO 590 IJ=1,50                                     INSTEP   134
                    IF(IS.EQ.1)GOTO 600                                INSTEP   135
135                 L=K+1                                              INSTEP   136
                    K=K+16                                             INSTEP   137
                    IF(K.GE.NA(I))610,620                              INSTEP   138
                 610 K=NA(I)                                           INSTEP   139
                    IS=1                                               INSTEP   140
140              620 WRITE(4,570)(IBDR(I,NI),NI=L,K)                   INSTEP   141
                 570 FORMAT(16I8)                                      INSTEP   142
                 590 CONTINUE                                          INSTEP   143
                 600 CONTINUE                                          INSTEP   144
                    IP=JJ=0                                            INSTEP   145
145             4000 IMA=MA(I)                                         INSTEP   146
                    DO 1000 IJ=1,IMA                                   INSTEP   147
                    IKI=ITEL(I)-MA(I)+IJ                               INSTEP   148
                    IP=IP+1                                            INSTEP   149
                    JJ=IP+15*MA(I)                                     INSTEP   150
150                 IF(JJ.GE.IKI)JJ=IKI                                INSTEP   151
                    WRITE(4,2000)(KOST(I,M),M=IP,JJ,IMA)               INSTEP   152
                2000 FORMAT(16I5)                                      INSTEP   153
                1000 CONTINUE                                          INSTEP   154
                    IP=JJ                                              INSTEP   155
155                 IF(JJ.EQ.IKI)50,4000                               INSTEP   156
                  50 CONTINUE                                          INSTEP   157
                    GOTO 222                                           INSTEP   158
                 997 PRINT 998                                         INSTEP   159
                 998 FORMAT(1H1,//,1X,*GEEN ENKELE RUILKLASSE VOOR STEPPING STONE BEREK INSTEP   160
160                1ENING*)                                            INSTEP   161
                 222 CONTINUE                                          INSTEP   162
                    END                                                INSTEP   163
```

```
 1    "BEGIN" "COMMENT"          TRANSPORT ALGOL MC PROGRAMMA ROB 28MRT.77;      STSTONE     2
 2                                                                               STSTONE     3
 3    "INTEGER"CM,RM,CN,RN,II,JJ,PIM,KK;                                         STSTONE     4
 4       "INTEGER" AAN,HAAN,ROB,HULP;                                            NEW         1
 5    "INTEGER" RPUNT,RNUL,R9,RMIN,RA;                                            STSTONE     5
 6    "INTEGER"CHAN,M,N,I,J,AI,BJ,ST,UP,ACTI;                                     STSTONE     6
 7      "REAL" AA,ANSWER,TIME;                                                    NEW         2
 8      "INTEGER""ARRAY"KAART(/1:128/);                                           NEW         3
 9    "BOOLEAN"ROWSOL;                                                            STSTONE     9
10    "REAL" "PROCEDURE" VELD(P,Q);"VALUE" P,Q; "INTEGER" P,Q;                    STSTONE    10
11    "BEGIN" "INTEGER" I,S,TELLER; "REAL" X; "BOOLEAN" PLUS,DECI;                STSTONE    11
12       TELLER.=0;  X.= 0; PLUS.= "TRUE"; DECI.= "FALSE";                        STSTONE    12
13       "FOR" I.= P "STEP" 1 "UNTIL" Q "DO"                                      STSTONE    13
14       "BEGIN" S.= KAART(/I/);                                                  STSTONE    14
15       "IF" S= RPUNT "THEN" DECI.= "TRUE" "ELSE"                                STSTONE    15
16       "IF" S= RMIN "THEN" PLUS.= "FALSE" "ELSE"                                STSTONE    16
17       "IF" S "NOTLESS" RNUL "AND" S "NOTGREATER" R9 "THEN"                     STSTONE    17
18    "BEGIN"X.=(S-RNUL)/RA+10*X;                                                 STSTONE    18
19          "IF" DECI "THEN" TELLER.= TELLER -1                                   STSTONE    19
20       "END"                                                                    STSTONE    20
21       "END" I;                                                                 STSTONE    21
22       VELD.= ("IF" PLUS "THEN" X "ELSE" -X)*                                   STSTONE    22
23             ("IF" DECI "THEN" 10"POWER" TELLER "ELSE" 1)                       STSTONE    23
24    "END" VELD;                                                                 STSTONE    24
25                                                                               STSTONE    25
26          "PROCEDURE"NEWPAGE;OUTPUT(61,"("*")");                               STSTONE    26
27          "PROCEDURE"PRINTTEXT(S);"STRING"S;OUTPUT(61,"("")",S);               STSTONE    27
28          "PROCEDURE"NLCR;OUTPUT(61,"("/")");                                  STSTONE    28
29          "PROCEDURE"FIXT(N,M,X);"VALUE"N,M,X;"INTEGER"N,M;"REAL"X;            STSTONE    29
30          "BEGIN""PROCEDURE"FORM;                                              STSTONE    30
31       "IF"M"NOT EQUAL"0"THEN"FORMAT("("XZD.XD,B")",N-1,M)                     NEW         4
32                      "ELSE"FORMAT("("XZD,B")",N-1);                           NEW         5
33                  "PROCEDURE"LIST(AB);"PROCEDURE"AB;                           STSTONE    33
34                  AB(X);                                                       STSTONE    34
35                  OUTLIST(61,FORM,LIST);                                       STSTONE    35
36          END..                                                                STSTONE    36
37          "END" FIXT;                                                          STSTONE    37
38                                                                               STSTONE    38
39    "PROCEDURE"CARD(CHAN);"INTEGER"CHAN;                                        STSTONE    39
40      INPUT(CHAN,"("128(A)")",KAART);                                          NEW         6
41                                                                               STSTONE    41
42    RPUNT.=EQUIV("(".")"); RNUL.=EQUIV("("0")");                                STSTONE    42
43    R9.=EQUIV("("9")"); RMIN.= EQUIV("("-")"); RA.= EQUIV("("A")");             STSTONE    43
44         CARD(10);                                                             NEW         7
45         AAN.=VELD(1,5);                                                       NEW         8
46       "FOR" HAAN:=1 "STEP" 1 "UNTIL"AAN"DO"                                   NEW         9
47       "BEGIN"                                                                 NEW        10
48    ROWSOL.="FALSE";                                                           STSTONE    44
49       CARD(10);                                                               NEW        11
50    TIME.=VELD(1,10);                                                          STSTONE    46
51    ST.=VELD(11,15);                                                           STSTONE    47
52    M.=VELD(61,63);                                                            STSTONE    48
53    N.=VELD(64,67);                                                            STSTONE    49
54    NEWPAGE;                                                                   NEW        12
55    NLCR;FIXT(3,0,M);FIXT(3,0,N);                                              STSTONE    50
56    FIXT(3,0,ST);                                                              STSTONE    51
57    UP.=M+N-1;ST.=0;                                                           STSTONE    52
58    "BEGIN"                                                                    STSTONE    53
59      "INTEGER""ARRAY"X,IX,JX(/1:UP/),A(/1:M/),B(/1:N/),VAK(/1:M/)             NEW        13
```

```
60        ,BEDR(/1:N/),                                                    NEW        14
61            U(/1:N/),V(/1:M/),C(/1:M,1:N/);                              NEW        15
62      "COMMENT" "CHECKON" BEDR,IX,U,V,JX;                                NEW        16
63                                                                         STSTONE    56
64    "REAL""ARRAY"D(/1:K/);                                               STSTONE    57
65                                                                         STSTONE    58
66                                                                         STSTONE    59
67    "INTEGER""PROCEDURE" CC(I,J);"INTEGER"I,J;                           STSTONE    60
68    "BEGIN"                                                              STSTONE    61
69    "IF" I "EQUAL" ACTI"THEN""GOTO"DOOR;                                 STSTONE    62
70    "IF" CLOCK"GREATER" TIME "THEN""GOTO" SLOT;                          STSTONE    63
71    DOOR:  CC.= C(/J,I/);                                                STSTONE    64
72    ACTI.=I;                                                             STSTONE    65
73    "END" CC;                                                            STSTONE    66
74                                                                         STSTONE    67
75     "REAL" "PROCEDURE"                                                  STSTONE    68
76     INTEGER TRANSPORT(M,N,I,J,AI,BJ,CIJ,U,V,X,IX,JX,ST,ROWSOL,INFEAS);  STSTONE    69
77     "VALUE" M,N,ST,ROWSOL;                                              STSTONE    70
78     "INTEGER" M,N,I,J,AI,BJ,CIJ,ST; "BOOLEAN" ROWSOL;                   STSTONE    71
79     "INTEGER" "ARRAY" X,IX,JX,U,V; "LABEL" INFEAS;                      STSTONE    72
80                                                                         STSTONE    73
81     "BEGIN" "INTEGER" E,F,G,G0,H,H0,I0,I1,J0,J1,K,NR,NS,P,Q,R,S,UBINT,UP; STSTONE  74
82       "REAL" Z; "BOOLEAN" W;                                            STSTONE    75
83       "INTEGER" "ARRAY" CHAIN[1:M+N],NEXTIX,NEXTJX[1:M+N-1],FIRSTX[1:N]; STSTONE   76
84       "BOOLEAN" "ARRAY" DIR[1:M+N-1];                                   STSTONE    77
85                                                                         STSTONE    78
86        "PROCEDURE" VECINDQSORT(V, IND, LVI, UVI);                       STSTONE    79
87            "VALUE" LVI, UVI; "INTEGER" LVI, UVI; "INTEGER" "ARRAY" IND; STSTONE    80
88            "ARRAY" V;                                                   STSTONE    81
89        "BEGIN" "INTEGER" P, Q, JX, JZ, H, N, K; "REAL" X, XX, Y, ZZ, Z; STSTONE    82
90                                                                         STSTONE    83
91            "PROCEDURE" VECINDSORT;                                      STSTONE    84
92            "BEGIN" "INTEGER" L, U; L:= LVI; U:= UVI;                    STSTONE    85
93                                                                         STSTONE    86
94            PART: P:= L; H:= IND[P]; X:= V[H]; Q:= U; N:= IND[Q]; Z:= V[N]; STSTONE  87
95                "IF" X > Z "THEN"                                        STSTONE    88
96                "BEGIN" K:= H; Y:= X;                                    STSTONE    89
97                    IND[P]:= H:= N; X:= Z; IND[Q]:= N:= K; Z:= Y         STSTONE    90
98                "END";                                                   STSTONE    91
99                                                                         STSTONE    92
100               "IF" U - L > 1 "THEN"                                    STSTONE    93
101               "BEGIN" XX:= X; JX:= P; ZZ:= Z; JZ:= Q;                  STSTONE    94
102                                                                        STSTONE    95
103               LEFT: "FOR" P:= P + 1 "WHILE" P < Q "DO"                 STSTONE    96
104                   "BEGIN" H:= IND[P]; X:= V[H];                        STSTONE    97
105       "IF" X "NOT LESS" XX "THEN" "GOTO" RIGHT                         NEW        17
106                   "END";                                               STSTONE    99
107                   P:= Q - 1; "GOTO" OUT;                               STSTONE   100
108                                                                        STSTONE   101
109               RIGHT: "FOR" Q:= Q - 1 "WHILE" Q > P "DO"                STSTONE   102
110                   "BEGIN" N:= IND[Q]; Z:= V[N];                        STSTONE   103
111       "IF" Z "NOT GREATER" ZZ "THEN" "GOTO" DIST                       NEW        18
112                   "END";                                               STSTONE   105
113                   Q:= P; N:= H; Z:= X; P:= P - 1; H:=IND[P]; X:= V[H]; STSTONE   106
114                                                                        STSTONE   107
115               DIST: "IF" X > Z "THEN"                                  STSTONE   108
116                   "BEGIN" K:= H; Y:= X;                                STSTONE   109
117                       IND[P]:= H:= N; X:= Z; IND[Q]:= N:= K; Z:= Y     STSTONE   110
118                   "END";                                               STSTONE   111
```

Bijlage 17.2

```
119                  "IF" X > XX "THEN" "BEGIN" XX:= X; JX:= P "END";
120                  "IF" Z < ZZ "THEN" "BEGIN" ZZ:= Z; JZ:= Q "END";
121                  "GOTO" LEFT;
122
123              OUT: "IF" P > JX "AND" X < XX "THEN"
124                  "BEGIN" IND[P]:= IND[JX]; IND[JX]:= H "END";
125                  "IF" Q < JZ "AND" Z > ZZ "THEN"
126                  "BEGIN" IND[Q]:= IND[JZ]; IND[JZ]:= N "END";
127                  "IF" U - Q > P - L "THEN"
128                  "BEGIN" LVI:= L; UVI:= P - 1; L:= Q + 1 "END" "ELSE"
129                  "BEGIN" UVI:= U; LVI:= Q + 1; U:= P - 1 "END";
130                  "IF" UVI > LVI "THEN" VECINDSORT;
131                  "IF" U > L "THEN" "GOTO" PART
132              "END" U - L > 1
133          "END" VECINDSORT;
134
135          "IF" UVI > LVI "THEN" VECINDSORT
136      "END" VECINDQSRT;
137     "PROCEDURE" HOR1(G); "INTEGER" G;
138     "BEGIN" "INTEGER" H;
139       H:= G;
140       "FOR" H:= NEXTJX[H] "WHILE" H^=G "DO"
141       "BEGIN" DIR[H]:= "FALSE";
142         "IF" JX[H]<0 "THEN" V[-JX[H]]:= CHAIN[H]-U[IX[H]] "ELSE"
143         "BEGIN" V[JX[H]]:= CHAIN[H]-U[ABS(IX[H])]; VERT1(H) "END"
144       "END"
145     "END";
146
147     "PROCEDURE" VERT1(H); "INTEGER" H;
148     "BEGIN" "INTEGER" G;
149       G:= H;
150       "FOR" G:= NEXTIX[G] "WHILE" G^=H "DO"
151       "BEGIN" DIR[G]:= "TRUE";
152         "IF" IX[G]<0 "THEN" U[-IX[G]]:= CHAIN[G]-V[JX[G]] "ELSE"
153         "BEGIN" U[IX[G]]:= CHAIN[G]-V[ABS(JX[G])]; HOR1(G) "END"
154       "END"
155     "END";
156
157     "PROCEDURE" HOR2(G); "INTEGER" G;
158     "BEGIN" "INTEGER" H;
159       H:= G;
160       "FOR" H:= NEXTJX[H] "WHILE" H^=G "DO"
161       "BEGIN"
162         "BEGIN" "INTEGER" J;
163           DIR[H]:= "FALSE"; J:= JX[H];
164           "IF" J<0 "THEN" "BEGIN" V[-J]:= V[-J]-Q; "GOTO" END "END";
165           V[J]:= V[J]-Q
166         "END";
167         VERT2(H);
168       END:
169       "END"
170     "END";
171
172     "PROCEDURE" VERT2(H); "INTEGER" H;
173     "BEGIN" "INTEGER" G;
174       G:= H;
175       "FOR" G:= NEXTIX[G] "WHILE" G^=H "DO"
176       "BEGIN"
177         "BEGIN" "INTEGER" I;
```

```
STSTONE  112
STSTONE  113
STSTONE  114
STSTONE  115
STSTONE  116
STSTONE  117
STSTONE  118
STSTONE  119
STSTONE  120
STSTONE  121
STSTONE  122
STSTONE  123
STSTONE  124
STSTONE  125
STSTONE  126
STSTONE  127
STSTONE  128
STSTONE  129
STSTONE  130
STSTONE  131
STSTONE  132
STSTONE  133
STSTONE  134
STSTONE  135
STSTONE  136
STSTONE  137
STSTONE  138
STSTONE  139
STSTONE  140
STSTONE  141
STSTONE  142
STSTONE  143
STSTONE  144
STSTONE  145
STSTONE  146
STSTONE  147
STSTONE  148
STSTONE  149
STSTONE  150
STSTONE  151
STSTONE  152
STSTONE  153
STSTONE  154
STSTONE  155
STSTONE  156
STSTONE  157
STSTONE  158
STSTONE  159
STSTONE  160
STSTONE  161
STSTONE  162
STSTONE  163
STSTONE  164
STSTONE  165
STSTONE  166
STSTONE  167
STSTONE  168
STSTONE  169
STSTONE  170
```

