
Y D R O L I N
versie 1.00
INTERACTIEF HYDROLOGISCH COMPUTERPROGRAMMA
TU DELFT/Ct (C) 880226 M.J. Vos
TECHNISCHE UNIVERSITEIT DELFT
FACULTEIT DER CIVIELE TECHNIEK

TU Delft

```
H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H
H                                                             H
H                                                             H
H             //  //                                          H
H            //  //                                           H
H           //--//                                            H
H          //--//  Y D R O L I N      versie  1.00            H
H         //  //                                              H
H                                                             H
H                                                             H
H         INTERACTIEF HYDROLOGISCH COMPUTERPROGRAMMA          H
H                                                             H
H              TU DELFT/Ct (C) 880226  M.J. Vos               H
H                                                             H
H                                                             H
H                                                             H
H              TECHNISCHE   UNIVERSITEIT  DELFT               H
H               FACULTEIT DER CIVIELE TECHNIEK                H
H            VAKGROEP WATERBOUWKUNDE EN HYDROLOGIE            H
H                                                             H
H                                                             H
H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H
```

INHOUD.

VOORWOORD.

In het kader van het afstuderen aan de Technische Universiteit te Delft, Faculteit der Civiele Techniek - Vakgroep Waterbouwkunde, Sectie Constructieve Waterbouwkunde - is naast

een hoofdonderzoek bij de Sectie der Constructieve Waterbouwkunde in samenwerking met Rijkswaterstaat t.w. een Energieanalyse van een Gepenetreerde Stortsteenglooiing,

en een deelopdracht bij de Vakgroep Hydrologie t.w. het ontwikkelen van interactief hydrologisch computerprogramma genaamd "HYDROLIN", dat als platform moet dienen voor diverse neerslag/afvoer-modellen, met als eerste de theorie van de Unit Hydrograph(Eenheidsafvoergolf)

als deelontwerp bij de Sectie der Constructieve Waterbouwkunde een ontwerp gemaakt van een Zeekade voor de noordzijde van de Brittaniehaven te Rotterdam.

Op deze plaats wil ik voor de begeleiding bij de ontwikkeling van het in dit rapport opgenomen computerprogramma HYDROLIN 1.00 mijn dank betuigen aan:

de hoogleraar Prof.Dr.Ir. J.C. van Dam

en de heer Ir. H. Vermeulen

SAMENVATTING.

HYDROLIN gaat de mogelijkheden bieden om meetseries van neerslag en afvoer te verwerken in diverse modellen. Na de invoer van de meetseries via het toetsenbord kunnen hieraan diverse bewerkingen worden gedaan, zoals het aanbrengen van verliezen en scheidingen, teneinde met een keuze uit een scala van modellen een neerslag -afvoerrelatie af te leiden. Als eerste model is de theorie van de UNIT HYDROGRAPH - EENHEIDSAFVOERGOLF - ingebracht.
Omdat de theorie van de UNIT HYDROGRAPH voor netto neerslag en de oppervlaktecomponent van de afvoer geldt, bevat HYDROLIN de mogelijkheid uit verschillende opties te kiezen om op bruto regen-gegevens verliezen in rekening te brengen en op de bruto afvoer een afvoerscheiding aan te brengen. Na invoer van de meetseries kunnen diverse manipulaties worden uitgevoerd met deze dataseries. Op vele wijzen is het mogelijk allerlei handige programma's aan HYDROLIN te koppelen om bewerkingen te doen. Alle daarmee samen-hangende databestanden worden op een extern geheugen vastgelegd, om tenslotte een selectie te kunnen doen ter bepaling van netto regen c.q. oppervlakteafvoer. Het "filemanagement" wordt door de computer zelf en interactief afgehandeld als van HYDROLIN-input-procedures gebruikt wordt gemaakt. Alle in HYDROLIN aangeboden opties worden in de vorm van menu's aan de gebruiker kenbaar ge-maakt - het gehele computerprogramma is menu-gericht ontwikkeld. Ter verkrijging van de UNIT HYDROGRAPH is een subprogramma voor de verwerking van matrices aanwezig. Weergave van de inhoud van bestanden op het beeldscherm en naar een "extern device", zoals een printer, kan geschieden in de vorm van data en grafisch.

1. INLEIDING.

De opdracht bestaat uit

- het maken van een gebruikersvriendelijk d.i. een interactief hydrologisch computerprogramma in TURBO PASCAL versie 3.01A voor een IBM compatible PC t.w. een Olivetti M24, dat
- als platform voor diverse neerslag-afvoer modellen moet dienen en
- als eerste implementatie de theorie van de UNIT HYDROGRAPH moet bevatten.

Met name de eerste twee eisen brengen met zich mee, dat de meest uiteenlopende "gereedschappen"(utilities) toegevoegd zijn.

Als computertaal is TURBO PASCAL van Borland gekozen op grond van

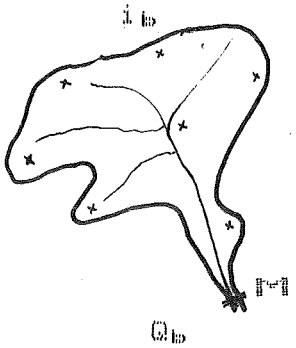
- zeer positieve ervaringen en een deels reeds beschikbare bibliotheek van utilities en
- goede vooruitzichten op nieuwe versies van TURBO PASCAL en daardoor van HYDROLIN en
- omdat PASCAL op de TU DELFT een algemeen bekende structurele taal is en
- omdat TURBO PASCAL sterk is door zijn ingebouwde editer en zeer goede I/O.

Het programma is voor gebruikersvriendelijkheid menugericht en ten behoeve van uitbreidingen in de vorm van procedures ontworpen. Aan vele wensen wordt tegemoet gekomen door het verschaffen van keuzemogelijkheden uit diverse opties in menu's en door vele beveiligingen aan te brengen.

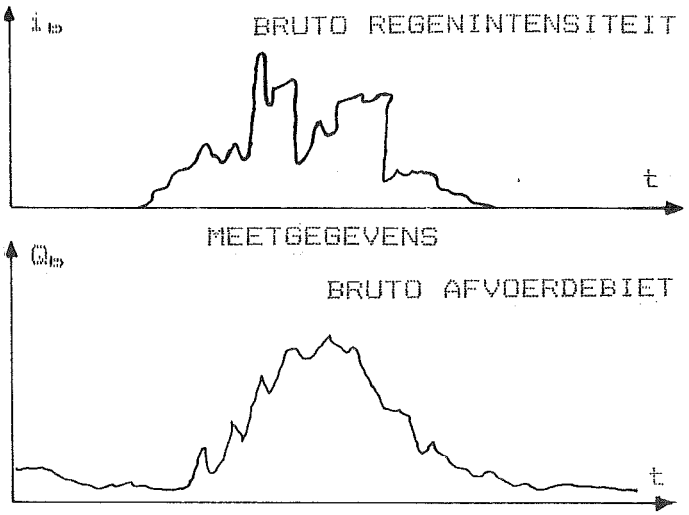
2. NEERSLAG EN AFVOER :

2.1. ALGEMEEN;

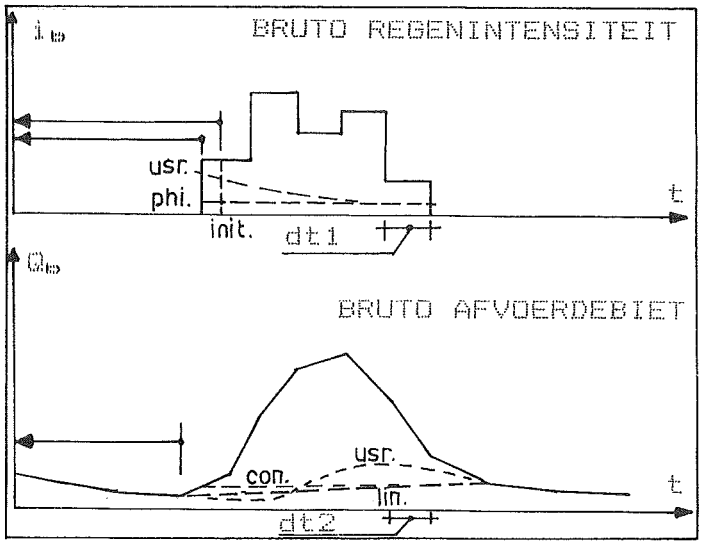
Voordat een neerslag/afvoermodel wordt gekozen zullen de gebiedsgegevens door de gebruiker op een rij worden gezet. Meetresultaten van afvoerdebieten en neerslagintensiteiten zullen doorgaans een grillig beeld vertonen.



Klimaat
Grootte
Vorm
Reliëf
Grondsoort
Gewas



Teneinde representatieve data in te voeren in een computerprogramma, zal het nodig zijn een discrete tijdstap te introduceren met daarbij behorende gemiddelde neerslag- resp. afvoerwaarden.
Niet strikt noodzakelijk maar wel bevorderlijk voor verdere verwerking kan de tijdas in gelijke tijdstappen worden ingedeeld.



zie HYDROLØ1.INC
HYDROLØ2.INC
HYDROLØ3.INC

De aldus verkregen basisdata worden ingevoerd en in principe als d:XXXXXXXØ-file bewaard.
Wijzigingen op deze datasets is mogelijk met behulp van bijvoorbeeld HYDREDIT; de naam en het volgnummer worden dan eveneens met behulp van het computerprogramma verzorgd. Voor bepaalde modellen zal het nodig zijn op de ingevoerde bruto data verliezen /

vervolg 2.1. Algemeen

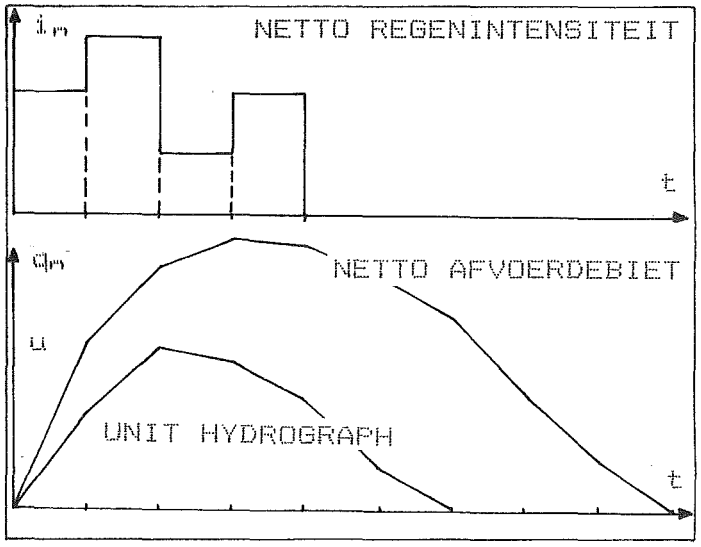
scheidingen aan te brengen, als netto gegevens worden gewenst.
Zo'n model is dat van de theorie van UNIT HYDROGRAPH.
Deze theorie is nml. van toepassing op de oppervlakteafvoer ten gevolge van netto neerslag.
Verliezen op de bruto neerslag worden veroorzaakt door b.v. verdamping, interceptie, infiltratie/percolatie, berging in terreindepressies, e.d. Grondwateraanvulling kan een vertraagde afvoer tot gevolg hebben; deze afvoer wordt samen met de basisafvoer in de waterloop, het aflaten van een reservoir, wateronttrekking voor irrigatie en drinkwatervoorziening e.d. op de bruto afvoerwaarden in rekening gebracht om de oppervlakte-afvoer te verkrijgen.
Neerslagverliezen kunnen worden ingevoerd in de vorm van:

aan het begin geconcentreerd gedachte verliezen de zgn. initiele verliezen;

gelijkmatig over de (samengestelde) bui verdeelde verliezen de zgn. phi-index;

een door de gebruiker zelf te definieren verdeling van de verliezen.

Alle nuldata vanaf het begin van de meetseries moeten na elke bewerking worden verwijderd.



zie HYDROL02.INC
HYDROL03.INC
HYDROL04.INC
HYDROCAL.PAS
HYDROGR .PAS

Voor de uiteindelijke afleiding van de UNIT HYDROGRAPH zijn nu netto data-sets te maken.
Met gelijke tijdstap voor netto neerslag- en afvoerseries en de grootte van het stroomgebied kan een UNIT HYDROGRAPH worden berekend voor een regenbui met een duur van deze tijdstap.
Deze UNIT HYDROGRAPH is representatief voor het afvoer-gedrag van het beschouwde stroomgebied.
Door het superpositiebeginsel bij de methode van de UNIT HYDROGRAPH is het verloop van het afvoerdebiet te bepalen na een gegeven bui met een geven samenstelling en duur.

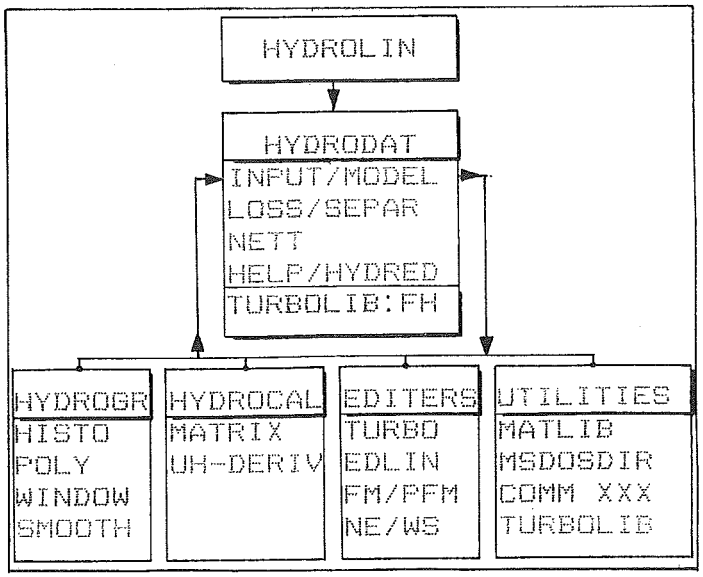
2.2. UNIT HYDROGRAPH. (EENHEIDSAFVOERGOLF)

De METHODE VAN DE EENHEIDSAFVOERGOLF berust op empirisch gevonden relaties - L.K. Sherman 1932 - en dient ter bepaling van oppervlakteafvoercomponent (van de afvoer), die ontstaat ten gevolge van een gemiddelde netto neerslag; deze neerslag wordt gelijkmatig verdeeld gedacht over het stroomgebied. Bij de METHODE VAN DE EENHEIDSAFVOERGOLF wordt uitgegaan van een EENHEIDSDIEPTE - UNIT DEPTH - b.v. 1 inch of 1 centimeter of een EENHEIDSVOLUME A, gelijkmatig verdeeld over het stroomgebied.
De UNIT HYDROGRAPH is een voor een gebied karakteristieke afvoerdebiet/tijd-relatie voor een gelijkmatig over het stroomgebied gedachte regen van een eenheidsdiepte/-volume.
De grootte(A) van het stroomgebied heeft een bovengrens van ca. 10.000 vierkante kilometer. De overige kenmerken van het stroomgebied worden door de gebruiker geinterpreteerd in responsie van het gebied op neerslag en afvoer.
De tijdsduur van een enkelvoudige bui wordt DT genoemd.
Vanaf het einde van deze bui tot het einde van de eenheidsafvoerkromme is de tijdsduur TR - de AFLOOPTIJD.
Het afvoerdebiet wordt weergegeven als Q of als q=Q/A.
Een over een bepaalde tijd gemeten neerslagkromme(samengestelde bui) kan worden opgedeeld in een aantal enkelvoudige buien van zo mogelijk dezelfde tijdsduur DT.
Met de UNIT HYDROGRAPH kunnen door superpositie afvoer/tijd-verbanden worden afgeleid voor allerlei voorkomende neerslagsituaties voor het beschouwde gebied.
Deze afleidingen zijn niet in het programma meegenomen.
De weg door het programma ter bepaling van de UNIT HYDROGRAPH wordt in het volgende hoofdstuk uiteen gezet.

zie MENU-STROOMDIAGRAM

2.3. HYDROLIN 1.00;

HYDROLIN is ontwikkeld in procedurevorm en menugericht. Hiertoe is een uitgebreide Library in het programma opgenomen als een zgn. included file. Vanwege de beperkte geheugenruimte voor source en code voor TURBO PASCAL 3.01A zijn een aantal activiteiten slechts realiseerbaar geworden door middel van linkbestanden zgn. chainfiles. Wel is een tweetal programma's ontwikkeld waarmee, bij voldoende overgebleven geheugenruimte naast HYDROLIN, willekeurige EXE en COM-bestanden kunnen worden opgestart en alle DOS-bevelen kunnen worden uitgevoerd.
Bij het opstarten van HYDROLIN ontstaat op het beeldscherm het logo en direct daarna het hoofdmenu. Het hoofdmenu is het centrale menu vanwaar naar volgende menu's wordt gesprongen. De menu's liggen als het ware achter elkaar; uit een menu stappen(="Quit") betekent terugkeer in het hoofdmenu. Na het volvoeren van een optie in een menu wordt in het menu gebleven, tenzij de aangeroepen procedure buiten het menuprocedureblok is geschreven. Wordt een bevel gegeven naar een chainfile dan wordt na beeindiging van de actie teruggekeerd naar het hoofdmenu van HYDROLIN. In menu's komen dikwijls utility-opties voor die voor dat stadium van het programma van belang kunnen zijn. Het data ver- en bewerkend gebeuren vindt plaats in HYDROLIN. De berekening van de UNIT HYDROGRAPH vindt plaats in HYDROCAL. Het grafische gebeuren wordt verzorgd door HYDROGR.
De op te starten EDITERS worden geactiveerd met een aan te roepen commandprocedure. Een overzicht van de modulus wordt hierna afgebeeld. Een lijn vanaf het hoofdmenu is in de Helpsheet getekend. Omdat alle data-activiteiten eindigen met het wegschrijven op schijf, elk menu via de quit-optie terugkeer naar het hoofdmenu verzorgd, strategische punten voorzien zijn van een extra beveiligin en er ruim is voorzien in editing/filehandling faciliteiten is herstel altijd mogelijk.



zie MENU-STROOMDIAGRAM
HYDROLIN.PAS

vervolg 2.3. HYDROLIN 1.00

De bruto meetgegevens worden via het toetsenbord ingebracht met de Keyboardoptie. Wijzigingen kunnen worden gedaan met de ingebouwde editer HYDREDIT, door een keuze uit editers of filemanagers. Allerlei bestandsmanipulaties worden uitgevoerd met opties uit Filehandling of door op DOS-niveau te opereren met de optie COMM. De optie Precip_Disch geeft aan, dat een menu volgt waarin of met regen- of afvoerdata verwerking wordt verder gegaan. Na invoer van bruto gegevens worden de verliezen voor de regen en de scheiding van oppervlakte en niet-directe afvoer in een dataset op schijf gezet.
De verlies opties voor regenintensiteit zijn :

phi-index t.w. een constante verliesintensiteit over de samengestelde regenbui,

initieel t.w. een aan het begin van de samengestelde regenbui verondersteld verlies en

user defined t.w. een door de gebruiker zelf samen te stellen verloop van verliesintensiteiten. Om de verliesopties uit te voeren wordt
steeds gevraagd de "aanvangsnullen" te vegen ("wiped")
De scheidingopties voor de afvoer zijn

een constant afvoerdebiet als niet-directe afvoer,

een lineair verloop van de afvoerscheiding en

een door de gebruiker zelf samen te stellen verloop van de afvoerscheiding. Hiervoor worden het einde van een mogelijk aan de hoogwatergolf voorafgaand uitputtingsverloop en het begin van het mogelijke uitputtingsverloop na de hoogwatergolf berekend en op de monitor vertoond.

De netto regen resp. afvoer kan nu worden verkregen. Voor een vergelijking van de totale hoeveelheid neerslag met die van de afvoer (optie Compare) worden de datasets allereerst onderworpen aan conversieprocedures naar mm/uur resp. m^3/uur. De oppervlakteafmeting van het gebied moet hierbij bekend zijn.
De gebruiker kan zelf zijn eerder gedane keuzen m.b.t. regenverliezen resp. afvoerscheiding beoordelen op aanvaardbaarheid. Voor de afleiding van de UNIT HYDROGRAPH wordt door het programma gevraagd naar de tijdstap voor regen- en afvoergegevens. Deze zijn gelijk te maken met de optie Timestep. De dan ontstane datasets kunnen worden aangeboden aan de optie Unit_hydrograph teneinde een UNIT HYDROGRAPH uit te rekenen.
Met de keuze "Graphics" kunnen histogrammen, polygonen en van beide in windows op het beeldscherm worden gebracht. Deze kunnen indien de computer is geinitialiseerd met Graphics.com via de prntscr-toets naar de printer worden gestuurd.
Door iteraties kan de nauwkeurigheid van een resulterende UNIT HYDROGRAPH worden verbeterd. Hiervoor leent een computerprogramma zich bij uitstek.

zie MENU-STROOMDIAGRAM

2.4. HYDROLIN X.XX.

Nieuwe versies van HYDROLIN zullen met een in 1988 uit te komen TURBO PASCAL versie 4.00 sneller kunnen worden door samenvoeging van alle programmacomponenten in één source- en codeprogramma. Er zullen procedures voor verdere manipulaties met de UNIT HYDROGRAPH kunnen worden ingebracht en nieuwe theorieen worden "aangehangen". Die versies zullen door gebrek aan externe geheugenruimte op 360 k-floppies noodgedwogen op harddisc-PC's moeten draaien. Met nieuw te komen TURBO PASCAL versies kunnen vele krachtige programma's middels een Linker worden gekoppeld.

Voor overige zij verwezen naar 5.3.

3. SYSTEEM CONFIGURATIE.

HYDROLIN versie 1.00 is geschreven in de PC DOS TURBO PASCAL versie 3.01A voor een Olivetti M24 met twee discdrives van 360 kBytes en het verwerkingssysteem MS DOS versie 2.11 en hoger. Dit betekent, dat HYDROLIN op elke IBM compatible PC kan draaien. De eisen voor het intern werkgeheugen zijn zeer gering, omdat bovengenoemde Pascal-versie slechts over 64 kBytes werkgeheugen voor source en code beschikt. Werken met chain-files omzeilt het probleem niet alles tegelijk in het werkgeheugen te kunnen laden. Door wijziging van door HYDROLIN standaard aangeroepen drives is deze HYDROLIN versie gemakkelijk te transformeren naar een hard-discversie; hierdoor krijgt HYDROLIN er meer extern (en werk-) geheugen bij b.v. bij pointergebruik en met TURBO EXTENDER. Het beeldscherm is monochroom en bij voorbaat niet fosforiserend, van-wege het lange nalichten.

4. PROGRAMMA.

4.1. LIJST VAN SYSTEEM-PROGRAMMA'S;

PROGRAMMA	OMSCHRIJVING
COMMAND.COM	MS DOS command-programma's
ANSI.SYS	
FORMAT.COM	
CONFIG.SYS	
PRINT.COM	
SYS.COM	
GRAPHICS.COM	
TURBO.COM	Turbo Pascal editer/compiler
TURBO.MSG	
TINST.COM	Installatie van Turbo Pascal op computertype
TGINST.COM	Installatie van Graphics-card
TGINST.MSG	
TYPEDEF.SYS	Turbo Graphics Toolbox
GRAPHIX.SYS	
KERNEL.SYS	
ERROR.MSG	
FINDWRLD.HGH	
HISTOGRM.HGH	
POLYGON.HGH	
WINDOWS.SYS	
AXIS.HGH	
SPLINE.HGH	
4X6.FON	
8X8.FON	
8X9.FON	

4.2. LIJST VAN HYDROLIN-PROGRAMMA'S;

PROGRAMMA	OMSCHRIJVING
HYDROLIN.BAT	Start met logo naar HYDRODAT
HYDRODAT.COM	Hoofdprogramma van HYDROLIN
COMM211.CHN	Command-programma voor DOS-commando's vanaf de versie MS DOS 2.11 voor twee drives
COMM300.CHN	Idem voor de harddiscversies vanaf de versie 3.00
MSDOSDIR.CHN	Zeer uitgebreid directory-programma (print-versie is niet geimplementeerd)
MSDOSLIB.INC	Bij MSDOSDIR behorende bibliotheek
HYDROCAL.CHN	Afleiding van de Unit Hydrograph via matrix rekening
HYDROGR.CHN	Grafisch programma: histogram,polygon,window
TURBOLIB.INC	Library van utilities
12345PB0.DAT	Test-databestand(regenintensiteit)
123450B0.DAT	Test-databestand(afvoerdebieten)
HYDROL00.INC	Included file van diverse procedures
HYDROL01.INC	idem
HYDROL02.INC	idem
HYDROL03.INC	idem
HYDROL04.INC	idem

DIRECTORY van FLOPPY A :

COMMAND.COM
SYS.COM
ANSI.SYS
CONFIG.SYS
GRAPHICS.COM
TURBO.COM
TURBO.MSG
TINST.COM
TINST.MSG
TGINST.BAT
ACCESS3.BOX
HYDROLIN.BAT
HYDRODAT.COM
HYDROGR.CHN
TYPEDEF.SYS
GRAPHIX.SYS
GRAPHICS.IBM
KERNEL.SYS
ERROR.MSG
WINDOWS.SYS
FINDWRLD.HGH
AXIS.HGH
HISTOGRM.HGH
POLYGON.HGH
SPLINE.HGH
HATCH.HGH
4X6.FON
8X9.FON
8X9.FON
14X9.FON
MSDOSDIR.CHN
COMM211.CHN
COMM300.CHN

DIRECTORY van FLOPPY B :

TURBOLIB.INC
HYDROCAL.CHN
HYDROL00.INC
HYDROL01.INC
HYDROL02.INC
HYDROL03.INC
HYDROL04.INC
HYDROGR.PAS
HYDROCAL.PAS
12345PB0.DAT
12345QB0.DAT
12345UH0.DAT
COMM211.PAS
COMM300.PAS
MSDOSDIR.PAS
MSDOSLIB.INC

4.3. LIJST VAN MENU'S;

LIJST VAN MENU'S.

MAIN MENU :
Comm Dir Editer File_hdl Graphics Help Input/data Model Quit

MODEL MENU :
Nash Quit Stochast Tank Unit_hydrograph

PRECIP/DISCH MENU :
Compare Discharge Precipitation Quit Timestep

PRECIPITATION MENU :
Accum_nett Bruto Loss Nett Quit Wiped

PRECIP CONVERS MENU :
0mm/hour 1mm/half_an_hour 2mm/day 3quit

PRECIP LOSS MENU :
Initial Phi_index Quit User_def

AREA DEFINE MENU :
Are Hectare M(sq) Km(sq) Quit

DISCHARGE MENU :
Accum_nett Bruto Nett Quit Reservoir Separ Wiped

DISCH CONVERS MENU :
0cub.m/sec 1cub.m/min 2m/sec 3quit

DISCH SEPAR MENU :
Constant Linear Quit User_def

vervolg Lijst van Menu's

EDITER MENU :
Edlin File_mngr Hydredit Norton Quit Turbo User_def Wordstar

HELP MENU :
Hydrolin Nash Quit Stochastic Tank Unit_hydrgr

INPUT MENU :
File_hdl Inputfile Keyboard Quit Test

GRAPHICS MENU :
Histo_p Poly_q Quit Uh_poly Smooth_uh Wind_pq

FILE_HANDLING MENU :
Copy Dir Erase Print Quit Rename Show User_def

4.4. LIJST VAN PROCEDURES+OMSCHRIJVINGEN;

LIJST VAN TURBOLIB-PROCEDURES EN -FUNCTIES

PROCEDURE/FUNCTIE	OMSCHRIJVING
LV	LowVideo(gedimde weergave van characters)
HV	NormVideo/HighVideo(normale characterweerg.)
BELL	Kort alarm
CLRLINE	Vegen van een lijn vanaf gegeven x,y
INIT_BITIMAGE	Initieren van de printer voor grafische taak
CLRLINES	Vegen van een aantal lijnen tussen y1 en y2
SCROLL	Beeld draait naar boven weg
CURSORON	Cursor zichtbaar
CURSOROFF	Cursor onzichtbaar
COPYFILE	Copieren van een bestand
WRITEXY	Tekst te plaatsen op x,y
PRESSKEY	Wacht op een toets, met tekst, wegdraaien van het beeld en schoonmaken van beeld
DISPLAYFILE	Weergave van een bestand op het beeldscherm
DRAWBLOC	Het tekenen van een kader op de monitor
EXIST	Controle van het bestaan van een bestand
JA	Correspondeert met 'y of 'Y'
NEE	Correspondeert met 'n' of 'N'
PRINTFILE	Het printen van de inhoud van een bestand
YES	Correspondeert met 'y' of'Y' met een tekst
MENU	Verzorgt de weergave van een menu met hoofd
MSDOSVOLUME	Geeft volumenaam van een schijf
FILEHANDLING	Een serie bewerkingen met bestanden
SORT	Sorteerprocedure
QUICKSORT	Snelle sorteerprocedure

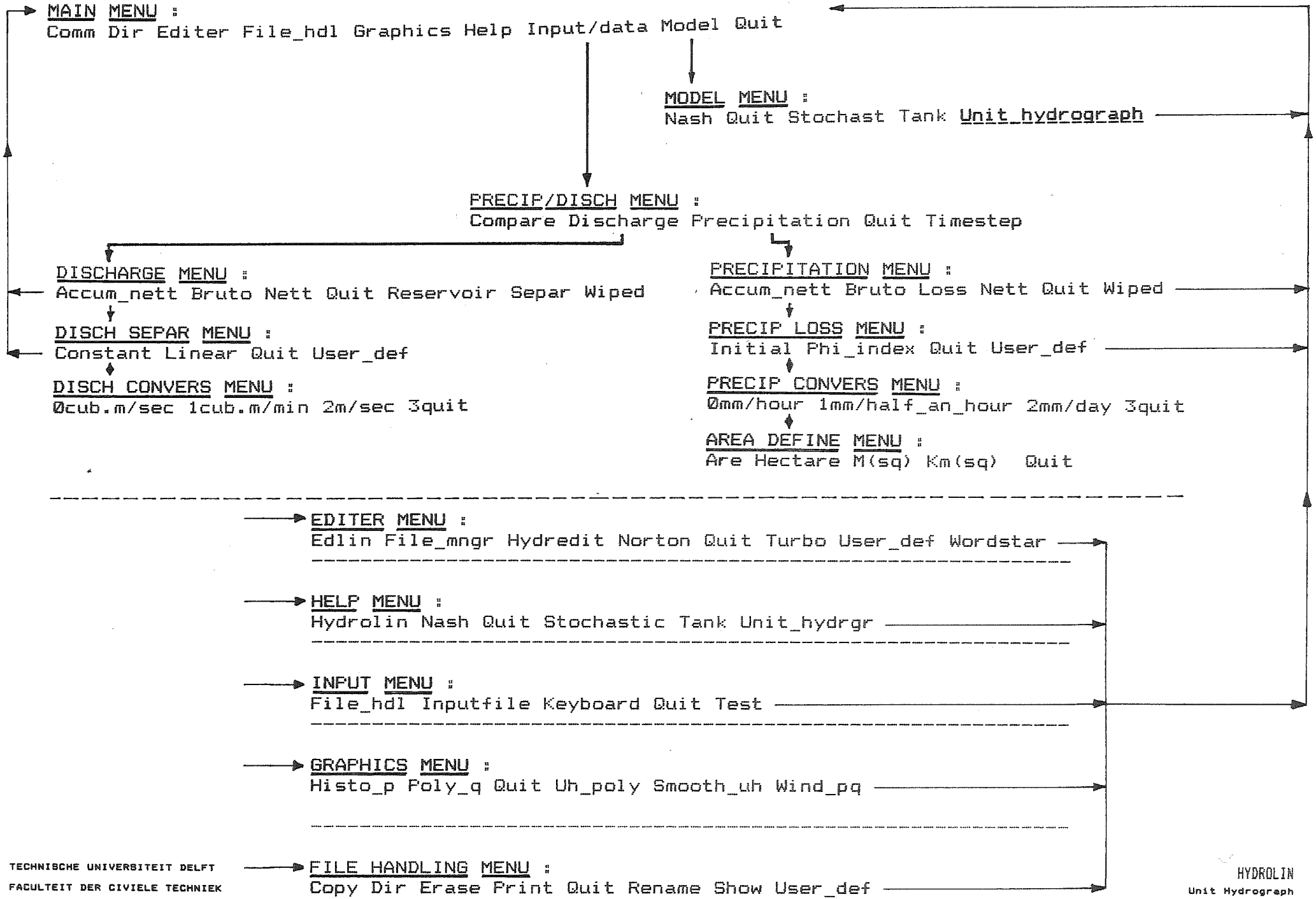
LIJST VAN PROCEDURES

PROCEDURE	OMSCHRIJVING
KEYBOARD_INPUT	Invoer van data middels het toetsenbord
INPUT_FILE	Invoer van data vanaf externe geheugen
HYDR_EDIT	Tekstbewerker van databestanden
FILE_NAME_NUM	Bestand wordt voorzien van een naam, specificatie en volgnummer
FILE_RENUM	Bewerkt bestand ontvangt een nieuw volgnummer
FILE_TEXT	Tekst in een databestand
HYDROLIN_HEADING	Logo bij start van Hydrolin
HEADING	Te implementeren heading voor uitvoer naar een extern device
DEVICE	Te implementeren menu van devices (in- en uitvoer)
HELP_HYDROLIN	Korte beschrijving van Hydrolin
HELP_UH	Beschrijving van de methode Unit Hydrograph in Hydrolin
TIME_STEP	Veranderen van de tijdstap in een datafile
AREA_DEFINE	Invoer en conversie van gebiedsgrootte
ZERO_WIPING	Nullen (regen of afvoer) worden geveegd tot de eerste ongelijk-nul-data
INITIAL_LOSS	Invoer en verwerking van initieel verlies tot een databestand
PHI_INDEX	Invoer en verwerking van een phi-index tot een databestand
USER_DEFINED	Invoer van zelf te kiezen verliezen en verwerking tot een databestand
PRECIP_CONVERSION	Conversie van regengegevens naar mm/uur
NETT_PRECIP	Bepaling van de netto regenintensiteit en verwerking tot een databestand
ACCUMUL_NETT_PRECIP	Geaccumuleerde netto regen
DEPLETION	Einde (voor de hoogwatergolf) en begin (na de hoogwatergolf) van het uitputt.verloop
SEPAR_DISCH	Menu van afvoerscheidingen t.b.v. bepaling van de opp.afvoer
DISCH_CONVERSION	Conversie van de afvoergegevens naar kubieke m/uur

vervolg LIJST VAN PROCEDURES

PROCEDURE	OMSCHRIJVING
NETT_DISCH	Bepaling van de netto debieten en verwerking tot een databestand
ACCUMUL_NETT_DISCH	Geaccumuleerde netto afvoer
COMPARE	Vergelijking van geaccumuleerde netto regen en afvoer
HYDROMAT	Invoer en oplossing van een stelsel dv met de methode van de kleinste kwadraten
MATPRINT	Printen van een matrix (bijgevoegde bibliotheek van matrixproc.)
VECPRINT	Printen van een vector
GAUSS	Gauss-eliminatie (andere in matrixbibliotheek)
P_HIST	Grafische weergave(van regengegevens)in een een histogram
Q_POLY	Grafische weergave(van afvoergegevens) in een veelhoek
PQ_WIND	Een histogram en een veelhoek tegelijk in afzonderlijk vensters in beeld
UH_SMOOTH	Een continue kromme door een set van punten (niet geimplementeerd)

