

**KONINKLIJK NEDERLANDS
METEOROLOGISCH INSTITUUT**

VERSLAGEN

V - 338

G. J. Prangma

Gegevensopslag voor CTD en XBT gegevens.

De Bilt 1980

Publikatienummer: K. N. M. I. V-338 (O. O.)

Koninklijk Nederlands Meteorologisch Instituut,
Oceanografisch Onderzoek,
Postbus 201,
3730 AE De Bilt,
Nederland.

Gegevensopslag voor CTD en XBT gegevens.

1. Inleiding.

De meetgegevens die bij een CTD- of XBT-meting worden verkregen doorlopen verschillende stadia van controle en bewerking (zie apart te verschijnen verslag). Elk van deze stadia zal in het algemeen een invoerstroom van meetgegevens nodig hebben en in vele gevallen een bewerkte stroom uitvoergegevens produceren, welke in een volgende bewerking als invoer gebruikt zal worden.

Teneinde de lees- en schrijf-procedures zo eenvoudig mogelijk te houden, is het gewenst voor alle gegevensopslag één standaardvorm te kiezen, die zonder tussenkomst van de programmeur/operateur gelezen kan worden en een minimaal ingrijpen van de programmeur bij het schrijven vereist.

Door de aard van het menglaagonderzoek zal het voorts nodig zijn, dat de lees- en schrijf-procedures eenvoudig gehanteerd kunnen worden op zeer verschillende computer-configuraties (thans PDP-8, HP-21 en B6700). Tenslotte zal de vorm van de gegevensopslag zodanig moeten zijn, dat een semi-automatische omzetting naar de internationale uitwisselingsstandaard (het GF3 format, zie ref. 1) op de genoemde configuraties (met name HP-21 en B6700) zonder al te veel programmeerwerk mogelijk is.

N.B. Voor meetreeksen die vòòr 1979 zijn ingezameld, hebben de ruwe gegevens een andere vorm dan in dit verslag beschreven wordt. Alle verdere bewerkingen produceren echter wel de beschreven standaardvorm.

2. Beschrijving van de file-structuur.

De vorm waarin de gegevens worden opgeslagen wordt bepaald door een keuze te maken uit de mogelijkheden die de ter beschikking staande configuraties bieden:

a. opslag medium:

- ponsband 8 bits per karakter (PDP en HP-21)
- magneetband 9 sporen 8 bits per karakter (PDP-8, HP-21 en B6700)
- magnetische schijf zeer afhankelijk van de machine (HP-21 en B6700)

b. woordlengte computergeheugen:

- PDP-8 12 bits equivalent met $1\frac{1}{2}$ karakter
- HP-21 16 bits equivalent met 2 karakters
- B6700 48 bits equivalent met 6 karakters

c. nauwkeurigheid van de CTD/XBT gegevens:

15 bits per meetgegeven (zie ref. 2)

Daarnaast spelen nog andere factoren een rol zoals: Efficiënt gebruik van de beschikbare ruimte, mogelijkheid om bepaalde gegevens vanaf ponskaart of dergelijke (80 karakters per regel) in te voeren voor commentaar en de hoeveelheid gegevens per data-cyclus.

Een en ander heeft tenslotte tot de keus geleid waarbij de gegevens worden verdeeld in "records" van 240 karakters lang, hetgeen overeenkomt met 3 ponskaarten, 240 PDP woorden, 120 HP woorden en 40 B6700 woorden. Om de capaciteit van een magneetband beter te benutten, worden bij opslag op magneetband dan nog eens 8 van dergelijke "records" samengevoegd tot 1 "block" van 1920 karakters lengte, wat de benodigde lengte op magneetband terugbrengt van ca. 7.2 inch tot ca. 3.0 inch. Voor opslag op magnetische schijf worden bij de B6700 3 "records" samengepakt tot een "block" van 120 woorden = 720 karakters.

De nauwkeurigheid van de meetgegevens bedraagt zoals aangegeven 15 bits, zodat deze als 2 karakters kunnen worden opgeslagen. Als we dit ook voor de andere, berekende gegevens kunnen realiseren, b.v. door een vaste schaalfactor al dan niet gecombineerd met een toegevoegde additieve constante, dan kan een data-cyclus bestaan uit een even aantal karakters, dat deelbaar is op 240. We komen dan tot cyclus-lengten van: 2, 4, 6, 8, 10, 12, 16, 20, 24, 30, 40, 48, 60 en 80 karakters per datacyclus. Op technische gronden zijn cycluslengten van 2 en 4 karakters echter verboden. Thans zijn in gebruik cyclus-lengten van 6 en 12 karakters (zie paragraaf 4). De gegevens in een datacyclus kunnen nog verder onderverdeeld worden als daar aanleiding toe bestaat, immers, we kunnen de gegevens interpreteren:

- als binaire getallen, hetzij 8 bits (1 karakter) hetzij 16 bits (2 karakters);
- als alfanumerieke gegevens, waarvoor gezien de standaard-codes van PDP-8 en HP-21 de ASCII code (7 bits met 8-ste bit = 0) gekozen is.

Uit de beschrijving van de datacycli die in de eerste records van een file wordt gegeven, moet blijken welke interpretatie gekozen moet worden (zie verder ook paragraaf 4).

Een file bestaat uit een reeks verschillende "records":

- een "header-record", bevattende alle gegevens die voor het interpreteren van de volgende gegevens van belang zijn, b.v. stationsnaam, begin-positie en -tijd, indeling van de data cycli, enz.;
- indien meer commentaar of beschrijvende details gewenst zijn, dan in het "header-record" kunnen worden ondergebracht, dan volgen direct aansluitend nog 1 of meer "comment-records";
- "data-records", bestaande uit cycli die zijn samengesteld zoals in het "header-record" (en eventueel volgende "comment-records") beschreven;
- "end-of-file-record", bevattende informatie die voor het correct lezen van de gegevens van belang zijn (b.v. aantal records in de file) alsmede gegevens die voor het interpreteren van belang zijn, zoals eind-tijd en -positie van de meting.

Tabel I geeft van elk van deze "record" types een gedetailleerde beschrijving. Op een magneetband komen naast de reeds genoemde records nog 2 andere types voor:

- "tape-header-record", welke de naam van de tape, het spoelnummer, en eventuele andere nuttige informatie bevat, zonodig verder ondergebracht in volgende "comment-records";
- "end-of-tape-record", het laatste record op de magneetband, dat aangeeft dat er geen gegevens meer volgen. Ook hier kan weer commentaar worden toegevoegd, dat in dit geval echter niet in volgende "comment-records" kan worden uitgebreid, immers het gaat om het laatste record op de band.

Samenvattend komen we tot de volgende file-structuren.

- a. Op ponsband 1 data-file, bestaande uit "header-record", 1 of meer "data-records" en een "end-of-file-record". Het geheel is zonodig over meerdere rollen ponsband verdeeld.
- b. Op magneetschijf kunnen meerdere data-files aanwezig zijn, elk samengesteld als hierboven reeds vermeld (zie ook fig. 1) en bekend onder een eigen unieke naam.
- c. Op magneetband zullen we in het algemeen meerdere data-files aantreffen, voorafgegaan door een "tapeheader-file" en gevolgd door een "end-of-tape" file zoals schematisch weergegeven in fig. 2, waarbij alle files van elkaar gescheiden zijn door een zgn. "tape-mark" en de "end-of-tape" file gevolgd wordt door 2 van deze "tape-marks" als aanduiding dat dit het einde van alle gegevens op deze magneetband is.

3. Indeling van de verschillende "record"-typen.

Bij de keuze van de indeling van de verschillende record-typen moet er op gelet worden, dat ze alle eenvoudig van elkaar onderscheiden moeten kunnen worden, terwijl tevens copiëren in GF3 format geen al te grote problemen mag opleveren. Zo komen we dan tot een indeling waarbij het begin van elk record identiek is met GF3 (namelijk de eerste 2 karakters een automatische copiëring mogelijk maken) gevolgd door een meer leesbare combinatie van 4 letters die de aard van het record aangeven. Om in overeenstemming met GF3 te blijven zijn de karakters 81 en 82 alsmede 161 en 162 identiek met de karakters 1 en 2, in alle record-typen, behalve het data-record, waar een en ander onnodig is en bovendien tot ernstige complicaties zou kunnen leiden.

Verder is het zo, dat de karakters 73 t/m 80, 153 t/, 160 en 223 t/m 240 gebruikt worden voor volgnummer-indicatie, behalve wederom in de data-records. Er is gekozen voor 8 karakters voor volgnummers (in plaats van 3 in GF3) om compatibel te zijn met de B6700 conventies. Dit staat een automatische omzetting naar GF3 echter niet in de weg.

Onderstaand overzicht geeft een algemene beschrijving van de indeling van de verschillende record-typen; een gedetailleerde beschrijving is te vinden in tabel I.

- "comment-record". Dient als uitbreiding van de ruimte voor commentaar in de "tape-header", "file-header" en "end-of-file" records en mag dan ook alleen direct aansluitend aan deze records voorkomen.

De karakters 1 t/m 6 vermelden het recordtype: "O COMM".

De karakters 81 t/m 161 bevatten een "O" en de karakters 82 en 162 een spatie.