```
178             DIR[G]:= "TRUE"; I:= IX[G];                                   STSTONE  171
179             "IF" I<0 "THEN" "BEGIN" U[-1]:= U[-I]+0; "GOTO" END "END";    STSTONE  172
180             U[I]:= U[I]+0                                                 STSTONE  173
181           "END";                                                          STSTONE  174
182           HOR2(G);                                                        STSTONE  175
183         END;                                                              STSTONE  176
184         "END"                                                             STSTONE  177
185       "END";                                                              STSTONE  178
186                                                                           STSTONE  179
187       "PROCEDURE" SCREENST;                                               STSTONE  180
188       "BEGIN" "INTEGER" G;                                                STSTONE  181
189         "FOR" G:= 1 "STEP" 1 "UNTIL" ST "DO" "IF" X[G]=-2 "THEN"          STSTONE  182
190         "BEGIN"                                                           STSTONE  183
191         A1: "IF" X[ST]=-2 "THEN"                                          STSTONE  184
192           "BEGIN" ST:= ST-1; "IF" ST<G "THEN" "GOTO" A2; "GOTO" A1 "END"; STSTONE  185
193           X[G]:= X[ST]; IX[G]:= IX[ST]; JX[G]:= JX[ST]; ST:= ST-1;        STSTONE  186
194         A2:                                                               STSTONE  187
195         "END"                                                             STSTONE  188
196       "END";                                                              STSTONE  189
197                                                                           STSTONE  190
198       UP:= M+N-1; Z:= 0; G:= NR:= NS:= 0;                                 STSTONE  191
199       "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO"                                 STSTONE  192
200       "BEGIN" H:= NEXTIX[I]:= AI; "IF" H<0 "THEN" "GOTO" INFEAS;          STSTONE  193
201         "IF" G<H "THEN" G:= H; Z:= Z+H                                    STSTONE  194
202       "END";                                                              STSTONE  195
203       "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO"                                 STSTONE  196
204       "BEGIN" H:= NEXTJX[J]:= BJ; "IF" H<0 "THEN" "GOTO" INFEAS;          STSTONE  197
205         "IF" G<H "THEN" G:= H; Z:= Z-H                                    STSTONE  198
206       "END";                                                              STSTONE  199
207       "IF" Z^=0 "THEN" "GOTO" INFEAS; UBINT:= G+1;                        STSTONE  200
208       "IF" ST>0 "THEN"                                                    STSTONE  201
209       "BEGIN" "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO" U[I]:= 0;               STSTONE  202
210         "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO" V[J]:= 0                      STSTONE  203
211       "END";                                                              STSTONE  204
212                                                                           STSTONE  205
213       W:= "FALSE";                                                        STSTONE  206
214       "FOR" G:= 1 "STEP" 1 "UNTIL" ST "DO"                                STSTONE  207
215       "BEGIN" I:= IX[G]; J:= JX[G]; "IF" I<0"OR"I>M"OR"J<0"OR"J>N "THEN"  STSTONE  208
216         "BEGIN" X[G]:= -2; W:= "TRUE" "END" "ELSE"                        STSTONE  209
217         "BEGIN" X[G]:= -1; U[I]:= U[I]+1; V[J]:= V[J]+1 "END"             STSTONE  210
218       "END";                                                              STSTONE  211
219       "IF" W "THEN" SCREENST; "IF" ST<-1 "THEN" ST:= 0; I1:= M; J1:= N;   STSTONE  212
220       "IF" ST<1 "THEN" "GOTO" B1; E:= 0;                                  STSTONE  213
221                                                                           STSTONE  214
222     A1: G:= 0; W:= "FALSE";                                               STSTONE  215
223     A2: G:= G+1; "IF" X[G]^=-1 "THEN" "GOTO" A4; I:= IX[G]; J:= JX[G];    STSTONE  216
224       I0:= U[I]; J0:= V[J]; "IF" I0>1&J0>1 "THEN" "GOTO" A4;              STSTONE  217
225           W:= "TRUE";                                                     NEW       19
226         "IF" I0 "NOT LESS" 1&J0 "NOT LESS" 1 "THEN" "GOTO" A3;            NEW       20
227         X[G]:= -2;                                                      NEW       21
228       "IF" I0 "NOT LESS" 1 "THEN" U[I]:= I0-1;                          NEW       22
229         "IF" J0 "NOT LESS" 1 "THEN" V[J]:= J0-1;                        NEW       23
230       "GOTO" A4;                                                          STSTONE  220
231                                                                           STSTONE  221
232     A3: H:= NEXTIX[I]; K:= NEXTJX[J];                                     STSTONE  222
233           "IF" "IF" H^=K "THEN" H<K "ELSE" I1 "NOT LESS" J1 "THEN"        NEW       24
234       "BEGIN" X[G]:= H; NEXTIX[I]:= -1; U[I]:= 0; I1:= I1-1;              STSTONE  224
235         NEXTJX[J]:= K-H; V[J]:= J0-1                                      STSTONE  225
236       "END" "ELSE"                                                        STSTONE  226
```

```
237       "BEGIN" X[G]:= K; NEXTJX[J]:= -1; V[J]:= 0; J1:= J1-1;                STSTONE   227
238         NEXTIX[I]:= H-K; U[I]:= I0-1                                        STSTONE   228
239       "END";                                                                STSTONE   229
240       E:= E+1; "IF" E=ST "THEN" "GOTO" "IF" E=UP "THEN" B3 "ELSE" B1;       STSTONE   230
241       "IF" E=UP "THEN" "BEGIN" SCREENST; "GOTO" B3 "END";                   STSTONE   231
242     A4: "IF" G<ST "THEN" "GOTO" A2; "IF" W "THEN" "GOTO" A1;                STSTONE   232
243                                                                             STSTONE   233
244       W:= "TRUE"; F:= UBINT;                                                STSTONE   234
245       "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO" "IF" U[I]>1 "THEN"                STSTONE   235
246       "BEGIN" H:= NEXTIX[I];                                                STSTONE   236
247         "IF" H<F "THEN" "BEGIN" F:= H; I0:= I "END"                         STSTONE   237
248       "END";                                                                STSTONE   238
249       "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO" "IF" V[J]>1 "THEN"                STSTONE   239
250       "BEGIN" K:= NEXTJX[J];                                                STSTONE   240
251         "IF" K<F "THEN" "BEGIN" F:= K; J0:= J; W:= "FALSE" "END"            STSTONE   241
252       "END";                                                                STSTONE   242
253       "IF" F=UBINT "THEN" "BEGIN" SCREENST; "GOTO" B1 "END";                STSTONE   243
254       "FOR" G:= 1 "STEP" 1 "UNTIL" ST "DO" "IF" X[G]=-1 "THEN"              STSTONE   244
255       "BEGIN" "IF" ("IF" W "THEN" IX[G]=I0 "ELSE" JX[G]=J0) "THEN"          STSTONE   245
256         "GOTO" A5                                                           STSTONE   246
257       "END";                                                                STSTONE   247
258     A5: I:= IX[G]; J:= JX[G];                                               STSTONE   248
259       "IF" W "THEN" "BEGIN" I0:= 1; J0:= V[J] "END" "ELSE" I0:= U[I];       STSTONE   249
260       W:= "TRUE"; "GOTO" A3;                                                STSTONE   250
261                                                                             STSTONE   251
262        B1: I:= 0; G:= "IF" ST "NOT LESS" 1 "THEN" ST "ELSE" 0;              NEW        25
263     B2: I:= I+1; "IF" I>M "THEN" I:= 1; H:= NEXTIX[I];                      STSTONE   253
264       "IF" H=-1 "THEN" "GOTO" B2; NR:= NR+1; J0:= 0;                        STSTONE   254
265       "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO" "IF" NEXTJX[J]^=-1 "THEN"         STSTONE   255
266       "BEGIN" R:= CIJ; "IF" ST=-1 "THEN" R:= R-V[J];                        STSTONE   256
267         "IF" "IF" J0=0 "THEN" "TRUE" "ELSE" R<P "THEN"                      STSTONE   257
268         "BEGIN" J0:= J; P:= R "END"                                         STSTONE   258
269       "END";                                                                STSTONE   259
270                                                                             STSTONE   260
271       J:= J0; G:= G+1; IX[G]:= I; JX[G]:= J;                                STSTONE   261
272       S:= CHAIN[G]:= "IF" ST=-1 "THEN" CIJ "ELSE" P; K:= NEXTJX[J];         STSTONE   262
273         "IF" ("IF" H^=K "THEN" H<K "ELSE" I1 "NOT LESS" J1) "THEN"          NEW        26
274       "BEGIN" X[G]:= H; NEXTJX[J]:= K-H; NEXTIX[I]:= -1; Z:= Z+H*S;         STSTONE   264
275         I1:= I1-1                                                           STSTONE   265
276       "END" "ELSE"                                                          STSTONE   266
277       "BEGIN" X[G]:= K; NEXTIX[I]:= H-K; NEXTJX[J]:= -1; Z:= Z+K*S;         STSTONE   267
278         J1:= J1-1                                                           STSTONE   268
279       "END";                                                                STSTONE   269
280       "IF" G<UP "THEN" "GOTO" B2;                                           STSTONE   270
281                                                                             STSTONE   271
282     B3: "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO" U[I]:= 0;                       STSTONE   272
283       "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO" FIRSTX[J]:= 0;                    STSTONE   273
284       "FOR" G:= 1 "STEP" 1 "UNTIL" UP "DO"                                  STSTONE   274
285       "BEGIN" I:= IX[G]; E:= U[I];                                          STSTONE   275
286         "IF" E=0 "THEN" U[I]:= NEXTJX[G]:= G "ELSE"                         STSTONE   276
287         "BEGIN" NEXTJX[G]:= NEXTJX[E]; U[I]:= NEXTJX[E]:= G "END";          STSTONE   277
288         J:= JX[G]; E:= FIRSTX[J];                                           STSTONE   278
289         "IF" E=0 "THEN" FIRSTX[J]:= NEXTIX[G]:= G "ELSE"                    STSTONE   279
290         "BEGIN" NEXTIX[G]:= NEXTIX[E]; FIRSTX[J]:= NEXTIX[E]:= G "END"      STSTONE   280
291       "END";                                                                STSTONE   281
292                                                                             STSTONE   282
293       "IF" ST>0 "THEN"                                                      STSTONE   283
294       "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO"                                   STSTONE   284
295       "BEGIN" H:= U[I]; W:= "FALSE";                                        STSTONE   285
```

```
296       "FOR" G:= H,NEXTJX[G] "WHILE" G^=H "DO" "IF" G "NOT GREATER" ST      NEW        27
297       "THEN"                                                               NEW        28
298         "BEGIN" J:= JX[G]; S:= CHAIN[G]:= C1J; Z:= Z+X[G]*S; W:= "TRUE"    STSTONE   287
299         "END";                                                             STSTONE   288
300         "IF" W "THEN" NR:= NR+1                                            STSTONE   289
301       "END";                                                               STSTONE   290
302                                                                            STSTONE   291
303       "FOR" G:= 1 "STEP" 1 "UNTIL" UP "DO" "IF" NEXTIX[G]=G "THEN"         STSTONE   292
304       JX[G]:= -JX[G] "ELSE" "IF" NEXTJX[G]=G "THEN" IX[G]:= -IX[G];        STSTONE   293
305       G:= 1;                                                               STSTONE   294
306       "FOR" H:= G+1 "WHILE" IX[G]>0&JX[G]>0 "DO" G:= H;                    STSTONE   295
307       "IF" JX[G]<0 "THEN"                                                  STSTONE   296
308       "BEGIN" U[IX[G]]:= 0; V[-JX[G]]:= CHAIN[G]; DIR[G]:= "TRUE";         STSTONE   297
309         HOR1(G)                                                            STSTONE   298
310       "END" "ELSE"                                                         STSTONE   299
311       "BEGIN" V[JX[G]]:= 0; U[-IX[G]]:= CHAIN[G]; DIR[G]:= "FALSE";        STSTONE   300
312         VERT1(G)                                                           STSTONE   301
313       "END";                                                               STSTONE   302
314       I0:= M; CHAIN[1]:= 0;                                                STSTONE   303
315                                                                            STSTONE   304
316     C1:                                                                    NEW        29
317         NS:= NS+1; I:= I0; J0:= 0;                                         STSTONE   306
318     C2: NR:= NR+1; I:= "IF" I=M "THEN" 1 "ELSE" I+1; Q:= U[I];             STSTONE   307
319       "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO"                                  STSTONE   308
320       "BEGIN" P:= C1J-V[J];                                                STSTONE   309
321         "IF" P<0 "THEN" "BEGIN" J0:= J; Q:= P "END"                        STSTONE   310
322       "END";                                                               STSTONE   311
323       "IF" J0=0 "THEN" "GOTO" "IF" I=I0 "THEN" F1 "ELSE" C2; Q:= Q-U[I];   STSTONE   312
324                                                                            STSTONE   313
325       I0:= I; K:= 1; G:= FIRSTX[J0];                                       STSTONE   314
326     D1: G:= NEXTIX[G]; I:= IX[G];                                          STSTONE   315
327       "IF" ABS(I)=I0 "THEN" "BEGIN" K:= K+1; CHAIN[K]:= G; "GOTO" E1 "END";STSTONE   316
328       "IF" I<0 "THEN" "GOTO" D1;                                           STSTONE   317
329       "IF" G=CHAIN[K] "THEN" K:= K-1 "ELSE"                                STSTONE   318
330       "BEGIN" K:= K+1; CHAIN[K]:= G "END";                                 STSTONE   319
331     D2: G:= NEXTJX[G]; "IF" JX[G]<0 "THEN" "GOTO" D2;                      STSTONE   320
332       "IF" G=CHAIN[K] "THEN" K:= K-1 "ELSE"                                STSTONE   321
333       "BEGIN" K:= K+1; CHAIN[K]:= G "END";                                 STSTONE   322
334       "GOTO" D1;                                                           STSTONE   323
335                                                                            STSTONE   324
336     E1: F:= UBINI;                                                         STSTONE   325
337       "FOR" H:= K "STEP" -2 "UNTIL" 2 "DO"                                 STSTONE   326
338       "BEGIN" E:= X[CHAIN[H]];                                             STSTONE   327
339         "IF" E<F "THEN" "BEGIN" F:= E; H0:= H "END"                        STSTONE   328
340       "END";                                                               STSTONE   329
341       "IF" F=0 "THEN" "GOTO" E2; Z:= Z+F*Q;                                STSTONE   330
342       "FOR" H:= 2 "STEP" 2 "UNTIL" K "DO"                                  STSTONE   331
343       "BEGIN" G:= CHAIN[H]; X[G]:= X[G]-F "END";                           STSTONE   332
344       "FOR" H:= 3 "STEP" 2 "UNTIL" K "DO"                                  STSTONE   333
345       "BEGIN" G:= CHAIN[H]; X[G]:= X[G]+F "END";                           STSTONE   334
346                                                                            STSTONE   335
347     E2: G0:= CHAIN[H0]; I1:= ABS(IX[G0]); J1:= ABS(JX[G0]); X[G0]:= F;     STSTONE   336
348                                                                            STSTONE   337
349       "IF" I0^=I1 "THEN"                                                   STSTONE   338
350       "BEGIN" IX[G0]:= I0; F:= G:= CHAIN[H0+1];                            STSTONE   339
351         "FOR" G:= NEXTJX[G] "WHILE" G^=G0 "DO" F:= G;                      STSTONE   340
352         H:= NEXTJX[F]:= NEXTJX[G0]; "IF" F=H "THEN" IX[F]:= -I1;           STSTONE   341
353         F:= CHAIN[K]; G:= NEXTJX[G0]:= NEXTJX[F]; NEXTJX[F]:= G0;          STSTONE   342
354         "IF" F=G "THEN" IX[F]:= I0                                         STSTONE   343
```