4.5. MENU-STROOMDIAGRAM;



TECHNISCHE UNIVERBITEIT DELFT
FACULTEIT DER CIVIELE TECHNIEK

4.6. COMPUTERPROGRAMMA.

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	PRECIPITATION Ibr mm/hr	TIME tIbr hour
01	.00	5
02	.00	10
03	.00	15
04	9.35	20
05	26.53	25
06	17.62	30
07	13.17	35
08	5.53	40
09	.00	45
10	.00	50
11	.00	55
12	.00	60
13	.00	65
14	.00	70
15	.00	75
16	.00	80
17	.00	85
18	.00	90
19	.00	95
20	.00	100

Project/Version/Nr. : HYDROLIN/test/1.00
Date (jjmmdd) : 880216

SEQ.NR	PHI-INDEX LOSS Iphi mm/hour	TIME tIphi hour
1	0.000	5
2	0.000	10
3	0.000	15
4	2.480	20
5	2.480	25
6	2.480	30
7	2.480	35
8	2.480	40
9	0.000	45
10	0.000	50
11	0.000	55
12	0.000	60
13	0.000	65
14	0.000	70
15	0.000	75
16	0.000	80
17	0.000	85
18	0.000	90
19	0.000	95
20	0.000	100

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	PRECIPITATION Ibr mm/hr	TIME tIbr hour
01	.00	5
02	.00	10
03	.00	15
04	5.00	20
05	.00	25
06	.00	30
07	.00	35
08	.00	40
09	.00	45
10	.00	50
11	.00	55
12	.00	60
13	.00	65
14	.00	70
15	.00	75
16	.00	80
17	.00	85
18	.00	90
19	.00	95
20	.00	100

Project/Version/Nr. : HYDROLIN/test/1.00
Date (jjmmdd) : 880216

SEQ.NR	USERDEF LOSS Iusr mm/hour	TIME tIusr hour
1	0.000	5
2	0.000	10
3	0.000	15
4	0.000	20
5	0.000	25
6	0.000	30
7	0.000	35
8	0.000	40
9	0.000	45
10	0.000	50
11	0.000	55
12	0.000	60
13	0.000	65
14	0.000	70
15	0.000	75
16	0.000	80
17	0.000	85
18	0.000	90
19	0.000	95
20	0.000	100

Project/Version/Nr. : HYDROLIN/test/1.00
Date (jjmmdd) : 880212

SEQ.NR	PRECIPITATION Ibr mm/hour	TIME tIbr hour
1	9.350	5
2	26.530	10
3	17.620	15
4	13.170	20
5	5.530	25

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	DISCHARGE Qbr cubic m/s	TIME tQbr hour
01	93.737	5
02	92.199	10
03	92.646	15
04	258.017	20
05	1045.862	25
06	2139.401	30
07	2412.913	35
08	2169.489	40
09	1206.810	45
10	575.472	50
11	215.298	55
12	91.482	60
13	91.696	65
14	92.932	70
15	92.555	75
16	91.555	80
17	93.829	85
18	93.708	90
19	92.512	95
20	91.283	100

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	DISCHARGE Qbr cubic m/s	TIME tQbr hour
01	93.737	5
02	92.199	10
03	92.646	15
04	98.017	20
05	145.862	25
06	339.401	30
07	512.913	35
08	669.489	40
09	406.810	45
10	175.472	50
11	115.298	55
12	91.482	60
13	91.696	65
14	92.932	70
15	92.555	75
16	91.555	80
17	93.829	85
18	93.708	90
19	92.512	95
20	91.283	100

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	UNIT HYDROGRAPH QUH	TIME tUH hour
01	0.0000	0
02	0.1650	5
03	0.3735	10
04	0.3714	15
05	0.0000	20

```
(*========================== MS DOS ============================== *)
PROGRAM Command_MSDOS_2_11;                    (* MS DOS >2.11 *)
(*========================== MS DOS ============================== ==
MCS                  (C)  1987  M.J. Vos         TU/DELFT/Ct *)

CONST
 Program_Size=$400;
TYPE
 Long_STRING =String[255];
 TReg        =RECORD
              CASE Boolean OF
               True :(AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags:Integer);
               False:(AL,AH,BL,BH,CL,CH,DL,DH         :Byte   );
             END;
VAR
 FilVar      : Text;
 s           : Long_STRING;
 Dummy       : Boolean;


FUNCTION Exec(CO:Long_STRING;Size:Integer):Boolean;

CONST
 SVSS        : Integer=0;
 SVSP        : Integer=0;
 Flags       : Integer=0;
VAR
 Regs        : TReg;
BEGIN
 CO:=CO+#13;
 Regs.AX:=$4A00;
 Regs.ES:=CSeg;
 Regs.BX:=SSeg-CSeg+Program_Size;
 MSDOS(Regs);
 IF (Regs.Flags AND 1)=1 THEN
 BEGIN
  Exec:=False;
  Exit
 END;

 INLINE
 ($50/$53/$51/$52/$1E/$06/$56/$57/$55/$9C/$FA/$2E/
 $8C/$16/SVSS/$2E/$89/$26/SVSP/$FB/
 $8C/$D0/$8E/$D8/$8D/$76/<CO/$CD/$2E/$9C/$58/$2E/$A3/Flags/
 $FA/$2E/$8E/$16/SVSS/$2E/$8B/$26/SVSP/$FB/$9D/$5D/$5F/$5E/$07/
 $1F/$5A/$59/$5B/$58);

 EXEC:=NOT ((Flags AND 1)=1);
END;

BEGIN
 ClrScr;
 LowVideo;Write('MCS ');Write('/ TU DELFT / Ct (C) 880226 M.J.Vos ');WriteLn;
 WriteLn('MSDOS version from 2.11 with two floppy-disc drives');
 WriteLn('-----------------------------------------------');WriteLn;
 REPEAT
  WriteLn;HighVideo;Write('Command : ');
```

```
  Read(s);Writeln;
  IF s<>'' THEN Dummy:= Exec(s,Program_Size);
 UNTIl (s='') OR NOT Dummy;
 Assign(FilVar,'a:HYDROLIN.COM');
 Execute(FilVar);
END.

(*END======================== COMM211.PAS ======================== *)
```

```
(*========================== MS DOS ============================== *)
PROGRAM CommPrgm300                    (* MS DOS 3.XX  *)
(*========================== MS DOS ================================
MCS                (c)  1987   M.J. Vos       TU/DELFT/Ct  *)

CONST
 Program_Size= $400;
TYPE
 Long_String = STRING[255];
VAR
 s          : Long_String;
 Dummy      : Boolean;


FUNCTION Exec(Command:Long_String;Size:Integer):Boolean;

CONST
 SVSS       : Integer=0;
 SVSP       : Integer=0;
TYPE
 String32    = STRING[32];
 String80    = STRING[80];
 TReg        = RECORD
               CASE Boolean OF
                True: (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags:Integer);
                False:(AL,AH,BL,BH,CL,CH,DL,DH         :Byte   );
              END;
 Dummy_FCB   =ARRAY[0..36] OF Byte;
 Env_Pack    =RECORD
              Env:Integer;
              Cmdl:^STRING80;
              fcb1,fcb2:^Dummy_FCB;
             END;

VAR
 Name       : String32;
 cmd        : String80;
 FCB1,FCB2   : Dummy_FCB;
 Paket      : Env_Pack;
 Regs       : TReg;


 FUNCTION GetEnviremont(s:STRING80):STRING80;


 TYPE
  Arr80      = ARRAY[1..80] OF Char;
  Cptr       = ^Arr80;
 VAR
  p1         : Cptr;
  EnvSeg     : Integer ABSOLUTE CSeg:$2C;
  l          : Integer;
  Found      : Boolean;
 BEGIN
  p1:=Ptr(EnvSeg,0);Found:=False;l:=Length(s)+1;
  WHILE (NOT Found) AND (p1^[1]<>#0) DO
  BEGIN
```

```
  Found:=(s+'=')=(Copy(p1^,1,l));
  IF NOT Found THEN
  p1:=Ptr(Seg(p1^),Ofs(p1^)+Pos(#0,p1^));
 END;
 IF Found THEN
 BEGIN
  p1:=Ptr(Seg(p1^),Ofs(p1^)+1);
  GetEnviremont:=Copy(p1^,1,Pos(#0,p1^)-1);
 END ELSE
 GetEnviremont:=' ';
END;

BEGIN
 Regs.AX:=$4A00;
 Regs.ES:=CSeg;
 Regs.BX:=SSeg-CSeg+Program_Size;
 MSDOS(Regs);
 IF (Regs.Flags AND 1)=1 THEN
 BEGIN
  Exec:=False;
  Exit;
 END;

 INLINE
 ($50/$53/$51/$52/$1E/$06/$56/$57/$55/$9C/$FA/$2E/
 $8C/$16/SVSS/$2E/$89/$26/SVSP/$FB);

 Name:=GetEnviremont('COMSPEC')+#0;
 Regs.DS:=Seg(Name);
 Regs.DX:=Ofs(Name)+1;
 cmd:='C/'+Command+#13;
 Paket.fcb1:=Addr(FCB1);
 Paket.fcb2:=Addr(FCB2);
 Paket.Env:=0;
 Paket.Cmdl:=Addr(cmd);
 Regs.ES:=Seg(Paket);
 Regs.BX:=Ofs(Paket);
 Regs.AL:=0;
 Regs.AH:=$4B;
 MSDOS(Regs);
 Exec:=NOT((Regs.Flags AND 1)=1);

 INLINE
 ($FA/$2E/$8E/$16/SVSS/$2E/$8E/$26/SVSP/$FB/$9D/$5D/$5F/$5E/
 $07/$1F/$5A/$59/$5B/$58);
END;

BEGIN
 ClrScr;
 LowVideo;Write ('MCS ');Write('/ (c)  1987  M.J. Vos   TU/DELFT/Ct ');
 WriteLn;WriteLn('MSDOS version from 3.00 with harddisc');
 WriteLn('-----------------------------------------------');WriteLn;
 REPEAT
  WriteLn;HighVideo;Write('Command : ');
  Read(s);Writeln;
  IF s<>' ' THEN Dummy:= Exec(s,Program_Size);
 UNTIl (s=' ') OR NOT Dummy;
```

```
END.

(*END ========================== COM300.PAS ========================== *)
```

```
PROGRAM Directory; (* Example program / M.J. Vos *)
                   (* HYDROLIN 1.00 extension    *)

VAR
  Path: STRING[64];
  Ch  : Char;

BEGIN
  Ch := '1';
  Repeat
    IF Upcase(Ch) IN ['1', 'M', '2', 'R', '3', 'C', '0', 'Q'] THEN
      BEGIN
        ClrScr;
        GetDir(0,Path);
        WriteLn('Current directory is ',Path);
        Writeln;
        WriteLn('Choose option: ');
        WriteLn('  1: Make a directory');
        WriteLn('  2: Remove a directory');
        WriteLn('  3: Change the current directory');
        WriteLn('  0: Quit');
        Writeln;
        Write('Option: ');
        Read(Kbd,Ch);

        {$I-}
        Case Upcase(Ch) OF
          '1','M': BEGIN
                     WriteLn('Make');
                     Write('Make what directory? ');
                     Readln(path);
                     MkDir(Path);
                   END;
          '2','R': BEGIN
                     WriteLn('Remove');
                     Write('Remove what directory? ');
                     Readln(path);
                     RmDir(Path);
                   END;
          '3','C': BEGIN
                     WriteLn('Change');
                     Writeln;
                     Write('Change to what directory? ');
                     Readln(path);
                     ChDir(Path);
                   END;
          '0','Q': WriteLn('Quit');
          ELSE
        END; { case }

        {$I+}
        IF IOResult<>0 THEN
          BEGIN
            Write('*** Error: ', path);
            Delay(3000);
          END;
      END { if }
```

```
    ELSE
      Read(kbd, ch)
   UNTIL Upcase(Ch) IN ['0','Q', #27];
END.
(* END ============================ DIRECT.PAS ============================*)
```

```
PROGRAM GetDate; (* Example program / M.J. Vos                    *)
                 (* HYDROLIN 1.00 Extension                      *)
                 (* Make a procedure of it by eliminating the Main-part *)


TYPE DateStr = STRING[10];

FUNCTION Date : DateStr;

TYPE
 RegPack = RECORD
   AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : Integer;
         END;
VAR
 RecPack      : RegPack;
 Month,Day    : STRING[2];
 Year         : STRING[4];
 Dx,Cx        : Integer;

BEGIN
 WITH RegPack DO
 BEGIN
  AX:=$2a SHL 8;
 END;
 MSDOS(RegPack);
 WITH RegPack DO
 BEGIN
  STR(CX,Year);
  STR(DX MOD 256,Day);
  STR(DX SHR 8,Month);
 END;
 Date:=Day+'/'+Month+'/'+Year;
END; (* PROCEDURE Date *)

BEGIN(* Main *)
 WriteLn(Date);
END.

(* END ============================== GETDATE.PAS ======================== *)
```

```
Program GWLister; (*Vakgroep Gezondheidstechniek en Waterbeheersing*)

Type DateStr              = STRING[10];

Var  LineTel,LinePag,DatTel,
     Page                 : Integer;
     FilVar               : Text;
     FileName             : STRING[30];
     Line                 : STRING[80];

(*=============================================================*)
FUNCTION Date             : DateStr;
(*=============================================================*)

Type RegPack              = RECORD
     AX,BX,CX,DX,BP,SI,DI,
     DS,ES,FLAGS          : Integer;
                       END;
VAR  RecPack              : RegPack;
     Month,Day            : STRING[2];
     Year                 : STRING[4];
     Dx,Cx                : Integer;

BEGIN
 WITH RecPack DO
 BEGIN
  AX:=$2a SHL 8;
 END;
 MSDOS(RecPack);
 WITH RecPack DO
 BEGIN
  STR(CX,Year);
  STR(DX MOD 256,Day);
  STR(DX SHR 8,Month);
 END;
 Date:=Day+'/'+Month+'/'+Year;
END; (* FUNCTION Date *)

BEGIN
 Write('Print file               : ');Read(FileName);WriteLn;
 Write('Pagelength (55 - 65)     : ');Read(LinePag);WriteLn;
 WriteLn;WriteLn('Printing ',FileName);
 Assign(FilVar,FileName);
 Reset(FilVar);
 Page:=1;
 REPEAT
  FOR DatTel:=1 TO 46 DO Write(LST,' ');
  Write(LST,Chr(15));
  Write(LST,'Vakgroep  Gezondheidstechniek en Waterbeheersing');
  WriteLn(LST,Chr(18));
  FOR DatTel:=1 TO 59 DO Write(LST,' ');
  Write(LST,Chr(15));Write(LST,Chr(27),'S',0);
  Write(LST,'       Sectie : HYDROLOGIE');
  Write(LST,Chr(27),'T');WriteLn(LST,Chr(18));
  WriteLn(LST);WriteLn(LST);
  WriteLn(LST,'File : ',FileName,'  /  Date : ',Date);
  WriteLn(LST);WriteLn(LST);
```

```
  FOR LineTel:=1 TO  LinePag-10 DO
  BEGIN
   ReadLn(FilVar,Line);
   WriteLn(LST,Line);
  END;
  WriteLn(LST);WriteLn(LST);WriteLn(LST);
  WriteLn(LST,'                          =  Page  ',Page,'  =');
  Write(LST,Chr(12));
  Page:=Page+1;
 UNTIL EOF(FilVar);
END.

(* END ======================= GWLISTER ======================*)
```

EEN INTRODUCTIE TOT HYDROLIN.

HYDROLIN is een interactief menugestuurd hydrologisch computerprogramma
Hierbij wordt U gevraagd om een keuze in menu's en invoer van data.
Stapt u uit een menu dan komt u terug in het hoofdmenu en kunt u opnieuw een
menu-traject kiezen.
Door deze programma-opzet is een manual niet strikt nodig.
Voorlopig is geimplementeerd :

DE METHODE VAN DE EENHEIDSAFVOERGOLF - UNIT HYDROGRAPH (UH) .