De posities 73 t/m 80, 153 t/m 160 en 223 t/m 240 zijn gereserveerd voor volgnummer aanduiding.

Alle posities zijn gevuld met leesbare (afdrukbare) karakters in 7 bit ASCII code met het 8-ste bit = 0.

- "file-header record". Bevat alle nodige informatie over de inhoud van de volgende (meet-) gegevens. Dit omvat ondermeer: File-naam, een copie van de naam en het nummer van de magneetband als de opslag op magneetband plaats vindt, datum en jaar van aanmaak van deze data-file, de afmetingen van de blokken, de records en de lengte van een data-cyclus, stationsnaam, begin-positie en -tijd, projectnaam, scheepsnaam, waterdiepte, plaats-bepalingsmethode en een beschrijving van de opbouw van een data-cyclus.

Eventueel kan deze beschrijving wordt voortgezet in een volgend "comment-record", terwijl verdere relevante inlichtingen in klare tekst gegeven kunnen worden.

De karakters 1 t/m 6 vermelden het record-type: "5 HDRL".

De karakters 81 en 161 bevatten een "5" en de karakters 82 en 162 een spatie.

De karakters 7 t/m 30 bevatten binaire informatie, de overige posities zijn alle leesbare ASCII code karakters.

- "data-record". Bevat in de karakters 1 t/m 6 "7 DATA" in ASCII code als aanduiding van het record-type en is overigens ingedeeld zoals vermeld moet zijn in de voorafgaande "file-header". In de regel zal het om 16-bit binaire (meet-)waarden gaan.

N.B. Als de data-cyclus langer is dan 6 karakters dan wordt achter de record-identificatie (7 DATA) een aantal binaire nullen geschreven om op een hele data-cycluslengte uit te komen.

- "end-of-file record". Bevat naast een copie van overeenkomstige gegevens uit de "file-header" ook het aantal data-records in de file, de eind-tijd en positie van het station, alsmede eventueel afsluitend commentaar van de operateur/bewerker van de gegevens, welk commentaar zonodig aangevuld kan worden in een volgend "comment record". Afgezien van de type-aanduiding "9 EOFL" in de karakters 1 t/m 6 en "9" in de karakters 81 en 161, is de indeling identiek aan die van de "file-header".

Voor gebruik op magneetband onderscheiden we voorts nog:

- "tape-header record". Bevat de naam en het nummer van de magneetband, de naam van het project, het nummer van het verslag waarin de indeling van de records beschreven is, en eventueel commentaar dat voor de gebruiker van de magneetband relevant kan zijn. Dit commentaar kan desgewenst ook worden voortgezet in een volgend "comment record".

De type aanduiding is te vinden in de karakters 1 t/m 6: "1 VOLL" en 81 en 161: "1". Het beschrijvende rapport is terug te vinden in de karakters 63 t/m 72, evenals de inhoud van dit record in ASCII code.

- "end-of-tape record". Afgezien van de type-aanduiding "8 EOTL" in de karakters 1 t/m 6 en "8" in de karakters 81 en 161, identiek aan de "tape-header", met dien verstande dat geen aanvullend commentaar in volgende "comment records" is toegestaan.

Tot slot van deze paragraaf nog enkele opmerkingen voor het gebruik van de verschillende records.

1. Aan het einde van een data-file wordt het laatste record aangevuld met binaire nullen tot een volledig record is verkregen, dat wordt weggeschreven.
2. Op magneetband worden blokken aan het einde van een file (een tape-header file, een data-file en de end-of-tape file) aangevuld met records die uit binaire nullen bestaan tot een volledig block van 1920 karakters is verkregen, alvorens dit wordt weggeschreven.

4. Indeling van datacycli.

Hoewel de indeling van de data-cycli in principe van geval tot geval in de file-header-records beschreven wordt, gelden op dit moment de volgende vier standaard indelingen:

- in file van ruwe gegevens kunnen voorkomen: Cycli van 6 karakters,
t.w. + CTD cycli
+ XBT cycli
+ tijd cycli
- in files van geschoonde en van karakteristieke meetpunten komen uitsluitend cycli voor van 12 karakters, d.w.z. 6 16-bit binaire getallen.

De indelingen van de CTD en XBT cycli gaan beide uit van 3 getallen per data-cylus, t.w. temperatuur, druk en conductiviteit resp. 3 opvolgende temperatuurwaarden. In de programma's op de PDP-8 die ook ponsbanduitvoer kunnen produceren, wordt dan nog een tweetal karakters toegevoegd om eventuele ponsfouten makkelijk te kunnen lokaliseren, n.l. twee ASCII karakters die tezamen een "carriage return line feed" actie geven op een teletype printer. Om dit patroon bovendien nog van de 15 bit nauwkeurigheid van de meetgegevens te kunnen onderscheiden is het 8-ste bit van het eerste (carriage return) karakter gezet. Details zijn samengevat in tabel II.

De tijdcyclus die temidden van CTD en XBT gegevens in ruwe gegevens files voorkomt, geeft de begintijd aan van een ononderbroken reeks meetgegevens, alsmede de onderlinge tijdafstand tussen twee opvolgende datacycli. Ook deze tijdcyclus wordt in de PDP-uitvoer gevolgd door een "carriage return line feed" combinatie met het 8-ste bit van de "carriage return" gezet ter herkenning van ponsfouten.

Bovendien is de tijdcyclus van "gewone" meetresultaten te onderscheiden, doordat ook het 8-ste bit van het eerste karakter in de cyclus gezet is. In tegenstelling tot de CTD en XBT cycli is de tijdcyclus opgebouwd uit 8-bit binaire getallen, zoals in tabel III te concluderen is.

N.B. Bij verdere bewerkingen worden allereerst de (verder overbodige) "carriage return/line feed" combinaties verwijderd, zodat die in files die door andere dan de PDP-inzamel-programma geproduceerd worden geheel ontbreken.

Cycli van geschoonde en karakteristieke meetgegevens bestaan uit 6 getallen van 16-bit elk. Allereerst zijn dit waarden voor temperatuur, druk en saliniteit, maar daarnaast ook informatie over de tijd die aan deze meetcyclus moet worden toegekend, alsook een aanduiding omtrent de plaats van deze cyclus in de gehele meting, b.v. is het het begin- of eind-punt van een ononderbroken serie metingen, is het een karakteristiek punt in de temperatuur of in de saliniteit, of is het een standaarddiepte, dan wel een keerpunt in een yoyo-meetreeks. Details zijn te vinden in tabel IV.

5. Computer-routines voor het lezen en schrijven van CTD en XBT gegevens.

De procedures voor het lezen en schrijven van CTD/XBT gegevens in de vorm als in de voorafgaande paragrafen beschreven, zijn ontwikkeld voor gebruik op de B6700 met opslag op magneetschijf (zie Appendix A1) en voor opslag op magneetband (zie Appendix A2).

De versies voor gebruik op de HP-21 en de PDP-8 zijn van dit basispakket afgeleid en worden elders in samenhang met de voor die configuraties bestemde verwerkingsprogramma's beschreven.

Het basispakket bestaat uit de volgende 6 procedures:

- OPENINPUT. Verzorgt het lezen van de file-header van de gevraagde file in een array (INPUTHEADER), leest het eerste datarecord in een buffer (JASINCHARS), en initialiseert tellers en indices voor het verdere inlezen.
- GETINPUT. Als er geen datacyclus meer beschikbaar is in de invoerbuffer, dan wordt het volgende record ingelezen en worden tellers en indices opnieuw geïnitieerd. Anders worden alleen index en cyclus-teller bijgewerkt.

Wordt bij het lezen een end-of-file record gedetecteerd, dan wordt dit door een boolean variabele ENDOFJASINFILE aan het hoofd-programma gemeld. Loze datacycli (uitsluitend nullen) worden automatisch overgeslagen.

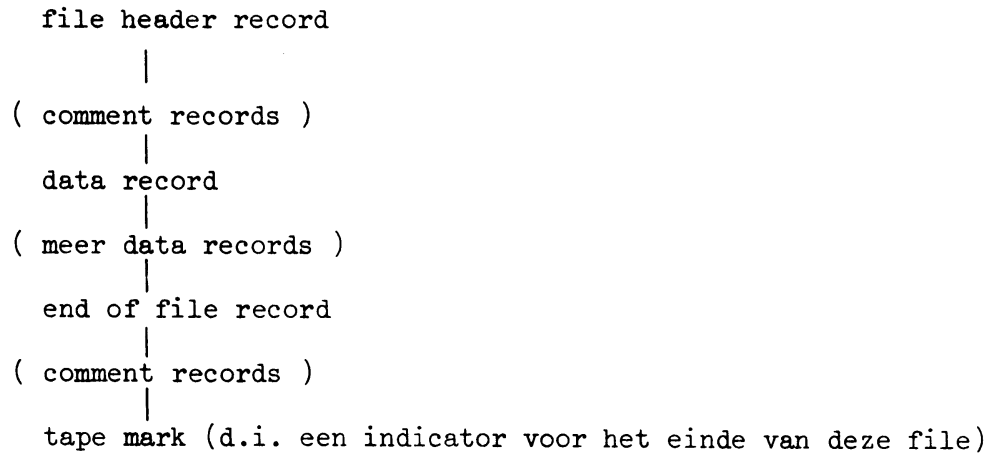
- CLOSE INPUT. De bijgehouden tellers worden vergeleken met de inhoud van het gelezen end-of-file record. Eventuele discrepanties alsmede het resultaat van de bijgehouden telling worden via de regeldrukker gemeld.
- OPENOUTPUT. Creëert een nieuwe file. Stelt de file-header samen (in OUTPUTHEADER) en schrijft deze weg. Initialiseert indices en tellers voor de verdere uitvoer.
- WRITEOUTPUT. Copieert de gegevens, welke het hoofd-programma in een buffertje (array OUTPUTDATA) geplaatst heeft, naar de uitvoerbuffer. Indien deze vol is, wordt de inhoud weggeschreven en worden indices en tellers opnieuw geïnitieerd. Anders worden alleen index en cyclusteller bijgewerkt.
- CLOSEOUTPUT. Vult de laatste buffer zonedig aan met nullen en schrijft de inhoud weg. Stelt het end-of-file record samen en schrijft dit weg. Maakt de file permanent en meldt relevante gegevens via de regeldrukker.