```
473           "FOR"JJ.=1"STEP"1"UNTIL"RN"DO"                    STSTONE   453
474           "BEGIN"                                           STSTONE   454
475           PIM.=PIM+1;                                       STSTONE   455
476           B(/PIM/).=VELD((JJ-1)*5+1,JJ*5)                   STSTONE   457
477           "END"JJ;                                          STSTONE   458
478           "END"GREATER 0;                                   STSTONE   459
479                                                             STSTONE   461
480       PIM.=0;                                               NEW        58
481        "FOR"II.=1"STEP"1"UNTIL"CN"DO"                       NEW        59
482        "BEGIN"                                              NEW        60
483        CARD(10);                                            NEW        61
484        "FOR"JJ.=1"STEP"1"UNTIL"16"DO"                       NEW        62
485        "BEGIN"                                              NEW        63
486        PIM.=PIM+1;                                          NEW        64
487        BEDR(/PIM/).=VELD((JJ-1)*8+1,JJ*8)                   NEW        65
488        "END"JJ;                                             NEW        66
489       "END"II;                                              NEW        67
490        "IF"RN"GREATER"0"THEN"                               NEW        68
491       "BEGIN"                                               NEW        69
492        CARD(10);                                            NEW        70
493        "FOR"JJ.=1"STEP"1"UNTIL"RN"DO"                       NEW        71
494       "BEGIN"                                               NEW        72
495        PIM.=PIM+1;                                          NEW        73
496        BEDR(/PIM/).=VELD((JJ-1)*8+1,JJ*8)                   NEW        74
497       "END"JJ;                                              NEW        75
498       "END"GREATER 0;                                       NEW        76
499            PIM.=0;                                         STSTONE   462
500            "FOR"II.=1"STEP"1"UNTIL"CN"DO"                  STSTONE   464
501            "BEGIN"                                         STSTONE   465
502            "FOR"JJ.=1"STEP"1"UNTIL"M"DO"                   STSTONE   466
503     "BEGIN"CARD(10);                                       STSTONE   467
504            "FOR"KK.=1"STEP"1"UNTIL"16"DO"                  STSTONE   468
505            "BEGIN" PIM.=(II-1)*16+KK;                      STSTONE   469
506            C(/JJ,PIM/).=VELD((KK-1)*5+1,KK*5)              STSTONE   470
507            "END"KK;                                        STSTONE   472
508            "END"JJ;                                        STSTONE   473
509            "END"II;                                        STSTONE   474
510                                                            STSTONE   475
511            "IF" RN "GREATER" 0 "THEN"                      STSTONE   476
512            "BEGIN"                                         STSTONE   477
513            "FOR"JJ.=1"STEP"1"UNTIL"M"DO"                   STSTONE   478
514            "BEGIN"                                         STSTONE   479
515     CARD(10);                                              STSTONE   480
516            "FOR" KK.=1"STEP"1"UNTIL"RN"DO"                 STSTONE   481
517            "BEGIN"PIM.=CN*16+KK;                           STSTONE   482
518            C(/JJ,PIM/).=VELD((KK-1)*5+1,KK*5)              STSTONE   483
519            "END"KK;                                        STSTONE   485
520            "END"JJ;                                        STSTONE   486
521            "END" GREATER 0;                                STSTONE   487
522     "COMMENT" INLEZEN PRIMAL;                              STSTONE   488
523     "IF" ST "GREATER" 0 "THEN"                             STSTONE   489
524     "BEGIN"                                                STSTONE   490
525     "FOR" II.= 1 "STEP" 1 "UNTIL" UP "DO"                  STSTONE   491
526     "BEGIN"                                                STSTONE   492
527     CARD(20);                                              STSTONE   493
528     IX(/II/).= VELD(1,5);                                  STSTONE   494
529     JX(/II/).=VELD(11,15);                                 STSTONE   495
530     X(/II/).=VELD(21,25);                                  STSTONE   496
531   "END";                                                   STSTONE   497
```

```
532   "END" S1 GREATER 0 DUS PRIMAL;                                                    STSTONE   498
533                                                                                     STSTONE   499
534   ACTI.=1;                                                                          STSTONE   533
535   ANSWER.=INTEGER TRANSPOPI(N,M,I,J,B(/I/),A(/J/),CC(I,J),U,V,X,IX,JX,ST,           STSTONE   534
536         ROWSOL,INFEAS);                                                           STSTONE   535
537   "GOTO"SLOT;                                                                       STSTONE   537
538   INFEAS: NLCR;PRINTTEXT("("INFEASIBLE")");                                         STSTONE   538
539   SLOT:                                                                             STSTONE   539
540      NLCR;NLCR;                                                                     NEW        78
541    PRINTTEXT("("                TABEL T 1.")");FIXT(2,0,ROB);                     NEW        79
542     PRINTTEXT("("   TOEDELING KUILKLASSE  ")"); FIXT(2,0,ROB);                     NEW        80
543   NLCR;NLCR;                                                                        NEW        81
544    PRINTTEXT("("                VAKNR.          BED.NR         OPPERVLAKTE")      NEW        82
545     ");                                                                           NEW        83
546      NLCR;                                                                        NEW        84
547     HULP.=1;                                                                      NEW        85
548   "FOR"II.=1"STEP"1"UNTIL"UP"DO"                                                  STSTONE   541
549       "BEGIN"NLCR;                                                               NEW        86
550       AA.=0.01*X(/II/);                                                          NEW        87
551      IX(/II/).=ABS(IX(/II/));   JX(/II/).=ABS(JX(/II/));                         NEW        88
552    "IF" VAK(/JX(/II/)/) "GREATER" HULP "THEN" "BEGIN" NLCR;                      NEW        89
553     HULP.=VAK(/JX(/II/)/);                                                        NEW        90
554     "END";                                                                        NEW        91
555     PRINTTEXT("("             ")");                                             NEW        92
556      FIXT(8,0,VAK(/JX(/II/)/));PRINTTEXT("("        ")");                       NEW        93
557      FIXT(8,0,BEDR(/IX(/II/)/));                                                NEW        94
558      FIXT(8,2,AA);                                                              NEW        95
559   IX(/II/).=ABS(IX(/II/));JX(/II/).=ABS(JX(/II/));                                STSTONE   543
560   OUTPUT(11,"("5D,5B,5D,5B,5D,556,/")",IX(/II/),JX(/II/),X(/II/));                STSTONE   544
561   OUTPUT(12,"("3D,8D,5D,/")",VAK(/JX(/II/)/),BEDR(/IX(/II/)/),X(/II/));           NEW        96
562     "END" ;                                                                     NEW        97
563     "COMMENT" "CHECKOFF" BEDR,IX,U,V,JX;                                        NEW        98
564   "END"                                                                           STSTONE   546
565     "END"                                                                       NEW        99
566   "END"                                                                           STSTONE   547
```

```
    PROGRAM PRTOED     74/175  OPT=1                    FTN 4.6+452        28/08/79  13.10.29        PAGE      1

 1                PROGRAM PRTOED (TAPE1,OUTPUT,TAPE2=OUTPUT,INPUT,TAPE3=INPUT)
                  DIMENSION IV(2000),IRU(10)
                  READ(3,10) (IRU(L),L=1,10)
                10 FORMAT(1014)
 5                L=2  $  LRU=1
                  J=NPB=ITEL=0  $  CALL PRINTT3(NPB,ITEL,LRU)
                20 J=J+1
                  READ(1,30) IV(J),I,IR
                  IF(EOF(1)) 90,40
10              30 FORMAT(13,I8,I5)
                40 R=IR/100.
                  IF(I.EQ.0) GOTO 65
                  IF(J.EQ.1) GOTO 50
                  IF(IV(J).EQ.IV(J-1)) GOTO 70
15                IF(IV(J).NE.IRU(L)) GOTO 50
                  LRU=LRU+1  $  L=L+1
                  NPB=ITEL=0  $  CALL PRINTT3 (NPB,ITEL,LRU)
                50 NPB=NPB+2  $  CALL PRINTT3 (NPB,ITEL,LRU)
                  WRITE(2,60) IV(J),I,R
20              60 FORMAT(1H0,15X,I5,I12,F8.2)
                  GOTO 20
                65 J=J-1  $  GOTO 20
                70 NPB=NPB+1  $  CALL PRINTT3 (NPB,ITEL,LRU)
                  WRITE(2,80) I,R
25              80 FORMAT(21X,I12,F8.2)
                  GOTO 20
                90 STOP
                  END


 SUBROUTINE PRINTT3    74/175  OPT=1                    FTN 4.6+452        28/08/79  13.10.29        PAGE      1

 1                SUBROUTINE PRINTT3 (NPB,ITEL,LRU)
                  IF(NPB.EQ.0) GOTO 5
                  IF(ITEL.GT.1) GOTO 4
                  IF(NPB.LT.55) RETURN
 5                GOTO 5
                 4 IF(NPB.LT.54) RETURN
                 5 ITEL=ITEL+1
                  WRITE(2,10) LRU,ITEL,LRU
                10 FORMAT(1H1//16X,"TABEL T 1.",I1,19X,"PAG",I3//16X,"TOEDELING RUILK
10                1LASSE",I2//16X,"VAKNR",6X,"BEDRNR",5X,"OPP"/)
                  NPB=0
                  RETURN
                  END
```

```
      C  MET PROGRAM TOEDKEN WORDEN VAN DIVERSE FILES DE BENODIGDE GEGEVENS
      C  BIJ ELKAAR GEZOCHT EN OP EEN FILE GEBRACHT. MET DE OP DEZE FILE
      C  VOORKOMENDE GEGEVENS WORDEN DE VERKAVELINGSKENMERKEN VAN DE
      C  TOEDELING BEREKEND
            PROGRAM TOEDKEN(OUTPUT,TAPE1=OUTPUT,TAPE2,TAPE3,TAPE4,TAPE5,
           1 TAPE6,TAPE7,TAPE8,INPUT,TAPE9=INPUT)
            INTEGER RU
            DIMENSION INDEX(4000),IBUR(3),IVAK(2),
           1 MA(10),NA(10),IAZ(16),IBD(4000),IAA(11),IBB(11),ICC(11),IAB(11),
           2 IBC(11),ICD(11),IAAA(10),SIOS(11,9),AKB(11,9),TOP(11,9),
           3 TOH(11,9),KAK(11,9),KAB(11,9),OBB1(11,9),KABB1(11,9),OBB2(11,9),
           4 KABB2(11,9),OBB(11,9),KABB(11,9),GKAS(11,9),POH(11,9),POHB(11,9),
           5 POHB1(11,9),POHB2(11,9),PAB2(11,9),PAB1(11,9),PAB(11,9),GBO(11,9)
           6 ,GKO(11,9),IKP(4000),IKN(4000),LU(4000),IBN(2000),IBNT(2000),
           7 IBTT(8),LAB(11),LBC(11),LCD(11)
            COMMON /INV/ IK(4000),IK1(4000),R(4000) /SORT/ ILR,IUR
      C  INLEZEN BEDRIJFSTYPEN VAN PONSKAARTEN
            M=0
          5 M=M+1
            READ(9,10) IAA(M),IBB(M),ICC(M)
            IF(EOF(9)) 20,5
           10 FORMAT(3A10)
           20 IT=M-1
            DO 25 M = 3,IT
           25 READ(9,10) IAB(M),IBC(M),ICD(M)
            ITA=IT-3
            DO 26 M = 1,ITA
           26 READ(9,27) IBTT(M)
           27 FORMAT(A10)
            DO 30 I = 1,10
           30 IAAA(I)=10H----------
      C  TOEDELING VAN ALLE RUILKLASSEN IN ARRAYS ZETTEN EN SORTEREN
      C  OP BEDRIJFSNUMMER
            I=KL=0
           40 I=I+1
            READ(3,50) IK1(I),IK(I),IR
            IF(EOF(3)) 90,60
           50 FORMAT (I3,I8,I5)
           60 IF(IK(I).EQ.0) GOTO 80
            R(I)=FLOAT(IR)/100.  $  IF(IK(I).LT.10**7) GOTO 70  $  GOTO 40
           70 IK(I)=IK(I)+9*10**7  $  KL=1  $  GOTO 40
           80 I=I-1  $  GOTO 40
           90 ILR=1  $  IUR=I-1  $  REWIND 3
            CALL INDCOL
      C  SAMENSTELLEN TABEL T 2 WAARIN PER BEDRIJF DE NIET UITRUILBARE EN DE
      C  TOEGEDEELDE OPPERVLAKTE STAAT AANGEGEVEN
            M=N=JZ=IEOF=0
            NPB=ITEL=0  $  CALL PRINTT2(NPB,ITEL)
          100 N=N+1
            READ(5,400) IBN(N),KVN,O,IVA,KVA
            IF(EOF(5)) 120,110
          110 IF(JZ.EQ.1) GOTO 140
            GOTO 130
          120 IEOF=1  $  IF(M.GT.IUR) GOTO 300 $  IF(JZ.EQ.1) GOTO 140
          130 M=M+1 $ IF(IEOF.EQ.1.AND.M.GT.IUR) GOTO 300 $ IF(M.GT.IUR) GOTO150
            IBNT(M)=IK(M)
            IF(IK(M).GT.9*10**7.AND.KL.EQ.1) IBNT(M)=IK(M)-9*10**7
```

Bijlage 19.1

```
      140 JZ=0  $  IF(IEOF.EQ.1) GOTO 250
          IF(IBN(N)-IBNT(M)) 150,200,250
  60  150 IF(N.EQ.1) GOTO 160
          IF(IBN(N).EQ.IBN(N-1)) GOTO 180
      160 NPB=NPB+2  $  CALL PRINTT2(NPB,ITEL)
          WRITE(1,170) IBN(N),IVN,O
      170 FORMAT(1HO,15X,2I8,F9.2)
  65      JZ=1  $  GOTO 100
      180 NPB=NPB+1  $  CALL PRINTT2 (NPB,ITEL)
          WRITE(1,190) IVN,O
      190 FORMAT(1H ,23X,I8,F9.2)
          JZ=1   $  GOTO 100
  70  200 IF(N.EQ.1) GOTO 210
          IF(IBN(N).EQ.IBN(N-1)) GOTO 230
      210 NPB=NPB+2  $  CALL PRINTT2 (NPB,ITEL)
          WRITE(1,220) IBN(N),IVN,O,IK1(M),R(M)
      220 FORMAT(1HO,15X,2I8,F9.2,I9,F9.2)
  75      GOTO 100
      230 NPB=NPB+1  $  CALL PRINTT2 (NPB,ITEL)
          WRITE(1,240) IVN,O,IK1(M),R(M)
      240 FORMAT(1H ,23X,I8,F9.2,I9,F9.2)
          GOTO 100
  80  250 IF(M.EQ.1) GOTO 260
          IF(IBNT(M).EQ.IBNT(M-1)) GOTO 280
      260 NPB=NPB+2  $  CALL PRINTT2 (NPB,ITEL)
          WRITE(1,270) IBNT(M),IK1(M),R(M)
      270 FORMAT(1HO,15X,I8,17X,I9,F9.2)
  85      GOTO 130
      280 NPB=NPB+1  $  CALL PRINTT2 (NPB,ITEL)
          WRITE(1,290) IK1(M),R(M)
      290 FORMAT(1H ,40X,I9,F9.2)
          GOTO 130
  90  300 REWIND 5
      C  TOEDELINGEN < 0,5 HA WORDEN INDIEN MOGELIJK BIJ EEN ANDERE TOEDELING
      C  VAN HETZELFDE BEDRIJF GEVOEGD
          K=J=0  $  LT=IUR
          DO 380 L = 1,IUR
  95      310 LT=IUR-J
          IF(L.GT.LT) GOTO 390
          IF(R(L).GE..5) GOTO 380
          IF(L.EQ.1) GOTO 320
          IF(IK(L-1).EQ.IK(L)) GOTO 330
 100      IF(L.EQ.LT) GOTO 390
      320 IF(IK(L+1).EQ.IK(L)) GOTO 340
          GOTO 380
      330 R(L-1)=R(L-1)+R(L)  $  GOTO 350
      340 R(L+1)=R(L+1)+R(L)
 105      350 LA=LT-1  $  IF(L.EQ.LT) GOTO 370
          DO 360 I = L,LA
          IK(I)=IK(I+1)  $  IK1(I)=IK1(I+1)  $  R(I)=R(I+1)
      360 CONTINUE
      370 J=J+1  $  GOTO 310
 110  380 CONTINUE
      C  NIET UITRUILBARE KAVELS WORDEN TOEGEVOEGD AAN DE ARRAYS MET
      C  TOEDELINGEN
      C  INDIEN IN EEN VAN EEN TOEDELING EN EEN NIET UITRUILBARE KAVEL
      C  VAN HETZELFDE BEDRIJF VOORKOMEN WORDEN DEZE SAMENGEVOEGD
```