EINDE VAN DE INTRODUCTIE.

```
(*************************************************************************
* HYDROCAL.PAS                                                          *
*                                                                       *
* CONTENTS                                                              *
*                                                                       *
* MISCELLANEOUS HYDROLIN-PROCEDURES        TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* HydroMat; MatPrint; VecPrint; Gauss;                                  *
*************************************************************************)

PROGRAM HydroCal;         (* HYDROLIN version 1.00 *)

CONST
  Arr_Afm          = 30; (* Dimension precip/disch datasets *)

LABEL
  Start,Exit;

TYPE

  Str1           = STRING[1];
  Str2           = STRING[2];
  Str5           = STRING[5];
  Str6           = STRING[6];
  Str10          = STRING[10];
  Str12          = STRING[12];
  Str14          = STRING[14];
  Str20          = STRING[20];
  Str80          = STRING[80];
  Arr            = ARRAY[0..Arr_Afm] OF Real;
  ArrInt         = ARRAY[0..Arr_Afm] OF Integer;
  ArrArr         = ARRAY[1..Arr_Afm,1..Arr_Afm] OF Real;

VAR

{Hydrolin variables}

  Ch             : Char;
  Quit           : Boolean;
  FileName,FN,FN0,
  FN1,FN2        : Str14;
  Filvar,FV,FV0,FV1,
  FV2            : Text;
  Line,Line1,Line2 : Str80;
  Menu_Heading,Name,
  Titel,Method   : Str20;
  jjmmdd,Seq,Spc : Str6;
  tt_tt          : Str5;
  Version,FileSpec : Str2;
  FNM            : Str5;
  FileNumE,FileNumR: Str1;
  PQ,rcQ,Qbr,Qbs,
  Qnett,Ibr,Iinit,
  Iphi,Inett,Iusr,
  U,Y,AY,B,C     : Arr;
  A,AA,TA        : ArrArr;
  t,tQbr,tQbs,tQnett,
```

```
  tIbr,tIinit,tIphi,
  tInett,tIusr     : ArrInt;
  tIinit1,tIinit2,
  tQs,tQu,Bf,Bpr,
  M,N,P,Q,I,II,J   : Integer;
  TEL,tTEL,fTEL,
  bTEL,sTEL,MaxTEL : Byte;
  Area,ha,sqkm,Af,
  Apr,Qs,Qu,Phi,MulP,
  Pbrtot,Qbrtot,MulQ,
  Pntot,R,Qntot,
  QT1,QT2,QT3,AAH  : Real;
  StrPQ,Strt,
  DimStrPQ,DimStrt : Str20;

{Compiler directives}
{$C+,I-,U+,R+}

{Included files: hydrolin & library}
{$I b:turbolib.inc    }
{$I b:hydrol01.inc    }
{$I b:hydrol02.inc    }


(*=========================================================================*)
PROCEDURE Gauss(Var A:ArrArr; Var B,X:Arr; Var N:Integer);
(*=========================================================================*)

Var J,K,ROW,COL,MAX,JP1,NM1,KP1 :Integer;
   MUL,TEMP                     :REAL;

BEGIN
   NM1 := N-1;
   FOR J:= 1 TO NM1 DO
   BEGIN
      JP1 := J+1;
      MAX := J;
      FOR ROW := JP1 TO N DO
      BEGIN
         IF A[ROW,J] > A[MAX,J] THEN MAX:=ROW;
      END;
      FOR COL:=J TO N DO
        BEGIN
           TEMP := A[J,COL];
           A[J,COL] := A[MAX,COL];
           A[MAX,COL] := TEMP
        END;
      TEMP := B[J];
      B[J] := B[MAX];
      B[MAX] := TEMP;
      FOR K:=JP1 TO N DO
        BEGIN
           MUL := -A[K,J]/A[J,J];
           FOR COL:=J TO N DO A[K,COL]:=A[K,COL]+MUL*A[J,COL];
           B[K] := B[K] + MUL*B[J]
        END;
   END;
```

```
   X[N] := B[N]/A[N,N];
   FOR K:= NM1 DOWNTO 1 DO
     BEGIN
        TEMP := 0.0;
        KP1 := K+1;
        FOR J:= KP1 TO N DO TEMP:=TEMP+A[K,J]*X[J];
        X[K] := (B[K] - TEMP)/A[K,K]
     END;
END; (* Gauss *)

(*=========================================================================*)
PROCEDURE MatPrint(Head : Str80; N,M : Integer; A : ArrArr);
(*=========================================================================*)
BEGIN
 WriteLn(Head);WriteLn;
 FOR I:=1 TO M DO
 BEGIN
  FOR J:=1 TO N DO
  WriteLn(A[I,J]:12:3);
 END;
END; (* MatPrint *)

(*=========================================================================*)
PROCEDURE VecPrint(Head : Str80; M : Integer; Y: Arr);
(*=========================================================================*)

BEGIN
 WriteLn(Head);WriteLn;
 FOR I:=1 TO M DO
 WriteLn(Y[I]:12:3);
END; (* VecPrint *)

(*=========================================================================*)
PROCEDURE Hydromat;
(*=========================================================================*)

BEGIN
 Write('Give wiped nett-precipitation datafilename  ');LV;
 Write('(d:XXXXXPNX.DAT)');HV;WriteLn;
 Input_File;
 FOR TEL:=1 TO MaxTEL DO PQ[TEL]:=PQ[TEL]*1E-3/3600;
 B:=PQ;P:=MaxTEL;
 WriteLn('Number of precipitation data P : ',P);

 Write('Give the extent of the area ');Delay(3000);
 Area_Define;PressKey;
 Write('Give wiped nett-discharge     datafilename  ');LV;
 Write('(d:XXXXXQWX.DAT)');HV;WriteLn;
 Input_File;
 FOR TEL:=1 TO MaxTEL DO PQ[TEL]:=PQ[TEL]/Area;
 C:=PQ;Q:=MaxTEL;
 WriteLn('Number of discharge data     Q : ',Q);WriteLn;WriteLn;

 WriteLn('Matrix MxN from : ');WriteLn;
 WriteLn('q1 = u1.i1                          ');
 WriteLn('q2 = u2.i1 + u1.i2                  ');
 WriteLn('q3 = u3.i1 + u2.i2 + u1.i3          ');
```

```
WriteLn('etc.                          ');
WriteLn;WriteLn;WriteLn;
WriteLn('i1                      u1    q1');
WriteLn('i2  i1                   u2    q2');
WriteLn('i3  i2  i1                u3    q3');
WriteLn('i4  i3  i2  i1             u4    q4');
WriteLn('.   .   .   .          .     . ');
WriteLn('in  .               *    .  =  . ');
WriteLn('                          qm');
WriteLn('etc.');PressKey;

M:=Q;
N:=M-P; (* Matrix-dimensions *)

FOR I:=1 TO M DO
BEGIN
 FOR J:=1 TO N DO
 BEGIN
  A[I,J]:=0.0E00;
 END;
END; (* Initialisation *)


MatPrint('INITIALISATION',N,M,A);

FOR J:=1 TO N DO       (* Column *)
BEGIN
 FOR I:=J TO (J+P) DO  (* Row *)
 BEGIN
  A[I,J]:=B[J];
  TA[J,I]:=A[I,J];
 END;
END; (* MatInp *)


MatPrint('INPUT MATRIX',N,M,A);PressKey;
MatPrint('TRANSFORM. MATRIX',N,M,TA);PressKey;

FOR I:=1 TO M DO
Y[I]:=C[I]; (* Vector *)


VecPrint('INPUT VECTOR',M,Y);PressKey;

FOR II:=1 TO N DO
BEGIN
 FOR I:=1 TO N DO
 BEGIN
  AAH:=0.0;
  FOR J:=1 TO M DO
  AAH:=AAH + TA[I,J] * A[J,II];
  AA[I,II]:=AAH;
 END;
END;  (* AT*A Matrix *)


MatPrint('AT*A MATRIX',N,M,AA);PressKey;
```

```
 FOR I:=1 TO N DO
 BEGIN
  AAH:=0.0;
  FOR J:=1 TO M DO
  AAH:=AAH + TA[I,J] * Y[J];
  AY[I]:=AAH;
 END;   (* AT*Y *)


 VecPrint('TRANSFORM.VECTOR',M,AY);PressKey;

 Gauss(AA,Y,U,N);
 WriteLn('Calculated Unit Hydrograph ordinates : ');WriteLn;
 FOR I:=1 TO M DO WriteLn('I = ',I:6,'U[',I,'] = ',U[I]:20:3);PressKey;

END; (* HydroMat *)

BEGIN  (* HydroCal *)
 Start:
 Scroll;ClrScr;LV;
 GotoXY(12,5);
 Write('*************************************************************');
 GotoXY(12,6);
 Write('*                                                           *');
 GotoXY(12,7);
 Write('*                                                           *');
 GotoXY(12,8);
 Write('*         HYDROLIN 1.00  TURBO  MATRIX-OPERATIONS      *');
 GotoXY(12,9);
 Write('*         -------------  -----  -----------------      *');
 GotoXY(12,10);
 Write('*                                                           *');
 GotoXY(12,11);
 Write('*                                                           *');
 GotoXY(12,12);
 Write('*       ');LV;Write(' M.J. Vos   880226    TU/DELFT/Ct');
 Write('             *');
 GotoXY(12,13);
 Write('*                                                           *');
 GotoXY(12,14);
 Write('*                                                           *');
 GotoXY(12,15);
 Write('*************************************************************');
 Delay(3000);HV;
 HydroMat;
 Write('Do you want to Save these data ?');LV;Write('  (Y/N)');HV;
 Read(Kbd,Ch);WriteLn;
 IF Ja THEN File_Name_Num;
 Assign(FV,FileName);
 ReWrite(FV);
 StrPQ:='UNIT DISCHARGE QUH';Strt:='TIME tUH';
 DimStrPQ:='m/s';DimStrt:='hour';
 File_Text;
 FOR I:=1 TO M DO WriteLn(I:6,U[I]:20:3,t[I]:20);
 Close(FV);Scroll;ClrScr;
 Write('Do you want to use GRAPHICS ? ');LV;Write(' (Y/N)');HV;Read(Kbd,Ch);
 WriteLn;
```

```
 IF Ja THEN
 BEGIN
  Assign(FV,'b:HYDROGR.CHN');
  Chain(FV);
 END ELSE
 BEGIN
  Assign(FV,'a:HYDROLIN.COM');
  Execute(FV);
 END;
END. (* HydroCal *)

(*END ======================HYDROCAL.PAS ==================================*)
```

```
PROGRAM HydroDat;

CONST
  Arr_Afm          = 20; (* Max. number in dataset *)

LABEL
  Start,Exit;

TYPE
  Str1           = STRING[1];
  Str2           = STRING[2];
  Str5           = STRING[5];
  Str6           = STRING[6];
  Str10          = STRING[10];
  Str12          = STRING[12];
  Str14          = STRING[14];
  Str20          = STRING[20];
  Str80          = STRING[80];
  Arr            = ARRAY[0..Arr_Afm] OF Real;
  ArrInt         = ARRAY[0..Arr_Afm] OF Integer;

VAR

{Hydrolin variables}

  Ch             : Char;
  Quit           : Boolean;
  FileName,FN,FN0,
  FN1,FN2        : Str14;
  Filvar,FV,FV0,FV1,
  FV2            : Text;
  Line,Line1,Line2 : Str80;
  Menu_Heading,Name,
  Titel,Method   : Str20;
  jjmmdd,Seq,Spc : Str6;
  tt_tt          : Str5;
  Version,FileSpec : Str2;
  FNM            : Str5;
  FileNumE,FileNumR: Str1;
  PQ,rcQ,Qbr,Qsc,Qsl,
  Qsu,Qnett,Ibr,Qbs,
  Iinit,Iphi,Inett,
  Iusr           : Arr;
  t,tQbr,tQbs,tQnett,
  tIbr,tIinit,tIphi,
  tInett,tIusr   : ArrInt;
  tIinit1,tIinit2,
  tQs,tQd,Bf,Bpr : Integer;
  TEL,tTEL,fTEL,
  bTEL,sTEL,MaxTEL : Byte;
  Area,Af,Apr,Qs,
  Pbrtot,Qbrtot,Qd,
  Pntot,R,Qntot,Phi,
  QT1,QT2,QT3,
  MulP,MulQ      : Real;
  StrPQ,Strt,
  DimStrPQ,DimStrt : Str20;
```

```
{Compiler directives}
{$C+,I-,U+,R+}


{Included files: command in Turbo }
(*{$I b:comm211.chn} *)
(*{$I b:comm310.chn} *)
(*{$I b:hydropac.001}*)
(*{$I b:hydropac.002}*)

{Included files: directory       }
(*{$I b:msdoslib.inc}*)
(*{$I b:msdosdir.inc}*)

{Included files: graphics        }
(*{$I b:m24.inc     }*)
(*{$I b:quickvid.inc}*)
(*{$I a:typedef.sys }*)
(*{$I a:graphix.sys }*)
(*{$I a:kernel.sys  }*)
(*{$I a:windows.sys }*)
(*{$I a:hatch.hgh   }*)
(*{$I a:axis.hgh    }*)
(*{$I a:histogrm.hgh}*)
(*{$I a:polygon.hgh }*)

{Included files: hydrolin & library}
{$I b:turbolib.inc    }
{$I b:hydrol00.inc    }
{$I b:hydrol01.inc    }
{$I b:hydrol02.inc    }
{$I b:hydrol03.inc    }
{$I b:hydrol04.inc    }


BEGIN
(*HYDROLIN_Heading;*)
 Start:
 Main_Menu; (* to subsequent menues *)
 ClrLines(1,5);GotoXY(20,12);
 Write(' Quit  /// HYDROLIN  ///   ???');LV;Write('    (Y/N)');
 HV;Read(KBD,Ch);WriteLn;
 IF Ja AND Yes('... are you sure to quit ?') THEN
 BEGIN
  Scroll;
  GOTO Exit;
 END ELSE
 GOTO Start;
 Exit:
 ClrScr;
END.

(*END ======================== HYDRODAT.PAS ========================== *)
```

```
PROGRAM HydroGr;

(*************************************************************************
* HYDROGR.PAS                                                           *
*                                                                       *
* CONTENTS :                                                            *
*                                                                       *
* MISCELLANEOUS HYDROLIN-GRAPHICS-PROCEDURES    TU DELFT/M.J.Vos/1988 *
*************************************************************************
* P_Histo; Q_Poly; PQ_Wind(P_Hist_Wind/Q_Poly_Wind); UH_grph;           *
* Smooth_UH; Graphics_Menu; Main program.                               *
*************************************************************************)

(*Hydrolin declarations*)
CONST
  Arr_Afm          = 30; (* Dimension measurement-datasets *)
LABEL
  Start,Exit;

TYPE

  Str1           = STRING[1];
  Str2           = STRING[2];
  Str5           = STRING[5];
  Str6           = STRING[6];
  Str10          = STRING[10];
  Str12          = STRING[12];
  Str14          = STRING[14];
  Str20          = STRING[20];
  Str80          = STRING[80];
  Arr            = ARRAY[0..Arr_Afm] OF Real;
  ArrInt         = ARRAY[0..Arr_Afm] OF Integer;

VAR

(*Hydrolin variables*)

  Ch             : Char;
  Quit           : Boolean;
  FileName,FN,FN0,
  FN1,FN2        : Str14;
  Filvar,FV,FV0,FV1,
  FV2            : Text;
  Line,Line1,Line2,
  HdrTxt,HdrTxt1,
  HdrTxt2,Hdr,Hdr1,
  Hdr2           : Str80;
  Menu_Heading,Name,
  Titel,Method   : Str20;
  jjmmdd,Seq,Spc : Str6;
  tt_tt          : Str5;
  Version,FileSpec : Str2;
  FNM            : Str5;
  FileNumE,FileNumR: Str1;
  PQ,rcQ,Qbr,Qbs,
  Qnett,Ibr,Iinit,
  Iphi,Inett     : Arr;
```

```
  t,tQbr,tQbs,tQnett,
  tIbr,tIinit,tIphi,
  tInett          : ArrInt;
  tIinit1,tIinit2,
  tQu,Bf,Bpr      : Integer;
  TEL,tTEL,fTEL,
  bTEL,sTEL,MaxTEL : Byte;
  Area,ha,sqkm,Af,
  Apr,Qu,Phi,
  Pbrtot,Qbrtot,
  Pntot,R,Qntot    : Real;
  StrPQ,Strt,
  DimStrPQ,DimStrt : Str20;

(*Included files: graphics*)

{$I b:turbolib.inc    }

(*{$I b:m24.inc     }*)
(*{$I b:quickvid.inc}*)

{$I a:typedef.sys     }
{$I a:graphix.sys     }
{$I a:kernel.sys      }
{$I a:windows.sys     }
{$I a:findwrld.hgh    }
{$I a:hatch.hgh       }
{$I a:axis.hgh        }
{$I a:histogrm.hgh    }
{$I a:polygon.hgh     }
{$I a:spline.hgh      }
{
(*=========================================================================*)
PROCEDURE P_Histo;                          (* Fixed World *)
(*=========================================================================*)

VAR    I, DisplyLen, HatchDen : Integer;
       A1,A2                  : PlotArray;
       T                      : ARRAY[1..6] OF Str80;
       R                      : Real;
       Ch                     : Char;
       Hatch                  : Boolean;

       FV1,FV2                : Text;
       FN1,FN2                : Str14;
       TEL,fTEL,MaxTEL,Bf     : Integer;
       Af                     : Real;
       Line1,Line2            : Str80;



PROCEDURE P_Histogram(A1,A2 : PlotArray);

BEGIN
 REPEAT
  Write('Give name of precipitation-datafile        ');LV;
  Write('(d:XXXXXPXX.DAT)  ');HV;Read(FN1);WriteLn;
```

```
 IF Exist(FN1)=False THEN
 BEGIN
  Bell;WriteLn(FN1,' ... does not exist !');
  WriteLn('Try again .');Delay(3000);
 END;
UNTIL (Exist(FN1)=True);
Assign(FV1,FN1);
Reset(FV1);
FOR I:=1 TO 6 DO
BEGIN
 Readln(FV1,Line1);
 T[I]:=Line1;
END;
I:=1;
REPEAT
 ReadLn(FV1,fTEL,Af,Bf);
 A1[I,2]:=Af;
 A1[I,1]:=Bf;
 I:=I+1;
UNTIL Eof(FV1);
MaxTEL:=fTEL;
Close(FV1);

Hatch := False;
HatchDen := 7;
Write('Give Header-text : ');Read(Hdr1);WriteLn;

Write('Give name of precipitation(-loss-)datafile ');LV;
Write('(d:XXXXXPXX.DAT)  ');HV;Read(FN2);WriteLN;
IF Exist(FN2)=True THEN
BEGIN
 Assign(FV2,FN2);
 Reset(FV2);
 FOR I:=1 TO 6 DO
 BEGIN
  Readln(FV2,Line2);
  T[I]:=Line2;
 END;
 I:=1;
 REPEAT
  ReadLn(FV2,fTEL,Af,Bf);
  A2[I,2]:=Af;
  A1[I,1]:=Bf;
  I:=I+1;
 UNTIL Eof(FV2);
 Close(FV2);

 Write('Give Header-text : ');Read(Hdr2);WriteLn;
END ELSE
BEGIN
 Bell;WriteLn(FN2,' ... does not exist !');Delay(1000);
 Hatch := True;
 HatchDen := 7;
 FOR I:=1 TO 6 DO
 T[I]:=Line1;
 FOR I:=1 TO MaxTEL-1 DO
 BEGIN
```

```
   A2[I,2]:=A1[I,2];
  END;
  Hdr2:='';
 END;
 IF Hdr2='' THEN HdrTxt:=ConCat(' ',Hdr1,' ') ELSE
 HdrTxt:=ConCat(' ',Hdr1,' / ',Hdr2,' ');

 InitGraphic;
 ClearScreen;
 DisplyLen:=fTEL;
 SetColorWhite;
 SetBackground(0);
 SetHeaderOn;
 DefineWindow(1,0,0,XMaxGlb,YMaxGlb);
 DefineHeader(1,HdrTxt);
 DefineWorld(1,0,0,100,100);
 SelectWorld(1);
 SelectWindow(1);

 DrawBorder;
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawHistogram(A1,-DisplyLen,Hatch,HatchDen);
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawHistogram(A2,-DisplyLen,Hatch,HatchDen);
 FOR I:=1 TO 6 DO
 DrawTextW(50,10+I*5,1,T[I]);
END; (* P_Histogram *)


BEGIN (* Main   *)
     (*M24(64);*)
 P_Histogram(A1,A2);
 REPEAT UNTIL KeyPressed;
 Clearscreen;
 LeaveGraphic;
 Close(FV1);
 Close(FV2);
END; (* P_Histo *)
}
(*=====================================================================*)
PROCEDURE P_Histo;                          (*Find a World *)
(*=====================================================================*)

VAR    I, DisplyLen, HatchDen : Integer;
       A1,A2                  : PlotArray;
       T                    : ARRAY[1..6] OF Str80;
       R                    : Real;
       Ch                    : Char;
       Hatch                  : Boolean;

       FV1,FV2                : Text;
       FN1,FN2                : Str14;
       dx,dy,TELdx,TELdy,
       TEL,fTEL,MaxTEL,Bf     : Integer;
       Af,Temp                : Real;
       Line1,Line2            : Str80;
```

```
PROCEDURE P_Histogram(A1,A2 : PlotArray);

BEGIN
 REPEAT
  Write('Give name of precipitation-datafile        ');LV;
  Write('(d:XXXXXPXX.DAT)  ');HV;Read(FN1);WriteLn;
  IF Exist(FN1)=False THEN
  BEGIN
   Bell;WriteLn(FN1,' ... does not exist !');
   WriteLn('Try again .');Delay(3000);
  END;
 UNTIL (Exist(FN1)=True);
 Assign(FV1,FN1);
 Reset(FV1);
 FOR I:=1 TO 6 DO
 BEGIN
  Readln(FV1,Line1);
  T[I]:=Line1;
 END;
 I:=1;
 REPEAT
  ReadLn(FV1,fTEL,Af,Bf);
  A1[I,2]:=Af;
  A1[I,1]:=Bf;
  I:=I+1;
 UNTIL Eof(FV1);
 MaxTEL:=fTEL;
 DisplyLen:=fTEL;
 Close(FV1);

 Hatch := False;
 HatchDen := 3;
 Write('Give Header-text : ');Read(Hdr1);WriteLn;

 Write('Give name of precipitation(-loss-)datafile ');LV;
 Write('(d:XXXXXPXX.DAT)  ');HV;Read(FN2);WriteLN;
 IF Exist(FN2)=True THEN
 BEGIN
  Assign(FV2,FN2);
  Reset(FV2);
  FOR I:=1 TO 6 DO
  BEGIN
   Readln(FV2,Line2);
   T[I]:=Line2;
  END;
  I:=1;
  REPEAT
   ReadLn(FV2,fTEL,Af,Bf);
   A2[I,2]:=Af;
   A2[I,1]:=Bf;
   I:=I+1;
  UNTIL Eof(FV2);
  Close(FV2);

  Write('Give Header-text : ');Read(Hdr2);WriteLn;
 END ELSE
```

```
 BEGIN
  Bell;WriteLn(FN2,' ... does not exist !');Delay(1000);
  FOR I:=1 TO 6 DO
  T[I]:=Line1;
  FOR I:=1 TO MaxTEL-1 DO
  BEGIN
   A2[I,2]:=A1[I,2];
   A2[I,1]:=A1[I,1];
  END;
  Hdr2:='';
 END;
 IF Hdr2='' THEN HdrTxt:=ConCat(' ',Hdr1,' ') ELSE
 HdrTxt:=ConCat(' ',Hdr1,' / ',Hdr2,' ');

 InitGraphic;
 ClearScreen;
 SetColorWhite;
 SetBackground(0);
 DefineWindow(1,0,0,XMaxGlb,YMaxGlb);
 DefineWindow(2,0,0,XMaxGlb,YMaxGlb);
 DefineHeader(2,HdrTxt);
 DefineWorld(1,0,1000,1000,0);
 SelectWindow(2);
 SetHeaderOn;

 FindWorld(2,A1,DisPlyLen,1,1);
 SelectWindow(2);
 DrawBorder;
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawHistogram(A1,-DisplyLen,Hatch,HatchDen);
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 Hatch := True;
 DrawHistogram(A2,-DisplyLen,Hatch,HatchDen);
 SelectWorld(1);
 SelectWindow(1);
 dy:=100;TELdy:=1000 DIV dy;
 FOR TEL:=2 TO TELdy-1 DO
 DrawLine(50,TEL*dy,500,TEL*dy);
 WITH World[I] DO
 BEGIN
  Temp:=Y1;Y2:=Y1;Y2:=Temp;
 END;
 FOR I:=1 TO 6 DO
 DrawTextW(550,600+I*25,1,T[7-I]);
(*DrawTextW(300,500+I*50,1,T[I]);*)
(*Delay(1000);InvertScreen;*)
END; (* P_Histogram *)


BEGIN
     (*M24(64);*)
 P_Histogram(A1,A2);
 REPEAT UNTIL KeyPressed;
 LeaveGraphic;
END; (* P_Histo *)
{
(*=================================================================*)
```

```
PROCEDURE Q_Poly;                          (* Fixed World *)
(*=================================================================*)

VAR
  N,I                : Integer;
  B1,B2                 : PlotArray;
  T                  : ARRAY[1..6] OF Str80;
  Ch                 : Char;
  X1, X2                : Integer;
  FV3,FV4               : Text;
  FN3,FN4               : Str14;
  TEL,fTEL,MaxTEL,Bf     : Integer;
  Af                 : Real;
  Line1,Line2           : Str80;


PROCEDURE Q_Polygon(B1,B2 : PlotArray);

BEGIN
 REPEAT
  Write('Give name of discharge-datafile             ');LV;
  Write('(d:XXXXXQXX.DAT)  ');HV;Read(FN3);WriteLN;
  IF Exist(FN3)=False THEN
  BEGIN
   Bell;WriteLn(FN3,' ... does not exist !');
   WriteLn('Try again .');Delay(3000);
  END;
 UNTIL (Exist(FN3)=True);
 Assign(FV3,FN3);
 Reset(FV3);
 FOR I:=1 TO 6 DO
 BEGIN
  ReadLn(FV3,Line1);
  T[I]:=Line1;
 END;
 I:=1;
 REPEAT
  ReadLn(FV3,fTEL,Af,Bf);
  B1[I,1]:=Bf;
  B1[I,2]:=Af;
  I:=I+1;
 UNTIL Eof(FV3);
 MaxTEL:=fTEL;
 N:=MaxTEL;
 Close(FV3);
 Write('Give Header-text : ');Read(Hdr1);WriteLn;
 HdrTxt:=Hdr1;

 Write('Give name of discharge(-separation-)datafile ');LV;
 Write('(d:XXXXXQXX.DAT)  ');HV;Read(FN4);WriteLN;
 IF Exist(FN4)=True THEN
 BEGIN
  Assign(FV4,FN4);
  Reset(FV4);
  FOR I:=1 TO 6 DO
  BEGIN
   Readln(FV4,Line2);
```

```
   T[I]:=Line2;
  END;
  I:=1;
  REPEAT
   ReadLn(FV4,fTEL,Af,Bf);
   B2[I,1]:=Bf;
   B2[I,2]:=Af;
   I:=I+1;
  UNTIL Eof(FV4);
  MaxTEL:=fTEL;
  N:=MaxTEL;
  Close(FV4);
  Write('Give Header-text : ');Read(Hdr2);WriteLn;
 END ELSE
 BEGIN
  Bell;WriteLn(FN4,' ... does not exist !');Delay(1000);
  FOR I:=1 TO 6 DO
  T[I]:=Line1;
  FOR I:=1 TO N-1 DO
  BEGIN
   B2[I,1]:=B1[I,1];
   B2[I,2]:=0.0;
  END;
  Hdr2:='';
 END;
 IF Hdr2='' THEN HdrTxt:=ConCat(' ',Hdr1,' ') ELSE
 HdrTxt:=ConCat(' ',Hdr1,' / ',Hdr2,' ');

 InitGraphic;
 SetBackground(0);
 SetHeaderOn;
 DefineWindow(1,0,0,XMaxGlb,YMaxGlb);
 DefineHeader(1,HdrTxt);
 DefineWorld(1,0,0,100,5000);
 SelectWorld(1);
 SelectWindow(1);
 DrawBorder;
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawPolygon(B1,1,N,0,0,0);
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawPolygon(B2,1,N,0,0,0);
 FOR I:=1 TO 6 DO
 DrawTextW(50,500+I*200,1,T[I]);
END; (* Q_Polygon *)

BEGIN (* Main   *)
     (*M24(64);*)
 Q_Polygon(B1,B2);
 REPEAT UNTIL KeyPressed;
 LeaveGraphic;
 Close(FV3);
 Close(FV4);
END; (* Q_Poly  *)
}
(*=================================================================*)
PROCEDURE Q_Poly;                          (*Find a World *)
(*=================================================================*)
```

```
VAR    I,N                  : Integer;
       B1,B2                : PlotArray;
       T                    : ARRAY[1..6] OF Str80;
       R                    : Real;
       Ch                   : Char;
       FV3,FV4              : Text;
       FN3,FN4              : Str14;
       dx,dy,TELdx,TELdy,
       TEL,fTEL,MaxTEL,Bf   : Integer;
       Af,Temp              : Real;
       Line1,Line2          : Str80;


PROCEDURE Q_Polygon(B1,B2 : PlotArray);

BEGIN
 REPEAT
  Write('Give name of discharge-datafile             ');LV;
  Write('(d:XXXXXQXX.DAT)  ');HV;Read(FN3);WriteLN;
  IF Exist(FN3)=False THEN
  BEGIN
   Bell;WriteLn(FN3,' ... does not exist !');
   WriteLn('Try again .');Delay(3000);
  END;
 UNTIL (Exist(FN3)=True);
 Assign(FV3,FN3);
 Reset(FV3);
 FOR I:=1 TO 6 DO
 BEGIN
  ReadLn(FV3,Line1);
  T[I]:=Line1;
 END;
 I:=1;
 REPEAT
  ReadLn(FV3,fTEL,Af,Bf);
  B1[I,1]:=Bf;
  B1[I,2]:=Af;
  I:=I+1;
 UNTIL Eof(FV3);
 MaxTEL:=fTEL;
 N:=MaxTEL;
 Close(FV3);
 Write('Give Header-text : ');Read(Hdr1);WriteLn;
 HdrTxt:=Hdr1;

 Write('Give name of discharge(-separation-)datafile ');LV;
 Write('(d:XXXXXQXX.DAT)  ');HV;Read(FN4);WriteLN;
 IF Exist(FN4)=True THEN
 BEGIN
  Assign(FV4,FN4);
  Reset(FV4);
  FOR I:=1 TO 6 DO
  BEGIN
   Readln(FV4,Line2);
   T[I]:=Line2;
  END;
  I:=1;
```

```
  REPEAT
   ReadLn(FV4,fTEL,Af,Bf);
   B2[I,1]:=Bf;
   B2[I,2]:=Af;
   I:=I+1;
  UNTIL Eof(FV4);
  MaxTEL:=fTEL;
  N:=MaxTEL;
  Close(FV4);
  Write('Give Header-text : ');Read(Hdr2);WriteLn;
 END ELSE
 BEGIN
  Bell;WriteLn(FN4,' ... does not exist !');Delay(1000);
  FOR I:=1 TO 6 DO
  T[I]:=Line1;
  FOR I:=1 TO N-1 DO
  BEGIN
   B2[I,1]:=B1[I,1];
   B2[I,2]:=0.0;
  END;
  Hdr2:='';
 END;
 IF Hdr2='' THEN HdrTxt:=ConCat(' ',Hdr1,' ') ELSE
 HdrTxt:=ConCat(' ',Hdr1,' / ',Hdr2,' ');

 InitGraphic;
 SetBackground(0);
 SetHeaderOn;
 DefineWindow(1,0,0,XMaxGlb,YMaxGlb);
 DefineWindow(2,0,0,XMaxGlb,YMaxGlb);
 DefineHeader(2,HdrTxt);
 DefineWorld(1,0,1000,1000,0);
 SelectWindow(2);
 SetHeaderOn;
 FindWorld(2,B1,N,1,1);
 SelectWindow(2);
 DrawBorder;
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawPolygon(B1,1,N,0,0,0);
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawPolygon(B2,1,N,0,0,0);
 SelectWorld(1);
 SelectWindow(1);
 dy:=100;TELdy:=1000 DIV dy;
 FOR TEL:=2 TO TELdy-1 DO
(*DrawLine(50,TEL*dy,1000,TEL*dy);*)
 DrawLine(50,TEL*dy,500,TEL*dy);
 WITH World[I] DO
 BEGIN
  Temp:=Y1;Y2:=Y1;Y2:=Temp;
 END;
 FOR I:=1 TO 6 DO
 DrawTextW(550,600+I*25,1,T[7-I]);
(*DrawTextW(50,300+I*50,1,T[I]);*)
(*Delay(1000);InvertScreen;*)
END; (* Q_Polygon *)
```

```
BEGIN
     (*M24(64);*)
 Q_Polygon(B1,B2);
 REPEAT UNTIL KeyPressed;
 LeaveGraphic;
END; (* Q_Poly *)

(*=================================================================*)
PROCEDURE UH_grph;
(*=================================================================*)

VAR
  N,I                : Integer;
  C                  : PlotArray;
  T                  : ARRAY[1..6] OF Str80;
  Ch                 : Char;
  X1, X2               : Integer;
  FV5                : Text;
  FN5                : Str14;
  TEL,fTEL,MaxTEL,Bf     : Integer;
  Af                 : Real;
  Line1,Line2          : Str80;


PROCEDURE UH_Poly(C : PlotArray);

BEGIN
 REPEAT
  Write('Give name of unit hydrograph-datafile      ');LV;
  Write('(d:XXXXXQXX.DAT)  ');HV;Read(FN5);WriteLN;
  IF Exist(FN5)=False THEN
  BEGIN
   Bell;WriteLn(FN5,' ... does not exist !');
   WriteLn('Try again .');Delay(3000);
  END;
 UNTIL (Exist(FN5)=True);
 Assign(FV5,FN5);
 Reset(FV5);
 FOR I:=1 TO 6 DO
 BEGIN
  ReadLn(FV5,Line1);
  T[I]:=Line1;
 END;
 I:=1;
 REPEAT
  ReadLn(FV5,fTEL,Af,Bf);
  C[I,1]:=Bf;
  C[I,2]:=Af;
  I:=I+1;
 UNTIL Eof(FV5);
 MaxTEL:=fTEL;
 N:=MaxTEL;
 Close(FV5);
 Write('Give Header-text : ');Read(Hdr);WriteLn;
 HdrTxt:=Hdr;

 InitGraphic;
```

```
 ClearScreen;
 DefineWindow(1, 0, 0, XMaxGlb, YMaxGlb);
 DefineHeader(1,HdrTxt);
 DefineWorld(1,0,0,50,1);
 SelectWorld(1);
 SelectWindow(1);
 SetBackground(0);
 SetHeaderOn;
 DrawBorder;
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawPolygon(C,1,N,0,0,0);
 FOR I:=1 TO 6 DO
 DrawTextW(25,0.1+I*0.05,1,T[I]);
END; (* U_H *)


BEGIN(* Main    *)
    (*M24(64); *)
 UH_Poly(C);
 REPEAT UNTIL KeyPressed;
 LeaveGraphic;
 Close(FV5);
END; (* U_Hgrph *)


(*=======================================================================*)
PROCEDURE Smooth_UH;
(*=======================================================================*)

VAR
  N,I                : Integer;
  D                  : PlotArray;
  T                  : ARRAY[1..6] OF Str80;
  Ch                 : Char;
  X1, X2             : Integer;
  FV6                : Text;
  FN6                : Str14;
  TEL,fTEL,MaxTEL,Bf     : Integer;
  Af                 : Real;
  Line1,Line2            : Str80;


PROCEDURE UH_smooth(D : PlotArray);

BEGIN
 REPEAT
  Write('Give name of unit hydrograph-datafile      ');LV;
  Write('(d:XXXXXQXX.DAT)  ');HV;Read(FN6);WriteLN;
  IF Exist(FN6)=False THEN
  BEGIN
   Bell;WriteLn(FN1,' ... does not exist !');
   WriteLn('Try again .');Delay(3000);
  END;
 UNTIL (Exist(FN6)=True);
 Assign(FV6,FN6);
 Reset(FV6);
 FOR I:=1 TO 6 DO
```

```
 BEGIN
  ReadLn(FV6,Line1);
  T[I]:=Line1;
 END;
 I:=1;
 REPEAT
  ReadLn(FV6,fTEL,Af,Bf);
  D[I,1]:=Bf;
  D[I,2]:=Af;
  I:=I+1;
 UNTIL Eof(FV6);
 MaxTEL:=fTEL;
 N:=MaxTEL;
 Close(FV6);
 Write('Give Header-text : ');Read(Hdr);WriteLn;
 HdrTxt:=Hdr;

 InitGraphic;
 ClearScreen;
 DefineWindow(1, 0, 0, XMaxGlb, YMaxGlb);
 DefineHeader(1,HdrTxt);
 DefineWorld(1,0,0,50,1);
 SelectWorld(1);
 SelectWindow(1);
 SetBackground(0);
 SetHeaderOn;
 DrawBorder;
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 Spline(D,N,D[2,1],D[N-1,1],D,50);
 DrawPolygon(D,1,N,0,0,0);
 FOR I:=1 TO 6 DO
 DrawTextW(45,0.1+I*0.05,1,T[I]);
END; (* UH_Smoooth *)


BEGIN(* Main      *)
    (*M24(64);   *)
 UH_Smooth(D);
 REPEAT UNTIL KeyPressed;
 LeaveGraphic;
 Close(FV6);
END; (* Smooth_uh *)


(*=====================================================================*)
PROCEDURE PQ_Wind;
(*=====================================================================*)

VAR

  I,N,DisplyLen,HatchDen,
  X1, X2                  : Integer;
  E,F                    : PlotArray;
  T1,T2                   : ARRAY[1..6] OF Str80;
  R                      : Real;
  Ch                      : Char;
```

```
  FV,FilVar                : Text;
  FN,FileName              : Str14;
  TEL,fTEL,MaxTEL,Bf       : Integer;
  Af                       : Real;
  Line                     : Str80;
  CharHeight,CharWidth     : Real;


PROCEDURE Total_Wind;

PROCEDURE P_Histo_Wind;

VAR Hatch : Boolean;

BEGIN
 DefineWindow(2,Trunc(XMaxGlb/10),Trunc(3*YMaxGlb/20),
          Trunc(XMaxGlb/2),Trunc(1*YMaxGlb/2));
 DefineWorld(2,0,0,100,100);

 SelectWorld(2);
 SelectWindow(2);
 SetHeaderOn;
 DefineHeader(2,HdrTxt1);
 SetBackground(0);
 DrawBorder;
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 Hatch := True;
 HatchDen := 2;

 DrawHistogram(E,-DisplyLen,Hatch,HatchDen);
END; (* P_Histo_Wind *)


PROCEDURE Q_Poly_Wind;

BEGIN
 DefineWindow(3,Trunc(XMaxGlb/10),Trunc((YMaxGlb*12)/20),
          Trunc(XMaxGlb*5/10),Trunc((YMaxGlb*19)/20));
 DefineWorld(3,0,0,100,5000);

 SelectWorld(3);
 SelectWindow(3);
 SetBackground(0);
 SetHeaderOn;
 DefineHeader(3,HdrTxt2);
 DrawBorder;
 DrawAxis(8,-8,0,0,0,0,0,0,True);
 DrawPolygon(F,1,N,0,0,0);

 DefineWindow(4,Trunc(XMaxGlb*21/40),Trunc(3*YMaxGlb/20),
          Trunc(XMaxGlb*19/20),Trunc((YMaxGlb*19)/20));
 DefineWorld(4,0,0,250,250);

 SelectWorld(4);
 SelectWindow(4);
 SetBackground(0);
 DrawBorder;
```

```
 FOR I:=1 TO 6 DO
 DrawTextW(5,10+I*10,1,T1[I]);
 FOR I:=1 TO 6 DO
 DrawTextW(5,150+I*10,1,T2[I]);
END;  (* Q_Poly_Wind *)

BEGIN
 REPEAT
  Write('Give name of precipitation-datafile  ');LV;Write('(d:XXXXXPXX.DAT)  ');
  HV;Read(FileName);WriteLn;
  IF Exist(FileName)=False THEN
  BEGIN
   Bell;WriteLn(FileName,' does not exist !');
   WriteLn('Try again .');Delay(3000);
  END;
 UNTIL (Exist(FileName)=True);
 Write('Give Header-text : ');Read(Hdr1);WriteLn;
 HdrTxt1:=Hdr1;
 Assign(Filvar,FileName);
 Reset(Filvar);
 FOR I:=1 TO 6 DO
 BEGIN
  Readln(Filvar,Line);
  T1[I]:=Line;
 END;
 I:=1;
 REPEAT
  ReadLn(Filvar,fTEL,Af,Bf);
  E[I,2]:=Af;
  E[I,1]:=Bf;
  I:=I+1;
 UNTIL Eof(Filvar);
 DisplyLen:=fTEL;

 REPEAT
  Write('Give name of discharge    -datafile  ');LV;Write('(d:XXXXXQXX.DAT)  ');
  HV;Read(FN);WriteLn;
  IF Exist(FN)=False THEN
  BEGIN
   Bell;WriteLn(FN,' does not exist !');
   WriteLn('Try again .');Delay(3000);
  END;
 UNTIL (Exist(FN)=True);
 Write('Give Header-text : ');Read(Hdr2);WriteLn;
 IF Hdr2='' THEN HdrTxt:=Hdr1 ELSE
 HdrTxt2:=Hdr2;
 HdrTxt:=ConCat('CONCATENATED ',Hdr1,'  /  ',Hdr2,' GRAPHICS');
 Assign(FV,FN);
 Reset(FV);
 FOR I:=1 TO 6 DO
 BEGIN
  ReadLn(FV,Line);
  T2[I]:=Line;
 END;
 I:=1;
 REPEAT
  ReadLn(FV,fTEL,Af,Bf);
```

```
  F[I,1]:=Bf;
  F[I,2]:=Af;
  I:=I+1;
 UNTIL Eof(FV);
 MaxTEL:=fTEL;
 N:=MaxTEL;

(*M24(64);*)
 InitGraphic;
 DrawBorder;
 DefineWindow(1,2,10,XmaxGlb-2,YMaxGlb-10);
 DefineHeader(1,HdrTxt);
 SetHeaderOn;
 SelectWindow(1);
 SelectWorld(1);
 DefineWorld(1,0,0,XmaxGlb,YMaxGlb);
 DrawBorder;
 SetBackground(51);
 P_Histo_Wind;
 Q_Poly_Wind;
END;

BEGIN(* Main    *)
 Total_Wind;
 REPEAT UNTIL KeyPressed;
 LeaveGraphic;
 Close(FilVar);
 Close(FV);
END; (* PQ_Wind *)

(*=====================================================================*)
PROCEDURE Graphics_Menu;
(*=====================================================================*)

BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='GRAPHICS MENU';
  CASE Menu('Histo_p Poly_q Unit_hydrogr Wind_pq Smooth_uh Quit') OF

    'H' :  BEGIN
            P_Histo;
           END;
    'P' :  BEGIN
            Q_Poly;
           END;
    'S' :  BEGIN
            Smooth_uh;
           END;
    'U' :  BEGIN
            UH_grph;
           END;
    'W' :  BEGIN
            PQ_Wind;
           END;
    'Q' :  BEGIN
```

```
          IF Yes('... are you sure ?') THEN
          Quit:=True;
         END; (* Quit *)
  END; (* Case *)
  HV;
 UNTIL Quit=True;
END;   (* Graphics_Menu *)

(*=====================================================================*)
(*                    MAINPROGRAM                      *)
(*=====================================================================*)

BEGIN
 Start:
 Scroll;ClrScr;
 GotoXY(12,5);
 Write('H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H');
 GotoXY(12,6);
 Write('H                                        H');
 GotoXY(12,7);
 Write('H                                        H');
 GotoXY(12,8);
 Write('H         HYDROLIN/1.00  TURBO  GRAPHICS             H');
 GotoXY(12,9);
 Write('H                                        H');
 GotoXY(12,10);
 Write('H                                        H');
 GotoXY(12,11);
 Write;
 GotoXY(12,12);
 Write('          ');LV;Write('TU DELFT / Ct   (C)    M.J. Vos   ');
 HV;GotoXY(12,13);
 Write;
 GotoXY(12,14);
 Write('H                                        H');
 GotoXY(12,15);
 Write('H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H');
  Delay(3000);HV;
 Graphics_Menu;
 GotoXY(20,10);
 Write('Do you want to use more Graphics ?');LV;Write('  (Y/N)');Read(Kbd,Ch);
 HV;WriteLn;
 IF Ja THEN GOTO Start ELSE
 IF Yes('... are you sure ?') THEN
 BEGIN
  Bell;Scroll;ClrScr;
  FOR TEL:=1 TO 2 DO
  BEGIN
   Bell;GotoXY(20,10);Write('Exit  ///  HYDROLIN 1.00 - GRAPHICS  ///');
   Delay(1000);ClrLine(1,10);Delay(1000);
  END;
  Scroll;ClrScr;
 END ELSE
 GOTO Start;
 Exit:
 Assign(FilVar,'a:HYDRODAT.COM');
 Execute(Filvar);
```

```
END. (* HYDROGR.PAS *)

(*END ====================== HYDROGR.PAS ============================ *)
```

```
(*************************************************************************
* HYDROL00.INC                                                          *
*                                                                       *
* CONTENTS                                                              *
*                                                                       *
* MISCELLANEOUS HYDROLIN-PROCEDURES        TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* Hydrolin_Heading; Device; Heading; Help_Hydrolin; Help_UH;            *
* Help_Menu.                                                            *
*************************************************************************)

{
(*=======================================================================*)
PROCEDURE HYDROLIN_Heading;
(*=======================================================================*)
(*          Possible heading with graphics drawbloc                      *)

BEGIN
CPMdrawbloc(1,2,66,21);
GotoXY(3,5);Write('              //  //');
GotoXY(3,6);Write('             //  //');
GotoXY(3,7);Write('            //--//');
GotoXY(3,8);Write('           //--//  Y D R O L I N      version 1.00');
GotoXY(3,9);Write('          //  //');
GotoXY(3,12);Write('          INTERACTIVE HYDROLOGICAL COMPUTERPROGRAM');
GotoXY(3,14);Write('             TU DELFT/Ct (C) 880226 M.J. Vos');
GotoXY(3,17);Write('             UNIVERSITY OF TECHNOLOGY  DELFT');
GotoXY(3,18);Write('             DEPARTMENT OF CIVIL ENGINEERING');
GotoXY(3,23);LV;Write('WAIT !');HV;
Delay(3000);
Scroll;
END;


(*=======================================================================*)
PROCEDURE HYDROLIN_Heading;
(*=======================================================================*)

BEGIN
FOR TEL:=1 TO 24 DO WriteLn;Delay(1000);
WriteLn('   H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H');
WriteLn('   H                                                             H');
WriteLn('   H                                                             H');
WriteLn('   H            //  //                                          H');
WriteLn('               //  //                                          ');
WriteLn('              //--//                                           ');
WriteLn('             //--//  Y D R O L I N      version 1.00            ');
WriteLn('   H         //  //                                          H');
WriteLn('   H                                                             H');
WriteLn('   H                                                             H');
WriteLn('   H        INTERACTIVE HYDROLOGICAL COMPUTERPROGRAM             H');
WriteLn('   H                                                             H');
WriteLn('   H             TU DELFT/Ct (C) 880226 M.J. Vos                 H');
WriteLn('   H                                                             H');
WriteLn('   H                                                             H');
WriteLn('   H                                                             H');
WriteLn('   H             UNIVERSITY OF TECHNOLOGY  DELFT                 H');
```

```
WriteLn('   H              DEPARTMENT OF CIVIL ENGINEERING              H');
WriteLn('   H                                                   H');
WriteLn('   H                                                   H');
WriteLn('   H                                                   H');
WriteLn('   H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H');
LV;WriteLn;WriteLn('   WAIT !');HV;
Delay(2500);
Scroll;
END;


(*==================================================================*)
PROCEDURE Device;
(*==================================================================*)
(*          Possible menu in case of several devices                *)

BEGIN
LV;
WriteLn('HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HH');
WriteLn;
WriteLn('choice  of  DEVICE:');
WriteLn('===================');
WriteLn;
WriteLn;
WriteLn('AUX:             = PUNCH RS232(Modem/Plotter)  ');
WriteLn('CON:          <= = CONSOLE(Buffered Input,Output)');
WriteLn('KBD:             = KEYBOARD(Input, no Echo)');
WriteLn('LST:          <= = PRINTER(Centronics, Serial)');
WriteLn('TRM:             = TERMINAL');
WriteLn('USR:             = USER DEFINED INPUT RS232(Modem)');
WriteLn('X:XXXXXXIN.DAT <= = INPUTFILENAME ON DISCDRIVE X:');
WriteLn('X:XXXXXOUT.DAT <= = OUTPUTFILENAME ON DISCDRIVE X:');
WriteLn;
WriteLn;
WriteLn;
WriteLn('HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HH');
WriteLn;WriteLn;WriteLn;HV;
Readln(FileName);
Assign(FV,FileName);
Scroll;
END; (* Device *)


(*=====================================================================*)
PROCEDURE Heading;
(*=====================================================================*)
(*          Possible heading in case of printing files                 *)

BEGIN
ClrScr;
LV;WriteLn('YOUR PROGRAMNAME :');HV;
Readln(Titel);
LV;WriteLn('METHOD :');HV;
Readln(Method);
LV;WriteLn('VERSIONNUMBER :');HV;
```

```
Readln(Version);
LV;WriteLn('YOUR NAME :');HV;
Readln(Name);
LV;WriteLn('DATE YOUR VERSION :');
Write('XXXXXX',Chr(22),Chr(13));HV;
Readln(jjmmdd);
LV;WriteLn('TIJD :');
Write('XX XX',Chr(13));HV;
Readln(tt_tt);LV;Write(' hour.');HV;
Write(Chr(22));
Scroll;ClrScr;
Device;
ReWrite(FV);
WriteLn(FV,'H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H');
WriteLn(FV,'                                                    ');
WriteLn(FV,'                                                    ');
WriteLn(FV,'                ///  HYDROLIN 1.00  ///                 ');
WriteLn(FV,'                                                    ');
WriteLn(FV,'                                                    ');
WriteLn(FV,'                 PROGRAM : ',Titel,'                  ');
WriteLn(FV,'                  METHOD : ',Method,'                   ');
WriteLn(FV,'                 VERSION : ',Version,'                 ');
WriteLn(FV,'                   NAME : ',Name,'                   ');
WriteLn(FV,'                                                    ');
WriteLn(FV,'                                                    ');
WriteLn(FV,'       UNIVERSITY OF TECHNOLOGY  DELFT                  ');
WriteLn(FV,'       DEPARTMENT OF CIVIL ENGINEERING                  ');
WriteLn(FV,'_____________________________________________________________________');
WriteLn(FV,'                                                    ');
WriteLn(FV,'       MS DOS                          IBM         ');
WriteLn(FV,'                                                    ');
WriteLn(FV,'       date : ',jjmmdd,'       time : ',tt_tt,' hour.        ');
WriteLn(FV,'                                                    ');
WriteLn(FV,'                                                    ');
WriteLn(FV,'H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H H');
LV;WriteLn('WAIT !');
Delay(3000);HV;
Scroll;
END; (* Heading *)

}
(*=================================================================*)
PROCEDURE Help_Hydrolin;   (* INTRODUCTION AND INFO HYDROLIN *)
(*=================================================================*)

BEGIN
ClrScr;
LV;WriteLn('HYDROLIN Help page');HV;
WriteLn;
WriteLn('HYDROLIN is een interactief hydrologisch analyseprogramma,   ');
WriteLn('dat menugericht U ten dienste wil staan. Hierbij wordt U ge- ');
WriteLn('vraagd om gegevens en een keuze uit opties.                  ');
WriteLn('Stapt u uit een menu dan komt u terug in het hoofdmenu en    ');
WriteLn('kunt u opnieuw een traject kiezen. Een manual is dienten-    ');
WriteLn('gevolge niet strikt nodig.                                   ');
WriteLn;
WriteLn('Voorlopig is geimplementeerd :                              ');
```

```
WriteLn;
WriteLn('DE METHODE VAN DE EENHEIDSAFVOERGOLF - UNIT HYDROGRAPH (UH) .');
Delay(500);PressKey;
END; (*Help_Hydrolin *)

(*=================================================================*)
PROCEDURE Help_UH;                (* INFO UNIT HYDROGRAPH *)
(*=================================================================*)

BEGIN
ClrScr;
LV;WriteLn('UNIT HYDROGRAPH Help page');HV;
WriteLn;
WriteLn('De METHODE VAN DE EENHEIDSAFVOERGOLF berust op empirisch gevon- ');
WriteLn('den relaties - L.K. Sherman 1932 - en dient ter bepaling van    ');
WriteLn('oppervlakteafvoercomponent (van de afvoer), die ontstaat ten ge-');
WriteLn('volge van de gemiddelde netto neerslag; deze neerslag wordt ge- ');
WriteLn('lijkmatig verdeeld gedacht over het stroomgebied. De grootte(A) ');
WriteLn('van het stroomgebied heeft een bovengrens van ca. 10.000 vier-  ');
WriteLn('kante kilometer. Bij de METHODE VAN DE EENHEIDSAFVOERGOLF wordt ');
WriteLn('uitgegaan van een EENHEIDSDIEPTE- UNIT DEPTH - b.v. 1 inch of   ');
WriteLn('1 centimeter of een EENHEIDSVOLUME A ,gelijkmatig verdeeld over ');
WriteLn('het stroomgebied.                                    ');
WriteLn('De tijdsDUUR van een enkelvoudige bui wordt DT genoemd.         ');
WriteLn('Vanaf het einde van de bui tot het einde van de eenheidsafvoer- ');
WriteLn('kromme is de tijdsduur TR - de AFLOOPTIJD.                      ');
WriteLn('Het afvoerdebiet  wordt weergegeven als Q of als q=Q/A.         ');
WriteLn('Een over een bepaalde tijd gemeten neerslagkromme kan worden op-');
WriteLn('gedeeld in een aantal enkelvoudige buien van zo mogelijk zelfde ');
WriteLn('tijdsduur DT.                                        ');
WriteLn;
Delay(500);PressKey;
Scroll;ClrScr;
WriteLn('Allereerst zult u data moeten invoeren. Dit geschiedt via INPUT ');
WriteLn('/  KEYBOARD naar een bestand op schijf. Veranderingen hieraan   ');
WriteLn('moeten worden gedaan met een EDITER.                            ');
WriteLn('Allerlei manipulaties met bestanden kunnen worden gedaan met    ');
WriteLn('COMMAND (= MSDOS commandos) of FILEHANDLING.                    ');
WriteLn('Vervolgens kunt u Regenverliezen PRECIP en Afvoerscheiding(en)  ');
WriteLn('DISCH invoeren, welke eveneens op schijf worden gezet onder een ');
WriteLn('voor u kenmerkende naam.                             ');
WriteLn('De bij elkaar behorende bruto en verlies-datafiles leveren      ');
WriteLn('netto-datafiles op. Vergelijking van geaccumuleerde netto regen ');
WriteLn('met afvoer is dan mogelijk. Repetitie van elk procedee is moge- ');
WriteLn('lijk tot een aantal van drie gemodificeerde gelijksoortige      ');
WriteLn('bestanden op schijf. Het beheer geschiedt door twee geimplemen- ');
WriteLn('procedures welke u vragen om characters t.b.v. identificatie ,  ');
WriteLn('specificatie en volgnummer.                          ');
WriteLn('Door filemanipulaties is het mogelijk meer bestanden te bewaren ');
WriteLn('b.v. onder een andere naam, op een andere schijf, op een andere ');
WriteLn('drive etc.                                           ');
WriteLn('De GRAPHICS-optie biedt de mogelijkheid tot graphische weergave ');
WriteLn('met HYDROGR en eventuele terugkeer in HYDROLIN.                 ');
Delay(500);PressKey;
END;  (* HELP_UH *)
```

```
(*=================================================================*)
PROCEDURE Help_Menu;
(*=================================================================*)

BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='HELP MENU';
  CASE Menu('Nash_cascade Hydrolin Stochastic Tank Unit_hydrograph Quit') OF

     'N' : BEGIN
           Bell;Writeln('Nash-cascade not yet implemented.');
           Delay(1000);ClrScr;
          END;
     'H' : BEGIN
          (*  Assign(FV,'b:HYDR.DOC'); *)
          (*  Reset(FV);              *)
          (*  REPEAT                  *)
          (*   FOR TEL:=1 TO 15 DO    *)
          (*   BEGIN                  *)
          (*    ReadLn(FV,Line);      *)
          (*    WriteLn(CON,Line);    *)
          (*   END;                   *)
          (*   PressKey;              *)
          (*  UNTIL Eof(FV);          *)
           Help_Hydrolin;
          END;
     'S' : BEGIN
           Bell;Writeln('Stochastic model not yet implemented.');
           Delay(1000);ClrSCr;
          END;
     'T' : BEGIN
           Bell;Writeln('Tank-model not yet implemented.');
           Delay(1000);ClrSCr;
          END;
     'U' : BEGIN
          (*  Assign(FV,'b:UH.DOC');   *)
          (*  Reset(FV);              *)
          (*  REPEAT                  *)
          (*   FOR TEL:=1 TO 15 DO    *)
          (*   BEGIN                  *)
          (*    ReadLn(FV,Line);      *)
          (*    WriteLn(CON,Line);    *)
          (*   END;                   *)
          (*   PressKey;              *)
          (*  UNTIL Eof(FV);          *)
           Help_UH;
          END;
     'Q' : BEGIN
           IF Yes('... are you sure ?') THEN
           Quit:=True;
          END;
  END; (* Case *)
 HV;
 UNTIL Quit=True;
END; (* Help_Menu *)
```