De wijze waarop deze routines gebruikt worden is globaal weergegeven in fig. 3. Voor de opslag op magneetband zijn nog extra routines nodig voor het lezen en schrijven van de tape-header records, voorts om een gevraagde file op de band te kunnen opzoeken. Deze routines (zie Appendix A2) spreken verder voor zichzelf.

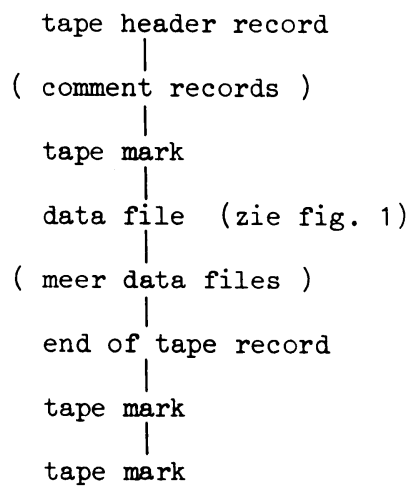
N.B. Omdat op magneetbanden een block-grootte van 1920 karakters gebruikt wordt voor een efficiënter gebruik van de beschikbare ruimte, vindt in de magneetband versie nog een verdere buffering plaats tot hele blokken van 1920 karakters. Dit leidt tot het resultaat dat een laatste block van een file altijd aangevuld moet worden met nullen tot de vereiste lengte verkregen is.

Referenties.

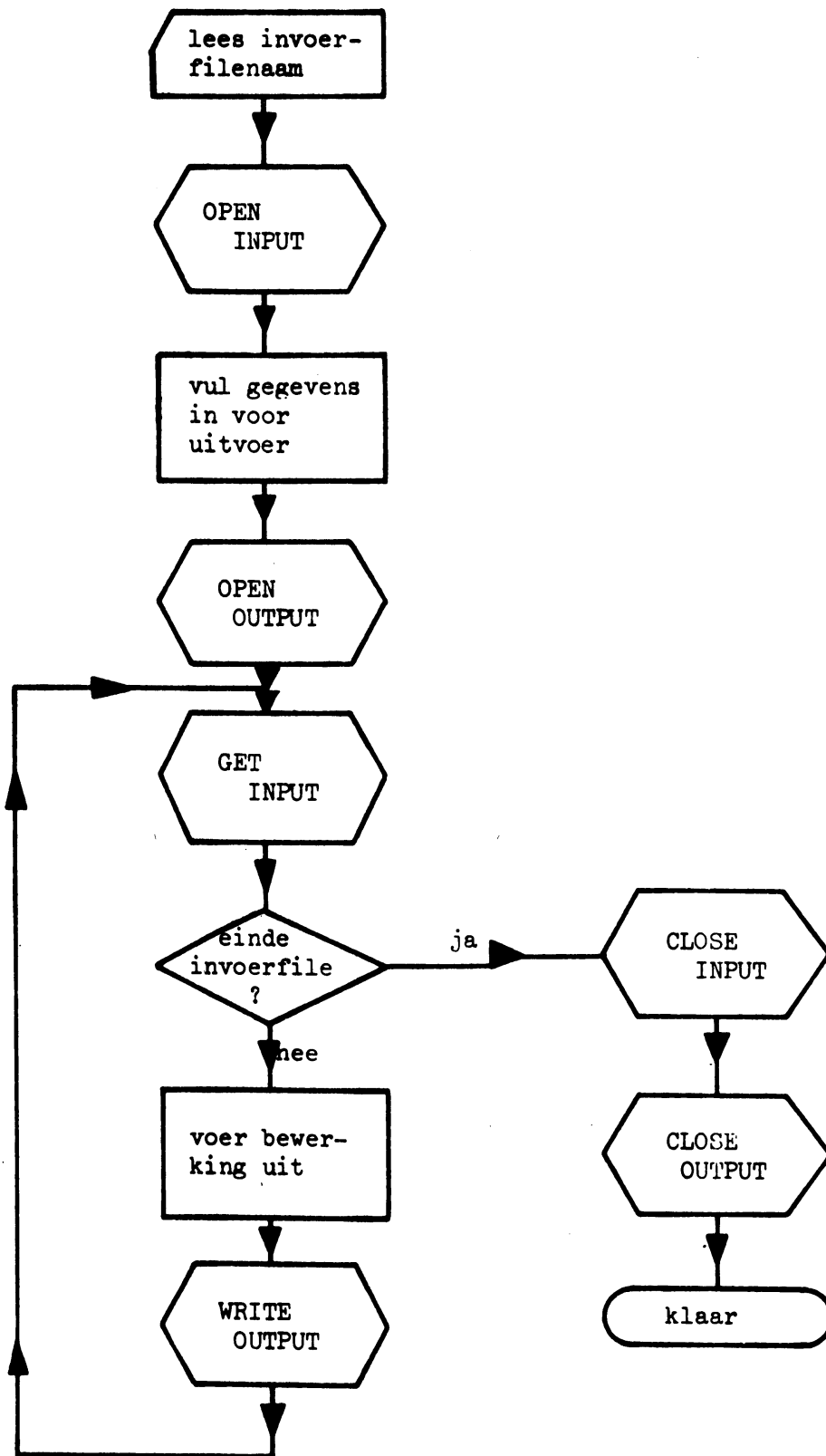
1. P. Winiarski, "Second report on the IOC general magnetic tape format containing a full documentation of GF3 and the parameter code table system", August 1978.
2. G.J. Prangma, "Instrumentatie voor het oceanisch menglaag-onderzoek", KNMI Verslagen V-291, 1977.



Figuur 1. Indeling van een gegevens file.



Figuur 2. Indeling van een magneetband.



Figuur 3. Basisschema voor de verwerkingsprogramma's.

TABEL I

Indeling van de verschillende typen "records".

a. Comment record.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-6	ASCII	0 COMM	record identificatie
7-72	ASCII	klare tekst	deel commentaar
73-80	idem	volgnummer	volgnummer van de kaart waarvan het commentaar werd ingelezen
81	idem	0	record identificatie
82	idem	spatie	idem
83-152	idem	klare tekst	vervolg commentaar
153-160	idem	volgnummer	
161	idem	0	record identificatie
162	idem	spatie	idem
163-232	idem	klare tekst	vervolg commentaar
233-240	idem	volgnummer	

b. Tape header record.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-6	ASCII	1 VOLL	record identificatie
7-30	binair	nullen	ongebruikt
31-36	ASCII	tape-nummer	nummer van de betreffende magneetband
37-42	idem	volume naam	"naam" van de magneetband N.B. Op de HP schijf: WERKnn, de naam van de door de PDP-8 aangemaakte "WERK" tape.
43	idem	spatie	
44	idem	/	voor gebruik in B6700/B6800 filenaam-conventie
45-52	idem	spaties	ongebruikt
53	idem	.	voor gebruik in B6700/B6800 filenaam-conventie.
54	idem	spatie	ongebruikt
55-62	idem	projectnaam	naam van het project waarvoor de gegevens verzameld worden, b.v. JASIN78
63-72	idem	rapport	naam en/of nummer van het rapport dat de indeling van de records van deze tape beschrijft (Hier dus: V-338)
73-80	idem	volgnummer	
81	idem	1	record identificatie
82	idem	spatie	idem