```
      C DE KAVELAFSTAND VAN NIET UITRUILBARE KAVELS WORDT OPGENOMEN
      C  IN DE ARRAY MET VAKNUMMERS
        390 IB=0  $  IV=0
        395 IBV=IB  $  IVV=IV
            READ(5,400) IB,KAV,0,IV,LTT
            IF(EOF(5)) 480,410
        400 FORMAT (I6,I3,F8.2,2I4)
        410 IF(IB.LT.10**7) IB=IB+9*10**7
            DO 440 I = 1,LT
            IA=IK(I)
            IF(IK(I).GT.10**8) IA=IK(I)/10
        420 IF(IA-IB) 440,430,450
        430 IF(IK1(I).NE.IV) GOTO 440
            IF(IB.EQ.IBV.AND.IV.EQ.IVV) GOTO 450
            R(I)=R(I)+0
            IF(IK(I).GT.10**8) GOTO 395
            IF(KAV.EQ.1) IK(I)=IK(I)*10+1  $  GOTO 395
        440 CONTINUE
            KN=LT+1  $  GOTO 470
        450 DO 460 K = I,LT
            KN=(LT+1)-K
            IK(KN+1)=IK(KN) $ IK1(KN+1)=IK1(KN) $ R(KN+1)=R(KN)
        460 CONTINUE
        470 IK(KN)=IB  $  IF(KAV.EQ.1) IK(KN)=IK(KN)*10+1
            IK1(KN)=IV*1000+LTT  $  R(KN)=0
            LT=LT+1  $  GOTO 395
      C  IN TWEE VAKKEN VOORKOMENDE DELEN VAN EEN HUISKAVEL WORDEN
      C  SAMENGEVOEGD TOT EEN KAVEL
        480 I=1
        490 I=I+1
            IF(I.GT.LT) GOTO 520
            IF(IK(I).EQ.IK(I-1).AND.IK(I).GT.10**8) GOTO 500
            GOTO 490
        500 M=I-1  $  R(I)=R(I-1)+R(I)  $  IK1(I)=IK1(I-1)
            DO 510 L = M,LT
            IK(L)=IK(L+1)  $  IK1(L)=IK1(L+1)  $  R(L)=R(L+1)
        510 CONTINUE
            LT=LT-1  $  I=I-1
            GOTO 490
      C  VAN FILE INLEZEN :  NAAM GEBIED, EERSTE DN MET BUITENBLOK BEDRIJVEN,
      C  TOTAAL AANTAL DN,AANTAL BEDRIJVEN EN AFMETINGEN MATRICES PER RUILKL.
        520 RU=NL=NT=0  $  REWIND 5
            READ(6,530) IM,IN,IP,IZ,NDA,NDG,MAB
        530 FORMAT(3A10,I5,2I2,I5)
            READ(6,540)(MA(I),I=1,10),(NA(I),I=1,10)
        540 FORMAT(20I5)
            REWIND 6
            CALL OPENMS (4,INDEX,4000,0)
            REWIND7
      C  VAN FILE MET BEDRIJVEN EN VAKKEN PER RUILKLASSE ALLE BEDRIJFS- EN
      C  VAKNUMMERS IN DE OORSPRONKELIJKE VOLGORDE IN EEN ARRAY ZETTEN
        550 RU=RU+1  $  IF(MA(RU).EQ.0.OR.RU.EQ.11) GOTO 580
            NB=NL+1  $  NL=NL+NA(RU)
            DO 560 I = NB,NL
            CALL READMS (4,IBDR,3,I)
            IBD(I)=IBDR(1)
            IF(IBD(1).LT.10**7) IBD(I)=IBD(I)+9*10**7
```

Bijlage 19.3

```
      560 CONTINUE
          MB=NL+1  $  NL=NL+MA(RU)
          DO 570 M = MB,NL
175       CALL READMS (4,IVAK,2,M)
          IBD(M)=IVAK(1)
      570 CONTINUE
          GOTO 550
C     IN ARRAYS MET TOEDELINGEN PER BEDRIJF DE RUISKAVEL ALS EERSTE
180   C  KAVEL IN DE ARRAYS ZETTEN
      580 DO 600 I = 2,LT
          IF(IK(I).LT.10**8) GOTO 600
          N=I-1  $  M=I-20  $  IF(I.LT.21) M=1  $  IL=IK(I)/10
          DO 590 J = M,N
185       IF(IK(J).NE.IL) GOTO 590
          JK=IK(J)  $  JK1=IK1(J)  $  RL=R(J)
          IK(J)=IK(I)  $  IK1(J)=IK1(I)  $  R(J)=R(I)
          IK(I)=JK  $ IK1(I)=JK1  $  R(I)=RL
          GOTO 600
190   590 CONTINUE
      600 CONTINUE
C     DE BENODIGDE VAKAFSTANDEN WORDEN BIJ DE TOEDELINGEN GEZOCHT,
C     WAARNA OP FILE WORDEN GEZET : BEDRIJFSNR,KAVELNR,OPP. EN AFSTAND
          DO 840 K = 1,LT
195       RU=NL=0
          IF(IK(K).GT.10**8) GOTO 790
          IF(IK1(K).GT.1000) GOTO 800
      610 RU=RU+1  $  IF(MA(RU).EQ.0.OR.RU.EQ.11) GOTO 630
          NB=NL+1  $  NL=NL+NA(RU)
200       DO 620 I = NB,NL
          IF(IK(K).NE.IBD(I)) GOTO 620
          IV=IK1(K)  $  GOTO 650
      620 CONTINUE
          NL=NL+MA(RU)  $  GOTO 610
205   630 WRITE (1,640) IK(K)
      640 FORMAT(1H ,"BEDRIJFSNUMMER :",I10,"  KOMT NIET VOOR BIJ TOEDELING"
         1)
          GOTO 840
      650 MB=NL+1  $  NL=NL+MA(RU)
210       DO 660 M = MB,NL
          IF(IV.NE.IBD(M)) GOTO 660
          KV=M  $  KBB=K  $  KB=1  $  GOTO 670
      660 CONTINUE
          GOTO 610
215   670 JL=JP=0  $  KR=RU-1
          IF(KR.NE.0) GOTO 680
          JP=(KB-1)*IFIX(FLOAT(MA(1))/16.+.99)+IFIX(FLOAT(KV-NA(1))/16.-.05)
          KK=KV-NA(1)  $  GOTO 700
      680 DO 690 J = 1,KR
220       JP=JP+IFIX(FLOAT(MA(J))/16.+.99)*NA(J)  $  JL=JL+NA(J)+MA(J)
      690 CONTINUE
          KK=KV-JL-NA(KR+1)
          JP=JP+(KB-JL-1)*IFIX(FLOAT(MA(KR+1))/16.+.99)+IFIX(FLOAT(KK)/16.-.
         A05)
225   700 IF(JP.EQ.0) GOTO 730
          DO 710 I = 1,JP
      710 READ(2,720)
      720 FORMAT (    )
```

```
      730 N=KK-(1FIX(FLOAT(KK)/16.-.05)*16)
230       READ(2,740)(IAZ(M),M=1,N)
      740 FORMAT(1615)
          IAZ(N)=1FIX(FLOAT(IAZ(N))/100.+.5)  $  REWIND2
          IF(KBB.EQ.1) GOTO 750
          IF(IK(KBB).EQ.IK(KBB-1)) GOTO 760
235       750 KA=2
          GOTO 770
          760 KA =KA+1
          770 IF(KL.EQ.1.AND.IK(KBB).GT.9*10**7) IK(KBB)=MOD(IK(KBB),10**7)
          WRITE(7,780) IK(KBB),KA,R(KBB),IAZ(N)
240       780 FORMAT(I8,I2,F6.2,I4)
          IF(KL.EQ.1.AND.IK(KBB).LT.10**7) IK(KBB)=IK(KBB)+9*10**7
          NT=NT+1
          GOTO 840
          790 IJ=IK(K)/10  $  KA=1  $  KAF=0  $  GOTO 830
245       800 IJ=IK(K)  $  KAL=IK1(K)/1000  $  KAF=IK1(K)-KAL*1000
          IF(K.EQ.1) GOTO 810
          IF(IK(K).EQ.IK(K-1)) GOTO 820
          810 KA=2  $  GOTO 830
          820 KA=KA+1
250       830 IF(KL.EQ.1.AND.IJ.GT.9*10**7) IJ=MOD(IJ,10**7)
          WRITE(7,780) IJ,KA,R(K),KAF
          NT=NT+1
          IF(KL.EQ.1.AND.IJ.LT.10**7) IJ=IJ+9*10**7
          840 CONTINUE
255       ENDFILE7  $  REWIND7
        C  OP FILE GEBRACHTE GEGEVENS WORDEN IN ARRAYS GEZET
          DO 860 LI = 1,NT
          READ(7,850) IKP(LI),IKN(LI),R(LI),LU(LI)
          850 FORMAT(I8,I2,F6.2,I4)
260       860 CONTINUE
          REWIND 7
        C  BEREKENEN VERKAVELINGSKENMERKEN
          ITEL=NPB=0  $  CALL PRINTT1 (NPB,ITEL,IM,IN,IP)
          NDGA=NDG+1  $  NDAT=NDA+1
265       DO 1320 I = 1,NDGA
          DO 870 J = 1,1T
          DO 870 N = 1,9
                    STOS(J,N)=TOP(J,N)=KAK(J,N)=KAB(J,N)=POH(J,N)=OBB(J,N)=
         1KABB(J,N)=OBH1(J,N)=DBB2(J,N)=KABB1(J,N)=KABB2(J,N)=0
270       AKB(J,N)=GBO(J,N)=GKO(J,N)=POH(J,N)          =GKAS(J,N)=PAB1(J,N)=
         1POHB1(J,N)=PAB2(J,N)=POHB2(J,N)=PAB(J,N)=POHB(J,N)=0
          870 CONTINUE
          SBOS=KAKB=TOPB=TOHB=OMAX=0
          DO 1060 K = 1,1T
275       LA=0
          DO 1050 M = 1,MAB
          LC=0
          READ( 8,880) ND,I1,1BT,1HD
          880 FORMAT(12,I10,2I3)
280       DBB=1H
          IF(I.EQ.1) GOTO 890
          IF(ND.NE.(I-1)) GOTO 1050
          890 IF(K.EQ.1) GOTO 910
          IF(I.GE.2) GOTO 900
285       IF(ND.GE.NDA) GOTO 1050
```

```
      900 IF(K.EQ.2) GOTO 910
          IF(K.EQ.3.AND.IBT.EQ.ITA) GOTO 1050
          IF(K.EQ.3) GOTO 910
          IF(IBT.NE.(K-3)) GOTO 1050
290   910 DO 1040 L = 1,NT
          LA=LA+1
          IK(LA)=IKP(L)
          IF(IK(LA).NE.I1) GOTO 1020
          IK1(LA)=IKN(L)  $  O=R(L)  $  LTOT=LU(L)
295       LC=1
          STOS(K,9)=STOS(K,9)+O*LTOT
          STOS(K,IBO)=STOS(K,IBO)+O*LTOT
          TOP(K,9)=TOP(K,9)+O $ TOP(K,IBO)=TOP(K,IBO)+O
          IF(IK1(LA).EQ.1) TOH(K,9)=TOH(K,9)+O
300       IF(IK1(LA).EQ.1) TOH(K,IBO)=TOH(K,IBO)+O
          IF(LA.EQ.1) GOTO 920
          IF(IK1(LA).EQ.IK1(LA-1).AND.IK(LA).EQ.IK(LA-1)) GOTO 930
      920 KAKB=KAKB+1
          KAK(K,9)=KAK(K,9)+1  $  KAK(K,IBO)=KAK(K,IBO)+1
305   930 SBOS=SBOS+O*LTOT
          TOPB=TOPB+O  $  IF(IK1(LA).EQ.1) GOTO 940
          IF(O.GT.OMAX) OMAX=O  $  GOTO 950
      940 TOHB=TOHB+O
      950 IF(L.EQ.NT) GOTO 960
310       GOTO 1040
      960 LA=LA-1
          KAB(K,9)=KAB(K,9)+1  $  KAB(K,IBO)=KAB(K,IBO)+1
          KAGS=SBOS/TOPB+.5  $  PH=TOHB/TOPB*100.  $  PV=OMAX/TOPB*100.
          IF(TOHB/TOPB.LT.1) GOTO 970
315       OBB1(K,9)=OBB1(K,9)+TOPB  $  OBB1(K,IBO)=OBB1(K,IBO)+TOPB
          KABB1(K,9)=KABB1(K,9)+1  $  KABB1(K,IBO)=KABB1(K,IBO)+1
      970 IF(TOHB/TOPB.LT..8) GOTO 980
          OBB2(K,9)=OBB2(K,9)+TOPB  $  OBB2(K,IBO)=OBB2(K,IBO)+TOPB
          KABB2(K,9)=KABB2(K,9)+1  $  KABB2(K,IBO)=KABB2(K,IBO)+1
320   980 IF(TOHB/TOPB.LT..6) GOTO 990
          OBB(K,9)=OBB(K,9)+TOPB  $  OBB(K,IBO)=OBB(K,IBO)+TOPB
          KABB(K,9)=KABB(K,9)+1  $  KABB(K,IBO)=KABB(K,IBO)+1
      990 IF(I.GT.1.OR.K.GT.1) GOTO 1010
          IF(ND.GE.NDA) LBB=1HX
325       NPB=NPB+1  $  CALL PRINTT1 (NPB,ITEL,IM,IN,IP)
          WRITE(1,1000) I1,KAKB,KAGS,TOPB,TOHB,PH,PV,IBTT(IBT),LBB
     1000 FORMAT(1H ,15X,I8,I6,I9,F11.2,F8.2,2F9.1,4X,A10,4X,A1)
     1010 KAKB=SBOS=TOPB=TOHB=OMAX=0
          IF(L.EQ.NT) GOTO 1040
330       GOTO 1050
     1020 IF(LC.NE.1) GOTO 1030
          GOTO 960
     1030 LA=LA-1
     1040 CONTINUE
335  1050 CONTINUE
          REWIND 6
     1060 CONTINUE
          DO 1090 KK = 1,IT
          DO 1090 LL = 1,9
340       IF(TOP(KK,LL).EQ.0) GOTO 1070
          GKAS(KK,LL)=STOS(KK,LL)/TOP(KK,LL)
          POH(KK,LL)=(TOH(KK,LL)/TOP(KK,LL))*100.
```