(*END======================== HYDROL00.INC ========================*)

```
(*************************************************************************
* HYDROL01.INC                                                          *
*                                                                       *
* CONTENTS :                                                            *
*                                                                       *
* MISCELLANEOUS HYDROLIN-PROCEDURES        TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* File_Name_Num; File_Renum; File_Text; Input_File; Keyboard_Input;   *
* Input_Menu; Hydr_Edit.                                                *
*************************************************************************)


(*=======================================================================*)
PROCEDURE File_Name_Num;
(*=======================================================================*)

BEGIN
 REPEAT
  Scroll;ClrScr;
  GotoXY(20,1);Write('*** DATA-FILE : NAME / NUMBER ***'); WriteLn;Delay(500);
  LV;Write('Default drive is');HV;Write(' b:');WriteLn;Delay(500);
  LV;WriteLn('Give your file-identification : ');WriteLn;
  Write('of exactly');HV;Write(' 5 ');LV;Write('characters ');WriteLn;
  WriteLn('without: drivenr.(d:), [|.\:";/<*]+> and filetype(.typ) : ',Chr(22));
  WriteLn;Write('XXXXX',Chr(13));HV;Read(FNM);WriteLn;
  Delay(500);
  LV;WriteLn('Give your file-specification : ');WriteLn;
  HV;Write('PB');LV;Write(' = bruto             precipitation;');
  WriteLn;
  HV;Write('PW');LV;Write(' = zeros wiped       precipitation ;');
  WriteLn;
  HV;Write('PI');LV;Write(' = initial losses    precipitation;');
  WriteLn;
  HV;Write('PF');LV;Write(' = phi-index loss    precipitation;');
  WriteLn;
  HV;Write('PU');LV;Write(' = user-def loss     precipitation;');
  WriteLn;
  HV;Write('PN');LV;Write(' = nett              precipitation;');
  WriteLn;
  HV;Write('PT');LV;Write(' = accumulated nett  precipitation;');
  WriteLn;WriteLn;Delay(500);
  HV;Write('QB');LV;Write(' = bruto             discharge;');
  WriteLn;
  HV;Write('QC');LV;Write(' = constant separ.  discharge ;');
  WriteLn;
  HV;Write('QL');LV;Write(' = linear separation discharge ;');
  WriteLn;
  HV;Write('QU');LV;Write(' = user-def separ.   discharge ;');
  WriteLn;
  HV;Write('QN');LV;Write(' = nett              discharge;');
  WriteLn;
  HV;Write('QW');LV;Write(' = zeros wiped       discharge');
  WriteLn;
  HV;Write('QT');LV;Write(' = accumulated nett  discharge;');
  WriteLn;WriteLn;
  HV;Write('UH');LV;Write(' = unit hydrograph           .');
  WriteLn;WriteLn;
```

```
  Write('XX',Chr(13));HV;Read(FileSpec);WriteLn;
  LV;Write('Give your file-number : (');HV;Write('0,1,2');
  LV;Write(' or');HV;Write(' 3) :');WriteLn;WriteLn;
  Write('X',Chr(13));HV;Read(FileNumE);WriteLn;
  WriteLn(Chr(22));
  FileName:=Concat('b:',FNM,FileSpec,FileNumE,'.DAT');
  IF Exist(FileName)=True THEN
  BEGIN
   Bell;
   WriteLn(FileName,' already exists !');Delay(1000);
   WriteLn('You should assemble another name ');
   Write('or do you want to overwrite an existing file ?');LV;
   Write(' (Y/N)');HV;WriteLn;Read(Kbd,Ch);
  END;
 UNTIL (Exist(FileName)=False OR (Ch IN ['y','Y']));
END; (* File_Name_Num *)


(*=================================================================*)
PROCEDURE File_Renum;
(*=================================================================*)

Label  DoIt;

BEGIN
 Scroll;ClrSCr;
 GotoXY(20,1);Write('*** DATA-FILE : RENUMBER ***');WriteLn;Delay(500);
 LV;Write('Default drive is ');HV;Write('b:');LV;WriteLn;WriteLn;Delay(500);
 WriteLn('If you desire substitutions in the source-program, consult the ');
 HV;WriteLn('               TU DELFT / HOLLAND.');WriteLn;Delay(500);
 WriteLn;
 Write('Do you want to use File_Handling ?  ');LV;Write('(Y/N)');HV;
 Read(Kbd,Ch);WriteLn;IF Ja THEN File_Handling;

 FNM:=Copy(FileName,3,7);
 FileSpec:=Copy(FileName,8,9);
 FileNumE:=Copy(FileName,10,10);

 IF FileNumE='0' THEN
 BEGIN FileNumR:='1';GOTO DoIt; END ELSE
 IF FileNumE='1' THEN
 BEGIN FileNumR:='2';GOTO DoIt; END ELSE
 IF FileNumE='2' THEN
 BEGIN FileNumR:='3';GOTO DoIt; END ELSE
 IF FileNumE='3' THEN
 BEGIN
  LV;Write('Overwrite an existing datafile ? ');HV;
  Write('(Y/N)');Read(Kbd,Ch);WriteLn;
  IF Ja THEN
  REPEAT
   LV;Write('Give datafile-');HV;Write('number : ');
   Read(FileNumR);WriteLn;
  UNTIL Yes('... are you sure ?') ELSE
  BEGIN
   WriteLn('This edited version will be saved as number 4,');
   WriteLn('but will be operable after: Copy/Delete/Rename');
   WriteLn('with sequence-number 1,2 or 3 in FileHandling.');WriteLn;
```

```
   FileNumR:='4';
  END;
 END;
 DoIt:
 FileName:=Concat('b:',FNM,FileSpec,FileNumR,'.DAT');
END; (* File_Renum *)


(*=====================================================================*)
PROCEDURE File_Text; {Implemented with six textlines}
(*=====================================================================*)

BEGIN
  Scroll;ClrScr;
  GotoXY(20,1);Write('*** DATA-FILE : TEXT ***');WriteLn;
  LV;Write('Project/version/Nr.                  : ');HV;
  Read(Line);WriteLn;
  LV;Write('Date      (jjmmdd)                   : ');HV;
  Read(jjmmdd);WriteLn;
  LV;Write('Dimension of ',StrPQ:20,'     : ');HV;
  Read(DimStrPQ);WriteLn;
  LV;Write('Dimension of ',Strt:20,'     : ');HV;
  Read(DimStrt);WriteLn;Seq:='SEQ.NR';Spc:='      ';
  WriteLn(FV,'Project/Version/Nr.      : ',Line);
  WriteLn(FV,'Date      (jjmmdd)       : ',jjmmdd);
  WriteLn(FV);
  WriteLn(FV,Seq:6,StrPQ:20,Strt:20);
  WriteLn(FV,Spc:6,DimStrPQ:20,DimStrt:20);
  WriteLn(FV);
  Scroll;ClrScr;
END; (* File_Text *)


(*=====================================================================*)
PROCEDURE Input_File;
(*=====================================================================*)

VAR  bTEL : Integer;

BEGIN
ClrScr;
REPEAT
 Write('Give input-filename ');LV;Write('(d:inputfile.dat) : ');HV;
 HV;Read(FileName);WriteLn;
 IF Exist(FileName)=False THEN
 BEGIN
  Bell;
  WriteLn('File does not exist !');
  Delay(3000);
 END;
UNTIL Exist(FileName)=True;Scroll;ClrScr;

Assign(FV,FileName);
Reset(FV);
WriteLn;
FOR TEL:=0 TO Arr_Afm DO
BEGIN
```

```
 PQ[TEL]:=0.0E00;
 t[TEL]:=00;
END; (* Initialisation *)

BEGIN
FOR tTEL:=1 TO 3 DO
 BEGIN
  Readln(FV,Line);
  WriteLn(CON,Line);
 END;
 ReadLn(FV,Seq,StrPQ,Strt);
 WriteLn(CON,Seq:6,StrPQ:20,Strt:20);
 ReadLn(FV,Spc,DimStrPQ,DimStrt);
 WriteLn(CON,Spc:6,DimStrPQ:20,DimStrt:20);
 Readln(FV,Line);
 WriteLn(CON,Line);
 Delay(200);
END; (* Text *)
fTEL:=0;bTEL:=1;
 REPEAT
  fTEL:=fTEL+1;
  IF fTEL=bTEL*15 THEN
  BEGIN
   bTEL:=bTEL+1;
   WriteLn;PressKey;
   Scroll;ClrScr;
   WriteLn(CON);
   WriteLn(CON,Seq:6,StrPQ:20,Strt:20);
   WriteLn(CON,Spc:6,DimStrPQ:20,DimStrt:20);
   WriteLn(CON,'-----------------------------------------------');
   WriteLn(CON);
  END;
  Readln(FV,fTEL,Af,Bf);
  TEL:=fTEL;
  PQ[TEL]:=Af;
  t[TEL]:=Bf;
  WriteLn(CON,TEL:6,PQ[TEL]:20:3,t[TEL]:20);
  Delay(200);
 UNTIL Eof(FV);
MaxTEL:=TEL;
Close(FV);

REPEAT
 WriteLn;
 Write('Repeated display of the file-contents ?  ');LV;
 Write('(Y/N)');HV;Read(Kbd,Ch);WriteLn;Scroll;ClrScr;
 IF JA THEN
 BEGIN
  WriteLn(CON);
  WriteLn(CON,Seq:6,StrPQ:20,Strt:20);
  WriteLn(CON,Spc:6,DimStrPQ:20,DimStrt:20);
  WriteLn(CON,'-----------------------------------------------');
  WriteLn(CON);
  BEGIN
   bTEL:=1;
   FOR TEL:=1 TO MaxTEL DO
   BEGIN
```

```
    IF TEL=bTEL*15 THEN
    BEGIN
     bTEL:=bTEL+1;
     WriteLn;PressKey;
     Scroll;ClrScr;
     WriteLn(CON);
     WriteLn(CON,Seq:6,StrPQ:20,Strt:20);
     WriteLn(CON,Spc:6,DimStrPQ:20,DimStrt:20);
     WriteLn(CON,'-----------------------------------------------');
     WriteLn(CON);
    END;
    WriteLn(CON,TEL:6,PQ[TEL]:20:3,t[TEL]:20);
    Delay(200);
   END;
  END;
 END;
UNTIL Nee;
END; (* Input_File *)


(*=====================================================================*)
PROCEDURE Keyboard_Input;
(*=====================================================================*)

LABEL Exit;

BEGIN
 ClrScr;

 FOR TEL:=0 TO Arr_Afm DO
 BEGIN
  PQ[TEL]:=0.0E00;
  t[TEL]:=00;
 END; (* Initialisering *)

 GotoXY(1,1);Write(Seq);
 GotoXY(15,1);Write(StrPQ);GotoXY(40,1);Write(Strt);LV;
 GotoXY(1,2);Write('-----------------------------------------------');
 HV;TEL:=0;
 REPEAT
  TEL:=TEL+1;
  GotoXY(1,TEL+3);Write(TEL:6);
  GotoXY(15,TEL+3);Read(Apr);
  GotoXY(40,TEL+3);Read(Bpr);
  PQ[TEL]:=Apr;
  t[TEL]:=Bpr;
  GotoXY(1,TEL+4);LV;
  Write('Continue wih <RETURN>   <<==>>   Stop with  S ');Read(Kbd,Ch);
  ClrLine(1,TEL+4);HV;
 UNTIL (Ch IN ['S','s']);
 MaxTEL:=TEL;
 Scroll;ClrScr;
 Write(Seq:6);Write(StrPQ:20);Write(Strt:20);WriteLn;
 LV;WriteLn('-----------------------------------------------');HV;
 WriteLn;
 REPEAT
  BEGIN
```

```
   bTEL:=1;
   FOR TEL:=1 TO MaxTEL DO
   BEGIN
    IF TEL=bTEL*15 THEN
    BEGIN
     bTEL:=bTEL+1;
     WriteLn;PressKey;
     Scroll;ClrScr;
     Write(Seq:6);Write(StrPQ:20);Write(Strt:20);WriteLn;
     LV;Write('-----------------------------------------------');HV;
     WriteLn;
    END;
    WriteLn(CON,TEL:6,PQ[TEL]:20:3,t[TEL]:20);
    Delay(200);
   END;
  END;
  WriteLn;
  Write('Repeated display of the file-contents ?  ');LV;
  Write('(Y/N)');HV;Read(Kbd,Ch);WriteLn;Scroll;ClrScr;
 UNTIL Nee;
 Write('SAVE keyboard-input ?');LV;Write(' (Y/N)');
 HV;Read(Kbd,Ch);WriteLn;ClrScr;
 IF Ja THEN
 REPEAT
  File_Name_Num;
  IF Exist(FileName)=True THEN
  BEGIN
   Bell;
   WriteLn(FileName,' exists !');
   Delay(1000);
   Write('Overwrite ',FileName,' ?');LV;Write(' (Y/N)');HV;
   Read(Kbd,Ch);WriteLn;
  END;
 UNTIL ((Exist(FileName)=False) OR Ja);
 Assign(FV,FileName);
 ReWrite(FV);
 File_Text;
 FOR TEL:=1 TO MaxTEL DO
 BEGIN
  fTEL:=TEL;
  Af:=PQ[TEL];
  Bf:=t[TEL];
  WriteLn(FV,fTEL:6,Af:20:3,Bf:20);
 END;
 Close(FV);
 PressKey;
END; (* Keyboard_Input *)


(*=================================================================*)
PROCEDURE Input_Menu;
(*=================================================================*)


BEGIN
ClrScr;
Quit:=False;
```

```
 REPEAT
  Menu_Heading:='INPUT MENU';
  CASE Menu('Keyboard Filehandling Inputfile Test Quit') OF

     'F' :  BEGIN
              File_Handling;
            END;
     'K' :  BEGIN
              WriteLn('Used for first input of data.');Delay(1000);
              WriteLn('For modifications use the I option.');Delay(1000);
              Write('Do you want to continue ? ');LV;Write('(Y/N)');
              HV;Read(Kbd,Ch);WriteLn;PressKey;
              IF Ja THEN Keyboard_Input ELSE
            END;
     'I' :  BEGIN
              WriteLn('Used for modifications. ');Delay(1000);
              WriteLn('Use for first input of data the K option.');Delay(1000);
              Write('Do you want to continue ? ');LV;Write('(Y/N)');
              HV;Read(Kbd,Ch);WriteLn;PressKey;
              IF Ja THEN Input_File ELSE
            END;
     'T' :  BEGIN
              LV;WriteLn('TestFiles :');
              Write('Precipitation : ');HV;Delay(1000);
              Write('b:12345PB0.DAT');WriteLn;LV;
              Write('Discharge     : ');HV;Delay(1000);
              Write('b:12345QB0.DAT');WriteLn;
              Write('Do you want to continue ?');LV;Write(' (Y/N) ');HV;
              WriteLn;Read(Kbd,Ch);
              IF Ja THEN Input_File ELSE
            END;
     'Q' :  BEGIN
              IF Yes('... are you sure ?') THEN Quit:=True;
            END;
  END; (* Case *)
  HV;
 UNTIL Quit=True;
 Scroll;ClrScr;
END; (* Input-Menu *)


(*=====================================================================*)
PROCEDURE Hydr_Edit;
(*=====================================================================*)

LABEL Exit;

VAR   bTEL :Integer;

BEGIN
 Scroll;GotoXY(20,12);
 Write('      /// HYDR');LV;Write('olin-');HV;Write('EDIT');
 LV;Write('er ');HV;Write('1.00 ///  ');
 Delay(3000);Scroll;ClrScr;
 WriteLn('You will see the contents of a chosen input-file.');WriteLn;
 WriteLn('Afterwards remember the sequential line-numbers to edit !');
 PressKey;
```

```
Input_File;
Write('Do you want to continue with HYDREDIT ?  ');LV;
Write('(Y/N)');HV;Read(Kbd,Ch);WriteLn;IF Nee THEN GOTO Exit;
REPEAT
 PressKey;
 GotoXY(1,1);LV;WriteLn('HYDREDIT 1.00');HV;
 GotoXY(1,5);Write(Seq:6);
 GotoXY(15,5);Write(StrPQ:20);GotoXY(40,5);Write(Strt:20);LV;
 GotoXY(1,6);Write('---------------------------------------------');
 HV;WriteLn;
 sTEL:=0;
 REPEAT
  GotoXY(1,7+sTEL);Write(Seq:6,' : ');Read(TEL);
  GotoXY(1,7+sTEL);ClrEol;
  GotoXY(1,7+sTEL);Write(TEL:6);
  GotoXY(15,7+sTEL);Write(PQ[TEL]:20);LV;Write(' wordt ');
  HV;Af:=PQ[TEL];Read(Af);PQ[TEL]:=Af;
  GotoXY(15,7+sTEL);ClrEol;Write(PQ[TEL]:20);
  GotoXY(40,7+sTEL);Write(t[TEL]:20);LV;Write(' wordt ');
  HV;Bf:=t[TEL];Read(Bf);t[TEL]:=Bf;
  GotoXY(40,7+sTEL);ClrEol;
  Write(t[TEL]:20);WriteLn;WriteLn;
  GotoXY(1,23);LV;Write('More ?   (Y/N)');Read(Kbd,Ch);
  GotoXY(1,23);ClrEol;
  WriteLn;HV;
  sTEL:=sTEL+1;
 UNTIL Nee;
 REPEAT
  Scroll;ClrScr;
  GotoXY(1,1);LV;WriteLn('HYDREDIT 1.00');HV;
  GotoXY(1,5);Write(Seq:6);
  Write(StrPQ:20);Write(Strt:20);LV;
  GotoXY(1,6);Write('---------------------------------------------');
  HV;WriteLn;
  TEL:=1;bTEL:=1;
  WHILE t[TEL]<>00 DO
  BEGIN
   WriteLn(CON,TEL:6,PQ[TEL]:20:3,t[TEL]:20);
   Delay(200);
   TEL:=TEL+1;
   IF TEL=bTEL*15 THEN
   BEGIN
    bTEL:=bTEL+1;
    WriteLn;PressKey;
   END; (* Block *)
  END;
  WriteLn;WriteLn;WriteLn('End of data-file.');
  WriteLn;Write('Repeated display of file-contents ?  ');LV;
  Write('(Y/N)');HV;Read(Kbd,Ch);WriteLn;
 UNTIL Nee;
 WriteLn;WriteLn;GotoXY(1,12);
 Write('      ||| End of Edit-session ||| ?  ');LV;
 Write('(Y/N)');HV;Read(Kbd,Ch);WriteLn;
UNTIL (Ja AND Yes('... o.k. ?'));
Write('Save data ?  ');LV;Write(' (Y/N)');Read(Kbd,Ch);WriteLn;
IF Ja THEN {Save_Data}
BEGIN
```

```
 LV;Write('Previous FileName                    : ');HV;
 Write(FileName);WriteLn;
 REPEAT
 Scroll;ClrScr;
 File_Renum;
 LV;Write('Next FileName in sequence    : ');HV;
 Write(FileName);WriteLn;
 IF Exist(FileName)=True THEN
  BEGIN
   Bell;WriteLn(FileName,' exists !');
   Delay(1000);
   WriteLn('Overwrite ',FileName,' ?');LV;Write(' (Y/N)');HV;
   Read(Kbd,Ch);WriteLn;
  END;
 UNTIL ((Exist(FileName)=False) OR Ja);
 Assign(FV,FileName);
 Rewrite(FV);
 File_Text;
 TEL:=1;
 WHILE t[TEL]<>0 DO
 BEGIN
  fTEL:=TEL;
  Af:=PQ[TEL];
  Bf:=t[TEL];
  WriteLn(FV,fTEL:6,Af:20,Bf:20);
  TEL:=TEL+1;
 END;
 Close(FV);
 PressKey;
 Scroll;ClrScr;
END;
 Exit:
END; (* Hydr_Edit *)

(*END ======================== HYDROL01.INC ========================
*)
```

```
(*************************************************************************
* HYDROL02.INC                                                          *
*                                                                       *
* CONTENTS                                                              *
*                                                                       *
* MISCELLANEOUS HYDROLIN-PROCEDURES        TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* Time_Step; Area_Define; Zero_Wiping; Initial_Loss; Phi_Index;         *
* Nett_Precip;Precip_Conversion; Accumul_Nett_Precip; Precip_Menu.      *
*************************************************************************)

(*=======================================================================*)
PROCEDURE Time_Step;
(*=======================================================================*)

VAR  dt,ts,nTEL,nMaxTEL,
     timestep          : Integer;
     nPQ               : Arr;
     nt                : ArrInt;
     iPQ               : Real;

BEGIN
 ClrScr;
 Quit:=False;
 WriteLn('WARNING !');Delay(1000);WriteLn;LV;
 WriteLn('The program will give a runtime error if the declared bounds ');
 WriteLn('of a new timestep-array  - initiated in the constant : Arr_Afm');
 WriteLn('in the declarationlist at the beginning of the program HYDRODAT -');
 WriteLn('will be to small.');HV;PressKey;
 REPEAT
  Menu_Heading:='TIME-STEP MENU';
  CASE Menu('Precipitation Discharge Quit') OF

'P' :
   BEGIN
    WriteLn;Write('Give filename ...');LV;Write('  (d:XXXXXPXXX.DAT)');HV;
    WriteLn;Delay(3000);
    Input_File;
    Write('Give a new timestep           : ');Read(timestep);WriteLn;

    ts:=timestep;
    dt:=t[2]-t[1];
    TEL:=0;nTEL:=0;
    PQ[0]:=0.0E00;nPQ[0]:=0.0E00;
    t[0]:=0;nt[0]:=0;

    IF ts<=dt THEN     (* Smaller timestep *)
    REPEAT
     TEL:=TEL+1;
     REPEAT
      nTEL:=nTEL+1;
      nt[nTEL]:=nt[nTEL-1]+ts;
      nPQ[nTEL]:=PQ[TEL];
     UNTIL nt[nTEL]>=t[TEL];
     nPQ[nTEL]:=PQ[TEL]*(t[TEL]-nt[nTEL-1])/dt+PQ[TEL+1]*(nt[nTEL]-t[TEL])/dt;
    UNTIL TEL=MaxTEL;
```

```
   IF ts>dt THEN     (* Bigger timestep *)
   REPEAT
    nTEL:=nTEL+1;
    nt[nTEL]:=nt[nTEL-1]+ts;
    REPEAT
     TEL:=TEL+1;
     nPQ[nTEL]:=nPQ[nTEL-1]+PQ[TEL];
    UNTIL t[TEL]>=nt[nTEL];
    iPQ:=((nt[nTEL]-t[TEL-1])/dt)*PQ[TEL];
    iPQ:=iPQ+((t[TEL]-nt[nTEL])/dt)*PQ[TEL+1];
    nPQ[nTEL]:=(nPQ[nTEL]-PQ[TEL]+iPQ)*dt/(nt[nTEL]-nt[nTEL-1]);
   UNTIL TEL=MaxTEL;
   nMaxTEL:=nTEL;

   FNM:=Copy(FileName,3,7);
   FileName:=Concat('b:',FNM,'PTS.DAT');
   WriteLn('Your timestep-filename will be b:',FNM,'PTS.DAT');
   Write('Do you agree ? ');LV;Write(' (Y/N)');HV;Read(Kbd,Ch);WriteLn;
   IF Nee THEN
    BEGIN
     LV;Write('Give your name of the file : ');HV;Read(FileName);WriteLn;
    END;
   Assign(FV,FileName);
   ReWrite(FV);
   File_Text;
   FOR nTEL:=1 TO nMaxTEL DO
   BEGIN
    Af:=nPQ[nTEL];
    Bf:=nt[nTEL];
    WriteLn(FV,nTEL:6,Af:20:3,Bf:20);
   END;
   Close(FV);
   WriteLn('Your filename has received a new number only !');
   WriteLn('If you want to rename : use the procedure Filehandling .');
   PressKey;
  END; (* Precipitation *)

'D' :

  BEGIN
   WriteLn;Write('Give filename ...');LV;Write('  (d:XXXXXQXX.DAT)');HV;
   WriteLn;Delay(3000);
   Input_File;
   Write('Give a new timestep   : ');Read(timestep);WriteLn;

   ts:=timestep;
   dt:=t[2]-t[1];
   TEL:=0;nTEL:=0;
   PQ[0]:=0.0E00;nPQ[0]:=0.0E00;
   t[0]:=0;nt[0]:=0;

   IF ts<=dt THEN     (* Smaller timestep *)
   REPEAT
    TEL:=TEL+1;
    REPEAT
     nTEL:=nTEL+1;
     nt[nTEL]:=nt[nTEL-1]+ts;
```

```
     nPQ[nTEL]:=nPQ[nTEL-1]+(PQ[TEL]-PQ[TEL-1])*ts/dt;
    UNTIL nt[TEL]>=t[TEL];
    nPQ[nTEL]:=nPQ[nTEL-1]+(PQ[TEL]-PQ[TEL-1])*(t[TEL]-nt[nTEL-1])/dt;
    nPQ[nTEL]:=nPQ[nTEL]+(PQ[TEL+1]-PQ[TEL])*(nt[nTEL]-t[TEL])/dt;
   UNTIL TEL=MaxTEL;

   IF ts>dt THEN     (* Bigger timestep *)
   REPEAT
    nTEL:=nTEL+1;
    nt[nTEL]:=nt[nTEL-1]+ts;
    REPEAT
     TEL:=TEL+1;
     nPQ[nTEL]:=nPQ[nTEL-1]+PQ[TEL]-PQ[TEL-1];
    UNTIL t[TEL]>=nt[nTEL];
    iPQ:=((nt[nTEL]-t[TEL-1])/dt)*(PQ[TEL]-PQ[TEL-1]);
    nPQ[nTEL]:=nPQ[nTEL-1]+PQ[TEL-1]-PQ[TEL-2]+iPQ;
   UNTIL TEL=MaxTEL;
   nMaxTEL:=nTEL;

   FNM:=Copy(FileName,3,7);
   FileName:=Concat('b:',FNM,'QTS.DAT');
   WriteLn('Your timestep-filename will be b:',FNM,'QTS.DAT');
   Write('Do you agree ? ');LV;Write(' (Y/N)');HV;Read(Kbd,Ch);WriteLn;
   IF Nee THEN
    BEGIN
     LV;Write('Give your name of the file : ');HV;Read(FileName);WriteLn;
    END;
   Assign(FV,FileName);
   ReWrite(FV);
   File_Text;
   FOR nTEL:=1 TO nMaxTEL DO
   BEGIN
    Af:=nPQ[nTEL];
    Bf:=nt[nTEL];
    WriteLn(FV,nTEL:6,Af:20:3,Bf:20);
   END;
   Close(FV);
   WriteLn('Your filename has received a new number only !');
   WriteLn('If you want to rename : use the procedure Filehandling .');
   PressKey;
  END; (* Discharge *)

'Q' :
  BEGIN
   IF Yes('... are you sure ?') THEN Quit:=True;
   ClrScr;
  END;
 END;  (* Case *)
 HV;
UNTIL Quit=True;
END; (* Time_Step *)


(*=================================================================*)
PROCEDURE Area_Define;
(*=================================================================*)
(*                Conversion to square m                    *)
```

```
VAR  are,sqm,ha,sqkm  :  Real;

BEGIN
 ClrScr;
 WriteLn('HYDROLIN/test/1.00 - Area       :  52510 ha  .....');
 PressKey;
 Quit:=False;
 REPEAT
  Menu_Heading:='AREA DEFINE MENU';
  CASE Menu('Are Hectare M(sq) Km(sq)  Quit') OF

     'A' : BEGIN
           Write('Area              (are) : ');Read(are);WriteLn;
           Area:=are*1E+2;{sq.m}
           WriteLn('Area             (sq.m) : ',AREA);
           Delay(5000);
           Quit:=True;
          END;
     'H' : BEGIN
           Write('Area              (ha) : ');Read(ha);WriteLn;
           Area:=ha*1E+4;{sq.m}
           WriteLn('Area             (sq.m) : ',AREA);
           Delay(5000);
           Quit:=True;
          END;
     'M' : BEGIN
           Write('Area            (sq.m) : ');Read(sqm);WriteLn;
           Area:=sqm;{sq.m}
           WriteLn('Area             (sq.m) : ',AREA);
           Delay(5000);
           Quit:=True;
          END;
     'K' : BEGIN
           Write('Area            (sq.km) : ');Read(sqkm);WriteLn;
           Area:=sqkm*1E+06;{sq.m}
           WriteLn('Area             (sq.m) : ',AREA);
           Delay(5000);
           Quit:=True;
          END;
     'Q' : BEGIN
           IF Yes('... are you sure ?') THEN  Quit:=True;
           ClrScr;
          END;
  END;  (* Case *)
  HV;
  UNTIL Quit=True;
  WriteLn;
  WriteLn('Further characteristics  : not yet implemented.');Delay(3000);
 END; (* Area_define *)


(*=============================================================*)
PROCEDURE Zero_Wiping;
(*=============================================================*)

VAR    WipeTEL : Integer;
```

```
BEGIN
 WriteLn('DATAFILE to apply zero-wiping to .....');
 WriteLn;PressKey;;
 Input_File;

 WriteLn('ZERO-WIPED DATAFILENAME ..... ');
 WriteLn;PressKey;
 File_Name_Num;

 BEGIN
  TEL:=0;
  REPEAT
   TEL:=TEL+1;
  UNTIL PQ[TEL]<>0.0E00;
  WipeTEL:=TEL;
  REPEAT
   TEL:=TEL+1;
  UNTIL PQ[TEL]=0.0E00;
  MaxTEL:=TEL;
 END;
 Assign(FV,FileName);
 Rewrite(FV);
 File_Text;
 PressKey;
 FOR TEL:=WipeTEL TO MaxTEL DO
 BEGIN
  Af:=PQ[TEL];Bf:=t[TEL]-t[WipeTEL-1];
  WriteLn(FV,TEL-WipeTEL+1:6,Af:20:3,Bf:20);
 END;
 Close(FV);
END; (* Zero_Wiping *)


(*========================= PRECIPITATION ============================*)

(*====================================================================*)
PROCEDURE Initial_Loss;
(*====================================================================*)

BEGIN
 Write('ZERO-WIPED PRECIPITATION-DATAFILE to apply initial losses to');
 LV;Write(' (d:XXXXXPWX.DAT)');HV;WriteLn;PressKey;
 Input_File;
 WriteLn('INITIAL-LOSS DATAFILE  :');Delay(3000);WriteLn;WriteLn;
 StrPQ:='INITIAL LOSS Iinit';Strt:='TIME tIinit';

 FOR TEL:=1 TO MaxTEL DO PQ[TEL]:=0.0E00;
 WriteLn('First an initialisation-datafile is written ..... ');
 WriteLn('INITIAL_LOSS_DATAFILE name ..... ');
 WriteLn('Use the PI-option of ..... ');
 PressKey;
 File_Name_Num;
 Assign(FV,FileName);
 ReWrite(FV);
 File_Text;
 FOR TEL:=1 TO MaxTEL DO
 WriteLn(FV,TEL:6,PQ[TEL]:20:3,t[TEL]:20);
```

```
 Close(FV);Delay(3000); (* Initialisation *)

 LV;
 WriteLn('Continue for datafile editing QUIT to use       : HYDREDIT');
 WriteLn('                                  or : TURBO   ');
 WriteLn('the initial_loss-datafile .....');
 Delay(3000);WriteLn;WriteLn;
 WriteLn('F for File_Handling');
 WriteLn('Q for Main-menu    ');HV;
 REPEAT
  Read(Kbd,Ch);
  IF Ch IN ['f','F'] THEN File_Handling;
 UNTIL Ch IN ['f','F','q','Q'];
 PressKey;
END; (* Initial_Loss *)


(*=================================================================*)
PROCEDURE Phi_Index;
(*=================================================================*)

 VAR       Pbrtot : Real;

BEGIN
 Write('ZERO-WIPED PRECIPITATION-DATAFILE to apply phi-index to');
 LV; Write(' (d:XXXXXPWX.DAT)');WriteLn;HV;PressKey;
 Input_File;

 WriteLn('HYDROLIN/test/1.00 - Phi-index  :  2.48 mm/hour .....');
 WriteLn;
 Write('Give estimated phi-index        : ');Read(Apr);WriteLn;Delay(1000);
 WriteLn;
 WriteLn('PHI_INDEX-DATAFILE name ..... ');
 WriteLn('Use the PF option of ..... ');
 PressKey;
 File_Name_Num;
 StrPQ:='PHI-INDEX LOSS Iphi';Strt:='TIME tIphi';
 {
 Write('Phi-index                   : ');Read(Phi);WriteLn;
 TEL:=0;Pbrtot:=0.0E00;t[0]:=0;
 BEGIN
  REPEAT
   TEL:=TEL+1;
   Pbrtot:=Pbrtot+PQ[TEL]*(t[TEL]-t[TEL-1]);
   MaxTEL:=TEL-1;
  UNTIL PQ[TEL]=0.0E00;
 END;
 Apr:=Phi*Pbrtot/t[MaxTEL];
 Write('Phi-index precipitation loss   : ',Apr);
 }
 Assign(FV,FileName);
 Rewrite(FV);
 File_Text;
 FOR TEL:=1 TO MaxTEL DO
 BEGIN
  Iphi[TEL]:=Apr;
  Af:=Iphi[TEL];Bf:=t[TEL];
```

```
  WriteLn(FV,TEL:6,Af:20:3,Bf:20);
 END;
 Close(FV);
 PressKey;
END; (* Phi_index *)

(*=====================================================================*)
PROCEDURE User_Def_Loss;
(*=====================================================================*)


BEGIN
 Write('ZERO-WIPED PRECIPITATION-DATAFILE to apply userdef. losses to');
 LV;Write(' (d:XXXXXPWX.DAT)');HV;WriteLn;PressKey;
 Input_File;
 WriteLn('USERDEF-LOSS DATAFILE  :');Delay(3000);WriteLn;WriteLn;
 StrPQ:='USERDEF LOSS Iusr';Strt:='TIME tIusr';

 FOR TEL:=1 TO MaxTEL DO PQ[TEL]:=0.0E00;
 WriteLn('First an initialisation-datafile is written ..... ');
 WriteLn('USERDEF_LOSS_DATAFILE name ..... ');
 WriteLn('Use the PU-option of ..... ');
 PressKey;
 File_Name_Num;
 Assign(FV,FileName);
 ReWrite(FV);
 File_Text;
 FOR TEL:=1 TO MaxTEL DO
 WriteLn(FV,TEL:6,PQ[TEL]:20:3,t[TEL]:20);
 Close(FV);Delay(3000); (* Initialisation *)

 LV;
 WriteLn('Continue for datafile editing in the Input_Menu : HYDREDIT');
 WriteLn('                    or Quit to use       : TURBO   ');
 WriteLn('the userdef_loss-datafile .....');
 Delay(3000);WriteLn;WriteLn;
 WriteLn('Give                 : I for Input_Menu   ');
 WriteLn('                     : F for File_Handling');
 WriteLn('                     : Q for Main-menu    ');HV;
 REPEAT
  Read(Kbd,Ch);
  IF Ch IN ['i','I'] THEN Input_Menu;
  IF Ch IN ['f','F'] THEN File_Handling;
 UNTIL Ch IN ['i','I','f','F','q','Q'];
 PressKey;
END; (* User_Def_Loss *)

(*=====================================================================*)
PROCEDURE Precip_Loss;
(*=====================================================================*)


BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='PRECIP. LOSS MENU';
```

```
  CASE Menu('Initial Phi_index User_def Quit') OF

    'I' :  BEGIN
           StrPQ:='PRECIP_LOSS Iinit';Strt:='TIME tIinit';
           PQ:=Iinit;t:=tIinit;
           Initial_Loss;
          END;
    'P' :  BEGIN
           StrPQ:='PRECIP_LOSS Iphi';Strt:='TIME tIphi';
           PQ:=Iphi;t:=tIphi;
           Phi_Index;
          END;
    'U' :  BEGIN
           StrPQ:='PRECIP_LOSS Iusr';Strt:='TIME tIusr';
           PQ:=Iusr;t:=tIusr;
           User_Def_Loss;
          END;
    'Q' :  BEGIN
           IF Yes('... are you sure ?') THEN Quit:=True;
           ClrScr;
          END;
  END; (* Case *)
  HV;
  UNTIL Quit=True;
END; (* Precip_Loss *)


(*=====================================================================*)
PROCEDURE Precip_Conversion;
(*=====================================================================*)

BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='PRECIP. CONVERSION';
  CASE Menu('0mm/hour 1mm/half_an_hour 2mm/day 3quit ') OF

(* Conversion Intensity to mm/hour counted ; time in hour.          *)
(* Not implemented l/s.ha= 8.64 mm/day , mm/year                    *)

    '0' :  BEGIN
           MulP:=1.0;
           Quit:=True;
          END;
    '1' :  BEGIN
           MulP:=0.5;
           Quit:=True;
          END;
    '2' :  BEGIN
           Mulp:=1/24;
           Quit:=True;
          END;
    '3' :  BEGIN
           IF Yes('... are you sure ?') THEN Quit:=True;
           ClrScr;
          END;
```

```
  END; (* Case *)
  HV;
UNTIL Quit=True;
END; (* Precip_Conversion *)

(*=====================================================================*)
PROCEDURE Nett_Write;
(*=====================================================================*)

BEGIN
 WriteLn('Give filename to the NETT PRECIPITATION file .....');
 WriteLn('Use now the PN-specification_option of .....');
 Delay(5000);
 File_Name_Num;
 Assign(FV,FileName);
 ReWrite(FV);
 File_Text;
 FOR TEL:=1 TO MaxTEL DO
 BEGIN
 Af:=Inett[TEL];
 Bf:=tInett[TEL];
 WriteLn(FV,TEL:6,Af:20:3,Bf:20);
 END;
Close(FV);
END; (* Nett_Write *)

(*=====================================================================*)
PROCEDURE Nett_Precip;
(*=====================================================================*)

BEGIN
 Write('ZERO-WIPED PRECIPITATION-DATAFILE   ');LV;
 Write(' (d:XXXXXPWX.DAT)');WriteLn; HV;PressKey;
 Input_File;
 Ibr:=PQ;
 tIbr:=t;


 WriteLn('PRECIPITATION_LOSS_DATAFILE(S)  ');LV;
 WriteLn('d:XXXXXPIX.DAT and/or d:XXXXXPFX.DAT and/or d:XXXXXPUX.DAT');
 WriteLn;HV;PressKey;
 Quit:=False;
 REPEAT
  Menu_Heading:='PRECIP_LOSS_DATAFILE';
  CASE Menu('Initial Phi_index User_def Quit') OF

'I' :
  BEGIN
   WriteLn('You should give a filename like d:XXXXXPIX.DAT .....');
   PressKey;
   Input_File;
   Iinit:=PQ;tIinit:=t;
   StrPQ:='NETT PRECIP Inett';Strt:='TIME tInett';
   Precip_Conversion;
   FOR TEL:=1 TO MaxTEL DO
   BEGIN
    IF Ibr[TEL]-Iinit[TEL]<=0.0E00 THEN Inett[TEL]:=0.0E00 ELSE
```

```
    BEGIN
     Inett[TEL]:=Ibr[TEL]-Iinit[TEL];
     Inett[TEL]:=Inett[TEL]*MulP;
     tInett[TEL]:=t[TEL];
    END;
   END;
   Nett_Write;
  END;

'P' :
  BEGIN
   WriteLn('You should give a filename like d:XXXXXPFX.DAT .....');
   PressKey;
   Input_File;
   Iphi:=PQ;tIphi:=t;
   StrPQ:='NETT PRECIP Inett';Strt:='TIME tInett';
   Precip_Conversion;
   FOR TEL:=1 TO MaxTEL DO
   BEGIN
    IF Ibr[TEL]-Iphi[TEL]<=0.0E00 THEN Inett[TEL]:=0.0E00 ELSE
    BEGIN
     Inett[TEL]:=Ibr[TEL]-Iphi[TEL];
     Inett[TEL]:=MulP*Inett[TEL];
     tInett[TEL]:=t[TEL];
    END;
   END;
   Nett_Write;
  END;

'U' :
  BEGIN
   WriteLn('You should give a filename like d:XXXXXPUX.DAT .....');
   PressKey;
   Input_File;
   Iusr:=PQ;tIusr:=t;
   StrPQ:='NETT PRECIP Inett';Strt:='TIME tInett';
   Precip_Conversion;
   FOR TEL:=1 TO MaxTEL DO
   BEGIN
    IF Ibr[TEL]-Iinit[TEL]<=0.0E00 THEN Inett[TEL]:=0.0E00 ELSE
    BEGIN
     Inett[TEL]:=Ibr[TEL]-Iusr[TEL];
     Inett[TEL]:=Inett[TEL]*MulP;
     tInett[TEL]:=t[TEL];
    END;
   END;
   Nett_Write;
   Close(FV);
  END;

'Q' :
  BEGIN
   IF Yes('... are you sure ?') THEN Quit:=True;
    ClrScr;
  END;

 END; (* Case *)
```

```
 HV;
 UNTIL Quit=True;(* Precip_Loss_Menu *)
END; (* Nett_Precip *)


(*=================================================================*)
PROCEDURE Accumul_Nett_Precip;
(*=================================================================*)
(*       Accumulated nett precipitation in cubic m                 *)

BEGIN
 Area_Define;
 WriteLn('WIPED NETT-PRECIPITATION-DATAFILE ..... ');
 PressKey;
 Line:=Line1;
 Input_File;
 Inett:=PQ;
 tInett:=t;
 Pntot:=0.0E00;
 FOR TEL:=1 TO MaxTEL DO
 Pntot:=Pntot+(Inett[TEL]+Inett[TEL-1])*0.5*(tInett[TEL]-tInett[TEL-1]);
 Pntot:=Pntot*1E-3*Area;
 WriteLn('Accumulated nett precipitation : ',Pntot,' cubic m.');
 Write('Do you want to save this in a datafile ?');LV;Write(' (Y/N) ');HV;
 Read(Kbd,Ch);WriteLn;
 IF Ja THEN
 BEGIN
  FNM:=Copy(FileName,3,7);
  FileSpec:='PT';
  FileNumE:=Copy(FileName,10,10);
  FileName:=Concat('b:',FNM,'PT',FileNumE,'.DAT');
  Assign(FV,FileName);
  ReWrite(FV);
  FOR TEL:=1 TO 6 DO WriteLn(FV,Line1);
  WriteLn(FV,'Accumulated Nett Precipitation ',Pntot,' cubic m.');
 END;
 Close(FV);
END; (* Accumul_Nett_Precip *)


(*=================================================================*)
PROCEDURE Precip_Menu;
(*=================================================================*)


BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='PRECIPITATION MENU';
CASE Menu('Bruto Zero_wiping Losses Nett Accum_nett Quit') OF

    'A' :  BEGIN
            Accumul_Nett_Precip;
           END;
    'B' :  BEGIN
            StrPQ:='PRECIPITATION Ibr';Strt:='TIME tIbr';
```

```
          PQ:=Ibr;t:=tIbr;
          Keyboard_Input;
         END;
    'L' :  BEGIN
          Precip_Loss;
         END;
    'N' :  BEGIN
          StrPQ:='PRECIPITATION Inett';Strt:='TIME tInett';
          PQ:=Inett;t:=tInett;
          Nett_Precip;
         END;
    'Z' :  BEGIN
          Zero_Wiping;
         END;
    'Q' :  BEGIN
          IF Yes('... are you sure ?') THEN Quit:=True;
          ClrScr;
         END;
  END; (* Case *)
  HV;
 UNTIL Quit=True;
END; (* Precip_Menu *)

(*END ======================== HYDROL02.INC ======================== *)
```

```
(*************************************************************************
* HYDROL03.INC                                                          *
*                                                                       *
* CONTENTS :                                                            *
*                                                                       *
* MISCELLANEOUS HYDROLIN-PROCEDURES        TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* Depletion; Separ_disch ;Nett_Disch; Disch_Conversion;                 *
* Accumul_Nett_Disch; Disch_Menu; Precip_Disc_Menu.                     *
*************************************************************************)

(*=======================================================================*)
PROCEDURE Depletion;
(*=======================================================================*)

BEGIN
 Scroll;ClrScr;
 WriteLn('BRUTO_DISCHARGE DATAFILE .....................');
 WriteLn('assumed to have one maximum discharge .....');
 WriteLn('to calculate end resp. start of depletion-curves .....');
 PressKey;
 Input_File;
 Qbr:=PQ;tQbr:=t;
 TEL:=1;
 BEGIN
  REPEAT
   TEL:=TEL+1;
   QT1:=PQ[TEL]/PQ[TEL-1];
   QT2:=PQ[TEL+1]/PQ[TEL];
   QT3:=QT1-QT2;
  UNTIL ((QT2>=0.0E00) OR (ABS(QT3)<=2E-2));
  Qs:=PQ[TEL+1];tQs:=t[TEL+1];
 END; (* Separation starting point *)

 TEL:=1;
 REPEAT
  TEL:=TEL+1;
 UNTIL (((PQ[TEL]-PQ[TEL-1])/(t[TEL]-t[TEL-1]))>=0.0E00) AND
      (((PQ[TEL+1]-PQ[TEL])/(t[TEL+1]-t[TEL]))< 0.0E00);
 LV;WriteLn('                    *** DISCHARGE SEPARATION  ***');HV;
 WriteLn;WriteLn;WriteLn('Maximum discharge              : ',PQ[TEL]:20:3);
 WriteLn('           time                : ',t[TEL]:20);
(* Maximum discharge *)
 WriteLn;
TEL:=TEL-1;
 BEGIN
  REPEAT
   TEL:=TEL+1;
   QT1:=PQ[TEL]/PQ[TEL-1];
   QT2:=PQ[TEL+1]/PQ[TEL];
   QT3:=QT1-QT2;
  UNTIL ABS(QT3)<=2E-2;
  Qd:=PQ[TEL-1];tQd:=t[TEL-1];
 END; (* Depletion starting point  *)

 WriteLn('Start separation curve at time     : ',tQs:20);
 WriteLn('               and discharge     : ',Qs:20:3);WriteLn;Delay(3000);
```

```
 WriteLn('Start depletion  curve at time     : ',tQd:20);
 WriteLn('                 and discharge     : ',Qd:20:3);WriteLn;
 PressKey;
END; (* Depletion *)


(*=====================================================================*)
PROCEDURE Separ_Disch;
(*=====================================================================*)

BEGIN
 Depletion;
 Quit:=False;
 REPEAT
  Menu_Heading:='DISCH. SEPARATION';
  CASE Menu('Constant Linear User_def Quit') OF

'C' :
   BEGIN
    WriteLn('Separation assumed to be constant with time .....');
    WriteLn;
    WriteLn('Use now the QC-filenamespecification_option .....');
    StrPQ:='CONSTANT SEPAR. Qc';Strt:='TIME tQc';
    PressKey;
    File_Name_Num;
    Assign(FV,FileName);
    ReWrite(FV);
    File_Text;
    TEL:=0;
    REPEAT
     TEL:=TEL+1;
     Qsc[TEL]:=PQ[TEL];
    UNTIL PQ[TEL]=Qs;   (* First depletion curve   *)
    REPEAT
     IF PQ[TEL]>=Qs THEN
     Qsc[TEL]:=Qs;
     TEL:=TEL+1;
    UNTIL PQ[TEL]<=Qs; (* Constant separationline *)
    TEL:=TEL-1;
    REPEAT
     TEL:=TEL+1;
     Qsc[TEL]:=PQ[TEL];
    UNTIL TEL=MaxTEL;
    FOR TEL:=1 TO MaxTEL DO
    WriteLn(FV,TEL:6,Qsc[TEL]:20:3,t[TEL]:20);
    Close(FV);
    PressKey;
   END; (* Linear constant *)

'L' :
   BEGIN
    WriteLn('Separation assumed to be linear with time .....');
    WriteLn;
    WriteLn('Use now the QL-filenamespecification_option .....');
    StrPQ:='LINEAR SEPAR. Ql';Strt:='TIME tQl';
    PressKey;
    File_Name_Num;
```

```
    Assign(FV,FileName);
    ReWrite(FV);
    File_Text;
    TEL:=0;
    REPEAT
     TEL:=TEL+1;
     Qsl[TEL]:=PQ[TEL];
    UNTIL PQ[TEL]=Qs;   (* First depletion curve   *)
    TEL:=TEL-1;
    REPEAT
     TEL:=TEL+1;
     Qsl[TEL]:=Qs+((t[TEL]-tQs)*(Qd-Qs)/(tQd-tQs));
    UNTIL PQ[TEL]=Qd;   (* Linear separation line  *)
    IF TEL<MaxTEL THEN
    REPEAT
     TEL:=TEL+1;
     Qsl[TEL]:=PQ[TEL];
    UNTIL t[TEL]=MaxTEL; (* Last depletion curve    *)
    FOR TEL:=1 TO MaxTEL DO
    WriteLn(FV,TEL:6,Qsl[TEL]:20:3,t[TEL]:20);
    Close(FV);
    PressKey;
   END; (* Linear up or down *)

'U' :
   BEGIN
    WriteLn('Separation assumed to be user defined .....');
    WriteLn;
    WriteLn('Use now the QU-filenamespecification_option .....');
    StrPQ:='USER DEF SEPAR. Qu';Strt:='TIME tQu';
    PressKey;
    File_Name_Num;
    Assign(FV,FileName);
    ReWrite(FV);
    File_Text;
    FOR TEL:=1 TO MaxTEL DO
    BEGIN
     IF t[TEL]<=tQs THEN Qsu[TEL]:=PQ[TEL];
     IF ((t[TEL]>tQs) AND (t[TEL]<tQd)) THEN Qsu[TEL]:=0.0E00;
     IF t[TEL]>=tQd THEN Qsu[TEL]:=PQ[TEL];
    END;
    FOR TEL:=1 TO MaxTEL DO
    WriteLn(FV,TEL:6,Qsu[TEL]:20:3,t[TEL]:20);
    Close(FV);
    Delay(3000);
    WriteLn('In ',FileName,' :');
    WriteLn('you are to give sequential discharge_separation_data .....');
    WriteLn('in a certain time-interval using an EDITER .....');
    Delay(1000);
    WriteLn('Take option QUIT and return to the MAIN MENU .....');
    PressKey;
   END; (* User-def *)

'Q' :
   BEGIN
    IF Yes('... are you sure ?') THEN Quit:=True;
    ClrScr;
```

```
   END;
  END; (* Case *)
  HV;
 UNTIL Quit=True;
END; (* Separ_disch *)


(*=================================================================*)
PROCEDURE Disch_Conversion;
(*=================================================================*)

BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='DISCHARGE CONVERSION';
  CASE Menu('0cub.m/sec 1cub.m/min 2m/sec 3quit') OF

(* Discharge-conversion to cubic m/s; time in hour                *)
(* Not implemented l/s , 10E6 cub.m/day or month or year , cf/s      *)

    '0' :  BEGIN
           MulQ:=1;
           Quit:=True;
          END;
    '1' :  BEGIN
           MulQ:=1/60;
           Quit:=True;
          END;
    '2' :  BEGIN
           Write('Give the cross-section in sq m of the river : ');Read(R);
           WriteLn;
           MulQ:=R;
           Quit:=True;
          END;
    '3' :  BEGIN
           IF Yes('... are you sure ?') THEN Quit:=True;
           ClrScr;
          END;
  END; (* Case *)
  HV;
 UNTIL Quit=True;
END; (* Discharge_Conversion *)


(*=================================================================*)
PROCEDURE Nett_Disch;
(*=================================================================*)

BEGIN
 WriteLn('BRUTO_DISCHARGE-DATAFILE ..... ');
 Delay(3000);
 Input_File;
 Qbr:=PQ;
 tQbr:=t;

 WriteLn('DISCHARGE-SEPARATION_DATAFILE ..... ');
```

```
 WriteLn('This must be in conjunction with the bruto_discharge-datafile');
 Delay(10000);
 Input_File;
 Qbs:=PQ;
 tQbs:=t;

 Disch_Conversion;
 FOR TEL:=1 TO MaxTEL DO
 BEGIN
  StrPQ:='NETT DISCHARGE Qnett';Strt:='TIME tQnett';
  Qnett[TEL]:=Qbr[TEL]-Qbs[TEL];
  Qnett[TEL]:=MulQ*Qnett[TEL];
  tQnett[TEL]:=t[TEL];
 END;
 WriteLn('Give filename to the NETT DISCHARGE file .....');
 WriteLn('Use now the QN-filenamespecification_option .....');
 Delay(5000);
 File_Name_Num;
 Assign(FV,FileName);
 ReWrite(FV);
 File_Text;
 FOR TEL:=1 TO MaxTEL DO
 BEGIN
  Af:=Qnett[TEL];
  Bf:=tQnett[TEL];
  WriteLn(FV,TEL:6,Af:20:3,Bf:20);
 END;
 Close(FV);
END; (* Nett_Disch *)



(*=====================================================================*)
PROCEDURE Accumul_Nett_Disch;
(*=====================================================================*)
(*       Accumulated Nett Discharge                    *)

BEGIN
 WriteLn('WIPED! NETT_DISCHARGE-DATAFILE ..... ');
 Delay(3000);
 Line:=Line2;
 Input_File;
 Qnett:=PQ;
 tQnett:=t;
 Qntot:=0.0E00;
 FOR TEL:=1 TO MaxTEL DO
 Qntot:=Qntot+(Qnett[TEL]+Qnett[TEL-1])*0.5*(tQnett[TEL]-tQnett[TEL-1]);
 Write('Do you want to save this in a datafile ?');LV;Write(' (Y/N) ');HV;
 Read(Kbd,Ch);WriteLn;
 IF Ja THEN
 BEGIN
  FNM:=Copy(FileName,3,7);
  FileSpec:='QT';
  FileNumE:=Copy(FileName,10,10);
  FileName:=Concat('b:',FNM,'QT',FileNumE,'.DAT');
  Assign(FV,FileName);
  ReWrite(FV);
```

```
  FOR TEL:=1 TO 6 DO WriteLn(FV,Line2);
  WriteLn(FV,'Accumulated nett discharge ',Qntot,' in cubic m.');
 END;
 Close(FV);
END; (* Accumul_Nett_Disch *)


(*=================================================================*)
PROCEDURE Disch_Menu;
(*=================================================================*)


BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='DISCHARGE MENU';
  CASE Menu('Bruto Separ Reservoir Nett Zero_wiping Accum_nett Quit') OF

    'A' :  BEGIN
           Accumul_Nett_Disch;
          END;
    'B' :  BEGIN
           StrPQ:='DISCHARGE Qbr';Strt:='TIME tQbr';
           PQ:=Qbr;t:=tQbr;
           Keyboard_Input;
          END;
    'R' :  BEGIN
           Bell;WriteLn('Reservoir in/outlet not implemented !');
           Delay(1000);
           WriteLn('With an EDITER you will manage in an PI-file.');
           Delay(3000);
          END;
    'S' :  BEGIN
           StrPQ:='DISCHARGE Qbs';Strt:='TIME tQbs';
           Separ_disch;
          END;
    'N' :  BEGIN
           StrPQ:='DISCHARGE Qnett';Strt:='TIME tQnett';
           Nett_Disch;
          END;
    'Z' :  BEGIN
           Zero_Wiping;
          END;
    'Q' :  BEGIN
           IF Yes('... are you sure ?') THEN Quit:=True;
           ClrScr;
          END;
  END; (* Case *)
  HV;
 UNTIL Quit=True;
END; (* Disch_Menu *)


(*=================================================================*)
PROCEDURE Precip_Disch;
(*=================================================================*)
```

```
BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='PRECIP./DISCH. MENU';
  CASE Menu('Precipition Discharge Compare Timestep Quit') OF

    'P' :  BEGIN
           Precip_Menu;
          END;
    'D' :  BEGIN
           Disch_Menu;
          END;
    'C' :  BEGIN
           WriteLn('Do you want to compare ');
           Write('accumulated precipitation and discharge ?  ');
           LV;Write('(Y/N)');HV;Read(Kbd,Ch);WriteLn;
           IF Ja THEN
           BEGIN
            Scroll;ClrScr;
            Accumul_Nett_Precip;
            WriteLn('Accumulated nett precipitation  : ',Pntot,' cubic m.');
            WriteLn;
            Accumul_Nett_Disch;
            WriteLn('Accumulated nett discharge       : ',Qntot,' cubic m.');
            WriteLn;
            WriteLn('Pntot-Qntot                      = ',Pntot-Qntot,' cubic m.');
            WriteLn('If you are satisfied you can QUIT and start GRAPHICS ');
            WriteLn('or derive a Unit Hydrograph .....................');
            WriteLn('Otherwise start again Precipitation/Discharge operations');
            PressKey;
           END; (* Compare *)
           Bell;WriteLn('Use another option ...');Delay(5000);ClrScr;
          END;
    'T' :  BEGIN
           Time_Step;
          END;
    'Q' :  BEGIN
           IF Yes('... are you sure ?') THEN Quit:=True;
           ClrScr;
          END;
  END; (* Case *)
  HV;
 UNTIL Quit=True;
END; (* Precip_Disch *)

(*======================== DISCHARGE ==============================*)

(*END ==================== HYDROL03.INC ============================*)
```

```
(*************************************************************************
* HYDROL04.INC                                                          *
*                                                                       *
* CONTENTS :                                                            *
*                                                                       *