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
83-152	idem	commentaar	commentaar
153-160	idem	volgnummer	
161	idem	1	record identificatie
162	idem	spatie	idem
163-232	idem	klare tekst	commentaar
233-240	idem	volgnummer	

c. File header record.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-6	ASCII	5 HDRL	record identificatie
7-8	16 bit binair	jaar	jaar (minus het honderdtal) waarin de file werd aangemaakt.
9-10	idem	dag	dag in dit jaar waarop de file werd aangemaakt
11-12	idem	recordsize	lengte van het record in karakters (=240)
13-14	idem	blocksize	lengte van een block in karakters (=240 of 1920)
15-16	idem	records	aantal records in deze file (=0, n.l. aan het begin van de file is nog niets geschreven)
17-28	idem	0	reserve
19-20	idem	data-type	soort gegevens in de file 0 ruwe gegevens 1 geschoonde gegevens 2 karakteristieke gegevens
21-22	idem	cyclus lengte	lengte van een data-cyclus in deze file in karakters
23-30	idem	nullen	reserve
31-44	ASCII	copie tape- header record	zie aldaar
45-52	ASCII	filenaam	naam van de file Uit de eerste 2 letters van de filenaam blijkt of het gaat om: XB XBT gegevens DE CTD gegevens van een diepe cast YY CTD gegevens van een yoyo station VE CTD gegevens van een vergelijking met een waterschepper
53-62	idem	copie tape- header record	
63-72	idem	stationsnaam	aanduiding van het station binnen het project
73-80	idem	volgnummer	nummer van de kaart waarvan de gegevens voor filenaam, station etc. werden gelezen.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
81	idem	5	record indentificatie
82	idem	spatie	idem
83-92	idem	datum-tijd groep	bij begin station
93-102	idem	breedte	bij begin station
103-112	idem	lengte	bij begin station
113-116	idem	waterdiepte	in hele meters bij begin station
117-118	idem	navigatie code	wijze van bepaling van de positie: SA satelliet OM Omega LA Loran A LC Loran C DC Decca DR Dead reckoning
119-122	idem	schip	naam van het platform
123-152	idem	klare tekst	beschrijving van de gegevens of commentaar
153-160	idem	volgnummer	van de ingelezen kaart met de gegevens
161	idem	5	record indentificatie
162	idem	spatie	idem
163-232	idem	klare tekst	vervolg beschrijving gegevens of com- mentaar
233-240	idem	volgnummer	

d. Data record.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-6	ASCII	7 DATA	record indentificatie
7-240	binair	gegevens	indeling zoals beschreven in de file header en eventuele volgende comment- records. N.B. Als de cycluslengte meer dan 6 karakters bedraagt, begint de eerste cyclus niet in karakter 7, maar zo- veel verder als de cyclus lengte - 6 bedraagt. De ontbrekende karakters worden gevuld met binaire nullen.

e. End-of-tape record.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-6	ASCII	8 EOTL	record indentificatie
81	ASCII	8	idem
161	idem	8	idem

Overigens is de indeling identiek aan die van de tape header record.

f. End-of-file record.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-6	ASCII	9 EOFL	record identificatie
15-16	16 bit binair	records	aantal data records in deze file
81	ASCII	9	record identificatie
161	idem	9	idem

Overigens is de indeling identiek aan die van de file header record, met dien verstande, dat datum-tijd groep, breedte en lengte en waterdiepte betrekking hebben op het einde van het station en dat een beschrijving van de data-cycli als regel vervangen zal zijn door relevant commentaar met betrekking tot de evaluatie van de kwaliteit van de gegevens in deze file.

TABEL II

Indeling van de data cycli van ruwe gegevens.

XBT cyclus.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-2	16 bit binair	temperatuur 1	3 opeenvolgende resultaten van de A/D conversie (15 bit) op het XBT kanaal, met een interval zoals blijkt uit de voorafgaande tijd-cyclus.
3-4	idem	temperatuur 2	
5-6	idem	temperatuur 3	
7	8 bit	4"8D"	ASCII carriage return met 8-ste bit = 1 ASCII line feed
8	8 bit	4"OA"	

CTD cyclus.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-2	16 bit binair	temperatuur	Alle meetgegevens zijn sommen van vier opeenvolgende A/D conversies op het betreffende kanaal, gedeeld door 4. De tijdsafstand tussen twee geregistreerde data cycli is als aangegeven in de voorafgaande tijd-cyclus.
3-4	idem	druk	
5-6	idem	conductiviteit	
7	8 bit	4"8D"	ASCII carriage return met 8-ste bit = 1 ASCII line feed
8	8 bit	4"OA"	

N.B. De karakters 7 en 8 komen alleen voor in files die aangemaakt worden door de PDP-8 tijdens de inzameling van de gegevens. Deze karakters worden bij de verdere verwerking verwijderd.

TABEL III

Indeling tijd-cyclus.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1	8 bit binair	interval + 128	Tijdsinterval tussen twee opvolgende meetgegevens in 1/100 sec. Het 8-ste bit is gezet (=1) in verband met het herkennen van deze tijdcyclus (8-ste bit heeft een waarde van 128).
2	idem	1/100 sec	Aantal stappen van 1/100 sec sinds de vorige gehele seconde
3	idem	seconde	
4	idem	minuut	
5-6	16 bit binair	512*uur +dag	Het dagnummer (in het lopende jaar) omvat de minst-significante 9 bits, het uur staat in de 7 meest-significante bits.
7	8 bit	4"8D"	ASCII carriage return met 8-ste bit = 1
8	idem	3"0A"	ASCII line feed

N.B. De karakters 7 en 8 worden alleen geproduceerd door het PDP-8 inzamelprogramma en worden bij de verdere bewerking verwijderd.

TABEL IV

Indeling van cycli van geschoonde en karakteristieke gegevens.

<u>Karakters</u>	<u>Type</u>	<u>Gebruik</u>	<u>Opmerkingen</u>
1-2	16 bit binair	temperatuur	In 1/1000 graden Celsius met een offset van 2 graden Celsius ter vermindering van negatieve waarden
3-4	idem	druk	In cbar
5-6	idem	saliniteit	Berekend uit de temperatuur, druk en conductiviteit, na correctie voor ijkings. In ppm met een offset van 30 ^o /oo 0 voor XBT gegevens
7	8 bit binair	0	
8	idem	vlag	0 alleen in geschoonde gegevens aanwezig 1 karakteristiek in de temperatuur 3 karakteristiek in de saliniteit 9 begin/einde van een aaneengesloten reeks meetgegevens 10 standaarddiepte 100 ontbrekende standaarddieptes
9	8 bit	uur	geteld vanaf de begin dag, dus eventueel groter dan 23
10	idem	minuut	
11	idem	seconde	
12	idem	1/100 sec	aantal stappen van 1/100 sec sinds de vorige hele seconde.

GENIO/DISK
=====

APPENDIX A 1

BEGIN FILE LP(KIND=PRINTER);

CARD(KIND=READER);

EBCDIC ARRAY LISTLINE(0:119),CARDLINE(0:79);
LABEL ABORTLABEL,ENDFCARD;
DEFINE CRLF = 4"800" #;
DO BEGIN COMMENT THE MAIN PROGRAM IS ESSENTIALLY A BIG LOOP;
READ(CARD+80,CARDLINE)ENDFCARD;
COMMENT THE INPUT IS ASSUMED TO START WITH A FILENAME CARD
CONTAINING IN COLUMNS,
1 - 15 FILENAME
21 - 30 STATION NAME/ACRONYM
31 - 40 LATITUDE AT BEGIN OF STATION
41 - 50 LONGITUDE AT BEGIN OF STATION
51 - 54 WATER DEPTH AT BEGIN OF STATION
55 - 64 DATE/TIME AS YYMMDDHHMM AT BEGIN OF STATION
65 - 66 NAVIGATIONAL AID
67 - 70 SHIP
73 - 80 CARD SEQUENCE NUMBER

IF CARDLINE = "YR0" OR
CARDLINE = "0ER0" OR
CARDLINE = "XBR0" THEN
BEGIN COMMENT GENERALIZED INPUT HANDLING ROUTINES FOR CTO/XBT DATA;
FILE JASINFILE(KIND=DISK,MAXRECSIZE=40,BLOCKSIZE=120,UNITS=WORDS);

EBCDIC ARRAY JASINCHARS,JASINHEADER(0:239);
ARRAY JASINWORDS(0)=JASINCHARS,JASINLABEL(0)=JASINHEADER;
COMMENT THIS EQUIVALENCE IS FOR EASE OF IO STATEMENTS;
POINTER JASINPTR;
INTEGER INCHARS,INRECS,INTYPE,INLGT,INTITLE,INCYCLES;
LABEL ENDOFJASINFILE;
BOOLEAN VALIDINPUT;
PROCEDURE OPENINPUT;

BEGIN REPLACE JASINFILE.TITLE BY CARDLINE;
READ(JASINFILE,40,JASINLABEL);
JASINPTR:=JASINHEADER(0);
IF JASINPTR NEQ 7"5 HDRL" THEN
BEGIN REPLACE LISTLINE BY " " FOR 120;
REPLACE LISTLINE BY "NO HEADER LABEL FOUND FOR ",
JASINFILE.TITLE, " FATAL ERROR. #";
WRITE(PLP,120,LISTLINE);
GO TO ABORTLABEL;
END;

INTYPE:=REAL(JASINPTR+18,2);
INLGT:=REAL(JASINPTR+20,2);
REPLACE LISTLINE BY " " FOR 120;
REPLACE LISTLINE BY "OPEN FILE ", JASINFILE.TITLE,
" CYCLE LENGTH: ", INLGT FOR 3 NUMERIC,
" DATA TYPE: ", INTYPE FOR 2 NUMERIC;
WRITE(PLP,SPACE 3,120,LISTLINE);
INRECS:=0;
INCYCLES:=0;
INCHARS:=0;