```
                POHB(KK,LL)=(OBB(KK,LL)/TOP(KK,LL))*100.
                POHB1(KK,LL)=(OBB1(KK,LL)/TOP(KK,LL))*100.
345             POHB2(KK,LL)=(OBB2(KK,LL)/TOP(KK,LL))*100.
             1070 IF(KAB(KK,LL).EQ.0) GOTO 1080
                PAB2(KK,LL)=(FLOAT(KABB2(KK,LL))/FLOAT(KAB(KK,LL)))*100.
                PAB1(KK,LL)=(FLOAT(KABB1(KK,LL))/FLOAT(KAB(KK,LL)))*100.
                PAB(KK,LL)=(FLOAT(KABB(KK,LL))/FLOAT(KAB(KK,LL)))*100.
350             GBO(KK,LL)=TOP(KK,LL)/KAB(KK,LL)
                AKB(KK,LL)=FLOAT(KAK(KK,LL))/FLOAT(KAB(KK,LL))
             1080 IF(KAK(KK,LL).EQ.0) GOTO 1090
                GKO(KK,LL)=TOP(KK,LL)/KAK(KK,LL)
             1090 CONTINUE
355          C  PRINTEN VERKAVELINGSKENMERKEN
                DO 1310 N = 1,IT
                IF(I.GT.1.AND.N.EQ.2) GOTO 1310
                IF(TOP(N,9).EQ.0) GOTO 1310
                III=I+1
360             IF(I.LT.NDAT.OR.N.EQ.1) GOTO 1100
                LAB(N)=IAB(N)  $  LBC(N)=IBC(N)  $  LCD(N)=ICD(N)
                GOTO 1110
             1100 LAB(N)=IAA(N) $ LBC(N)=IBB(N) $ LCD(N)=ICC(N)
             1110 IF(I.GT.1) GOTO 1130
365             WRITE(1,1120) III,N,IM,IN,IP,LAB(N),LBC(N),LCD(N)
             1120 FORMAT(1H1//16X,"TABEL",I3,".",I2,2X,"GEHELE RUILVERKAVELING"//16X
                 1,"RUILVERKAVELING :",3A10//16X,"VERKAVELINGSKENMERKEN TOEDELING"//
                 216X,3A10)
                GOTO 1150
370          1130 IIA=I-1
                WRITE(1,1140) III,N,IIA,IM,IN,IP,LAB(N),LBC(N),LCD(N)
             1140 FORMAT(1H1//16X,"TABEL",I3,".",I2,2X,"DORPSBEHOREN",I3//16X,"RUILV
                 1ERKAVELING :",3A10//16X,"VERKAVELINGSKENMERKEN TOEDELING"//16X,3A1
                 20)
375          1150 WRITE(1,1160) IAAA,IAAA
             1160 FORMAT(1H0,15X,10A10/16X,"BEDRIJFSOPPERVLAKTE IN HA     < 5     5-
                 110    10-15    15-20    20-30    30-40    40-50    > 50     TOTAAL "/16X,
                 210A10/)
                WRITE(1,1170)(KAB(N,L),L=1,9)
380          1170 FORMAT(1H ,15X,"AANTAL BEDRIJVEN",11X,9(16,2X)/)
                WRITE(1,1180)(KAK(N,L),L=1,9)
             1180 FORMAT(1H ,15X,"AANTAL KAVELS",14X,9(I6,2X)/)
                WRITE(1,1190)(TOP(N,L),L=1,9)
             1190 FORMAT(1H ,15X,"OPP CULTUURGROND IN HA",5X,9(F8.1)/)
385             WRITE(1,1200)(AKB(N,L),L=1,9)
             1200 FORMAT(1H ,15X,"AANTAL KAVELS PER BEDRIJF",2X,9(F8.1)/)
                WRITE(1,1210)(GBO(N,L),L=1,9)
             1210 FORMAT(1H ,15X,"GEM BEDRIJFSOPP IN HA",6X,9(F8.1)/)
                WRITE(1,1220)(GKO(N,L),L=1,9)
390          1220 FORMAT(1H ,15X,"GEM KAVELOPP IN HA",9X,9(F8.1)/)
                WRITE(1,1230)(TOH(N,L),L=1,9)
             1230 FORMAT(1H ,15X,"OPP INGENOMEN DOOR HUISKAV",1X,9(F8.1)/)
                WRITE(1,1240)(POH(N,L),L=1,9)
             1240 FORMAT(1H ,15X,"PERC VAN DE OPP INGENOMEN"/16X,"DOOR HUISKAVELS",
395             112X,9(F8.1)/)
                WRITE(1,1250)(GKAS(N,L),L=1,9)
             1250 FORMAT(1H ,15X,"GEM SCHIJNB KAVAFST IN HM",2X,9(F8.1)/)
                IF(I.GE.NDAT) GOTO 1310
                WRITE(1,1260)(KABB1(N,L),L=1,9)
```

```
400          1260 FORMAT(1H0,15X,"BEDRIJVEN WAARVAN DE HUISKAVEL 100 PROC VAN DE BED
                 1ROPP UITMAAKT :"//16X,"AANTAL BEDRIJVEN",11X,9(I6,2X)/)
                  WRITE(1,1270)(PAB1(N,L),L=1,9)
             1270 FORMAT(1H ,15X,"PERC VAN TOT AANT BEDR",5X,9(F8.1)/)
                  WRITE(1,1280)(POHB1(N,L),L=1,9)
405          1280 FORMAT(1H ,15X,"PERC VAN DE TOT OPP",8X,9(F8.1)/)
                  WRITE(1,1290)(KABB2(N,L),L=1,9)
             1290 FORMAT(1H0,15X,"BEDRIJVEN WAARVAN DE HUISKAVEL 80 PROC OF MEER VAN
                 1 DE BEDROPP UITMAAKT :"//16X,"AANTAL BEDRIJVEN",11X,9(16,2X)/)
                  WRITE(1,1270)(PAB2(N,L),L=1,9)
410               WRITE(1,1280)(POHB2(N,L),L=1,9)
                  WRITE(1,1300)(KABB(N,L),L=1,9)
             1300 FORMAT(1H0,15X,"BEDRIJVEN WAARVAN DE HUISKAVEL 60 PROC OF MEER VAN
                 1 DE BEDROPP UITMAAKT :"//16X,"AANTAL BEDRIJVEN",11X,9(I6,2X)/)
                  WRITE(1,1270)(PAB(N,L),L=1,9)
415               WRITE(1,1280)(POHB(N,L),L=1,9)
             1310 CONTINUE
             1320 CONTINUE
                  CALL EXIT
                  END
```



```
1                 SUBROUTINE PRINTT1 (NPB,ITEL,IM,IN,IP)
                  IF(NPB.EQ.0) GOTO 5
                  IF(ITEL.GT.1) GOTO 4
                  IF(NPB.LT.50) RETURN
5                 GOTO 5
                4 IF(NPB.LT.49) RETURN
                5 ITEL=ITEL+1
                  WRITE(1,10) ITEL,IM,IN,IP
               10 FORMAT(1H1//16X,"TABEL 1  ",73X,"PAG",13//16X,"RUILVERKAVELING :",
10               13A10//16X,"BEDRIJFSGEGEVENS TOEDELING"//34X,"GEM",56X,"X="/26X,"AA
                 2NTAL  SCHIJNB",11X,"OPP      PERC     PERC",9X,"BEDR     BEDR"/
                 3 18X,"BEDRNR  KAVELS  KAVAFST  BEDROPP  HUISKAV  HUISKAV  GROOTSTE
                 4     TYPE     BUITEN"/34X,"IN HM",31X,"VELDKAV",15X,"BLOK"/)
                  NPB=0
15                RETURN
                  END
```



```
1                 SUBROUTINE PRINTT2 (NPB,ITEL)
                  IF(NPB.EQ.0) GOTO 5
                  IF(ITEL.GT.1) GOTO 4
                  IF(NPB.LT.50) RETURN
5                 GOTO 5
                4 IF(NPB.LT.48) RETURN
                5 ITEL =ITEL+1
                  WRITE(1,10) ITEL
               10 FORMAT(1H1//16X,"TABEL 1 2",35X,"PAG",13//16X,"OPGAVE PER BEDRIJF
10               1VAN NIET UITRUILBARE"/16X,"EN TOEGEDEELDE OPPERVLAKTEN"//27X,"NIET
                 2 UITRUILBAAR     TOEGEDEELD"//17X,"BEDRNR      VAKNR     OPP      VAK
                 3NR     OPP"/36X,"IN HA",13X,"IN HA"/)
                  NPB=0
                  RETURN
15                END
```

Bijlage 19.8

```
      PROGRAM CONTROL (TAPE1,TAPE2,OUTPUT,INPUT,TAPE3=OUTPUT,TAPE4=INPUT
     1)
      DIMENSION I(4000),K(4000),IA(4000),KA(4000),JB(4000)
      READ(4,1) NDA
    1 FORMAT(I2)
      M=0
    5 M=M+1
      READ(1,10) I(M),K(M),JB(M)
      IF(EOF(1)) 20,5
   10 FORMAT (2X,I8,2X,I2,02X,I2)
   20 MT=M-1  $  N=0
   25 N=N+1
      READ(2,30) IA(N),KA(N)
      IF(EOF(2)) 40,25
   30 FORMAT (I8,I3)
   40 NT=N-1
      DO 50 L = 1,MT
      IF(JB(L).GE.NDA) GOTO 50
      DO 60 J = 1,NT
      IF(I(L).EQ.IA(J).AND.K(L).EQ.KA(J)) GOTO 50
   60 CONTINUE
      WRITE(3,70) I(L),K(L)
   70 FORMAT (1H ,"KAVEL",I10,I4,"  KOMT NIET VOOR BIJ AANV GEG")
   50 CONTINUE
      DO 100 L = 1,NT
      DO 110 J = 1,MT
      IF(JB(J).GE.NDA) GOTO 110
      IF(IA(L).EQ.I(J).AND.KA(L).EQ.K(J)) GOTO 100
  110 CONTINUE
      WRITE (3,120) IA(L),KA(L)
  120 FORMAT (1H ,"KAVEL",I10,I4,"  KOMT NIET VOOR IN CIN")
  100 CONTINUE
      END
```

Bijlage 20

```
      PROGRAM HUISKAV(TAPE1,TAPE2)
      COMMON /INV/ IK(4000),IK1(4000),R(4000)  /SORT/ ILR,IUR
      DATA R/4000*0/,IK1/4000*0/
      L=0
    5 L=L+1
      READ(1,100) I,K
      IF(EOF(1)) 10,6
  100 FORMAT(I8,I3)
    6 IK(L)=I*100+K
      GOTO 5
   10 ILR=1  $  IUR=L-1  $  CALL INDCOL
      DO 12 N = 1,IUR
      JM=IK(N)  $  IK(N)=JM/100  $  IK1(N)=JM-IK(N)*100
   12 CONTINUE
      DO 20 J = 1,IUR
      KB=0
      IF(J.EQ.1.AND.IK1(J).NE.1) GOTO 15
      IF(J.EQ.1) GOTO 16
      IF(IK(J).NE.IK(J-1).AND.IK1(J).NE.1) GOTO 15
      IF(IK(J).NE.IK(J-1)) GOTO 16
      GOTO 20
   15 KB=1
   16 WRITE(2,110) IK(J),KB
  110 FORMAT(I8,38X,I2)
   20 CONTINUE
      REWIND2
      STOP
      END
```

```
      PROGRAM TRKAV(INPUT,TAPE1=INPUT,OUTPUT,TAPE2=OUTPUT,TAPE3,TAPE4,
     1 TAPE5,TAPE6,TAPE7)
      IMPLICIT INTEGER (A-Z)
      REAL DX,AXI,BR,DY,AYI,AR
      DIMENSION BUF(3),COOR(3),IND(10000),HULP(24),TRA(3),HELP(3),
     1 KAV(2300),KAVA(2300),WEG(3000),WCR(4),IX(2300),IY(2300)
      DATA STER/"F"/,VRAG/"?"/,BLANC/" "/
      CALL OPENMS (4,IND,10000,0)
      CALL OPENMS (6,WEG,3000,0)
      READ(1,101) ITT,NDA,KB
  101 FORMAT (I5,I2,I1)
      IT=IREC=0  $  WRITE(2,100)
  100 FORMAT(1H1)
    5 READ(3,102) DX,AXI,BR,DY,AYI,AR
  102 FORMAT(2(F9.1,2F9.6))
      READ(7,104) NRB
  104 FORMAT(I3)
      NRT=NRB/100
    7 IREC=IREC-IT
   10 READ(7,106)(BUF(I),I=1,3)
      IF(EOF(7)) 40,12
  106 FORMAT(3A10)
   12 IF(BUF(3).NE.BLANC) IT=0
      FOUT=BLANC
      IREC=IREC+1  $  IT=IT+1
      DECODE(24,108,BUF)(HULP(I),I=1,24)
  108 FORMAT(24A1)
      DO 200 K = 1,24
      IF(HULP(K).EQ.STER) GOTO 7
  200 CONTINUE
      IF(HULP(2).NE.BLANC.AND.HULP(11).NE.BLANC) GOTO 18
      IF(HULP(11).EQ.BLANC) GOTO 15
      DECODE(15,110,BUF)(HELP(I),I=1,3)
  110 FORMAT(A1,2A7)
      DECODE(30,111,HELP)(COOR(I),I=1,3)
  111 FORMAT(3I10)
      TRA(1)=COOR(1)/10**9 $ X=COOR(2)/1000  $  Y=COOR(3)/1000
      GOTO 20
   15 DECODE(24,112,BUF)(HELP(I),I=1,3)
  112 FORMAT(A10,2A7)
      DECODE(30,113,HELP)(COOR(I),I=1,3)
  113 FORMAT(3I10)
      TRA(1)=COOR(1)  $  X=COOR(2)/1000  $  Y=COOR(3)/1000
      TRA(2)=TRA(3)=0      $  GOTO 19
   18 FOUT=VRAG  $  TRA(1)=TRA(2)=TRA(3)=0
   19 CALL WRITMS(4,TRA,3,IREC)
      WRITE(2,114) IREC,TRA(1),FOUT
  114 FORMAT(I5,I12,7X,A1)
      IF(FOUT.EQ.VRAG) GOTO 10
      IREC=IREC+1  $  TRA(1)=0      $  IT=IT+1
   20 TRA(2)=IFIX(DX+AXI*X+BR*Y+.5)
      TRA(3)=IFIX(DY+AYI*Y+AR*X+.5)
      CALL WRITMS(4,TRA,3,IREC)
      WRITE(2,115) IREC,TRA
  115 FORMAT(I5,I2,2I8)
      GOTO 10
   40 NRTT=NRB-NRT*100  $  IF(NRTT.LT.NRT) GOTO 5
```

Bijlage 22.1

```
      L=0
      DO 202 I = 1,IREC
60    CALL READMS(4,TRA,3,I)
      WRITE(2,116) TRA
  116 FORMAT(I12,2I8)
      IF(TRA(1).GT.1000) GOTO 50
      GOTO 202
65     50 L=L+1  $  KAVA(L)=TRA(1)
  202 CONTINUE
      I=0
     60 I=I+1
      READ(5,120) ND,IB,IK
70    IF(EOF(5)) 80,70
  120 FORMAT(I2,I8,2X,I2)
     70 IF(IK.EQ.1) GOTO 75
      IF(KB.NE.1) GOTO 71
      IF(ND.LT.NDA) GOTO 75
75    71 KAV(I)=IB*100+IK
      GOTO 60
     75 I=I-1
      GOTO 60
     80 NT=I-1
80    DO 210 I = 1,L
      DO 220 K = 1,NT
      IF(KAVA(I).EQ.KAV(K)) GOTO 210
  220 CONTINUE
      WRITE(2,130) KAVA(I)
85  130 FORMAT (1H ,"KAVEL :",I12,"  KOMT NIET VOOR BIJ GEGEVENS")
  210 CONTINUE
      DO 230 K = 1,NT
      DO 240 I = 1,L
      IF(KAV(K).EQ.KAVA(1)) GOTO 230
90    240 CONTINUE
      WRITE(2,140) KAV(K)
  140 FORMAT (1H ,"KAVEL :",I12,"  NIET GEDIGITALISEERD")
  230 CONTINUE
      DO 250 I = 1,ITT
95    CALL READMS (6,WCR,4,I)
      IX(I)=WCR(1)  $  IY(I)=WCR(2)
  250 CONTINUE
      P=Q=R=PP=0
      DO 260 I = 1,IREC
100   CALL READMS (4,TRA,3,I)
      IF(TRA(1).GT.1000) GOTO 85
      P=TRA(1)  $  Q=TRA(2)  $  R=TRA(3)
      IF(I.EQ.IREC) GOTO 85
      GOTO 260
105    85 PPP=PP  $  PP=TRA(1)
      IF(I.EQ.1) GOTO 260
      DO 270 L = 1,ITT
      IF(IABS(Q-IX(L)).LT.9.AND.IABS(R-IY(L)).LT.9) GOTO 260
  270 CONTINUE
110   WRITE(2,150) Q,R,PPP
  150 FORMAT(1H ,"COORDINATEN :",2I8,"  VAN KAVEL :",I12,"  KOMEN NIET V
     1OOR IN WEGENNET")
  260 CONTINUE
      CALL EXIT
```