* MISCELLANEOUS HYDROLIN-PROCEDURES        TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* Editer_Menu; Unit_Hydrograph; Theory_Menu; Main_Menu.                 *
*************************************************************************)

(*=======================================================================*)
PROCEDURE Editer_Menu;
(*=======================================================================*)

BEGIN
WriteLn('Required : all editers/filehandlers in drive A !');Delay(5000);
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='EDITER MENU';
  CASE Menu('Edlin File_mngr Hydr_edit Norton Turbo User_def Wordstar Quit') OF

     'E' : BEGIN
           FileName:='a:EDLIN.COM';
           IF NOT Exist(FileName) THEN
           BEGIN
            Bell;
            WriteLn(FileName,' does not exist !');Delay(1000);
           END ELSE
           BEGIN
            WriteLn('Input command : ',FileName);Delay(3000);
            Assign(FV,'a:COMM211.CHN');
            Chain(FV);
           END;
           ClrScr;
          END;
     'H' : BEGIN
           Hydr_Edit;
          END;
     'N' : BEGIN
           FileName:='a:NE.EXE';
           IF NOT Exist(FileName) THEN
           BEGIN
            Bell;
            WriteLn(FileName,' does not exist !');Delay(1000);
           END ELSE
           BEGIN
            WriteLn('Input command : ',FileName);Delay(3000);
            Assign(FV,'a:COMM211.CHN');
            Chain(FV);
           END;
           ClrScr;
          END;
     'T' : BEGIN
           FileName:='a:TURBO.COM';
           IF NOT Exist(FileName) THEN
           BEGIN
```

```
        Bell;
        WriteLn(FileName,' does not exist !');Delay(1000);
       END ELSE
       BEGIN
        WriteLn('Input command : ',FileName);Delay(3000);
        Assign(FV,'a:COMM211.CHN');
        Chain(FV);
       END;
       ClrScr;
      END;
'U' : BEGIN
       LV;Write('Give filename  (d:filename.typ) :  ');
       HV;Read(FileName);WriteLn;
       IF NOT Exist(FileName) THEN
       BEGIN
        Bell;
        WriteLn(FileName,' does not exist !');Delay(1000);
       END ELSE
       BEGIN
        WriteLn('Input command : ',FileName);Delay(3000);
        Assign(FV,'a:COMM211.CHN');
        Chain(FV);
       END;
       ClrScr;
      END;
'W' : BEGIN
       FileName:='a:WS.COM';
       IF NOT Exist(FileName) THEN
       BEGIN
        Bell;
        WriteLn(FileName,' does not exist !');Delay(1000);
       END ELSE
       BEGIN
        WriteLn('Input command : ',FileName);Delay(3000);
        Assign(FV,'a:COMM211.CHN');
        Chain(FV);
       END;
       ClrScr;;
      END;
'F' : BEGIN
       FileName:='a:PFM.COM';
       IF NOT Exist(FileName) THEN
       BEGIN
        FN:='a:FM.COM';
        IF NOT Exist(FileName) THEN
        BEGIN
         Bell;
         WriteLn(FileName,'or ',FN,' not present on A: !');Delay(1000);
        END ELSE
        BEGIN
         WriteLn('Input command : ',FileName);Delay(3000);
         Assign(FV,'a:COMM211.CHN');
         Chain(FV);
         END;
        END;
        BEGIN
        BEGIN
```

```
            WriteLn('Input command : ',FileName);Delay(3000);
            Assign(FV,'a:COMM211.CHN');
            Chain(FV);
           END;
          END;
          ClrScr;
         END;
     'Q' : BEGIN
          IF Yes('... are you sure ?') THEN Quit:=True;
          ClrScr;
         END;
  END;  (* Case *)
  HV;
 UNTIL Quit=True;
END; (* Editer_Menu *)


(*=====================================================================*)
PROCEDURE Model_Menu;
(*=====================================================================*)

BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='MODEL MENU';
  CASE Menu('Nash_cascade Stochast Tank_model Unit_hydrograph Quit') OF

    'N' :  BEGIN
          {Nash_cascade;}
           Bell;WriteLn('Nash Cascade not yet implemented !');
           Delay(1000);ClrScr;
          END;
    'S' :  BEGIN
          {Stochastic model}
           Bell;WriteLn('Stochastic models not yet implemented !');
           Delay(1000);ClrScr;
          END;
    'T' :  BEGIN
          {Tank_model;}
           Bell;WriteLn('Tank Model not yet implemented !');
           Delay(1000);ClrScr;
          END;
    'U' :  BEGIN
           WriteLn('Derivation of the Unit Hydrograph is possible only with :');
           WriteLn;WriteLn('1. Zero-wiped datafiles;');
           WriteLn('2. Files with nett data;');
           WriteLn('3. Same timestep for precipitation and discharge data;');
           WriteLn('4. Datafiles which are compared : accumulated nett.');
           WriteLn;
           IF Yes('... are you sure these conditions are fulfilled ?  (Y/N)') THEN
           BEGIN
            Assign(FV,'b:HYDROCAL.CHN');
            Chain(FV);
           END ELSE
           WriteLn('Choose again the precip_disch-menu ...');PressKey;
          END;
```

```
    'Q' :  BEGIN
           IF Yes('... are you sure ?') THEN Quit:=True;
           ClrScr;
          END;
  END; (* Case *)
  HV;
 UNTIL Quit=True;
END; (* Model_Menu *)


(*=================================================================*)
PROCEDURE Main_Menu;
(*=================================================================*)

BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='MAIN MENU';
  CASE Menu('Comm Dir Edit File_hnd Graphics Help Input Model Quit') OF

     'C' : BEGIN
           Assign(FV,'a:COMM211.CHN');
           Chain(FV);
          END;
     'D' : BEGIN
           Assign(FV,'a:MSDOSDIR.CHN');
           Chain(FV);
          (*alternative :                  *)
          (*WriteLn('DOS-Command is PDIR ...');*)
          (*Assign(FV,'a:COMM211.CHN');       *)
          (*Chain(FV);                       *)
           END;
     'E' : BEGIN
           Editer_Menu;
          END;
     'F' : BEGIN
           File_Handling;
          END;
     'G' : BEGIN
           Assign(FV,'a:HYDROGR.CHN');
           Chain(FV);
          END;
     'H' : BEGIN
           Help_Menu;
          END;
     'I' : BEGIN
           Precip_Disch;
          END;
     'M' : BEGIN
           Model_Menu;
          END;
     'Q' : BEGIN
           IF Yes('... are you sure ?') THEN Quit:=True;
           ClrScr;
          END;
  END; (* Case *)
```

```
  HV;
 UNTIL Quit=True;
END; (* Main-Menu *)

(*END ======================== HYDROL04.INC ============================ *)
```

```
(*************************************************************************
* MSDOSDIR.PAS                                                          *
*                                                                       *
* CONTENTS                                                              *
*                                                                       *
* MISCELLANEOUS MSDOS DIRECTORY-PROCEDURES  TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* Beep; Display; DecodeDTA; Display_Data; Lees_Volume_Label;            *
* Bladhoofd; Directory.                                                 *
*************************************************************************)

CONST
  Papierlengte      = 72;

TYPE
  Str2          = STRING[2];
  Str3          = STRING[3];
  Str4          = STRING[4];
  Str5          = STRING[5];
  Str9          = STRING[9];
  Str11         = STRING[11];
  Str12         = STRING[12];
  Str80         = STRING[80];
  Char80Arr     = ARRAY [1..80] OF Char;

  Regpack       = RECORD
                  AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
                  END;

  Filegegevens  = RECORD
                  Attribute : Integer;
                  Tijd      : Str5;
                  Datum     : Str11;
                  Lengte    : Real;
                  FileNaam  : Str12;
                  END;


VAR
  Regs          : Regpack;
  Filedata      : Filegegevens;
  DTA           : ARRAY [1..43] OF Byte;
  OldDTA_Seg,
  OldDTA_Ofs,
  Error,I,Blad,
  Aantal_Files,
  Regel         : Integer;
  Path          : Char80Arr;
  AsciizString,
  InvoerString  : Str80;
  Drive         : Char;
  UsedBytes     : Real;
  AttribuutTekst: Str80;
  VolumeLabel   : Str12;
  VolumeTijd    : Str5;
  VolumeDatum   : Str11;
  Space         : Real;
```

```
  Fout          : Boolean;
  Ch            : Char;
  FV            : Text;

{$I b:MSDOSLIB.INC }


(*=====================================================================*)
{PROCEDURE Beep;                                    }
(*=====================================================================*)

{BEGIN                      }
{  Sound(1200);Delay(300);NoSound;}
{  Sound(1600);Delay(500);NoSound;}
{  Sound(1200);Delay(300);NoSound;}
{END;                       }


(*=====================================================================*)
PROCEDURE Display(X,Y(*,KleurIn,KleurUit*): Integer; Txt : Str80);
(*=====================================================================*)

BEGIN
 {TextColor(KleurIn);                               }
 {IF KleurIn=8 THEN TextBackground(15) ELSE TextBackground(8);}
  GotoXY(X,Y);
  Write(Txt);
 {IF KleurIn <> KleurUit THEN                       }
 {BEGIN                                  }
 {  TextColor(KleurUit);                            }
 {  TextBackground(8);                              }
 {END;                                  }
END; {Display}


(*=====================================================================*)
PROCEDURE DecodeDTA( VAR Filedata : Filegegevens );
(*=====================================================================*)

TYPE
  Maanden       = ARRAY [1..12] OF Str3;

CONST
  Maand : Maanden  =
    ('Jan','Feb,','Mrt','Apr','Mei','Jun','Jul','Aug','Sep','Okt','Nov','Dec');

VAR
  Uur,Minuut,Dag   : Str2;
  Jaar          : Str4;
  I,Spaties,Punt,
  Plek          : Integer;
  M             : Byte;

BEGIN {DTA}
  WITH Filedata DO
  BEGIN
    Attribute := DTA [22];
```

```
    AttribuutTekst:='';
    IF Attribute AND 1 = 1 THEN AttribuutTekst:='Read-Only ';
    IF Attribute AND 2 = 2 THEN AttribuutTekst:=AttribuutTekst+'Hidden ';
    IF Attribute AND 4 = 4 THEN AttribuutTekst:=AttribuutTekst+'System ';
    IF Attribute AND 32 = 32 THEN AttribuutTekst:=AttribuutTekst+'Archive ';

    Str(((DTA [24] Shl 3) AND $3F)+((DTA [23] Shr 5) AND $07),Minuut);
    IF Length(Minuut)=1 THEN Insert('0',Minuut,1);
    Str((DTA [24] Shr 3),Uur);
    IF Length(Uur)=1 THEN Insert('0',Uur,1);
    Tijd := Uur + ':' + Minuut;

    Str(DTA [25] AND $1F,Dag);
    IF (Length(Dag)<2) AND (Attribute<>40) THEN Insert(' ',Dag,1);
    M:=((DTA [26] Shl 3)AND $0F)+((DTA [25] Shr 5) AND $07);
    Str(((DTA [26] Shr 1)+1980),Jaar);
    Datum:=Dag + ' ' + Maand [M]+ ' ' + Jaar;

    Lengte := DTA [27]+(256.0 * DTA [28])+(65536.0 * DTA [29]);

    I := 1;
    FillChar(FileNaam,SizeOf(FileNaam),' ');
    REPEAT
      FileNaam [I] := Chr(DTA [I+30]);
      I := I+1;
    UNTIL   (NOT (FileNaam [I-1]  IN [' '..'~']));
    IF NOT (Attribute IN [8,40,16]) THEN
    BEGIN
      FileNaam [0] :=Chr(12);
      Spaties := 14-I;
      Punt := Pos('.',FileNaam);
      Delete(FileNaam,Punt,1);
      FOR I := Punt TO Punt+Spaties DO Insert(' ',FileNaam,I);
    END ELSE
    Filenaam[0]:=Chr(I-1);
    IF Attribute=40 THEN Delete(FileNaam,9,1);
    IF Attribute=16 THEN
    BEGIN
      IF Length(FileNaam)<8 THEN
      BEGIN
        Plek:=Length(FileNaam);
        FOR I:=8-Length(FileNaam) TO 8 DO Insert(' ',FileNaam,Plek);
      END;
    END;
  END;{FileData}
END;{DecodeDTA}


(*=================================================================*)
PROCEDURE Display_Data(Filedata:Filegegevens);
(*=================================================================*)

BEGIN
  DecodeDTA(Filedata);
  WITH FileData DO
  BEGIN
    IF Attribute=16 THEN
```

```
    BEGIN
      Write(Filenaam);
      Writeln('<DIR>         ',Datum,'  ',Tijd,'  ',AttribuutTekst);
    END ELSE
    BEGIN
      UsedBytes:=UsedBytes+Lengte;
    Writeln
      (Filenaam,'  ',Lengte:6:0,'  ',Datum,'  ',Tijd,'  ',AttribuutTekst);
    END;
   END;
END;{ Display_Data}


(*=====================================================================*)
PROCEDURE Lees_Volumelabel;
(*=====================================================================*)

BEGIN
  FindFirst(Path,8,Error);
  IF Error=0 THEN
  BEGIN
    DecodeDTA(FileData);
    WITH FileData DO
    BEGIN
      Volumelabel:=FileNaam;
      VolumeDatum:=Datum;
      VolumeTijd:=Tijd;
    END;
   END ELSE
   BEGIN
     VolumeLabel:='';
     VolumeDatum:='';
     VolumeTijd:='';
   END;
END;{ Lees_VolumeLabel }


(*=====================================================================*)
PROCEDURE Bladhoofd;
(*=====================================================================*)

BEGIN
  Writeln('VolumeLabel : ',VolumeLabel,'                    ',Datum);
  Write('Directory   : ');
  I:=1;
  WHILE Path[I] <> '*' DO
  BEGIN
    Write(Path[I]);
    I:=I+1;
  END;

 Writeln;
  Writeln('Created on         : ',VolumeDatum,'  ',VolumeTijd);

  FOR I:=1 TO 73 DO
 BEGIN
  Write('=');
```

```
 END;
 Writeln;
END; {BladHoofd }


(*======================== MSDOS ==================================*)
(*                  MAIN PROGRAM                                   *)
(*======================== = = = ==================================*)

BEGIN
  GetDTA(OldDTA_Seg,OldDTA_Ofs);
  SetDTA(Seg(DTA),Ofs(DTA));
  ClrScr;
  FillChar(DTA,43,0); {init DTA}

  REPEAT
    ClrScr;
    Display(20,1,{9,2,}'///  M  S  D  O  S    D  I  R  E  C  T  O  R  Y  ///');
    LowVideo;Display(1,2,{2,2,}'Give :');
    Display(17,4,{10,10,}'[ Drive: ][ \PATH ][ \PATH ][ etc. ]');
    Display(17,5,{10,10,}' i.e. a: or b:Name\Name\ etc.');
    UsedBytes:=0;
    Aantal_Files := 0;
    InvoerString:='';
    FillChar(Path,80,Chr(0));
    Blad:=1;
    GotoXY(35,8);ClrEol;
    Display(17,8,{2,10,}'Directory path : ');HighVideo;
    Read(InvoerString);Writeln;
    IF Length(InvoerString)=0 THEN
    BEGIN
      CurrentDisk(Drive);
      AsciizString:=Drive+':\*.*';
      Display(34,8,{2,10,}Drive+':\*.*');
    END ELSE
    BEGIN
      Invoerstring[1]:=Upcase(Invoerstring[1]);
      Drive:=Invoerstring[1];
      IF NOT (Drive in ['A'..'C']) THEN CurrentDisk(Drive);
      AsciizString:=InvoerString+'\*.*';
      Display(33,8,{2,10,}Drive+':\*.*');
    END;

    FOR I:=1 TO Length(AsciizString) DO Path[I]:=Upcase(AsciizString[I]);

    Display(1,10,{2,10,}'MSDOS DIRECTORY :');
    I:=1;
    WHILE Path[I] <> '*' DO
    BEGIN
      Write(Path[I]);
      I:=I+1;
    END;

    Display(1,12,{10,2,}'D');LowVideo;Write('isplay MSDOS Directory');HighVideo;
    Display(1,13,{10,2,}'G');LowVideo;Write('ive path again');HighVideo;
    Display(1,14,{10,2,}'S');LowVideo;Write('top procedure');
    Display(1,16,{10,10,}'Your choice :  ');HighVideo;
```

```
  REPEAT
   Read(Kbd,Ch);
  UNTIL Upcase(Ch) in ['D','G','S'];Writeln;
  Ch:=Upcase(Ch);
  IF Ch='D' THEN
  BEGIN
    Writeln;Writeln;
    ClrScr;
    Lees_VolumeLabel;
    FindFirst(Path,22,Error);
    Aantal_Files:=0;
    IF Error=0 THEN
    BEGIN
      WHILE Error=0 DO
      BEGIN
        BladHoofd;
        Regel:=10;
        WHILE (Error = 0) AND (Regel < PapierLengte-5) DO
        BEGIN
          Display_Data(FileData);
          Regel:=Regel+1;
          Aantal_Files := Aantal_Files + 1;
          FindNext(Error);
        END;
        IF Error=18 THEN
        BEGIN
          Writeln;
          Writeln(UsedBytes:6:0,' bytes in ',Aantal_Files,' files.');
          I:=Ord(Drive)-64;
          IF NOT I IN [1..3] THEN I:=0;
          GetDiskFreeSpace(I,Space,Fout);
          IF NOT Fout THEN
          BEGIN
            Writeln;Writeln(Space:6:0,' bytes free.');
          END;
        END;
        Writeln; Writeln;
        Write(^G,'press <RETURN> ....');Read(Kbd,Ch);
      END;
    END ELSE
    BEGIN
      Display(5,22,{25,10,}'* Bad input * ');
      Writeln('Path not found !');
      Writeln('Try again......');
     {Beep;}
      Delay(4000);
    END;
  END;
  FOR I:=10 TO 25 DO
  BEGIN
    GotoXY(1,I);
    ClrEol;
  END;
UNTIL Ch='S';
SetDTA(OldDTA_Seg,OldDTA_Ofs);
Assign(FV,'a:HYDRODAT.COM');
Execute(FV);
```

```
END.

(*END ====================== MSDOSDIR.PAS ============================*)
```

```
(*************************************************************************
* MSDOSLIB.INC                                                          *
*                                                                       *
* CONTENTS                                                              *
*                                                                       *
* MISCELLANEOUS MSDOS DIRECTORY-PROCEDURES  TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* SetCurSor; CurrentDisk; SetDTA; Datum; GetDTA; GetDiskFreeSpace     *
* FindFirst; FindNext.                                                  *
*************************************************************************)

(*=======================================================================*)
PROCEDURE SetCursor(CursorSize:Integer);
(*=======================================================================*)

VAR
  Regs : RECORD
           AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
         END;

BEGIN
  WITH Regs DO
   BEGIN
     AX:=$0100;
     CX:=CursorSize;
  END;
  Intr($10,Regs);
END;{ SetCursor }


(*=======================================================================*)
PROCEDURE CurrentDisk(VAR Drive:Char);
(*=======================================================================*)

CONST
  Disk : ARRAY [0..2] OF Char = 'ABC';

VAR
  Regs : RECORD
           AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
         END;

BEGIN
  WITH Regs DO
  BEGIN
    AX := $1900;
    MSDOS(Regs);
    Drive:=Disk[(AX AND $00FF)];
  END;
END; { CurrentDisk }


(*=======================================================================*)
PROCEDURE SetDTA(Segment,Offset:Integer);
(*=======================================================================*)

VAR
```

```
  Regs : RECORD
           AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
         END;
BEGIN
  Regs.AX := $1A00;
  Regs.DS := Segment;
  Regs.DX := Offset;
  MSDOS( Regs );
END; { SetDTA  }


(*=======================================================================*)
FUNCTION Datum:Str11;
(*=======================================================================*)

TYPE
  Str3      = String [3];
  Str4      = String [4];

CONST
  Maand     : ARRAY [1..12] OF Str3  = ('Jan','Feb','Mrt','Apr','Mei','Jun',
                                        'Jul','Aug','Sep','Okt','Nov','Dec');
VAR
  Regs      : RECORD
                AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
              END;
  Dag,Jaar  : Str4;

BEGIN
  WITH Regs DO
  BEGIN
    AX:=$2A00;
    MSDOS(Regs);
    Str(Lo(DX),Dag);
    Str(CX,Jaar);
    Datum:=Dag+'-'+Maand[Hi(DX)]+'-'+Jaar;
  END;
END; { Datum }

(*=======================================================================*)
PROCEDURE GetDTA(VAR Segment,Offset:Integer);
(*=======================================================================*)

VAR
  Regs : RECORD
           AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
         END;
BEGIN
  Regs.AX:=$2F00;
  MSDOS(Regs);
  Segment:=Regs.ES;
  Offset:=Regs.BX;
END; { GetDTA }

(*=======================================================================*)
PROCEDURE GetDiskFreeSpace(Drive:Integer; VAR Space:Real; VAR Error:Boolean);
(*=======================================================================*)
```

```
VAR
  Regs : RECORD
           AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
         END;

BEGIN
  WITH Regs DO
  BEGIN
    AX:=$3600;
    DX:=Drive;
    MSDOS(Regs);
    IF AX=$FFFF THEN Error:=True ELSE
    BEGIN
      Error:=False;
      Space:=AX*CX;
      Space:=Space*BX;
    END;
  END;
END; { GetDiskFreeSpace }


(*=====================================================================*)
PROCEDURE FindFirst(Mask:Char80arr; Attribute:Integer; Var Error:Integer);
(*=====================================================================*)

VAR
  Regs : RECORD
           AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
         END;
BEGIN
  Error:=0;
  Regs.AX:=$4E00;
  Regs.DS:=Seg( Mask );
  Regs.DX:=Ofs( Mask );
  Regs.CX:=Attribute;
  MSDOS(Regs);
  Error:=Regs.AX AND $FF;
END; { FindFirst }

(*=====================================================================*)
PROCEDURE FindNext( VAR Error:Integer);
(*=====================================================================*)

VAR
 Regs : RECORD
           AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
         END;
BEGIN
  Error:=0;
  Regs.AX:=$4F00;
  MSDOS(Regs);
  Error:=Regs.AX AND $FF;
END; { FindNext }

(* END ================ DOS-FUNCTIONS LIBRARY  ====================*)
```

```
echo off
dir %1 | sort > lpt1:
```

```
PROGRAM UsePointers; (* Example program / M.J. Vos *)
                (* HYDROLIN 1.00 extension    *)

CONST N=2000;

TYPE OneDim    = ARRAY[1..N] OF Real;
     PtrOneDim = ^OneDim;
     TwoDim    = ARRAY[1..N] OF PtrOneDim;
     PtrTwoDim = ^TwoDim;

VAR  A         : PtrTwoDim;
     I,J       : Integer;

BEGIN
 New(A);
 FOR I:=1 TO N DO New(A^[I]);
 FOR I:=1 TO N
 BEGIN
  FOR J:=1 TO N DO
  BEGIN
   A^[I]^[J]:=1.0;
   WriteLn('i , j, A[I,J] = ',I:6,J:6,'':6,A^[I]^[J]:12);
  END;
 END;
END.

(* END ========================== POINTERS.PAS ========================== *)
```

```
program SortExampleOne {Customer File};

type
  CustRec = record
              Number: integer;
              Name:   string[30];
              Addr:   string[20];
              City:   string[12];
              State:  string[3];
              Zip:    string[5];
            end;

var
  CustFile: file of CustRec;
  Customer: CustRec;

{$ISORT.BOX}

procedure Inp; {this procedure is forward declared in SORT.BOX}
begin
  repeat
    Read(CustFile,Customer);
    SortRelease(Customer);
  until EOF(CustFIle);
end;

function Less; {this boolean function has two parameters, X and Y}
               {and is forward declared in SORT.BOX}
var
  FirstCust:  CustRec absolute X;
  SecondCust: CustRec absolute Y;
begin
  Less := FirstCust.Number < SecondCust.Number;
end;

procedure OutP;
var
  I:  Integer;
begin
  repeat
    SortReturn(Customer);
    with Customer do
    begin
      Write(Number, ' ',Name,' ');
      for I := Length(Name) to 30 do Write(' ');
      Write(Addr);
      for I := Length(Addr) to 20 do Write(' ');
      Write(City);
      for I := Length(City) to 12 do Write(' ');
      WriteLn(State,' ',Zip);
    end;
  until SortEOS;
end;

begin {program SortExampleOne}
  Assign(CustFile,'CUSTOMER.DTA');
  Reset(Custfile);
```

```
  WriteLn(TurboSort(SizeOf(CustRec)));
end.
```

```
program SortExampleTwo { Customer File and Stock File };

type
  CustRec = record
              Number: integer;
              Name:   string[30];
              Addr:   string[20];
              City:   string[12];
              State:  string[3];
              Zip:    string[5];
            end;

  ItemRec = record
              Number:  integer;
              Descrip: string[30];
              InStock: integer;
              Price:   real;
            end;

var
  CustFile:   file of CustRec;
  Customer:   CustRec;
  StockFile:  file of ItemRec;
  Item:       ItemRec;
  Choice:     Char;

{$ISORT.BOX}

procedure Inp; {this procedure is forward declared in SORT.BOX}
begin
  case Choice of
    'C': begin
           repeat
             Read(CustFile,Customer);
             SortRelease(Customer);
           until EOF(CustFile);
         end;
    'S': begin
           repeat
             Read(StockFile,Item);
             SortRelease(Item);
           until EOF(StockFile);
         end;
  end; {case}
end;

function Less; {this boolean function has two parameters, X and Y}
               {and is forward declared in SORT.BOX}
var
  FirstCust:  CustRec absolute X;
  SecondCust: CustRec absolute Y;
  FirstItem:  ItemRec absolute X;
  SecondItem: ItemRec absolute Y;
begin
  case Choice of
    'C': Less := FirstCust.Number < SecondCust.Number;
    'S': Less := (FirstItem.InStock < SecondItem.InStock) or
```

```
            ((FirstItem.InStock = SecondItem.InStock) and
             (FirstItem.Price < SecondItem.Price));
  end;
end;

procedure OutP;
var
  I:  Integer;
begin
  case Choice of
    'C': begin
           repeat
             SortReturn(Customer);
             with Customer do
             begin
               Write(Number, ' ',Name,' ');
               for I := Length(Name) to 30 do Write(' ');
               Write(Addr);
               for I := Length(Addr) to 20 do Write(' ');
               Write(City);
               for I := Length(City) to 12 do Write(' ');
               WriteLn(State,' ',Zip);
             end;
           until SortEOS;
         end;
    'S': begin
           repeat
             SortReturn(Item);
             with Item do
             begin
               Write(Number, ' ',Descrip,' ');
               for I := Length(Descrip) to 30 do Write(' ');
               WriteLn(InStock:5,Price:8:2);
             end;
           until SortEOS;
         end;
  end; {case}
end;

begin {program SortExampleOne}
  Write('Sort Customers or Stock? (enter C or S): ');
  repeat
    read(Kbd,Choice);
    Choice := UpCase(Choice);
  until Choice in ['C','S'];
  WriteLn(Choice);
  case Choice of
    'C': begin
           Assign(CustFile,'CUSTOMER.DTA');
           Reset(CustFile);
           WriteLn(TurboSort(SizeOf(CustRec)));
         end;
    'S': begin
           Assign(StockFile,'STOCK.DTA');
           Reset(StockFile);
           WriteLn(TurboSort(SizeOf(ItemRec)));
         end;
```

```
  end; {case}
end.
```

```
echo off
cls
echo ÿ
echo This program will install Turbo Graphix Toolbox for the IBM Color/Graphics
echo Adapter, the Hercules (Monochrome) Graphics Card, or the Heath/Zenith Z100
echo series of computers.
echo ÿ
for %%x in (ibm IBM hgc HGC z10 Z10) do if [%1]==[%%x] goto gotname
echo You must type "TGINST IBM" for the IBM color card, "TGINST HGC"
echo for the Hercules monochrome echo card, or "TGINST Z10" for the
echo Heath/Zenith Z100.
goto quit
:gotname
if exist graphix.%1 goto gotfile
echo Turbo Graphix cannot be installed because GRAPHIX.%1 is not available on
echo the current drive.
goto quit
:gotfile
if not exist graphix.sys goto inst%1
echo GRAPHIX.SYS already exists.  If you do not want to continue (and overwrite
echo it with GRAPHIX.%1), please type CTRL-Break followed by a 'Y' right now:
echo ÿ
pause
goto inst%1
:instibm
echo ÿ
copy graphix.ibm graphix.sys >nul
echo Turbo Graphix Toolbox has been installed for the IBM Color/Graphics Adapter.
goto quit
:insthgc
echo ÿ
copy graphix.hgc graphix.sys >nul
echo Turbo Graphix Toolbox has been installed for the Hercules Graphics Card.
goto quit
:instz10
echo ÿ
copy graphix.z10 graphix.sys >nul
echo Turbo Graphix Toolbox has been installed for the Heath/Zenith Z100 computer.
:quit
```

```
(*************************************************************************
*  TURBOLIB.INC                                                         *
*                                                                       *
*                                                                       *
*  MISCELLANEOUS TURBO PASCAL PROCEDURES    TU DELFT / M.J.Vos / 1988 *
*************************************************************************
* LV; HV; Bell; ClrLine; ClrLines; Scroll; CursorOff; CursorOn;       *
* CopyFile; WriteXY, PressKey; DisplayFile; DrawBloc; Exist; Ja; Nee; *
* File_Text; PrintFile; Yes; Menu; MSDOS_Volume; File_Handling;       *
* Init_BitImage; File_Handling; Sort; Quicksort.                      *
*************************************************************************)

(* See also the example programs on floppy B and report-appendices   *)

(*=======================================================================*)
PROCEDURE LV;
(*=======================================================================*)

BEGIN
  LowVideo;
END; (* LV *)


(*=======================================================================*)
PROCEDURE HV;
(*=======================================================================*)

BEGIN
  HighVideo;
END; (* HV *)


(*=======================================================================*)
PROCEDURE Bell;
(*=======================================================================*)

BEGIN
  Write(Chr(7));
  Delay(500);
END; (* Bell *)

{
(*=======================================================================*)
PROCEDURE Init_BitImage;   (* CRT-Hardcopy 640X400 with M24- PROCEDURE*)
(*=======================================================================*)

BEGIN
 Write(LST,#$27,'*',#$4,#$128,#$2);
END; (* Init_BitImage *)
}

(*=======================================================================*)
PROCEDURE ClrLine(X,Y:Integer);
(*=======================================================================*)

BEGIN
 GotoXY(X,Y);
```

```
 ClrEol;
END; (* ClrLine *)


(*=====================================================================*)
PROCEDURE ClrLines(First,Last:Integer);
(*=====================================================================*)

VAR   i :Integer;

BEGIN
 FOR i:=First TO Last DO ClrLine(1,i);
END;  (* ClrLines *)


(*=====================================================================*)
PROCEDURE Scroll;
(*=====================================================================*)

VAR  i : Integer;

BEGIN
 FOR i:=1 TO 32 DO
 BEGIN
  WriteLn;
  Delay(10);
 END;
END;  (* Scroll *)
{
(*=====================================================================*)
PROCEDURE CursorOn;
(*=====================================================================*)

BEGIN
  Write(Chr(27),'B',4);
END; (* CursorOn *)

(*=====================================================================*)
PROCEDURE CursorOff;
(*=====================================================================*)

BEGIN
  Write(Chr(27),'C',4);
END; (* CursorOff *)

}
(*=====================================================================*)
PROCEDURE CopyFile;
(*=====================================================================*)

VAR LineTel,LineTelMax : Integer;
    LineArr            : ARRAY[1..Arr_Afm] OF Str80;

BEGIN
 Assign(Filvar,FileName);
 Reset(FilVar);
 Assign(FV,FN);
```

```
 ReWrite(FV);

 LineTel:=0;
 REPEAT
  LineTel:=LineTel+1;
  ReadLn(Filvar,LineArr[LineTel]);
  IF Eof(Filvar) THEN LineTelMax:=LineTel;
 UNTIL Eof(Filvar);
 FOR LineTel:=1 TO LineTelMax DO
 WriteLn(FV,LineArr[LineTel]);

 Close(Filvar);
 Close(FV);
END; (* CopyFile *)

(*=======================================================================*)
PROCEDURE WriteXY(X,Y{,video}:Integer;Str:Str80);
(*=======================================================================*)

BEGIN
 GotoXY(X,Y);
 {VideoMode(video,True;}
 Write(Str);
 {VideoMode(video,False);}
END; (* WriteXY *)


(*=======================================================================*)
PROCEDURE Presskey;
(*=======================================================================*)

VAR  Ch : Char;

BEGIN
  ClrLines(23,24);
  LV;WriteXY(1,23,{1,}'Continue with a key ...');
  Read(Kbd,Ch);
  Scroll;ClrScr;HV;
END; (* PressKey *)


(*=======================================================================*)
PROCEDURE DisplayFile(FileName:Str14);
(*=======================================================================*)

VAR Line   : Str80;
    F      : Text;
    Nr,bNr : Integer;

BEGIN
 Assign(F,FileName);
 Reset(F);
 Nr:=0;bNr:=1;
 REPEAT
   Nr:=Nr+1;
   ReadLn(F,Line);
   WriteLn(Line);
```

```
   Delay(200);
   IF Nr=BNr*20 THEN
   BEGIN
    bNr:=bNr+1;
    PressKey;
   END;
 UNTIL Eof(F);
 Close(F);
END; (* DisplayFile *)

{
(*=====================================================================*)
PROCEDURE DrawBloc(X1,Y1,X2,Y2:Integer);          (* MS DOS *)
(*=====================================================================*)

BEGIN
  ClrScr;
  Write(Chr(27),'L',Chr(Y1+31),Chr(X1+31),Chr(Y1+31),Chr(X2+31));
  Write(Chr(27),'L',Chr(Y1+31),Chr(X2+31),Chr(Y2+31),Chr(X2+31));
  Write(Chr(27),'L',Chr(Y2+31),Chr(X2+31),Chr(Y2+31),Chr(X1+31));
  Write(Chr(27),'L',Chr(Y2+31),Chr(X1+31),Chr(Y1+31),Chr(X1+31));
END;  (* DrawBloc *)
}

(*=====================================================================*)
FUNCTION Exist(FileName:Str14) : Boolean;
(*=====================================================================*)

VAR  F : File;

BEGIN
  Exist:=True;
  IF (FileName<>'con:') AND (FileName<>'CON:') THEN
  BEGIN
    (*$I-*)
    Assign(FV,FileName);
    Reset(FV);
    (*$I+*)
    Exist:=(IOresult=0);
  END;
END; (* Exist *)


(*=====================================================================*)
FUNCTION Ja:Boolean;
(*=====================================================================*)

BEGIN
 Ja:=False;
 IF (Ch IN ['Y','y']) THEN Ja:=True;
END; (* Ja *)


(*=====================================================================*)
FUNCTION Nee:Boolean;
(*=====================================================================*)
```

```
BEGIN
 Nee:=False;
 IF (Ch IN ['N','n']) THEN Nee:=True;
END; (* Nee *)


(*=====================================================================*)
PROCEDURE PrintFile(FileName : Str14);
(*=====================================================================*)

VAR Line : Str80;
    F    : Text;

BEGIN
  Assign(FV,FileName);
  Reset(FV);
  WriteLn(LST,#$64);
  REPEAT
    ReadLn(FV,Line);
    WriteLn(LST,Line);
  UNTIL Eof(FV);
  WriteLn(LST,#$12);
  Close(FV);
END; (* PrintFile *)


(*=====================================================================*)
FUNCTION Yes(question:Str80):Boolean;
(*=====================================================================*)

VAR Ch : Char;

BEGIN
  Write(question);LV;Write(' (Y/N) : ');HV;
  REPEAT Read(Kbd,Ch); UNTIL Ch IN ['y','Y','n','N'];
  WriteLn;
  Yes:=(Ch IN ['y','Y']);
END; (* Yes *)


(*=====================================================================*)
FUNCTION Menu(MenuStr:Str80):Char;
(*=====================================================================*)

VAR  Ch           : Char;
     ChPos,Lengte,i: Integer;
     Found        : Boolean;
     Line         : Str80;


 PROCEDURE PrintMenu(Menu_Heading:Str20;MenuStr:Str80);

BEGIN
  GotoXY(1,1);ClrEol;LV;
  Write('                *****      ',Menu_Heading,'      *****');
  HV;
  GotoXY(1,3); ClrEol;
```

```
  Lengte:=LENGTH(MenuStr);
  FOR i:=1 TO Lengte DO
  BEGIN
    Ch:=MenuStr[i];
    IF Ch IN ['A'..'Z','0'..'9'] THEN HV ELSE LV;
    Write(Ch);
  END;
  HV; Write(' ? ');
END;
BEGIN
  PrintMenu(Menu_Heading,MenuStr); Read(Kbd,Ch); WriteLn(Ch);
  IF Ch IN ['a'..'z'] THEN Ch:=UpCase(Ch);
  ChPos:=1; Found:=False; Line:=MenuStr;
  REPEAT
    Lengte:=LENGTH(Line);
    Found:=(Ch = Line[1]);
    ChPos:=POS(' ',Line)+1;
    Line:=Copy(Line,ChPos,Lengte);
  UNTIL ((ChPos=1) OR Found);
  IF Found THEN Menu:=Ch ELSE
  BEGIN
    Menu:=' ';
    Bell;WriteLn(' ... is no choice !');Delay(500);
  END;
  ClrScr;PrintMenu(Menu_Heading,MenuStr);WriteLn(Ch);
END; (* Menu *)

{
(*========================= MSDOS ==================================*)
FUNCTION MSDOS_Volume(drive_nr : Byte) : Str255;
(*======================= = = = = = ================================*)

VAR
   nam : Str11;
   k   : Byte;
   FCB : RECORD
         flag            : Byte;
         dos_reserved    : ARRAY[1..5] OF Byte;
         attribute,
         drive           : Byte;
         name            : ARRAY[1..11] OF Char;
         not_used        : ARRAY[12..36] OF Byte;
        END;
   reg : RECORD
         ax,bx,cx,dx,bp,si,di,ds,es,flags
                         : Integer;
        END;

BEGIN
WITH FCB,reg DO
 BEGIN
  ds:= Seg(FCB);
  dx:= Ofs(FCB);
  ax:= $1A00;
  MSDOS(reg);
  flag:=$FF;
  attribute:=$08;
```

```
  drive:=drive_nr;
  FillChar(name,11,'?');
  ax:=$1100;
  MSDOS(reg);
  nam:=' ';
  k:=1;
  IF Lo(ax)=0 THEN
  WHILE (k<=11) AND (name[k]<>' ') DO
  BEGIN
   nam:=nam+name[k];
   k:=Succ(k);
  END;
  MSDOSVolume:=nam;
 END;
END; (* MSDOS_Volume *)

}
(*=========================================================================*)
PROCEDURE File_Handling;
(*=========================================================================*)

VAR Dr,LineTel,LineTelMax : Integer;
    St                    : Str14;

BEGIN
ClrScr;
Quit:=False;
 REPEAT
  Menu_Heading:='FILE HANDLING MENU';
  CASE Menu('Copy Directory Erase File_manager Print Rename Show User_defined Quit ')
  OF
     'C' : BEGIN
           REPEAT
             Write('Give filename');LV;Write('  (d:filename.typ) : ');
             HV;Read(FileName);WriteLn;
             IF Exist(FileName)=True THEN
            BEGIN
             REPEAT
              LV;Write('New filename                  : ');
              HV;Read(FN);WriteLn;
              IF Exist(FN)=True THEN
              BEGIN
               Bell;WriteLn('File exists !');
               Delay(1000);
               Write('Replace ',FileName,' ?');
              END;
             UNTIL Yes('... o.k. ?');
             LV;WriteLn('WAIT !');HV;
             CopyFile;
             LV;Write(FileName,'  copied to  ');
             HV;Write(FN);WriteLn;
            END;
            IF Exist(FileName)=False THEN
            BEGIN
             Bell;
             WriteLn('File does not exist  !');
             Delay(1000);
```

```
        END;
        Write('More COPY ?');LV;
        Write('  (Y/N) ');HV;
        Read(Kbd,Ch);WriteLn;WriteLn;
       UNTIL Nee;
       Delay(1000);ClrScr;
      END; (* Copy *)
'D' : BEGIN
       Assign(FV,'b:MSDOSDIR.CHN');
       Chain(FV);
       Delay(3000);PressKey;
      END; (* Directory *)
'E' : BEGIN
       REPEAT
        Write('Give filename');LV;Write('  (d:filename.typ) : ');
        HV;Read(FileName);WriteLn;Assign(Filvar,FileName);
        IF Exist(FileName)=True THEN
        BEGIN
         Write('Erase ',FileName,' ?');LV;
         Write('  (Y/N) ');HV;
         Read(Kbd,Ch);WriteLn;IF Ja THEN
         BEGIN
          Erase(Filvar);
          LV;Write(FileName);
          HV;Write(' erased');WriteLn;
         END;
        END ELSE
        BEGIN
         Bell;
         WriteLn('File does not exist !');
         Delay(1000);
        END;
        Write('More ERASE ?');LV;
        Write('  (Y/N) ');HV;
        Read(Kbd,Ch);WriteLn;WriteLn;
       UNTIL Nee;
       Delay(1000);ClrScr;
      END; (* Erase *)
'F' : BEGIN
       FileName:='a:PFM.COM';
       IF NOT Exist(FileName) THEN
       BEGIN
        FN:='a:FM.COM';
        IF NOT Exist(FN) THEN
        BEGIN
         Bell;
         WriteLn(FileName,' and ',FN,' not yet implemented !');Delay(1000);
        END ELSE
        BEGIN
         WriteLn('Input command : FN ');Delay(3000);
         Assign(FV,'a:COMM211.CHN');
         Chain(FV);
        END;
       END;
       BEGIN
        WriteLn('Input command : PFM');Delay(3000);
        Assign(FV,'a:COMM211.CHN');
```

```
        Chain(FV);
       END;
       Delay(3000);
       ClrScr;
      END; (* File_Manager *)
'P' : BEGIN
       Write('Do you want to use a:PRINT.COM ?');
       LV;Write(' (Y/N)');HV;Read(Kbd,Ch);WriteLn;
       IF Nee THEN
       BEGIN
        Bell;
        LV;WriteLn('You will use the ... ');HV;
        Delay(1000);GotoXY(20,12);
        WriteLn('/// HYDROLIN /// print-program : ');
        Delay(2000);Scroll;ClrScr;
        REPEAT
         Write('Give filename');LV;Write('  (d:filename.typ) : ');
         HV;Read(FileName);WriteLn;
         IF Exist(FileName)=True THEN
         PrintFile(FileName) ELSE
         BEGIN
          Bell;WriteLn('File does not exist');
          Delay(1000);
         END;
         Write('More PRINT ?');LV;
         Write('  (Y/N) ');HV;
         Read(Kbd,Ch);
        UNTIL Nee;
       END ELSE
       BEGIN
        WriteLn('Input command : d:PRINT  d:FileName.typ');Delay(500);
        WriteLn('          or : d:PDIR   d:FileName.typ');Delay(500);
        WriteLn('          or : d:GWLIST d:FileName.typ');Delay(3000);
        Assign(FV,'a:COMM211.CHN');
        Chain(FV);
       END;
       Delay(1000);ClrScr;
      END; (* Print *)
'Q' : BEGIN
       IF Yes('... are you sure ?') THEN Quit:=True;
       ClrScr;
      END; (* Quit *)
'R' : BEGIN
       REPEAT
        Write('Give filename');LV;Write('  (d:filename.typ) : ');
        HV;Read(FileName);WriteLn;Assign(Filvar,FileName);
        IF Exist(FileName)=True THEN
        BEGIN
         LV;Write('Rename ');HV;
         Write(FileName,' ?  ');LV;Write('  (Y/N) ');
         HV;Read(Kbd,Ch);WriteLn;
         IF Ja THEN
         BEGIN
          REPEAT
           LV;Write('New name                    : ');
           HV;Read(FN);WriteLn;
           IF Exist(FN)=True THEN
```

```
            BEGIN
             Bell;
             WriteLn('File exists !');
            END;
           UNTIL Exist(FN)=False;
           Rename(Filvar,FN);
           WriteLn(FileName,' renamed.');
          END ELSE
         END ELSE
         BEGIN
          Bell;WriteLn('File does not exist !');
          Delay(1000);
         END;
         Write('More RENAME ?');LV;
         Write('  (Y/N) ');HV;
         Read(Kbd,Ch);WriteLn;WriteLn;
        UNTIL Nee;
        Delay(1000);ClrScr;
       END; (* Rename *)
   'S' : BEGIN
        LV;WriteLn('You will use the ... ');HV;
        Delay(1000);GotoXY(20,12);
        WriteLn('/// HYDROLIN /// display-program : ');
        Delay(2000);Scroll;ClrScr;
        REPEAT
         Write('Give filename');LV;Write('  (d:filename.typ):  ');
         HV;Read(FileName);WriteLn;
         IF Exist(FileName)=True THEN
         DisplayFile(FileName) ELSE
         BEGIN
          Bell;WriteLn(FileName,' does not exist !');
          Delay(1000);
         END;
         WriteLn;Write('SHOW more ?');
         LV;Write('  (Y/N) ');HV;
         Read(Kbd,Ch);WriteLn;ClrScr;
        UNTIL Nee;
       END; (* Show *)
   'U'  : BEGIN
        Write('Give filename');LV;Write('  (b:filename.typ) :  ');
        HV;Read(FileName);WriteLn;
        IF NOT Exist(FileName) THEN
        BEGIN
         Bell;
         WriteLn(FileName,' not yet implemented !');Delay(1000);
        END ELSE
        BEGIN
         LV;WriteLn('Input command : ',FileName);HV;Delay(3000);
         Assign(FV,'a:COMM211.CHN');
         Chain(FV);
        END;
        ClrScr;
       END; (* User_defined *)
 END; (* Case *)
HV;
UNTIL Quit=True;
END; (* File_Handling *)
```

```
{
(*=================================================================*)
PROCEDURE SORT(Var X : ArrInt ; Var N :Integer);
(*=================================================================*)
(* See also the report-appendices and files on floppy B           *)
Var j,k,p : Integer;


PROCEDURE WISSEL(Var X,Y : Real;);

Var H     : Real;

BEGIN
 H:=X;X:=Y;Y:=H;
END; (*WISSEL *)


BEGIN
 j:=0;
 WHILE j<>N DO
 BEGIN
  j:=j+1;
  k:=j;p:=j;
  WHILE p<>N DO
  BEGIN
   p:=p+1;
   IF X[p]>X[k] THEN k:=p
  END;
  WISSEL(X[j],X[k]);
 END;
END; (* SORT *)


(*=================================================================*)
PROCEDURE QUICKSORT (Var X : Arr; I1,I2 : Integer);
(*=================================================================*)

Var     R,S    : Integer;
        G      : Real;


PROCEDURE WISSEL(Var X,Y : Real);

Var     H      : Real;

BEGIN
H:=X;X:=Y;Y:=H;
END; (*WISSEL *)


BEGIN
 IF I1<I2 THEN
 BEGIN
  G:=X[(I1+I2) DIV 2];
  R:=I1;S:=I2;
  WHILE R<=S DO
  BEGIN
```

```
   WHILE X[R]>G DO R:=R+1;
   WHILE X[S]<G DO S:=S-1;
   IF R<=S THEN
   BEGIN
    WISSEL(X[R],X[S]);
    R:=R+1;S:=S-1;
   END;
  END;
  QUICKSORT(X,I1,S);
  QUICKSORT(X,R,I2);
 END;
END; (* QUICKSORT *)
}
(*END ======================== TURBOLIB.INC ======================== *)
```