END;
PROCEDURE GETINPUT;
BEGIN VALIDINPUT:=FALSE;
DO
BEGIN IF INCHARS = 0 THEN
BEGIN READ(JASINFILE,40,JASINWORDS)(ENDOFJASINFILE);
INCHARS:=240;
JASINPTR:=JASINCHARS(0);
WHILE JASINPTR = 7"0 COMM" DO
READ(JASINFILE,40,JASINWORDS)(ENDOFJASINFILE);
IF JASINPTR = 7"9 EOF" THEN GO TO ENDOFJASINFILE;
INRECS:=INRECS + 1;
END;
JASINPTR:=JASINPTR + INLGT;
IF JASINPTR NEQ 4"00" FOR INLGT THEN VALIDINPUT:=TRUE;
INCHARS:=INCHARS - INLGT;

END UNTIL VALIDINPUT;
INCYCLES:=INCYCLES + 1;
END;
PROCEDURE CLOSEINPUT;
BEGIN REPLACE LISTLINE BY " " FOR 120;
IF INRECS NEQ REAL(JASINCHARS(14),2) THEN
BEGIN REPLACE LISTLINE BY "RECORD COUNT IN EOF LABEL: ",
REAL(JASINCHARS(14),2) FOR 6 NUMERIC;
WRITE(PLP,120,LISTLINE);
REPLACE LISTLINE BY " " FOR 120;

END;
REPLACE LISTLINE BY "CLOSED FILE ", JASINFILE.TITLE,
" AFTER ",INRECS FOR 6 NUMERIC, " RECORDS AND ",
INCYCLES FOR 7 NUMERIC, " CYCLES. #";
WRITE(PLP,SPACE 3,120,LISTLINE);
END;
COMMENT GENERALIZED OUTPUT ROUTINES FOR CTO/XBT DATA;
FILE JASOUTFILE(KIND=DISK,MAXRECSIZE=40,BLOCKSIZE=120,UNITS=WORDS,
FLEXIBLE=TRUE,MYUSE=OUT);

EBCDIC ARRAY JASOUTCHARS,JASOUTHEADER(0:239),HEADERCOMMENT(0:135);
ARRAY JASOUTWORDS(0)=JASOUTCHARS,JASOUTLABEL(0)=JASOUTHEADER;
COMMENT THIS EQUIVALENCE IS FOR EASE OF IO STATEMENTS;
INTEGER ARRAY OUTPUTDATA(0:9);
ARRAY DATE(0:0);

EBCDIC ARRAY OUTDATPTR(0)=OUTPUTDATA;
COMMENT WE NEED A POINTER TO THE OUTPUTDATA ARRAY, SO THIS IS
A CONVENIENT WAY TO GET ONE;
POINTER JASOUTPTR;
INTEGER YEAR,DAY;
PROCEDURE OPENOUTPUT;
BEGIN COMMENT CHECK DATA TYPE;
IF OUTTYPE < 0 OR OUTTYPE > 3 THEN
BEGIN REPLACE LISTLINE BY " " FOR 120;
REPLACE LISTLINE BY "ILLEGAL OUTPUT DATA TYPE (",
OUTTYPE FOR 2 NUMERIC, ") OPENING FILE ",
JASOUTFILE.TITLE, " FATAL ERROR. #";
WRITE(PLP,SPACE 3,120,LISTLINE);
GO TO ABORTLABEL;
END;

COMMENT COMPOSE AND WRITE HEADER LABEL;
DATE(0)=TIME(10);
YEAR:=INTEGER(POINTER(DATE),3);
DAY:=INTEGER(POINTER(DATE),3,3);
OUTPUTDATA(0)=YEAR;
OUTPUTDATA(1)=DAY;
OUTPUTDATA(2)=240;
OUTPUTDATA(3)=OUTTYPE;

Table with 2 columns: Address (e.g., 0001000, B.CC00 IS) and Hexadecimal values (e.g., 000:000:0, SEGMENT 00003).

```

OUTPUTDATA(4):=OUTLGT;
REPLACE JASOUTFILE.TITLE BY CAROLINE;
REPLACE JASOUTHEADER BY 7"5 HDRL";
OUTDATAPTR(4) FOR 2, OUTDATAPTR(10) FOR 2,
OUTDATAPTR(16) FOR 2, OUTDATAPTR(16) FOR 2,
4"00" FOR 4,
OUTDATAPTR(22) FOR 2, OUTDATAPTR(28) FOR 2,
4"00" FOR 8,
7" " FOR 6,
CAROLINE(6) FOR 6 WITH EBCDICTOASCII,
7" " FOR 9,
CAROLINE(6) FOR 10 WITH EBCDICTOASCII,
7" " FOR 9,
CAROLINE(20) FOR 10 WITH EBCDICTOASCII,
7" " FOR 5,
CAROLINE(54) FOR 10 WITH EBCDICTOASCII,
CAROLINE(30) FOR 24 WITH EBCDICTOASCII,
CAROLINE(64) FOR 6 WITH EBCDICTOASCII,
7" " FOR 38, 7"5", 7" " FOR 79;
WRITE(JASOUTFILE,40,JASOUTLABEL);
FILL JASOUTWORDS(1) WITH 7"7 DATA",39(0);
OUTCHARS:=OUTLGT;
OUTRECS:=OUTCYCLES:=0;
JASOUTPTR:=JASOUTCHARS(OUTLGT);
END;
PROCEDURE WRITEOUTPUT;
BEGIN IOUT:=0;
WHILE IOUT < OUTLGT DO
BEGIN REPLACE JASOUTPTR + IOUT BY
OUTDATAPTR(4 + 3*IOUT) FOR 2;
IOUT:=IOUT + 2;
END;
OUTCYCLES:=OUTCYCLES+1;
OUTCHARS:=OUTCHARS + OUTLGT;
IF OUTCHARS = 240 THEN
BEGIN WRITE(JASOUTFILE,40,JASOUTWORDS);
OUTRECS:=OUTRECS + 1;
OUTCHARS:=OUTLGT;
JASOUTPTR:=JASOUTCHARS(OUTLGT);
FILL JASOUTWORDS(1) WITH 7"7 DATA",39(0);
END ELSE JASOUTPTR:=JASOUTPTR + OUTLGT;
END;
PROCEDURE CLOSEOUTPUT;
BEGIN COMMENT FILL LAST RECORD WITH ZERO'S;
IF OUTCHARS NEQ 0 THEN
BEGIN WRITE(JASOUTFILE,40,JASOUTWORDS);
OUTRECS:=OUTRECS + 1;
END;
REPLACE JASOUTHEADER BY 7"9 EOF1";
REPLACE JASOUTHEADER(80) BY 7"9";
REPLACE JASOUTHEADER(160) BY 7"9";
REPLACE JASOUTHEADER(40) BY CAROLINE(30) FOR 34
WITH EBCDICTOASCII;
REPLACE JASOUTHEADER(82) BY
CAROLINE(54) FOR 10 WITH EBCDICTOASCII,
CAROLINE(30) FOR 24 WITH EBCDICTOASCII,
CAROLINE(64) FOR 6 WITH EBCDICTOASCII,
7" " FOR 38, 7"5", 7" " FOR 79;
COMMENT READ OR GENERATE COMMENT;
COMMENT COPY OUT RECORD COUNT IN JASOUTHEADER;
OUTPUTDATA(C):=OUTRECS;
REPLACE JASOUTHEADER(14) BY OUTDATAPTR(4) FOR 2;
WRITE(JASOUTFILE,40,JASOUTLABEL);
LOCK(JASOUTFILE,*) ;
REPLACE LISTLINE BY " " FOR 120;
REPLACE LISTLINE BY "CLOSED OUTPUT FILE ",
JASOUTFILE.TITLE, " AFTER ",
OUTCYCLES FOR 7 NUMERIC,
" CYCLES IN ",
OUTRECS FOR 6 NUMERIC, " RECORDS.";
WRITE(LEFTSPACE 3,120,LISTLINE);
END;
COMMENT GENERAL DATA HANDLING STRUCTURE;
REPLACE CAROLINE(2) BY "RM";
COMMENT OR ANY OTHER CONVENIENT INDICATION FOR THE FILENAME;
OPENINPUT;
COMMENT READ AND/OR SET HEADER COMMENT;
REPLACE CAROLINE(2) BY "CL";
COMMENT OR ANY OTHER CONVENIENT INDICATION FOR THE FILENAME;
OPENOUTPUT;
DO BEGIN GETINPUT;
COMMENT DEAL WITH DATA;
WRITEOUTPUT;
END UNTIL FALSE;
COMMENT THE DATA HANDLING LOOP IS EXITED AS SOON AS
- AN END-OF-FILE IS ENCOUNTERED ON THE INPUT FILE
- A CYCLE OF ALL ZERO'S IS DETECTED
- AN EOF1 END-OF-FILE LABEL IS FOUND.
IN EITHER CASE A JUMP TO THE ENDOFJASINFILE LABEL IS
EXECUTED;
ENDOFJASINFILE:
CLOSEINPUT;
COMMENT READ AND/OR SET HEADER COMMENT FOR END LABEL;
CLOSEOUTPUT;
END ELSE BEGIN REPLACE LISTLINE BY " " FOR 120;
REPLACE LISTLINE BY "ILLEGAL FILENAME: ",
CAROLINE FOR 16, " SKIPPED";
WRITE(LEFTSPACE 4,120,LISTLINE);
END;
END UNTIL FALSE;
COMMENT THE MAIN LOOP IS EXITED WHEN THE CARD READER IS EXHAUSTED
CAUSING A JUMP TO ENDOFCARD;
ENDOFCARD:
ABORTLABEL:
END OF PROGRAM.