```
      PROGRAM MUTATKA (INPUT,TAPE1=INPUT,OUTPUT,TAPE2=OUTPUT,TAPE3,
     1 TAPE4)
      IMPLICIT INTEGER (A-Z)
      DIMENSION TRA(3),PUR(3),HULP(3),IND(10000),WEG(2500),X(2500),
     1 Y(2500),COOR(4)
      READ(1,100) IT,(PUR(I),I=1,3),ITT
  100 FORMAT(I5,3I1,I5)
      CALL OPENMS (3,IND,10000,0)
      CALL OPENMS (4,WEG,2500,0)
      WRITE(2,101)
  101 FORMAT(1H1)
      M=ML=0
    2 READ(1,102) NR,T,(TRA(I),I=1,3)
      IF(EOF(1)) 40,5
  102 FORMAT(I5,A1,I12,2I8)
    5 NR=NR+(M-ML)
      IF(T.EQ.1H ) GOTO 10
      IF(T.EQ.1H+) GOTO 20
      IF(T.EQ.1H-) GOTO 30
      WRITE(2,105) NR
  105 FORMAT(1H ,I5,3X,"OPGAVE FOUT")
      GOTO 2
   10 CALL WRITMS (3,TRA,3,NR)
      IF(NR.LE.IT) GOTO 2
      IT=IT+1  $  GOTO 2
   20 J=NR+1
      DO 201 I = J,IT
      K=(IT+J)-I
      CALL READMS (3,HULP,3,K)
      L=K+1
      CALL WRITMS (3,HULP,3,L)
  201 CONTINUE
      CALL WRITMS (3,TRA,3,J)
      M=M+1  $  IT=IT+1
      GOTO 2
   30 J=NR+1
      DO 202 I = J,IT
      CALL READMS (3,HULP,3,I)
      L=I-1
      CALL WRITMS (3,HULP,3,L)
  202 CONTINUE
      CALL WRITMS (3,PUR,3,IT)
      ML=ML+1  $  IT=IT-1
      GOTO 2
   40 DO 203 K = 1,ITT
      CALL READMS (4,COOR,4,K)
      X(K)=COOR(1)  $  Y(K)=COOR(2)
  203 CONTINUE
      P=Q=R=PP=0
      DO 205 I = 1,IT
      CALL READMS (3,TRA,3,1)
      IF(TRA(1).GT.100) GOTO 45
      P=TRA(1)  $  Q=TRA(2)  $  R=TRA(3)
      IF(I.EQ.IT) GOTO 45
      GOTO 205
   45 PPP=PP  $  PP=TRA(1)
      IF(I.EQ.1) GOTO 205
```

```
      DO 206 L = 1,ITT
       IF(IABS(Q-X(L)).LT.9.AND.IABS(R-Y(L)).LT.9) GOTO 50
  206 CONTINUE
      WRITE(2,106) PPP
  106 FORMAT(1H ,"COORDINATEN VAN KAVEL :",I12,3X,"KOMEN NIET VOOR IN WE
     1GENNET")
      GOTO 205
   50 TRA(2)=X(L)  $  TRA(3)=Y(L)  $  TRA(1)=P
      S=I-1
      IF(I.EQ.IT) S=IT
      CALL WRITMS (3,TRA,3,S)
  205 CONTINUE
      DO 210 L = 1,IT
      CALL READMS (3,TRA,3,L)
  210 WRITE(2,104) TRA,L
  104 FORMAT(112,218,I8)
      WRITE(2,110) IT
  110 FORMAT (I5)
      STOP
      END
```

```
      PROGRAM INKAF (TAPE1,TAPE2,INPUT,TAPE3=INPUT,OUTPUT,TAPE4=OUTPUT)
      IMPLICIT INTEGER (A-Z)
      DIMENSION X(5),Y(5),WK(5),TRA(3),A(3),IND(10000)
      REWIND2
 5    READ(3,100) IT
  100 FORMAT(I5)
      CALL OPENMS (1,IND,10000,0)
      I=LASTX=LASTY=LASTW=0
    2 L=0
10    4 I=I+1
      CALL READMS (1,TRA,3,I)
      IF(TRA(1).LT.100) GOTO 20
      IF(LASTX.NE.0) GOTO 20
      IF(L.NE.0) GOTO 40
15    6 WNR2=0 $ KNR=MOD(TRA(1),100) $ L=0 $ KNW=KNR*100+WNR2
      WRITE(2,102) TRA(1),KNW
  102 FORMAT(I10,T77,I4)
      GOTO 4
   20 IF(LASTX.EQ.0) GOTO 30
20    X(1)=LASTX  $  Y(1)=LASTY  $  L=L+1
      LASTX=LASTY=LASTW=0
      IF(TRA(1).LT.100) GOTO 30
      WK(2)=1RW
      GOTO40
25    30 L=L+1
      X(L)=TRA(2)  $  Y(L)=TRA(3)  $  WK(L)=TRA(1)+27
      IF(I.EQ.IT.AND.L.EQ.5) GOTO 35
      IF(I.EQ.IT) GOTO 40
      IF(L.LT.5) GOTO 4
30    35 LASTX=X(L)  $  LASTY=Y(L)
   40 WNR2=WNR2+1  $  KNW=KNR*100+WNR2  $  WK(L+1)=1RW
      AA=10H(1(I6,1X,I  $  AB=10H(2(I6,1X,I
      AC=10H(3(I6,1X,I  $  AD=10H(4(I6,1X,I
      A(2)=10H6,1X,R1,1X  $  A(3)=9H),T77,I4)
35    IF(L.EQ.5) L=4
      IF(L.EQ.1) A(1)=AA  $  IF(L.EQ.2) A(1)=AB
      IF(L.EQ.3) A(1)=AC  $  IF(L.EQ.4) A(1)=AD
      WRITE(2,A)(X(J),Y(J),WK(J+1),J=1,L),KNW
      IF(TRA(1).GT.100) GOTO 6
40    IF(I.LT.IT) GOTO 2
      IF(LASTX.EQ.0) GOTO 50
      A(1)=AA  $  LASTW=1RW  $  KNW=KNW+1
      WRITE(2,A) LASTX,LASTY,LASTW,KNW
   50 ENDFILE2  $  REWIND2
45    STOP
      END
```

```
      PROGRAM KAVAF (TAPE1,TAPE2,OUTPUT,TAPE3=OUTPUT,TAPE4,TAPE5,TAPE6)
      IMPLICIT INTEGER (A-Z)
      DIMENSION X(5),Y(5),WK(5),C(4),BE(4),WEG(5),FIFO(2100),
     C  FIBA(2100),D(2100),R(2100),NEFO(2200),NEBA(2200),
     C  BAR(2200),EAR(2200),LAR(2200),WAR(2200),DIST(5000),
     C  TODO(1450),TREEC(4200)
      LOGICAL LIST(2200)
      COMMON /TREE/ TREEK(2500,3) /SORTVAR/ ILR,IUR,ILC,IUC
      DATA TREEC/4200*0/,BAR/2200*0/,EAR/2200*0/,LAR/2200*0/,
     C WAR/2200*0/,DIST/5000*0/,WEG/5*0/,
     C  PTRC,PTRP,PTRK,PTRD,NRC,RNR2,WNR2/7*0/,Z/20000/,
     C  NC/2100/,NP/2200/,NK/2500/
C     INLEZEN WEGENNET (VAN WEGCO)
  100          WNR1=WNR2
      READ(1,102) X(1),Y(1),BE(1),W,(X(I),Y(I),BE(I),I=2,4),RNR2,WNR2
  102 FORMAT(218,2X,A1,I1,3(2I8,2X,A1,1X),2X,I4,I2)
      IF (EOF(1).NE.0) GO TO 200
      DO 110 I=1,4
      C(I)=X(I)*1000000+Y(I)
      IF (BE(I).NE.1HE) GO TO 110
      L=I
      GO TO 120
  110 CONTINUE
      L=4
  120 CALL VULCOWE(C(1),TREEC,NC,NRC,PTRC)
      IF (BE(1).EQ.1HB) GO TO 130
      CALL VULPIJL(LASTC,C(1),LASTW,PTRP,TREEC,BAR,EAR,LAR,WAR,NP)
      GO TO 140
  130 WNR1=0
  140 IF (WNR1+1 .NE. WNR2) CALL FOUT (1,RNR2*100+WNR2)
      IF (L.EQ.1) GO TO 100
      DO 150 I=2,L
      CALL VULCOWE(C(I),TREEC,NC,NRC,PTRC)
  150 CALL VULPIJL(C(I-1),C(I),W,PTRP,TREEC,BAR,EAR,LAR,WAR,NP)
      IF (BE(L).EQ.1HE) GO TO 100
      LASTC=C(4) $ LASTW=W
      GO TO 100
C     INLEZEN BEDRIJFSGEGEVENS (VAN BEDRCO EN UIT INPUT)
  200 READ(4,160) B,AX,AY
      IF(EOF(4).NE.0) GOTO 201
  160 FORMAT (3I8)
      PTRK=PTRK+1  $  PTRD=PTRD+1
      TREEK(PTRK,1)=B*100+1  $  TREEK(PTRK,2)=AX*10**6+AY
      TREEK(PTRK,3)=PTRD  $  DIST(PTRD)=-PTRK  $  GOTO 200
  201 CODEW=23
      READ(2,202) BEDRNR,KAVELNR,KNR,WNR2
  202 FORMAT(18,I2,T77,2I2)
      IF (EOF(2).NE.0) GO TO 400
      BKP=BEDRNR*100+KAVELNR
      IF (KAVELNR.NE.KNR) CALL FOUT (2,BKP)
  204 WNR1=WNR2
      READ(2,205)(X(I),Y(I),WK(I),I=1,4),KNR,WNR2
  205 FORMAT(4(I6,1X,I6,1X,R1,1X),I77,2I2)
      IF (EOF(2).NE.0) GO TO 400
      IF(KNR.NE.KAVELNR.OR.WNR1+1.NE.WNR2) CALL FOUT  (2,BKP)
      IF (WNR2.EQ.1) GO TO 220
      W=WK(5)-27
```

```
      WEG(W)=WEG(W) +
     C IFIX(SQRT(FLOAT((X(5)-X(1))*(X(5)-X(1)) +
     C (Y(5)-Y(1))*(Y(5)-Y(1)))))
  220 L=1
      IF (WK(1).EQ.CODEW) GO TO 240
      DO 230 I=2,4
      W=WK(I-1)-27
      WEG(W)=WEG(W) +
     C IFIX(SQRT(FLOAT((X(I-1)-X(I))*(X(I-1)-X(I)) +
     C (Y(I-1)-Y(I))*(Y(I-1)-Y(I)))))
      IF (WK(I).NE.CODEW) GO TO 230
      L=I
      GO TO 240
  230 CONTINUE
      X(5)=X(4) $ Y(5)=Y(4) $ WK(5)=WK(4)
      GO TO 204
  240 LAST=X(L)*1000000+Y(L)
      PTRK=PTRK+1
      TREEK(PTRK,1)=BKP $ TREEK(PTRK,2)=LAST
      PTRD=PTRD+1 $ TREEK(PTRK,3)=PTRD $ DIST(PTRD)=-PTRK
      DO 250 I = 1,5
      IF (WEG(I).EQ.0) GO TO 250
      PTRD=PTRD+1
      DIST(PTRD)=WEG(I)*10+I
  250 CONTINUE
      WEG(1)=WEG(2)=WEG(3)=WEG(4)=WEG(5)=0  $  GOTO 200
  400 WRITE 410,PTRC,PTRP,PTRK,PTRD,NRC
  410 FORMAT(*1  PTRC = *,I5,*  PTRP = *,I5,*  PTRK = *,I5,
     F *  PTRD = *,I5,*  NRC = *,I5,//)
C     VERWERKING INGELEZEN GEGEVENS
C     INITIALISEREN EN SORTEREN
      DO 501 I=1,NRC
  501 FIFO(I)=FIBA(I)=0
      DO 502 K=1,PTRP
      I=BAR(K) $ NEFO(K)=FIFO(I) $ FIFO(I)=K
      I=EAR(K) $ NEBA(K)=FIBA(I) $ FIBA(I)=K
  502 CONTINUE
      ILR = 1 $ IUR = PTRK $ ILC = 1 $ IUC = 3
      CALL INDSOR
C     BEREKENING KORTSTE AFSTANDEN
      WRITE(3,550)
  550 FORMAT(1H1,31X,"KAVELAFSTAND"//"   BEDRNR  KAVNR    VERH  SEMIVH
     1ONVERH   WATER    LAND"/)
      IND1=1
  600 DO 601 I=1,NRC
  601 LIST(I)=.FALSE.
      BEDRNR=TREEK(IND1,1)/100  $  KAVPERC=MOD(TREEK(IND1,1),100)
      IND=IND3=IND1
      IND=IND+1
      IF (IND .GT. PTRK) GO TO 620
      IF(TREEK(IND,1)/100.NE.BEDRNR) GOTO 620
  615 IND3=IND
      IND=IND+1
      IF (IND .GT. PTRK) GO TO 620
      IF(TREEK(IND,1)/100.NE.BEDRNR) GOTO 620
      GO TO 615
  620 IF(KAVPERC.NE.1) CALL FOUT (5,BEDRNR),RETURNS(650)
```

Bijlage 25.2

```
      NR=0  $  IND=IND1+1
      IF (IND .GT. IND3) GO TO 640
      DO 630 I=IND,IND3
      CALL NUMMER(N,TREEK(I,2),TREEC),RETURNS(o30) $ LIST(N)=.TRUE.
      NR=NR+1
  630 CONTINUE
      CALL NUMMER(ONTSLPB,TREEK(IND1,2),TREEC),RETURNS(650)
      CALL AFSTAND(Z,NRC,ONTSLPB,NR,LIST,FIFO,FIBA,NEFO,NEBA,D,R,
     C TODO,BAR,EAR,LAR)
  640 CALL RESTAT(IND1,     IND3,Z,D,R,NK,PTRP,TREEC,TREEK,DIST,
     C BAR,EAR,LAR,WAR)
  650 IF (IND3 .EQ. PTRK) GO10 660
      IND1=IND3+1 $ GO TO o00
  660 ENDFILE5  $  REWIND5
      CALL EXIT
      END
```



```
      SUBROUTINE FOUT (NR,VAR),RETURNS(A)
      IMPLICIT INTEGER (A-Z)
      GO TO (10,20,30,40,50,o0,70,80) NR
   10 WRITE (3,11) VAR
   11 FORMAT(* ----- FOUT IN VOLGNUMMER WEGENNET *,I7,/)
      CALL EXIT
   20 WRITE (3,21) VAR
   21 FORMAT(* ----- FOUT IN VOLGNUMMER BEDRIJFSGEGEVENS *,I12,/)
      CALL EXIT
   30 WRITE (3,31) VAR
   31 FORMAT(* ----- FOUT IN INTERNE NUMMERING COORDINATEN *,I12,/)
      CALL EXIT
   40 WRITE(3,41) VAR
   41 FORMAT(* ----- PUNT OP DE WEG BIJ PERCEEL OF BOERDERIJ*,
     F * NIET VERMELD IN WEGENNET *,I12,/)
      RETURN A
   50 WRITE(3,51) VAR
   51 FORMAT(* ----- VAN BEDRIJF *,I8,* IS GEEN HUISBEDRIJFSKAVEL*,
     F * AANWEZIG *,/)
      RETURN A
   60 WRITE(3,61) VAR
   61 FORMAT(* ----- ER IS GEEN WEG VAN DE HUISBEDRIJFSKAVEL*,
     F * NAAR HET PERCEEL *,I12,/)
      RETURN A
   70 WRITE(3,71) VAR
   71 FORMAT(* ----- DE BOVENGRENS VOOR HET AANTAL PUNTEN IN HET*,
     F * WEGENNET IS TE KLEIN *,I6,/)
      CALL EXIT
   80 WRITE(3,81) VAR
   81 FORMAT(* ----- DE BOVENGRENS VOOR HET AANTAL PIJLEN IN HET*,
     F * WEGENNET IS TE KLEIN *,I6,/)
      CALL EXIT
      END
```

```
 SUBROUTINE VULCOWE    74/175  OPT=1                    FTN 4.6+452        16/08/79  14.59.36        PAGE      1

 1               SUBROUTINE VULCOWE(C,TREEC,N,NRC,PTRC)
                 IMPLICIT INTEGER(A-Z)
                 DIMENSION TREEC(1)
                 ADDR=1
 5           100 ADDR1=TREEC(ADDR)
                 IF (ADDR1.EQ.0) GO TO 300
                 IF (C-ADDR1) 110,400,120
             110 ADDR1=TREEC(ADDR+1)/10000
                 IF (ADDR1.EQ.0) GO TO 111
10               ADDR=ADDR1 $ GO TO 100
             111 TREEC(ADDR+1)=(PTRC+1)*10000 + TREEC(ADDR+1)
                 GO TO 300
             120 ADDR1=MOD(TREEC(ADDR+1),10000)
                 IF (ADDR1.EQ.0) GO TO 121
15               ADDR=ADDR1 $ GO TO 100
             121 TREEC(ADDR+1)=PTRC+1 + TREEC(ADDR+1)
             300 NRC=NRC+1
                 IF (NRC .GT. N) CALL FOUT (7,N)
                 TREEC(PTRC+1)=C
20               PTRC=PTRC+2
             400 RETURN
                 END