INFO VAN DE METHODE VAN DE UNIT HYDROGRAPH.

De METHODE VAN DE EENHEIDSAFVOERGOLF berust op empirisch gevonden relaties
- L.K. Sherman 1932 - en dient ter bepaling van oppervlakteafvoercomponent
(van de afvoer), die ontstaat ten gevolge van de gemiddelde netto neerslag;
deze neerslag wordt gelijkmatig verdeeld gedacht over het stroomgebied.
De grootte(A) van het stroomgebied heeft een bovengrens van ca. 10.000 vier
kante kilometer. Bij de METHODE VAN DE EENHEIDSAFVOERGOLF wordt uitgegaan
van een EENHEIDSDIEPTE- UNIT DEPTH - b.v. 1 inch of 1 centimeter of een
EENHEIDSVOLUME A ,gelijkmatig verdeeld over het stroomgebied.
De tijdsDUUR van een enkelvoudige bui wordt DT genoemd.
Vanaf het einde van de bui tot het einde van de eenheidsafvoerkromme is de
tijdsduur TR - de AFLOOPTIJD.
Het afvoerdebiet wordt weergegeven als Q of als q=Q/A.
Een over een bepaalde tijd gemeten neerslagkromme kan worden opgedeeld in
een aantal enkelvoudige buien van zo mogelijk zelfde tijdsduur DT.
Allereerst zulle data ingevoerd moeten worden. Dit geschiedt via INPUT
/ KEYBOARD naar een bestand op schijf. Veranderingen hieraan moeten worden
gedaan met een EDITER.
Allerlei manipulaties met bestanden kunnen worden gedaan met COMMAND (= MSDOS
commandos) of FILEHANDLING.
Vervolgens kunnen regenverliezen en afvoerscheiding(en) worden ingevoerd,
welke eveneens op schijf worden weggeschreven onder een door HYDROLIN
ondersteunde karakteristieke naam. De bij elkaar behorende bruto- en verlies-
datafiles leveren netto-datafiles op.
Het is dan mogelijk om te zien of de geaccumuleerde netto regen en afvoer
met elkaar in overeenstemming zijn.