```

```

00104200 005:000E:0
00105000 005:000F:3
00106000 005:000D:3
00107000 005:0004:4
00108000 005:0007:4
00109000 005:000A:4
00110000 005:000B:3
00111000 005:000E:3
00112000 005:000F:2
00113000 005:000E:1:4
00114000 005:000E:3:5
00115000 005:000E:4:5
00116000 005:000E:7:1
00117000 005:000E:9:4
00118000 005:000E:C:0
00119000 005:000F:1:0
00120000 005:000F:3:2
00121000 005:000F:5:4
00122000 005:000F:8:0
00123000 005:000F:E:3
00133000 005:0104:2
00134000 005:0108:5
00135000 005:0109:5
00136000 005:0108:1
00137000 005:010C:3
4 00138000 005:010C:4
4 00139000 005:010C:4
5 00139100 005:010D:2
5 00139200 005:010E:1
5 00139300 005:010F:5
5 00139400 005:0112:1
5 00139500 005:0113:4
5 00140000 005:0114:1
5 00141000 005:0115:3
5 00141100 005:0117:0
5 00142000 005:0117:3
5 00143000 005:011E:2
5 00144000 005:011F:4
5 00145000 005:0120:4
5 00145100 005:0122:0
5 00146000 005:0126:3
5 00147000 005:0129:4
4 00148000 005:0129:5
4 00149000 005:0128:5
4 00149100 005:0128:5
5 00151000 005:0129:3
5 00152000 005:0130:2
5 00153000 005:0131:4
5 00154000 005:0131:4
5 00155000 005:0134:3
5 00156000 005:0136:3
5 00157000 005:0138:3
5 00158000 005:0138:1
5 00159000 005:0138:5
5 00160000 005:0138:4
5 00161000 005:013F:0
5 00161100 005:0141:2
5 00161200 005:0143:4
5 00162000 005:0149:3
5 00163000 005:0149:3
5 00164000 005:0149:3
5 00165000 005:014A:5
5 00167000 005:014D:0
5 00168000 005:0153:2
5 00169000 005:0154:5
5 00170000 005:0157:3
5 00171000 005:015A:0
5 00172000 005:0160:0
5 00173000 005:0163:1
5 00174000 005:0168:0
5 00175000 005:016F:5
5 00176000 005:0176:2
5 00177000 005:0176:3
5 00177010 005:0176:3
5 00177020 005:0179:3
5 00178000 005:0179:3
5 00179000 005:017A:1
5 00179010 005:017A:1
5 00179020 005:017C:3
5 00180000 005:017C:3
5 00181000 005:017D:5
5 00182000 005:017D:5
5 00183000 005:017D:5
5 00184000 005:017E:3
5 00184100 005:017F:0
5 00184200 005:017F:0
5 00184300 005:017F:0
5 00184400 005:017F:0
5 00184500 005:017F:0
5 00184600 005:017F:0
5 00184700 005:017F:0
5 00185000 005:017F:0
5 00186000 005:017F:4
5 00187000 005:017F:4
5 00188000 005:0180:2
B.0001(005) IS 01A3 LONG
3 00189000 003:0015:3
3 00190000 003:0018:0
3 00191000 003:001D:5
3 00192000 003:0024:2
3 00193000 003:0024:2
2 00194000 003:0024:5
2 00195000 003:0024:5
2 00196000 003:0024:5
2 00197000 003:0024:5
2 00198000 003:0024:5
B.0001(003) IS 0035 LONG
DATA IS 0062 LONG

```

```

=====
***** THIS PROGRAM IS NOT EXECUTABLE BECAUSE OF COMPILING FOR SYNTAX ONLY. *****
=====
NUMBER OF ERRORS DETECTED = 0.
NUMBER OF SEGMENTS = 13. TOTAL SEGMENT SIZE = 616 WORDS. CORE ESTIMATE = 4095 WORDS. STACK ESTIMATE = 53
PROGRAM SIZE = 221 CARDS, 1221 SYNTACTIC ITEMS, 44 DISK SEGMENTS.
PROGRAM FILE NAME = GENJ/DISK.
COMPILATION TIME = 14.933 SECONDS ELAPSED; 2.896 SECONDS PROCESSING; 0.798 SECONDS I/O.
=====

```

GENIO / MAGTAPE
 = = = = =

BEGIN COMMENT ROUTINES AND SKELETON PROGRAM FOR CTD/XBT DATA USING

```

    THE 86700 WITH MAGTAPES;
    FILE LP(KIND=PRINTER);
    CARD(KIND=READER);

    EBCDIC ARRAY LISTLINE(0:119), CAROLINE(0:79);
    LABEL ABORTLABEL, ENDOFCARD;
    COMMENT GENERALIZED INPUT HANDLING ROUTINES FOR CTD/XBT DATA;
    FILE JASINFILE(KIND=PETAPE, MAXRECSIZE=240, BLOCKSIZE=1%0,
        UNITS=CHARACTERS, LABELTYPE=OMITTED EOF);

    EBCDIC ARRAY JASINCHARS, JASINHEADER(0:239);
    POINTER JASINPTR;
    INTEGER INCHARS, INRECS, INLGT, INLGT, INTITLE, INCYCLES;
    LABEL ENDOFJASINFILE, ENDOFJASINDATA;
    PROCEDURE OPENINPUT;
    BEGIN READ(JASINFILE, 240, JASINHEADER) (ENDOFJASINDATA);
        READ(JASINFILE, 240, JASINCHARS);
        JASINPTR:=JASINCHARS(0);
        IF JASINPTR NEQ 4"CB" THEN
            BEGIN REPLACE LISTLINE BY " " FOR 120;
                REPLACE LISTLINE BY "NO HEADER CYCLE FOUND FOR ",
                    JASINFILE.TITLE, " FATAL ERROR.";
                    WRITE(LP, 120, LISTLINE);
                    GO TO ABORTLABEL;
                END;
            INLGT:=REAL(JASINPTR+1);
            INTYPE:=REAL(JASINPTR+7);
            INTITLE:=REAL(JASINPTR+2);
            REPLACE LISTLINE BY " " FOR 120;
            REPLACE LISTLINE BY "OPEN FILE ", JASINFILE.TITLE,
                " CYCLE LENGTH: ", INLGT FOR 3 NUMERIC,
                " DATA TYPE: ", INTYPE FOR 2 NUMERIC;
            WRITE(LP, SPACE 3, 120, LISTLINE);
            INRECS:=1;
            INCYCLES:=0;
            INCHARS:=240 - INLGT;
        END;
    PROCEDURE GETINPUT;
    BEGIN IF INCHARS = 0 THEN
        BEGIN READ(JASINFILE, 240, JASINCHARS) (ENDOFJASINFILE);
            INCHARS:=240;
            JASINPTR:=JASINCHARS(0);
            IF JASINPTR = 7"9 EOF" THEN GO TO ENDOFJASINFILE;
            INRECS:=INRECS + 1;
        END;
        JASINPTR:=JASINPTR + INLGT;
        IF JASINPTR = 4"00" FOR INLGT THEN GO TO ENDOFJASINFILE;
        INCHARS:=INCHARS - INLGT;
        INCYCLES:=INCYCLES + 1;
    END;
    PROCEDURE CLOSEINPUT;
    BEGIN REPLACE LISTLINE BY " " FOR 120;
        CLOSE(JASINFILE);
        IF JASINCHARS(0) NEQ 7"9 EOF" THEN
            BEGIN REPLACE LISTLINE BY "MISSING EOF LABEL";
                WRITE(LP, 120, LISTLINE);
                REPLACE LISTLINE BY " " FOR 120;
            END ELSE
            IF INRECS NEQ REAL(JASINCHARS(226), 2) THEN
                BEGIN REPLACE LISTLINE BY "RECORD COUNT IN EOF LABEL: ",
                    REAL(JASINCHARS(226), 2) FOR 6 NUMERIC;
                    WRITE(LP, 120, LISTLINE);
                    REPLACE LISTLINE BY " " FOR 120;
                END;
            REPLACE LISTLINE BY "CLOSED FILE ", JASINFILE.TITLE,
                " AFTER ", INRECS FOR 6 NUMERIC, " RECORDS AND ",
                INCYCLES FOR 7 NUMERIC, " CYCLES.";
            WRITE(LP, SPACE 3, 120, LISTLINE);
        END;
    COMMENT GENERALIZED OUTPUT ROUTINES FOR CTD/XBT DATA;
    FILE JASOUTFILE(KIND=PETAPE, MAXRECSIZE=240, BLOCKSIZE=1920,
        UNITS=CHARACTERS, LABELTYPE=OMITTED EOF, SAVEFACTOR=1);

    EBCDIC ARRAY JASOUTCHARS, JASOUTHEADER(0:239), HEADERCOMMENT(0:135);
    ARRAY JASOUTWORDS(0)=JASOUTCHARS, JASOUTLABEL(0)=JASOUTHEADER;
    ARRAY JATE(0);
    INTEGER ARRAY OUTPUTDATA(0:9);
    EBCDIC ARRAY OUTDATAPTR(0)=OUTPUTDATA;
    COMMENT WE NEED A POINTER TO THE OUTPUTDATA ARRAY, SO THIS IS
        A CONVENIENT WAY TO GET ONE;
    POINTER JASOUTPTR;
    BOOLEAN WRITE TAPE HEADER;
    INTEGER OUTCHARS, OUTLGT, OUTTYPE, OUTRECS, OUTCYCLES, OUTTITLE, IOUT;
    INTEGER YEAR, DAY, REELNR;
    PROCEDURE OPENTAPE;
    BEGIN REPLACE JASOUTHEADER BY 7"1 VOL" FOR 6, 4"00" FOR 24,
        REELNR FOR 6 DIGITS,
        CAROLINE FOR 6 WITH EBCDICTOASCII,
        7" / " FOR 2, 7" " FOR 8, 7" ." FOR 1,
        7" " FOR 27, 7"5" FOR 1,
        7" " FOR 79, 7"5" FOR 1,
        7" " FOR 79;
        WRITE(JASOUTFILE, 40, JASOUTLABEL);
        WRIETAPEHEADER:=FALSE;
    END;
    PROCEDURE WPEOUTPUT;
    BEGIN COMMENT CHECK DATA TYPE;
        IF OUTTYPE < 0 OR OUTTYPE > 3 THEN
            BEGIN REPLACE LISTLINE BY " " FOR 120;
                REPLACE LISTLINE BY "ILLEGAL OUTPUT DATA TYPE (",
                    OUTTYPE FOR 2 NUMERIC, ") OPENING FILE ",
                    JASOUTFILE.TITLE, " FATAL ERROR.";
                    WRITE(LP, SPACE 3, 120, LISTLINE);
                    GO TO ABORTLABEL;
            END;