 SUBROUTINE VULPIJL    74/175  OPT=1                    FTN 4.6+452        16/08/79  14.59.36        PAGE      1

 1               SUBROUTINE VULPIJL(C1,C2,W,PTRP,TREEC,BAR,EAR,LAR,WAR,N)
                 IMPLICIT INTEGER (A-Z)
                 DIMENSION TREEC(1),BAR(1),EAR(1),LAR(1),WAR(1)
                 CALL NUMMER(B,C1,TREEC),RETURNS(100)
 5               CALL NUMMER(E,C2,TREEC),RETURNS(100)
                 IF (B.EQ.E) CALL FOUT (3,C1)
                 PTRP=PTRP+1
                 IF (PTRP .GT. N) CALL FOUT (8,N)
                 BAR(PTRP)=B $ EAR(PTRP)=E $ WAR(PTRP)=W
10               X1=C1/1000000 $ Y1=MOD(C1,1000000)
                 X2=C2/1000000 $ Y2=MOD(C2,1000000)
                 LAR(PTRP)=IFIX(SQRT(FLOAT((X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2))))
                 RETURN
             100 CALL EXIT
15               END
```

```
      SUBROUTINE AFSTAND(Z,N,A,NR,H ON LIST,FIFO,FIBA,NEFO,NEBA,D,R,
     C TODO,BAR,EAR,LAR)
      IMPLICIT INTEGER (A-Z)
      DIMENSION FIFO(1),FIBA(1),NEFO(1),NEBA(1),D(1),R(1),TODO(1),
     C          BAR(1),EAR(1),LAR(1)
      LOGICAL H ON LIST(1)
      DO 100 J=1,N
      R(J)=-A $ D(J)=Z
  100 CONTINUE
      H=A $ DH=0 $ D(H)=0 $ R(H)=A $ UP=0
      DO 400 K=2,N
      G=FIFO(H)
  210 IF (G .EQ. 0) GO TO 250
      J=EAR(G)
      IF (R(J) .GE. 0) GO TO 240
      DJ=D(J) $ TDJ=DH+LAR(G)
      IF (TDJ .GE. DJ) GO TO 240
      IF (DJ .NE. Z) GO TO 230
      UP=UP+1 $ TODO(UP)=J
  230 D(J)=TDJ $ R(J)=-H
  240 G=NEFO(G) $ GO TO 210
  250 G=FIBA(H)
  260 IF (G .EQ. 0) GO TO 300
      J=BAR(G)
      IF (R(J) .GE. 0) GO TO 290
      DJ=D(J) $ TDJ=DH+LAR(G)
      IF (TDJ .GE. DJ) GO TO 290
      IF (DJ .NE. Z) GO TO 280
      UP=UP+1 $ TODO(UP)=J
  280 D(J)=TDJ $ R(J)=-H
  290 G=NEBA(G) $ GO TO 260
  300 IF (UP .EQ. 0) RETURN
      DH=Z
      DO 310 G=1,UP
      IF (D(TODO(G)) .GE. DH) GO TO 310
      J=G $ DH=D(TODO(G))
  310 CONTINUE
      H=TODO(J) $ R(H)=-R(H) $ TODO(J)=TODO(UP) $ UP=UP-1
      IF (.NOT. H ON LIST(H)) GO TO 400
      NR=NR-1
      IF (NR .EQ. 0) RETURN
  400 CONTINUE
      END
```

```
      SUBROUTINE RESTAT(IND1,     IND3,Z,D,R,N,M,TREEC,TREEK,DIST,
     C BAR,EAR,LAR,WAR)
      IMPLICIT INTEGER (A-Z)
      DIMENSION  WW(5),WEG(5),D(1),R(1),TREEC(1),
     C          TREEK(N,1),DIST(1),BAR(1),EAR(1),LAR(1),WAR(1)
      BKP=TREEK(IND1,1)
      BEDRNR=BKP/100  $  KAVNR=MOD(TREEK(IND1,1),100)
    2 READ(6,10) BEDR,KB
      IF(EOF(6)) 4,3
   10 FORMAT(I8,38X,I2)
    3 IF(BEDRNR.EQ.BEDR) GOTO 6
      GOTO 2
    4 WRITE(3,12) BEDRNR
   12 FORMAT (1H ,I10," KOMT NIET VOOR BIJ BEDRIJFSNUMMERS")
      GOTO 7
    6 IF(KB.EQ.1) GOTO 7
      WRITE(3,5) BEDRNR,KAVNR
      WRITE(5,5) BEDRNR,KAVNR
    5 FORMAT(I9,5X,I2,5(7X,1H0))
    7 CALL NUMMER(ONTSLPB,TREEK(IND1,2),TREEC) $ PTRD=TREEK(IND1,3)
   20 PTRD=PTRD+1 $ DD=DIST(PTRD)
      IF (DD .LE. 0) GO TO 100
      W=MOD(DD,10) $ WW(W)=DD/10
      GO TO 20
  100 IND1=IND1+1  $  IF(IND1.GT.IND3) RETURN
      DO 400 IND=IND1,IND3
      WEG(1)=WEG(2)=WEG(3)=WEG(4)=WEG(5)=0
      BKP=TREEK(IND,1)  $  BDR=BKP/100  $  KNR=MOD(TREEK(IND,1),100)
      CALL NUMMER(ONTSLPP,TREEK(IND,2),TREEC),RETURNS(400)
      PTRD=TREEK(IND,3)
  210 PTRD=PTRD+1 $ DD=DIST(PTRD)
      IF (DD .LE. 0) GO TO 300
      W=MOD(DD,10) $ WEG(W)=DD/10
      GO TO 210
  300 J=ONTSLPP
      IF (D(J) .EQ. Z) CALL FOUT (6,BKP),RETURNS(400)
  310 I=J  $  J=R(I)
      DO 320 K=1,M
      IF (.NOT.(BAR(K).EQ.I .AND. EAR(K).EQ.J .OR.
     C          BAR(K).EQ.J .AND. EAR(K).EQ.I)) GO TO 320
      ARR=K $ GO TO 321
  320 CONTINUE
  321 WARR=WAR(ARR) $ LARR=LAR(ARR)
      WEG(WARR)=WEG(WARR)+LARR
      IF(J.NE.ONTSLPB) GOTO 310
  326 DO 330 I = 1,5
  330 WEG(I)=IFIX((FLOAT(WEG(I)+WW(I)))/100.+.5)
      WRITE(3,340) BDR,KNR,(WEG(W),W=1,5)
      WRITE(5,340) BDR,KNR,(WEG(W),W=1,5)
  340 FORMAT(I9,5X,I2,5(5X,I3))
  400 CONTINUE
      RETURN
      END
```

Bijlage 25.6

```
      C  PROGRAMMA TOEDELING MET CIN EN UIT DIGITALISERING BEREKENDE KAVAFS.
      C
            PROGRAM TOEDEL(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT,TAPE3,TAPE4,
           1TAPE5,TAPE6,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12,TAPE13,TAPE14,
           2TAPE15,TAPE16,TAPE17,TAPE18,TAPE19,TAPE20,TAPE21,TAPE22)
            INTEGER FOUT
            DIMENSION INDEX(4000),MB(50),MK(50),OTU(50),OT(50),MA(50),MR(50),
           1 MV(50),MU(50),IAA(11),IBB(11),ICC(11),IAAA(10),STU(11,9),
           2 TOP(11,9),TUH(11,9),KAK(11,9),KAB(11,9),OBB1(11,9),KABB1(11,9),
           3 OBB2(11,9),KABB2(11,9),OBB(11,9),KABB(11,9),GKA(11,9),GKAS(11,9),
           4 AKB(11,9),GKU(11,9),PUH(11,9),GBU(11,9),PUHB(11,9),PAB(11,9),
           5 PUHB1(11,9),PAB1(11,9),PUHB2(11,9),PAB2(11,9),IAB(11),IBC(11),
           6 ICD(11),LU(4000),IKP(4000),IKN(4000),STUS(11,9),IBTT(8),LAB(11),
           7 LBC(11),LCD(11)
            COMMON /INV/ IK(4000),IK1(4000),R(4000)  /SORT/ ILR,IUR
            EQUIVALENCE (INDEX,IK)
            REWIND3 $ REWIND4 $ REWIND5  $ REWIND17
            REWIND10  $  REWIND18  $  REWIND19
      C OVERNEMEN INPUT CIN VAN MAGNEETTAPE, SORTEREN OP BEDRIJFS- EN
      C KAVELNUMMER EN OP FILE ZETTEN
            READ(1,10)  NDA,IA,IB,IC,IZ
         10 FORMAT(I2,3A10,I5)
            M=0
         20 M=M+1
            READ(1,30) IAA(M),IBB(M),ICC(M)
            IF(EOF(1)) 40,20
         30 FORMAT(3A10)
         40 IS=M-1
            DO 50 M = 3,IS
         50 READ(1,30) IAB(M),IBC(M),ICD(M)
            IST=IS-3
            DO 60 M = 1,IST
         60 READ(1,70) IBTT(M)
         70 FORMAT (A10)
            J=0
         80 J=J+1
            READ(1,200) IKH,IKK,UG,IRH,IVH,IUH
            IF(EOF(1)) 100,90
        200 FORMAT(I8,I3,F9.4,I3,I4,I2)
         90 WRITE(10,200) IKH,IKK,UG,IRH,IVH,IUH
            GOTO 80
        100 JA=J-1  $  REWIND 10
            DO 110 I = 1,10
        110 IAAA(I)=10H----------
            M=0  $  NDG=1
        120 M=M+1
            READ(17,130) ND,I,K,IO,IVC,LV,LS,LO,LW,LL
            IF(EOF(17)) 160,140
        130 FORMAT (I2,I8,2X,I2,8X,I7,47X,I2,21X,I3,4I2)
        140 U=IO*1./10000.
            IF(IVC.GE.NDA) GOTO 150
            IF(ND.GT.NDG) NDG=ND
            IK(M)=I*100+K  $  IK1(M)=LV*10**8+LS*10**6+LO*10**4+LW*100+LL
            R(M)=ND*1000+U $ GOTO 120
        150 M=M-1  $  GOTO 120
        160 MI=M-1  $  ILR=1  $  IUR=MI
      C MI = AANTAL KAVELS IN CIN
```

```
                CALL INLCOL                                                    TOEDOC1    59
                DO 170 M = 1,MT                                                TOEDOC1    60
 60             JL=IK1(M) $ LV=JL/10**8 $ JL=JL-LV*10**8 $ LS=JL/10**6         TOEDOC1    61
                JL=JL-LS*10**6 $ LO=JL/10**4 $ JL=JL-LO*10**4 $ LW=JL/100      TOEDOC1    62
                LL=JL-LW*100 $ BY=R(M) $ ND=BY/1000 $ O=BY-ND*1000             TOEDOC1    63
              170 WRITE(3,180)ND,IK(M),O,LV,LS,LO,LW,LL                        TOEDOC1    64
              180 FORMAT(I3,I10,F9.4,I4,4I3)                                   TOEDOC1    65
 65             ENDFILE 3  $  REWIND 3  $  REWIND 17                           TOEDOC1    66
            C INLEZEN AANVULLENDE GEGEVENS,SORTEREN OP BEDRIJFS- EN KAVELNR    TOEDOC1    67
            C EN OP FILE ZETTEN                                                TOEDOC1    68
            C BEREKENDE KAVELAFSTANDEN TOEVOEGEN                               TOEDOC1    69
            C NDA = NUMMER VAN HET EERSTE DORPSBEHOREN WAARIN BUITEN BLOK BEDRIJVEN TOEDOC1    70
 70         C VOORKOMEN                                                        TOEDOC1    71
                IVM=1                                                          TOEDOC1    72
                N=NS=NSS=0                                                     TOEDOC1    73
              190 N=N+1                                                        TOEDOC1    74
                READ(21,210) I1,K1,IR,IV,IO,O1,LV1,LS1,LO1,LW1,LL1             TOEDOC1    75
 75             IF(EOF(21)) 230,220                                            TOEDOC1    76
              210 FORMAT(I6,2I3,I4,I2,F9.4,I4,4I3)                             TOEDOC1    77
              220 IK(N)=I1*100+K1 $ IK1(N)=LV1*10**8+LS1*10**6+LO1*10**4+LW1*100+LL1 TOEDOC1    78
                R(N)=IR*10**7+IV*10**4+IO*1000+O1                              TOEDOC1    79
                IF(IV.LE.IVM) GOTO 190                                         TOEDOC1    80
 80             IVM=IV $ GOTO 190                                              TOEDOC1    81
              230 NT=N-1  $  ILR=1  $  IUR=NT  $  REWIND 21                    TOEDOC1    82
            C NT = AANTAL KAVELS BINNEN HET BLOK                               TOEDOC1    83
                CALL INDCOL                                                    TOEDOC1    84
                WRITE(2,240)                                                   TOEDOC1    85
 85             240 FORMAT(1H1)                                                TOEDOC1    86
              250 READ(19,260) NRB,NRK,LV,LS,LO,LW,LL                          TOEDOC1    87
                IF(EOF(19)) 310,270                                            TOEDOC1    88
              260 FORMAT(I9,5X,I2,5(5X,I3))                                    TOEDOC1    89
              270 NRBK=NRB*100+NRK                                             TOEDOC1    90
 90             IF(NRK.EQ.1) GOTO 250                                          TOEDOC1    91
                DO 280 M = 1,NT                                                TOEDOC1    92
                IF(NRBK.EQ.IK(M)) GOTO 300                                     TOEDOC1    93
              280 CONTINUE                                                     TOEDOC1    94
                NS=NS+1                                                        TOEDOC1    95
 95             WRITE(2,290) NRBK                                              TOEDOC1    96
              290 FORMAT(1H ,"KAVEL :",I12,3X,"KOMT NIET VOOR BIJ AANVULLENDE GEG") TOEDOC1    97
              300 IK1(M)=LV*10**8+LS*10**6+LO*10**4+LW*100+LL                  TOEDOC1    98
                IF(IK(M).EQ.IK(M+1)) IK1(M+1)=IK1(M)                           TOEDOC1    99
                IF(IK(M).EQ.IK(M+2)) IK1(M+2)=IK1(M)                           TOEDOC1   100
100             IF(IK(M).EQ.IK(M+3)) IK1(M+3)=IK1(M)                           TOEDOC1   101
                GOTO 250                                                       TOEDOC1   102
              310 REWIND 19                                                    TOEDOC1   103
                DO 330 M = 1,NT                                                TOEDOC1   104
                JL=IK1(M) $ LV1=JL/10**8 $ JL=JL-LV1*10**8 $ LS1=JL/10**6      TOEDOC1   105
105             JL=JL-LS1*10**6 $ LO1=JL/10**4 $ JL=JL-LO1*10**4 $ LW1=JL/100  TOEDOC1   106
                LL1=JL-LW1*100 $ BY=R(M) $ IR=BY/10**7 $ BY=BY-IR*10**7        TOEDOC1   107
                IV=BY/10**4 $ BY=BY-IV*10**4 $ IO=BY/1000 $ O1=BY-IO*1000      TOEDOC1   108
                WRITE(4,320)IK (M),IR,IV,IO,O1,LV1,LS1,LO1,LW1,LL1             TOEDOC1   109
              320 FORMAT(111,I3,I4,I2,F9.4,I4,4I3)                             TOEDOC1   110
110           330 CONTINUE                                                     TOEDOC1   111
                ENDFILE4 $ REWIND4                                             TOEDOC1   112
                IF(NS.NE.0.OR.NSS.NE.0) GOTO 2380                              TOEDOC1   113
                WRITE(2,340)                                                   TOEDOC1   114
              340 FORMAT(1H1,36HKAVELS KOMEN NIET VOOR BIJ INPUT CIN//9H   BEDRNR,  TOEDOC1   115
```

Bijlage 26.2