Repetitie van elk procedee is mogelijk waarbij een aantal van maximaal drie
gemodificeerde gelijksoortige bestanden op schijf kunnen bestaan. Het beheer
geschiedt door twee procedures welke vragen om characters voor identificatie,
specificatie en volgnummer.
Door filemanipulaties is het mogelijk meer bestanden te bewaren b.v. onder een
andere naam, op een andere schijf, op een andere drive etc.
Uit bij elkaar behorende datasets van neerslag en afvoer voor gelijke tijdstap
- een aparte procedure voor gelijke tijdstap is aanwezig - kan het computer-
programma nu een Unit Hydrigraph uitrekenenen.
Er is een mogelijkheid tot graphische weergave op beeldscherm en papier.

EINDE VAN DE UNIT HYDROGRAPH-INFO.

5. SLOTOPMERKINGEN.

5.1. FACILITEITEN EN MOGELIJKHEDEN.

Het platvorm van Utilities in HYDROLIN kan ook van toepassing zijn voor andere programma's. Ook volkomen andere modellen kunnen gebruik maken van diverse gereedschappen, die zich in HYDROLIN bevinden.
HYDROLIN draait op een IBM compatible PC met twee discdrives en, voor niet al te grote aantallen data, met weinig extern en intern geheugen. Dit geeft aan, dat zeer kleine systemen al goed zijn om HYDROLIN te laten draaien. Met de toename van de aantallen extra utilities groeit de noodzaak tot meer intern en extern geheugen. Draagbare computers bieden gelegenheid eventueel op reis c.q. in het veld dit programma te laten draaien. Gedacht kan hier ook worden aan het gebruik van één floppy per model met een deel van het hoofdprogramma.

zie TURBOLIB.INC
COMM211 .PAS
MATLIB .INC
MSDOSDIR.PAS
COMM300 .PAS

5.2. BEPERKINGEN.

HYDROLIN versie 1.00 is ontwikkeld met TURBO PASCAL versie 3.01A met de restrictie van een klein werkgeheugen van 64 kBytes.
HYDROLIN/ versie 1.00 kent duidelijk beperkingen voor het aantal te verwerken meetgegevens door beperkte opslag op floppies.
Te grote arrays veroorzaken te gauw overschrijding van de grenzen van het werkgeheugen.
Vanwege de noodzaak tot het maken van op zichzelf staande subprogramma's is er weinig ruimte over op de floppies.
Het grafisch zichtbaar maken van bestanden heeft zijn grenzen in nauwkeurigheid van aflezing door de resolutie van 640X200 pixels.

zie HYDROGR .PAS
POINTERS.PAS
M24 .INC

5.3. AANBEVELINGEN.

Een TURBO PASCAL/ versie 4.XX kan door een groter aanspreekbaar werkgeheugen nml. 640 kBytes voor een meer efficient werkend programma zorgen doordat er meer ruimte is voor het in zijn geheel compileren van de programma-source en er meer ruimte is voor de dataopslag in arrays in het werkgeheugen. De beperkingen van de HYDROLIN-versie met twee discdrives kunnen worden opgelost door het werken met een harddisc. Tijdelijke opslag van data, meer dan het nog beschikbare beschikbare werkgeheugen toelaat, kan - ook voor de 64k-versie - geschieden door toepassing van pointers, waartoe een listing is bijgevoegd. Het programma TURBO EXTENDER verschaft de mogelijkheid tot het beschikken over externe geheugenruimte - zoals een harddisc - als werkgeheugen. TURBO POWER TOOLKIT biedt de mogelijkheid voor een cursorbestuurd menu, opstarten van externe programma's en terugkeer in HYDROLIN op de plaats van vertrek. De standaard-resolutie van een Olivetti/IBM Graphics Card van 640X200 pixels is met de bijgevoegde procedure M24 te verhogen tot 640X400 punten.
Het zelf direct in een grafische weergave tekenen met de cursor wordt door TURBO GRAPHICS TOOLBOX geboden. Vergrotingen van grafieken is in dit pakket eveneens mogelijk.
Voor de weergave van de UNIT HYDROGRAPH met een vloeiende kromme is het mogelijk de veelhoeksvorm te "smooth-en"(spline).
Zoals al enigszins aangegeven kan met het aansturen van diverse devices communicatie worden gepleegd met een scala van perifierie-apparatuur; kleine systemen kunnen dan door verbinding met grotere systemen toch grote opdrachten uitvoeren.
Het bijgevoegde directory-moduul is een in principe in kleur weer te geven bestandsprogramma.
Door een zgn. "intermediate plotfile-program" is het mogelijk om zeer compact op schijf data weg te schrijven en via een menu naar een modem, printer of plotter te verzenden.
Niet gebruikt maar resident in TURBOLIB bevindt zich een procedure voor het opvragen van Volume-naam van schijven ten behoeve van snelle contrôle van het gebruik van de juiste floppy.
Ten behoeve van andere gebruikers is een matrixbibliotheek meegeleverd alsook enige sorteerroutines.
Een eventueel TURBO OPTIMIZER-programma biedt de mogelijkheid in code te verkleinen door slechts benodigde delen van de bibliotheken mee te compileren. De toepassing van pointers - ook voor de uitbreiding van datasets(records en fields) - bespaart onnodig continu beslag op het werkgeheugen. Vooral bij nieuwe versies van HYDROLIN met veel theorieen kan dit erg belangrijk zijn.