```

```

0001000 000:0000:0
E. COCO IS SEGMENT 00003
0002000 003:0000:1
0003000 003:0000:1
    DATA IS 0004 LONG
0004000 003:0000:1
    DATA IS 0005 LONG
0005000 003:0000:1
0006000 003:0000:1
0007000 003:0000:1
0008000 003:0000:1
0009000 003:0000:1
0010000 003:0000:1
    DATA IS 0008 LONG
0011000 003:0000:1
0012000 003:0003:4
0013000 003:0003:4
0014000 003:0003:4
0015000 003:0003:4
0016000 003:0003:4
0017000 003:0000:5
0018000 003:0013:2
0019000 003:0014:3
0020000 003:0016:0
0021000 003:0019:3
0022000 003:0018:5
0023000 003:0015:5
0024000 003:0026:2
0025000 003:0027:3
0026000 003:0027:3
0027000 003:0029:3
0028000 003:0028:4
0029000 003:0028:0
0030000 003:0030:3
0031000 003:0038:2
0032000 003:0030:2
0033000 003:0042:3
0034000 003:0049:2
0035000 003:0044:0
0036000 003:0044:4
0037000 003:0040:1
0038000 003:0040:2
0039000 003:0040:2
0040000 003:0040:0
0041000 003:0056:5
0042000 003:0057:4
0043000 003:0058:5
0044000 003:0050:2
0045000 003:0050:4
0046000 003:0050:4
0047000 003:0057:2
0048000 003:0061:5
0049000 003:0063:2
0050000 003:0064:4
0051000 003:0064:5
0052000 003:0064:5
0053000 003:0067:3
0054000 003:0069:0
0055000 003:0068:4
0056000 003:0068:3
0057000 003:0074:2
0058000 003:0077:3
0059000 003:0077:3
0060000 003:0079:5
0061000 003:0070:5
0062000 003:0081:1
0063000 003:0087:2
0064000 003:0084:3
0065000 003:0084:3
0066000 003:0092:2
0067000 003:0090:0
0068000 003:0092:5
0069000 003:0092:5
0070000 003:0092:2
0071000 003:0092:3
0072000 003:0092:3
    DATA IS 0019 LONG
    DATA IS 0008 LONG
0073000 003:0092:3
0074000 003:0092:4
0075000 003:0092:2
0076000 003:0092:2
0077000 003:0092:0
0078000 003:0092:0
0079000 003:0092:0
0080000 003:0092:0
0081000 003:0092:0
0082000 003:0092:0
0083000 003:0092:0
0084000 003:0092:0
0085000 003:0092:0
0086000 003:0092:0
0087000 003:0092:0
0088000 003:0092:0
0089000 003:0092:0
0090000 003:0092:0
0091000 003:0092:0
0092000 003:0092:0
0093000 003:0092:0
0094000 003:0092:0
0095000 003:0092:0
0096000 003:0092:0
0097000 003:0092:0
0098000 003:0092:0
0099000 003:0092:0