```
115           14H KAV,4H VAK/)                                          TOEDOC1   116
          C FOUTIEVE KAVELNUMMERS WORDEN UITGESORTEERD EN GEPRINT       TOEDOC1   117
              DO 350 L = 1,MT                                           TOEDOC1   118
            350 READ(3,360)  IK1(L)                                     TOEDOC1   119
            360 FORMAT(3X,I10)                                          TOEDOC1   120
120           NTT=0                                                     TOEDOC1   121
              DO 410 N =1,NT                                            TOEDOC1   122
              READ(4,370) IV                                            TOEDOC1   123
            370 FORMAT(14X,I4)                                          TOEDOC1   124
              DO 380 M =1,M1                                            TOEDOC1   125
125           IF(IK(N).EQ.IK1(M)) GOTO 400                              TOEDOC1   126
            380 CONTINUE                                                TOEDOC1   127
              JM=IK (N) $ I1=JM/100 $ K1=JM-I1*100                      TOEDOC1   128
              WRITE(2,390)I1,K1,IV                                      TOEDOC1   129
            390 FORMAT(1H ,I8,2I4)                                      TOEDOC1   130
130           NTT=NTT+1                                                 TOEDOC1   131
            400 IF(N.LT.NT) GOTO 410                                    TOEDOC1   132
              IF(NTT.NE.0) GOTO 2380                                    TOEDOC1   133
              GOTO 420                                                  TOEDOC1   134
            410 CONTINUE                                                TOEDOC1   135
135         420 M=N=0 $ REWIND3 $ REWIND4                               TOEDOC1   136
          C INPUT CIN EN AANVULLENDE GEGEVENS WORDEN SAMENGEVOEGD       TOEDOC1   137
            430 N=N+1                                                   TOEDOC1   138
              READ(4,320)IK (N),IR,IV,IU,O1,LV1,LS1,LO1,LW1,LL1         TOEDOC1   139
            440 M=M+1                                                   TOEDOC1   140
140           READ(3,180)ND,IK1(M),O,LV,LS,LO,LW,LL                     TOEDOC1   141
              IF(IK(N).NE.IK1(M)) GOTO 440                              TOEDOC1   142
          C WAAR NODIG WORDEN KAVELS GESPLITST                          TOEDOC1   143
              IF(IK (N).EQ.IK (N+3)) GOTO 460                           TOEDOC1   144
              IF(IK (N).EQ.IK (N+2)) GOTO 500                           TOEDOC1   145
145           IF(IK (N).EQ.IK (N+1)) GOTO 540                           TOEDOC1   146
              JM=IK (N) $ I1=JM/100 $ K1=JM-I1*100                      TOEDOC1   147
              IF(ND.GE.NDA) GOTO 450                                    TOEDOC1   148
              WRITE(5,680)I1,K1,O,LV,LS,LO,LW,LL,IR,IV,IU               TOEDOC1   149
            680 FORMAT(I8,I3,F9.4,I4,5I3,I4,I2)                         TOEDOC1   150
150           GOTO 580                                                  TOEDOC1   151
            450 WRITE(5,680)I1,K1,O,LV1,LS1,LO1,LW1,LL1,IR,IV,IU        TOEDOC1   152
              GOTO 580                                                  TOEDOC1   153
            460 LE=N $ LK=N+3                                           TOEDOC1   154
              DO 490 L=LE,LK                                            TOEDOC1   155
155           JM=IK (N) $ I1=JM/100 $ K1=JM-I1*100                      TOEDOC1   156
              IF(ND.GE.NDA) GOTO 470                                    TOEDOC1   157
              WRITE(5,680)I1,K1,O1,LV,LS,LO,LW,LL,IR,IV,IU              TOEDOC1   158
              GOTO 480                                                  TOEDOC1   159
            470 WRITE(5,680)I1,K1,O1,LV1,LS1,LO1,LW1,LL1,IR,IV,IU       TOEDOC1   160
160         480 IF(N.EQ.LK) GOTO 490                                    TOEDOC1   161
              N=N+1                                                     TOEDOC1   162
              READ(4,320)IK (N),IR,IV,IU,O1,LV1,LS1,LO1,LW1,LL1         TOEDOC1   163
            490 CONTINUE                                                TOEDOC1   164
              GOTO 580                                                  TOEDOC1   165
165         500 LE=N $ LK=N+2                                           TOEDOC1   166
              DO 530 L=LE,LK                                            TOEDOC1   167
              JM=IK (N) $ I1=JM/100 $ K1=JM-I1*100                      TOEDOC1   168
              IF(ND.GE.NDA) GOTO 510                                    TOEDOC1   169
              WRITE(5,680)I1,K1,O1,LV,LS,LO,LW,LL,IR,IV,IU              TOEDOC1   170
170           GOTO 520                                                  TOEDOC1   171
            510 WRITE(5,680)I1,K1,O1,LV1,LS1,LO1,LW1,LL1,IR,IV,IU       TOEDOC1   172
```

```
                520 IF(N.EQ.LK) GOTO 530                                          TOEDOC1   173
                    N=N+1                                                         TOEDOC1   174
                    READ(4,320) IK(N),IR,IV,IO,O1,LV1,LS1,LO1,LW1,LL1             TOEDOC1   175
175             530 CONTINUE                                                      TOEDOC1   176
                    GOTO 580                                                      TOEDOC1   177
                540 LE=N $ LK=N+1                                                 TOEDOC1   178
                    DO 570 L=LE,LK                                                TOEDOC1   179
                    JM=IK (N) $ I1=JM/100 $ K1=JM-I1*100                          TOEDOC1   180
180                 IF(ND.GE.NDA) GOTO 550                                        TOEDOC1   181
                    WRITE(5,680)I1,K1,O1,LV,LS,LO,LW,LL,IR,IV,IO                  TOEDOC1   182
                    GOTO 560                                                      TOEDOC1   183
                550 WRITE(5,680)I1,K1,O1,LV1,LS1,LO1,LW1,LL1,IR,IV,IO             TOEDOC1   184
                560 IF(N.EQ.LK) GOTO 570                                          TOEDOC1   185
185                 N=N+1                                                         TOEDOC1   186
                    READ(4,320) IK(N),IR,IV,IO,O1,LV1,LS1,LO1,LW1,LL1             TOEDOC1   187
                570 CONTINUE                                                      TOEDOC1   188
                580 IF(N.LT.NT) GOTO 430                                          TOEDOC1   189
                    ENDFILE5 $ REWIND3 $ REWIND4 $ REWIND5                        TOEDOC1   190
190         C  PER BEDRIJF GROOTTEKLASSE BEREKENEN                                TOEDOC1   191
                    M=FOUT=OP=LBD=MAB=MTT=0                                       TOEDOC1   192
                590 M=M+1  $  MTT=MTT+1                                           TOEDOC1   193
                    READ(5,680) IK(M),K,O,LV,LS,LO,LW,LL,IR,IV,IO                 TOEDOC1   194
                    IF(EOF(5)) 630,600                                            TOEDOC1   195
195             600 IF(M.EQ.1) GOTO 620                                           TOEDOC1   196
                    IF(IK(M).NE.IK(M-1)) GOTO 640                                 TOEDOC1   197
                620 OP=OP+O  $  GOTO 690                                          TOEDOC1   198
                630 LBD=1                                                         TOEDOC1   199
                640 IF(OP.LT.5.) IBO=1 $ IF(OP.GE.5..AND.OP.LT.10.) IBO=2         TOEDOC1   200
200                 IF(OP.GE.10..AND.OP.LT.15.) IBO=3                             TOEDOC1   201
                    IF(OP.GE.15..AND.OP.LT.20.) IBO=4                             TOEDOC1   202
                    IF(OP.GE.20..AND.OP.LT.30.) IBO=5                             TOEDOC1   203
                    IF(OP.GE.30..AND.OP.LT.40.) IBO=6                             TOEDOC1   204
                    IF(OP.GE.40..AND.OP.LT.50.) IBO=7 $ IF(OP.GE.50.) IBO=8       TOEDOC1   205
205                 READ(22,650) NE,IKB,IBT                                       TOEDOC1   206
                650 FORMAT (I2,I10,I3)                                            TOEDOC1   207
                    IF(IK(M-1).NE.IKB) GOTO 660                                   TOEDOC1   208
                    MAB=MAB+1                                                     TOEDOC1   209
                    WRITE(20,610) NE,IKB,IBT,IBO                                  TOEDOC1   210
210             610 FORMAT (I2,I10,2I3)                                           TOEDOC1   211
                    OP=0  $  OP=OP+O  $  GOTO 690                                 TOEDOC1   212
                660 WRITE(2,670) IK(M-1)                                          TOEDOC1   213
                670 FORMAT (1H ,"BEDRIJFSNUMMER",I10,"KOMT NIET VOOR OP LIJST MET BEDR TOEDOC1   214
                   1IJFSNUMMERS")                                                 TOEDOC1   215
215                 OP=0  $  OP=OP+O  $  FOUT=1                                   TOEDOC1   216
                690 IF(LBD.EQ.0) GOTO 590                                         TOEDOC1   217
                    MTT=MTT-1                                                     TOEDOC1   218
                    IF(FOUT.EQ.1) GOTO 2380                                       TOEDOC1   219
                    ENDFILE 20  $  REWIND 5  $  REWIND 20  $  REWIND 22           TOEDOC1   220
```

Bijlage 27.1

Instructies voor het gereed maken van de te digitaliseren kaart

Te gebruiken materiaal

Gebruikt wordt een maatvaste calque van de topografische kaart schaal 1 : 10 000. Afhankelijk van de grootte van het gebied kunnen meerdere kaartbladen nodig zijn.

Alle beschrijvingen op de kaarten worden met watervaste inkt uitgevoerd.

Nummering kaartbladen

Het bladnummer bestaat uit drie cijfers. Het 1e cijfer geeft het totaal aantal bladen aan, het 2e cijfer is een nul en het 3e cijfer is het eigenlijke bladnummer. Indien b.v. 2 kaartbladen worden gebruikt hebben deze de nummers 201 en 202.

Aangeven wegennet

Het wegennet wordt overgenomen van de gebruikerskaarten van de CIN. Wegen die niet nodig zijn voor het berekenen van de afstanden van de bedrijven naar de vakken kunnen worden weggelaten. Dit betreft voornamelijk onverharde wegen, insteekwegen en wegen binnen dorpskommen. Indien ook kavelafstanden moeten worden berekend dient er rekening mee te worden gehouden dat de wegen voor het ontsluiten van de kavels op de kaart voorkomen. Wegen buiten het ruilverkavelingsblok die nodig zijn voor het berekenen van afstanden moeten op de kaart worden aangegeven. Verharde wegen worden aangegeven in rood, semi-verharde wegen in groen en onverharde wegen in geel.

Snijpunten van wegen en knikpunten in de wegen worden met een stip gemarkeerd, voor zover deze punten niet voldoende zijn aangegeven door de plaats van bedrijven, vakken en kavels. De punten die in het wegennet worden aangegeven mogen niet dichter dan 2 mm bij elkaar liggen.

Plaats bedrijven

De plaats van de bedrijven wordt aangegeven met een stip in het wegennet op de plaats waar vanaf het bedrijfsgebouw de weg wordt bereikt. Het nummer van het bedrijf wordt op de kaart geschreven

en aangepijld naar de stip in het wegennet.

Buiten blok bedrijven die op slechts één plaats het blok kunnen binnenkomen worden aangegeven op de plaats waar de blokgrens wordt gepasseerd. Buiten blok bedrijven die meerdere mogelijkheden hebben om het blok te bereiken worden aangegeven op de plaats waar zij zijn gevestigd of op de plaats waar de keuze van routes zich voordoet.

Meerdere bedrijven die dicht bij elkaar liggen kunnen met dezelfde stip in het wegennet worden aangegeven. Dit voorkomt een te groot aantal punten in het wegennet.

Plaats vakken

De plaats van de vakken wordt eveneens met een stip in het wegennet aangegeven. Het vaknummer wordt met grote cijfers op de kaart geschreven, met een volle lijn omcirkeld en aangepijld met een volle lijn naar het punt in het wegennet.

Punten van vakken mogen nooit samenvallen met punten van bedrijven.

Plaats kavels

De plaats van kavels wordt aangegeven indien kavelafstanden moeten worden berekend. Alleen de plaats van kavels waarvoor de afstand moet worden berekend wordt op de kaart gezet.

Voor kavels die niet aan een weg zijn gelegen wordt met een stip het ontsluitingspunt aangegeven en vervolgens de knikpunten in de ontsluitingsroute tot en met het punt in het wegennet. Ook de wegkwaliteit van de ontsluitingsroute wordt aangegeven. Kavels die wel aan een weg liggen worden alleen met een stip in het wegennet aangegeven.

Het kavelnummer wordt op de kaart geschreven en aangepijld naar het ontsluitingspunt van de kavel of naar het punt in de weg. De notatie van het kavelnummer mag in vereenvoudigde vorm of volledig zijn, b.v. 780-971/$_2$ of 0780097102.

Het punt in het wegennet van een kavel mag nooit samenvallen met het punt van het bedrijf waar de kavel bij hoort.

Bijlage 27.3

Instructies voor digitalisering

De instructies zijn bestemd voor het digitaliseren met een Haromat die voorzien is van ponsbanduitvoer.

1. Ruitennet

a. intoetsen nummer blad CR LF. Het bladnummer bestaat altijd uit 3 cijfers

b. code kaartschaal intoetsen CR LF

codes: 1 = schaal 1 : 10 000
2 = schaal 1 : 5 000
3 = schaal 1 : 2 000

c. intoetsen RD coördinaten ZW ruitpunt CR LF

d. beginnend met het ZW ruitpunt de ruitpunten digitaliseren. Bij het beginpunt moet de Haromat op X = 10 000 en Y = 10 000 worden ingesteld. Bij grotere kaarten is het voldoende de ruitpunten om en om te digitaliseren. Slecht zichtbare ruitpunten in ieder geval overslaan

Indien tijdens de meting een fout wordt gemaakt, opnieuw beginnen.

2. Wegennet

a. intoetsen bladnummer CR LF

b. intoetsen B, 1e punt van de 1e route digitaliseren, wegkwaliteits-code naar volgende punt intoetsen, 2e punt, code, enz. tot en met laatste punt van de 1e route, intoetsen E. Vervolgens 1e punt 2e route digitaliseren, code enz. Per route mogen maximaal 100 punten voorkomen. In een route mag slechts één wegkwaliteit voorkomen. Wordt tijdens de meting van een route een fout gemaakt, dan F CR LF intoetsen en de route opnieuw meten, beginnend met een B, 1e punt route, code enz.
Na het intoetsen van de E na de laatste route wordt geëindigd met het intoetsen van CR LF.

3. Plaats bedrijfsgebouwen

a. nummer blad CR LF
b. bedrijfsnummer intoetsen. Dit nummer moet uit 8 cijfers bestaan, zodat eventueel aanvulling met nullen nodig is. Bijvoorbeeld 777-815 wordt 07770815
c. punt in weg bij bedrijfsnummer digitaliseren
d. alle bedrijven afwerken. Indien tijdens de meting van een bedrijf een fout wordt gemaakt F CR LF intoetsen en het bedrijf opnieuw meten

4. Plaats vakken

a. nummer blad CR LF
b. nummer 1e vak intoetsen. Dit nummer moet uit 3 cijfers bestaan, dus vak 1 wordt ingetoetst als 001
c. punt in weg bij vaknummer digitaliseren
d. vakken in numerische volgorde afwerken. In het algemeen zullen de vakken vanaf 1 doorlopend genummerd zijn. Het is echter mogelijk dat enkele nummers ontbreken

Na een gemaakte fout F CR LF intoetsen en het vak opnieuw meten.

5. Plaats kavels

a. bladnummer CR LF
b. kavelnummer intoetsen. Het nummer moet uit 10 cijfers bestaan, zodat ook hier eventueel aanvulling met nullen nodig is, bijvoorbeeld 780-971/2 wordt 0780097102
c. ontsluitingspunt digitaliseren. Indien de kavel aan een weg ligt is dit punt niet op de kaart aangegeven. Er moet dan een punt in de buurt van het punt in de weg worden gedigitaliseerd

d. code wegkwaliteit naar volgende punt, coördinaten volgende punt enz. tot en met het punt in de weg

codes: 1 = verhard
2 = semi-verhard
3 = onverhard
4 = over water
5 = over land

e. kavels achtereenvolgens afwerken. Indien tijdens de meting van een kavel een fout wordt gemaakt, F CR LF intoetsen en de kavel, beginnend met het kavelnummer, opnieuw meten. Wordt al tijdens het intoetsen van het kavelnummer een fout geconstateerd, dan eerst het kavelnummer tot 10 cijfers volmaken, coördinaten registreren en pas daarna F CR LF intoetsen.