```

```

END;
DATE(C):=TIME(10);
YEAR:=INTEGER(PINTER(DATE),3);
DAY:=INTEGER(PINTER(DATE)+3,7);
OUTPUTDATA(0):=YEAR;
OUTPUTDATA(1):=DAY;
OUTPUTDATA(2):=240;
OUTPUTDATA(3):=OUTTYPE;
OUTPUTDATA(4):=OUTLGT;
REPLACE JASOUTHEADER BY 7"5 HORL";
  OUTDATAPTR(4) FOR 2, OUTDATAPTR(10) FOR 2,
  OUTDATAPTR(16) FOR 2, OUTDATAPTR(16) FOR 2,
  4"00" FOR 4,
  OUTDATAPTR(22) FOR 2, OUTDATAPTR(28) FOR 2,
  4"00" FOR 8,
  7" " FOR 6,
  CARLINE(0) FOR 6 WITH EBCDICTOASCII,
  7" " FOR 9,
  CARLINE(6) FOR 10 WITH EBCDICTOASCII,
  7" " FOR 9,
  CARLINE(20) FOR 10 WITH EBCDICTOASCII,
  7" " FOR 5,
  CARLINE(54) FOR 10 WITH EBCDICTOASCII,
  CARLINE(30) FOR 24 WITH EBCDICTOASCII,
  CARLINE(64) FOR 6 WITH EBCDICTOASCII,
  7" " FOR 38, 7"5", 7" " FOR 79;
COMMENT THE HEADERCOMMENT IS EITHER PROGRAM-GENERATED OR
READ FROM INPUTCARDS OR COPIED FROM INPUTHEADER OR
A (PROGRAM-DEPENDANT) COMBINATION;
WRITE(JASOUTFILE,240,JASOUTHEADER);
OUTCHARS:=OUTLGT;
OUTRECS:=OUTCYCLES:=0;
FILL JASOUTWORDS(+) WITH 7"7 DATA", 39(0);
JASOUTPTR:=JASOUTCHARS(OUTLGT);
END;
PROCEDURE WRITEOUTPUT;
BEGIN IOUT:=0;
  WHILE IOUT < OUTLGT DO
  BEGIN REPLACE JASOUTPTR + IOUT BY
    OUTDATAPTR(4 + 3*IOUT) FOR 2;
    IOUT:=IOUT + 2;
  END;
  OUTCYCLES:=OUTCYCLES + 1;
  OUTCHARS:=OUTCHARS + OUTLGT;
  IF OUTCHARS = 240 THEN
  BEGIN WRITE(JASOUTFILE,240,JASOUTCHARS);
    OUTRECS:=OUTRECS + 1;
    OUTCHARS:=OUTLGT;
    JASOUTPTR:=JASOUTCHARS(OUTLGT);
    FILL JASOUTWORDS(+) WITH 7"7 DATA", 39(0);
  END ELSE JASOUTPTR:=JASOUTPTR + OUTLGT;
END;
PROCEDURE CLOSEOUTPUT;
BEGIN IF OUTCHARS NEQ 0 THEN
  BEGIN WRITE(JASOUTFILE,240,JASOUTCHARS);
    OUTRECS:=OUTRECS + 1;
  END;
  REPLACE JASOUTHEADER BY 7"9 EOFL";
  REPLACE JASOUTHEADER(80) BY 7"9";
  REPLACE JASOUTHEADER(160) BY 7"9";
  REPLACE JASOUTHEADER(40) BY CARLINE(30) FOR 34
  WITH EBCDICTOASCII;
  REPLACE JASOUTHEADER(82) BY CARLINE(54) FOR 10
  WITH EBCDICTOASCII,
  CARLINE(30) FOR 24
  WITH EBCDICTOASCII;
  COMMENT FILL OUTDATA WITH 16 BIT BINARIES OF LABEL RECORDS;
  OUTPUTDATA(0):=OUTRECS;
  OUTPUTDATA(1):=OUTRECS;
  OUTPUTDATA(11):=OUTRECS DIV 8;
  REPLACE JASOUTHEADER(226) BY OUTDATAPTR(4) FOR 2;
  REPLACE JASOUTHEADER(230) BY OUTDATAPTR(10) FOR 2;
  WRITE(JASOUTFILE,240,JASOUTHEADER);
  REPLACE LISTLINE BY " " FOR 120;
  REPLACE LISTLINE 4Y "CLOSED OUTPUT FILE ",
  JASOUTFILE.TITLE, " AFTER ",
  OUTCYCLES FOR 7 NUMERIC,
  " CYCLES IN ",
  OUTRECS FOR 6 NUMERIC, " RECORDS.";
  WRITE(LP(SPACE 3),120,LISTLINE);
  COMMENT FORCE SYSTEM TO WRITE A FULL 1920 CHARACTERS BLOCK;
  REPLACE JASOUTCHARS BY 4"00" FOR 240;
  WHILE OUTRECS + 1 MOD 8 NEQ 0 DO
  BEGIN WRITE(JASOUTFILE,240,JASOUTCHARS);
    OUTRECS:=OUTRECS + 1;
  END;
  CLOSE(JASOUTFILE,);
END;
PROCEDURE CLOSETAPE;
BEGIN REPLACE JASOUTHEADER BY 7"8 EOFL" FOR 6, 4"00" FOR 24,
  REELNR FOR 6 DIGITS,
  CARLINE FOR 6 WITH EBCDICTOASCII,
  7" /" FOR 2, 7" " FOR 8, 7" " FOR 1,
  7" " FOR 27, 7"8" FOR 1,
  7" " FOR 79, 7"8" FOR 1,
  7" " FOR 79;
  WRITE(JASOUTFILE,40,JASOUTLABEL);
END;
COMMENT START OF MAIN PROGRAM.
CREATE AND WRITE TAPE HEADER FILE;
WRITETAPEHEADER:=TRUE;
READ(CARD,<16>);REELNR;
OO BEGIN COMMENT THE MAIN PROGRAM IS ESSENTIALLY A BIG LOOP;
  READ(CARD,<80>);CARLINE(ENDDFCARD);
  COMMENT THE INPUT IS ASSUMED TO START WITH A FILENAME CARD
  CONTAINING IN COLUMNS,
  1 - 16 FILENAME
  21- 30 STATION ACRONYM AT BEGIN OF STATION
  31 - 40 LATITUDE AT BEGIN OF STATION
  41 - 50 LONGITUDE AT BEGIN OF STATION
  51- 54 WATER DEPTH AT BEGIN OF STATION
  55- 64 DATE/TIME AS YYMMDDHHMM AT BEGIN OF STATION
  72- 80 CARD SEQUENCE NUMBER
  00084700 003:00EA:3
  00095000 003:00EA:3
  00096000 003:00EC:2
  00097000 003:00EE:4
  00098000 003:00F1:3
  00099000 003:00F2:5
  00100000 003:00F4:1
  00101000 003:00F5:3
  00102000 003:00F8:3
  00103000 003:00FE:4
  00103100 003:00FE:4
  00103200 003:0101:4
  00103300 003:0102:3
  00103400 003:0105:3
  00103500 003:0106:2
  00103600 003:0108:4
  00103700 003:010A:5
  00103800 003:010B:5
  00103900 003:010E:1
  00104000 003:0110:4
  00104100 003:0113:0
  00104200 003:011A:0
  00104300 003:011A:2
  00104400 003:011C:4
  00104500 003:011F:0
  00105000 003:0125:3
  00106000 003:0125:3
  00107000 003:0125:3
  00108000 003:0125:3
  00109000 003:012B:2
  00110000 003:012C:2
  00111000 003:012D:4
  00112000 003:0132:1
  00113000 003:0133:3
  00114000 003:0133:4
  00115000 003:0133:4
  00116000 003:0134:2
  00117000 003:0135:1
  00118000 003:0136:5
  00119000 003:0139:1
  00120000 003:013A:4
  00121000 003:013B:1
  00122000 003:013C:3
  00123000 003:013E:0
  00124000 003:013E:3
  00125000 003:0145:2
  00126000 003:0146:4
  00127000 003:0147:4
  00128000 003:0149:0
  00129000 003:014D:3
  00130000 003:014F:4
  00131000 003:014F:5
  00132000 003:0157:2
  00133000 003:0158:4
  00134000 003:0158:3
  00135000 003:015D:3
  00136000 003:015F:3
  00137000 003:0161:1
  00138000 003:0162:5
  00139000 003:0164:3
  00140000 003:0166:0
  00141000 003:0166:3
  00142000 003:0168:3
  00143000 003:0168:5
  00144000 003:0168:1
  00145000 003:016D:0
  00146000 003:016F:1
  00147000 003:0171:2
  00148000 003:0177:2
  00149000 003:017A:3
  00150000 003:017D:0
  00151000 003:0183:0
  00152000 003:0186:1
  00153000 003:0188:0
  00154000 003:0192:5
  00155000 003:0199:2
  00156000 003:0199:2
  00157000 003:019A:4
  00158000 003:019A:4
  00159000 003:019B:4
  00160000 003:01A2:2
  00161000 003:01A3:4
  00162000 003:01A4:1
  00163000 003:01A5:4
  00164000 003:01A5:5
  00165000 003:01A9:3
  00166000 003:01A8:0
  00167000 003:01A0:1
  00168000 003:01B3:3
  00169000 003:01B7:3
  00170000 003:01B8:3
  00171000 003:01B0:3
  00172000 003:01C3:2
  00173000 003:01C3:3
  00174000 003:01C3:3
  00175000 003:01C4:1
  00176000 003:01C4:1
  00177000 003:01D4:1
  00178000 003:01D4:1
  00179000 003:01D4:1
  00180000 003:01D4:1
  00181000 003:01D4:1
  00182000 003:01D4:1
  00183000 003:01D4:1
  00184000 003:01D4:1
  00185000 003:01D4:1
  00186000 003:01D4:1
  00187000 003:01D4:1
  00188000 003:01D4:1
  00189000 003:01D4:1
  00190000 003:01D4:1
  00191000 003:01D4:1
  00192000 003:01D4:1
  00193000 003:01D4:1
  00194000 003:01D4:1
  00195000 003:01D4:1
  00196000 003:01D4:1
  00197000 003:01D4:1
  00198000 003:01D4:1
  00199000 003:01D4:1
  00200000 003:01D4:1

```



```

IF CAROLINE = "YYRO" OR                                002C1000 003:01D4:1
CAROLINE = "OERO" OR                                  002C2000 003:01D6:4
CAROLINE = "XBRO" THEN                                002C3000 003:01D9:5
BEGIN REPLACE CAROLINE(?) BY "RW";
COMMENT OR ANY OTHER CONVENIENT INDICATION FOR THE FILENAME; 3 002C4000 003:01DC:5
OPENINPUT;
COMMENT READ AND/OR SET HEADER COMMENT;                002C5000 003:01E0:3
REPLACE CAROLINE(?) BY "RW";                          002C6000 003:01E1:1
COMMENT OR ANY OTHER CONVENIENT INDICATION FOR THE FILENAME; 002C7000 003:01E1:1
OPENOUTPUT;
DO BEGIN GETINPUT;
COMMENT DEAL WITH DATA;                                4 002C8000 003:01E3:3
WRITEOUTPUT;
END UNTIL FALSE;
COMMENT THE DATA HANDLING LOOP IS EXITED WHEN:        4 002C9000 003:01E3:3
- AN END-OF-FILE CONDITION OCCURS READING THE INPUT FILE
- AN EOF LABEL IS DETECTED
IN EITHER CASE A JUMP TO THE ENDOFJASINFILE LABEL IS
EXECUTED;
ENDOFJASINFILE:
CLOSEINPUT;
COMMENT READ AND/OR SET HEADER COMMENT FOR END LABEL; 002D0000 003:01E4:1
CLOSEOUTPUT;
END ELSE BEGIN REPLACE LISTLINE BY " " FOR 120;
REPLACE LISTLINE BY "ILLEGAL FILENAME: ",
CAROLINE FOR 16, " SKIPPED";
WRITE(LP(SPACE 4),120,LISTLINE);
END;
END UNTIL FALSE;
COMMENT THE MAIN LOOP IS EXITED WHEN THE CARD READER IS EXHAUSTED 3 002D1000 003:01E4:5
CAUSING A JUMP TO ENDOFCARD;
COMMENT WHEN A TAPE MARK IS REAC OPENING A NEW INPUT FILE 2 002D2000 003:01E5:3
A JUMP TO ENDOFJASINDATA IS EXECUTED;
ENDOFJASINDATA:
REPLACE LISTLINE BY " " FOR 120;
REPLACE LISTLINE BY "UNEXPECTED END OF INPUT, OPENING ",
CAROLINE(?) FOR 16;
WRITE(LP,120,LISTLINE);
ENDOFCARD:
CLOSETAPE;
ABORTLABEL;
END OF PROGRAM.

```

```

H.000C003) IS C208 LONG
STACKCODE IS SEGMENT 0000F
STACKCODE00F) IS 0036 LONG
DATA IS 0068 LONG

```

***** THIS PROGRAM IS NOT EXECUTABLE BECAUSE OF COMPILING FOR SYNTAX ONLY. *****

```

NUMBER OF ERRORS DETECTED = 0.
NUMBER OF SEGMENTS = 13. TOTAL SEGMENT SIZE = 731 WORDS. CORE ESTIMATE = 4210 WORDS. STACK ESTIMATE = 52
PROGRAM SIZE = 258 CARDS, 1456 SYNTACTIC ITEMS, 48 DISK SEGMENTS.
PROGRAM FILE NAME: GENIO/MAGTA.P.
COMPILATION TIME = 12.720 SECONDS ELAPSED; 3.344 SECONDS PROCESSING; 0.973 SECONDS I/O.

```