LITTERATUUR

1. HYDROLOGIE
Basiscollege f15N
door Prof.Dr.Ir. J.C. van Dam
Technische Universiteit Delft
DELFT,september 1986;

2. HYDROLOGISCHE MODELLEN f15D
door Prof.Dr.Ir. J.C. van Dam
Technische Universiteit Delft
DELFT,september 1987;

3. VOORTGEZETTE HYDROLOGIE f15G
door Prof.Dr.Ir. J.C. van Dam
Technische Universiteit Delft
DELFT,september 1987;

4. TURBO PASCAL (3.01A) MANUAL
door Borland International
CALIFORNIA, mei 1984;

5. TURBO GRAPHICS TOOLBOX MANUAL
door Borland International
CALIFORNIA, mei 1985.

TECHNISCHE UNIVERSITEIT DELFT
FACULTEIT DER CIVIELE TECHNIEK

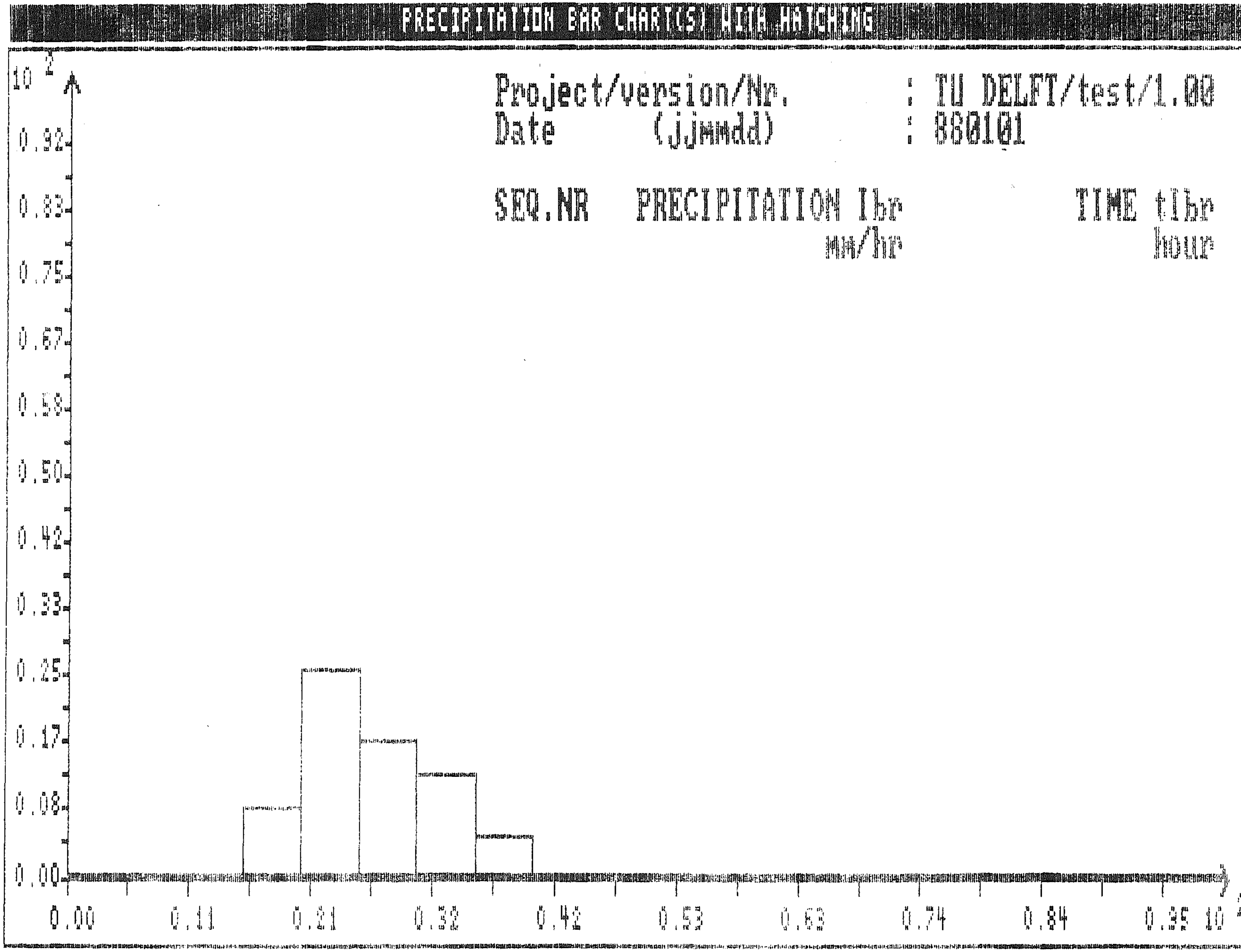


Fig. 1

HYDROLIN
Unit Hydrograph

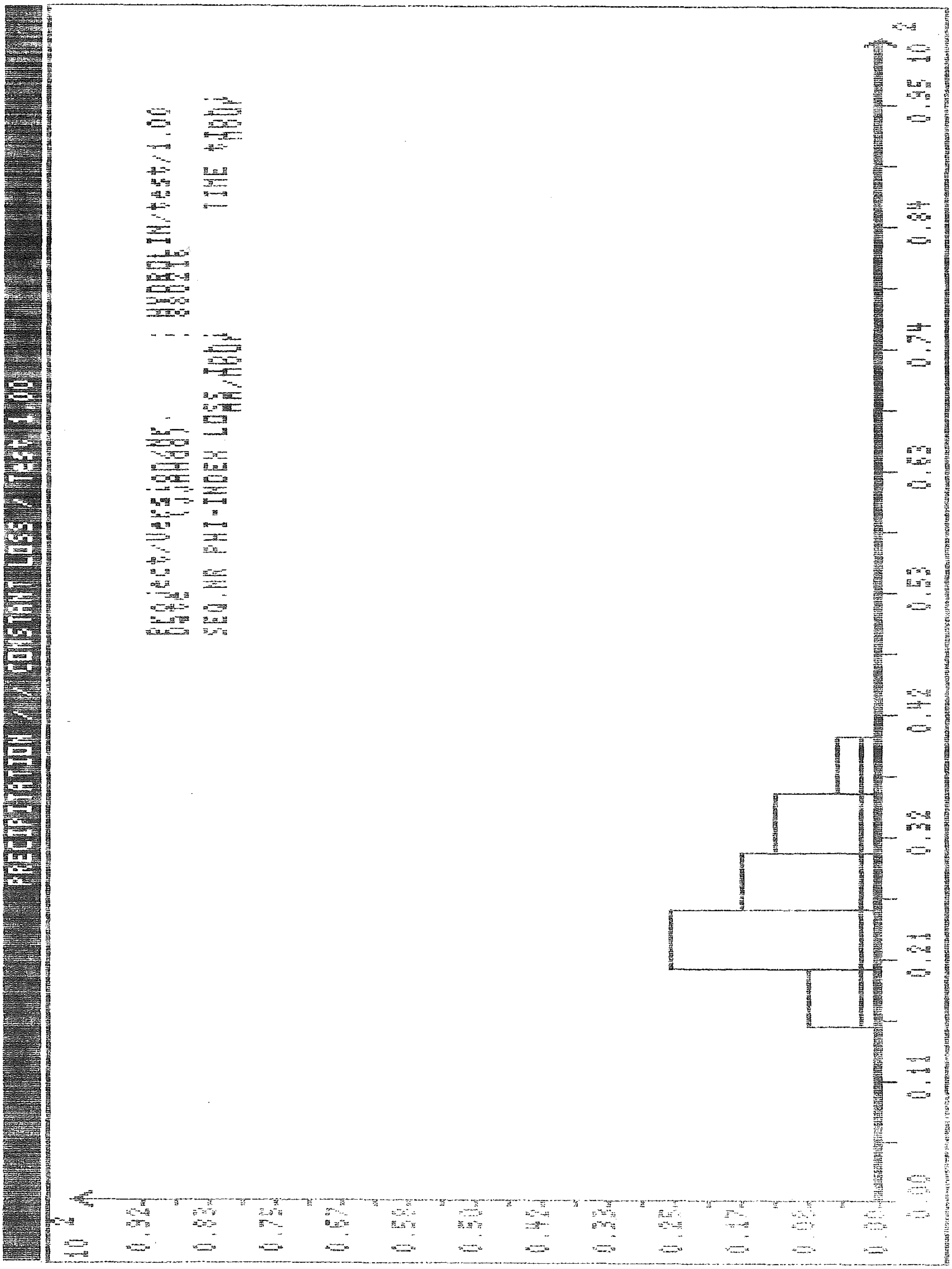


Fig. 2

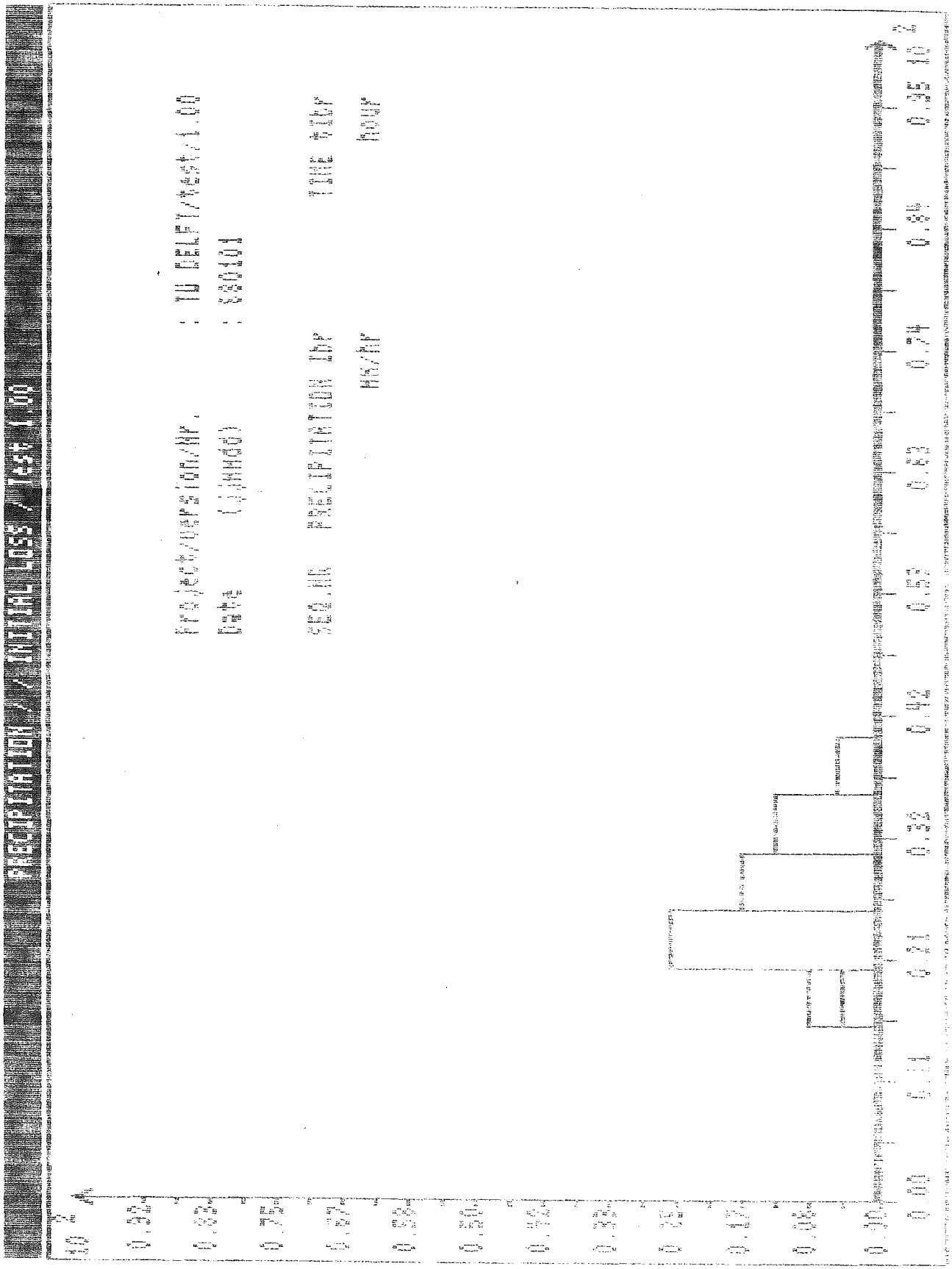


Fig. 3

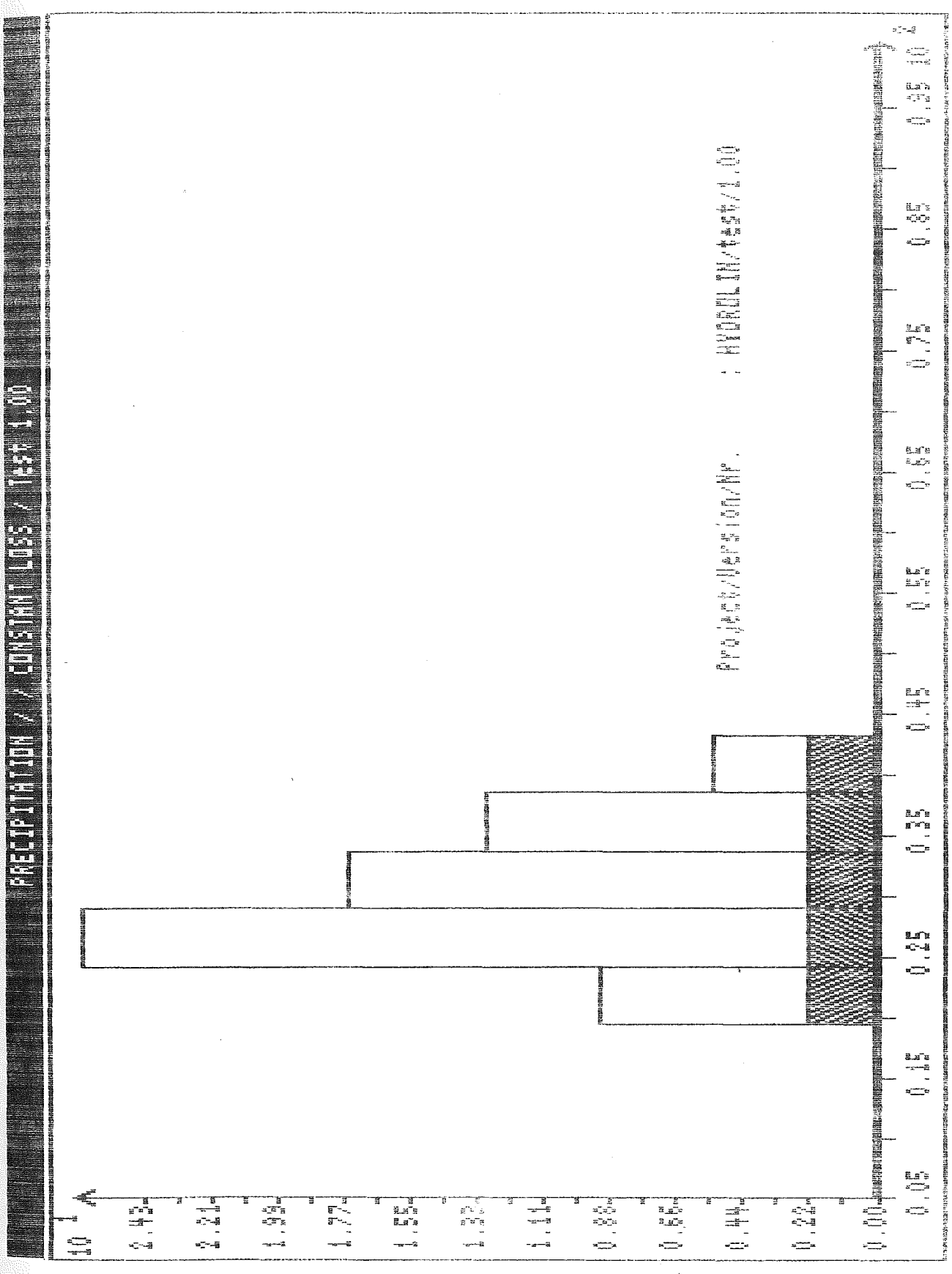


Fig. 4

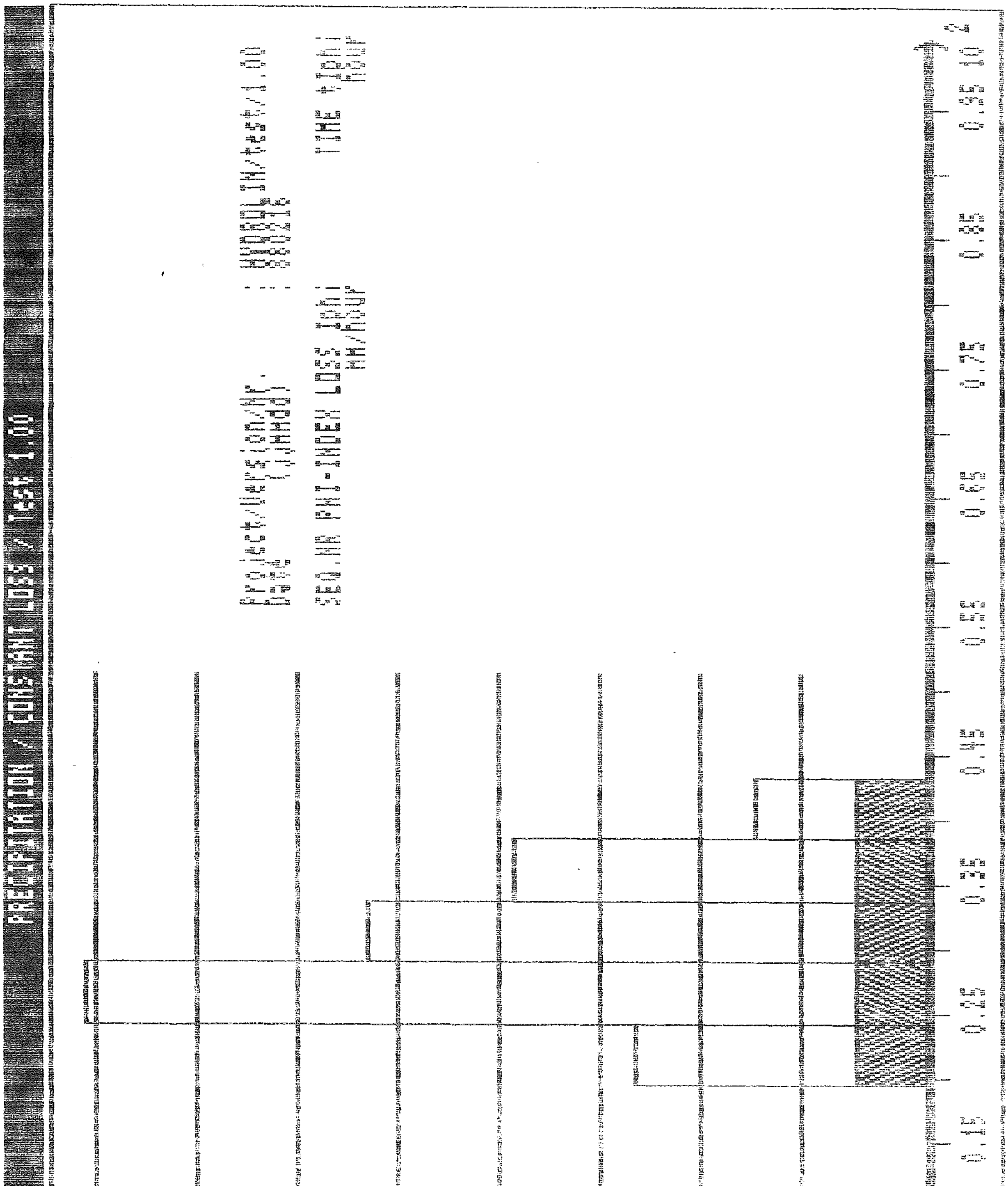


Fig. 5

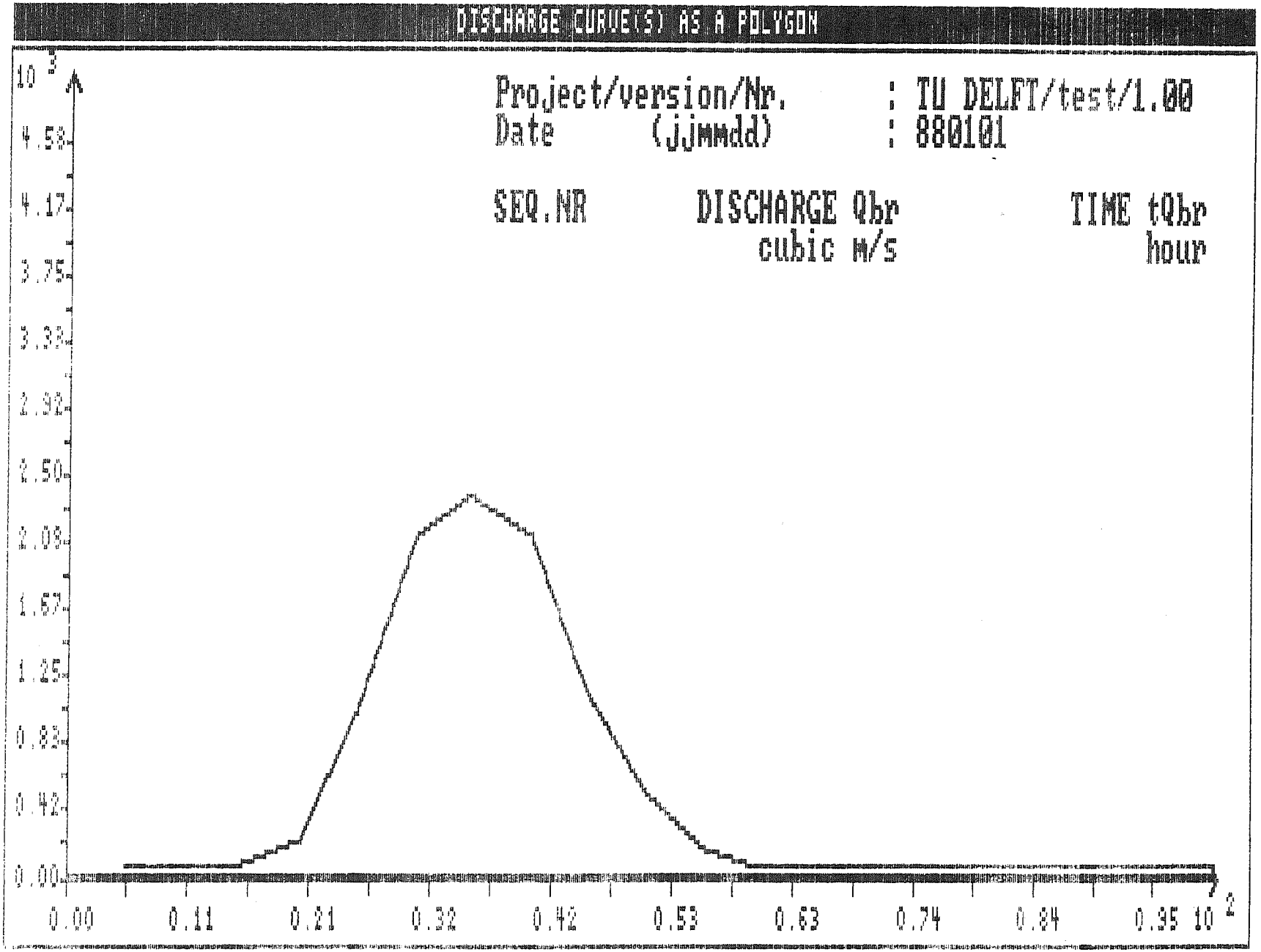

DISCHARGE CURVE(S) AS A POLYGON
Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101
SEQ.NR
DISCHARGE Qbr
cubic m/s
TIME tQbr
hour
10 -3
4.58
4.17
3.75
3.33
2.92
2.50
2.08
1.67
1.25
0.83
0.42
0.00
0.00
0.11
0.21
0.32
0.42
0.53
0.63
0.74
0.84
0.95 10 2

Fig. 6

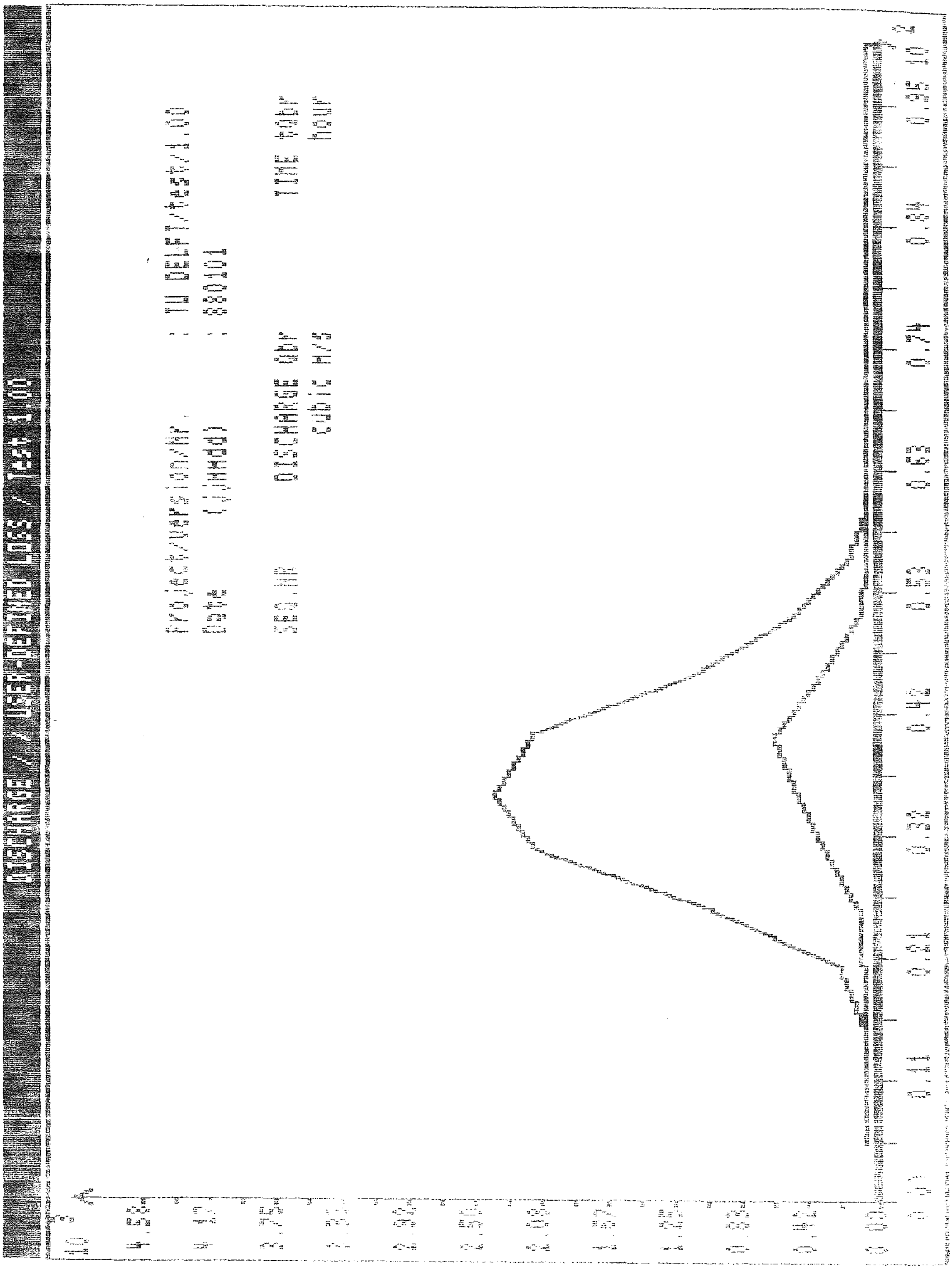


Fig. 7

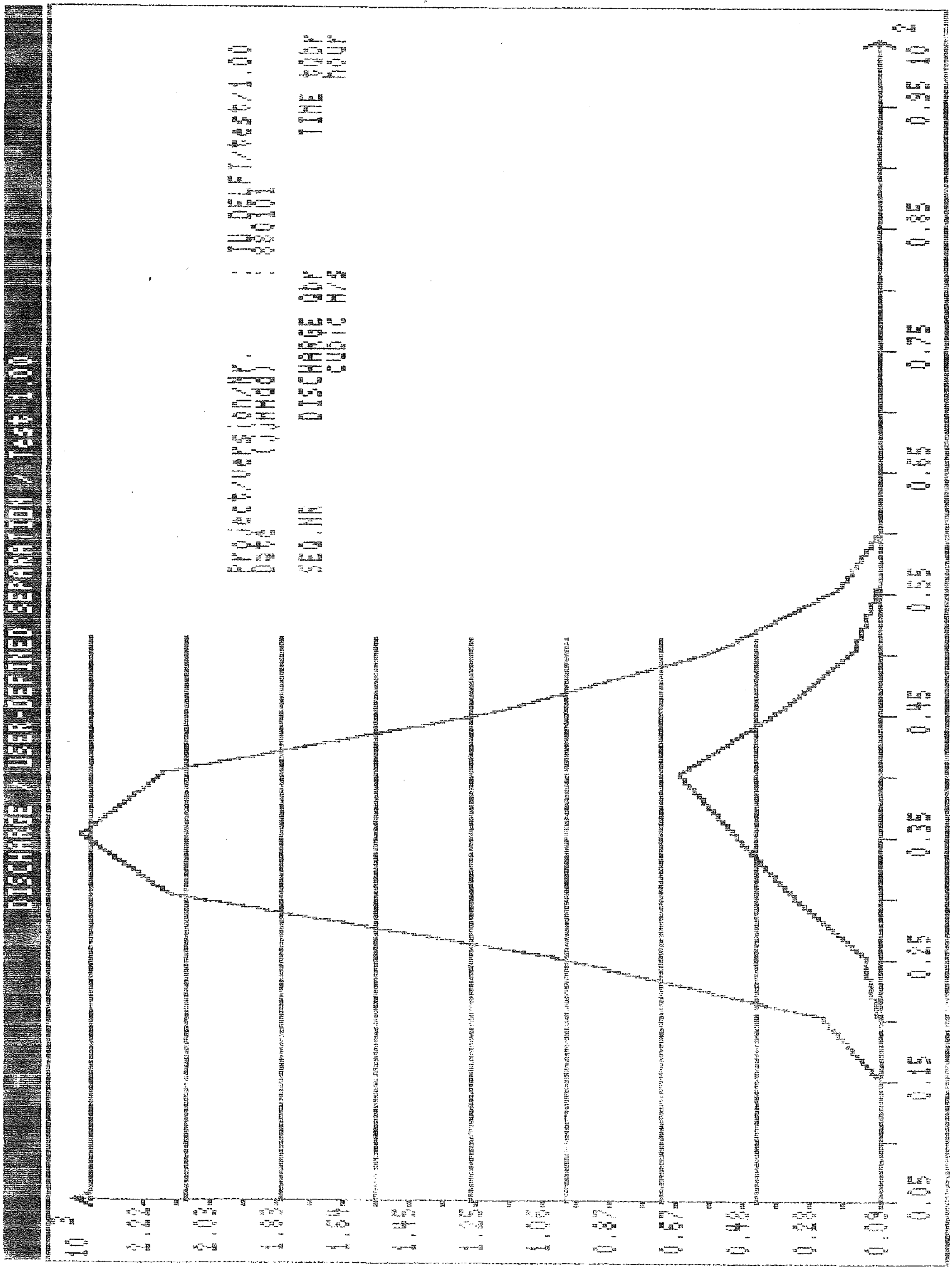


Fig. 8

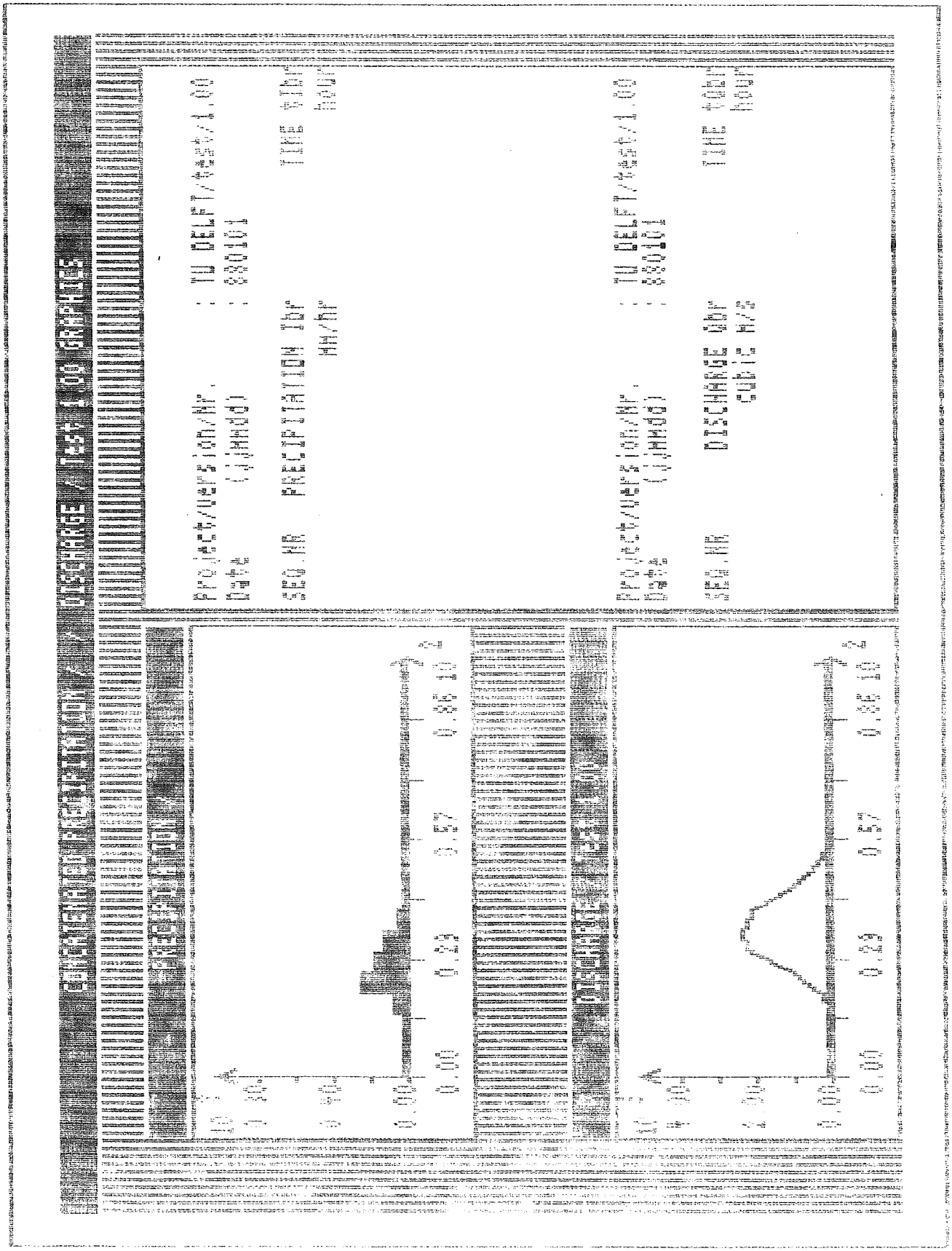


Fig. 9

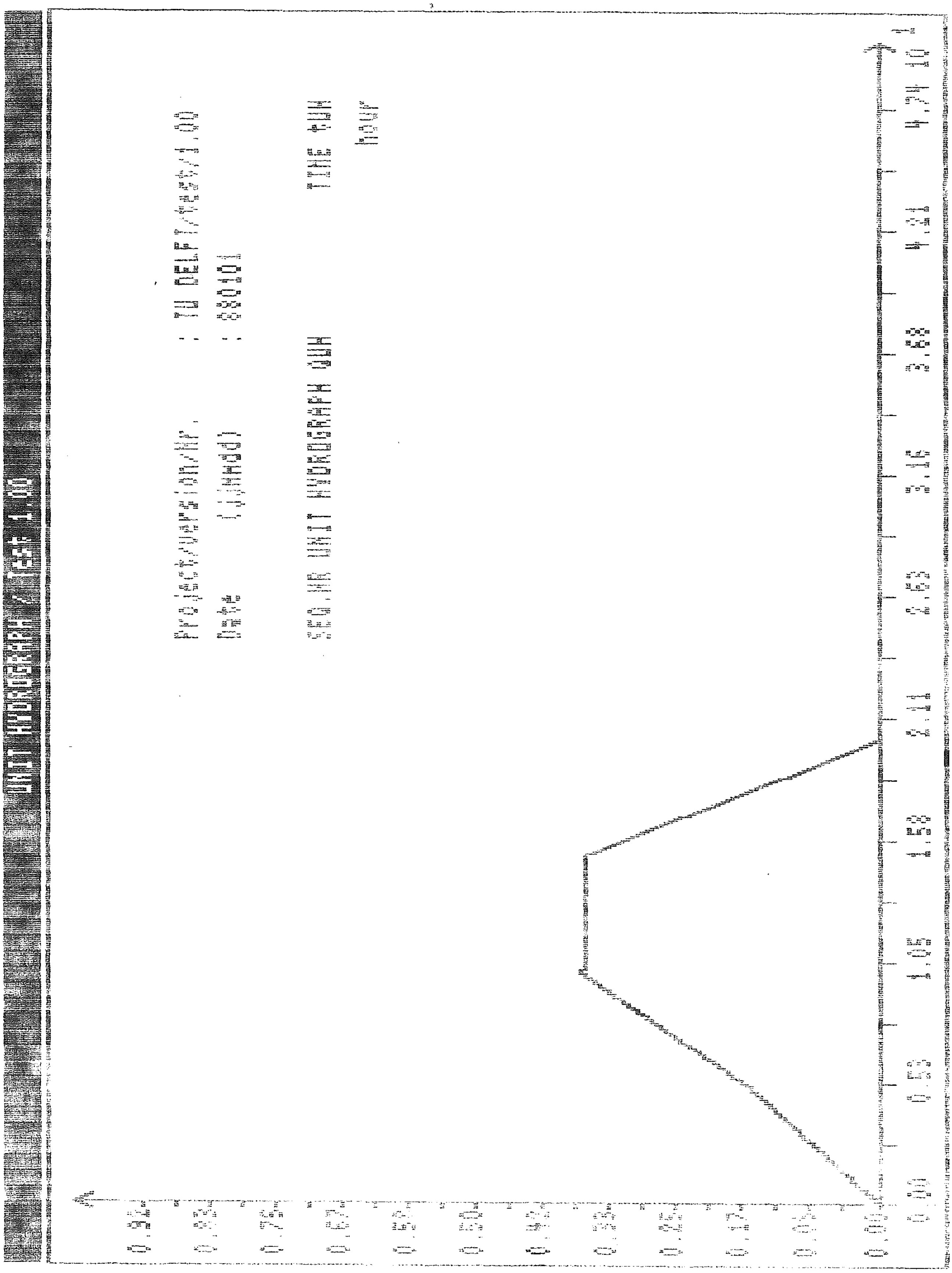

UNIT HYDROGRAPH
Project/Version/Nr.
: TU DELFT
Date (yymmdd)
: 880401
SCS UNIT HYDROGRAPH UUH
TIME UUH
hour

Fig. 10

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	PRECIPITATION Ibr mm/hr	TIME tIbr hour
01	.00	5
02	.00	10
03	.00	15
04	9.35	20
05	26.53	25
06	17.62	30
07	13.17	35
08	5.53	40
09	.00	45
10	.00	50
11	.00	55
12	.00	60
13	.00	65
14	.00	70
15	.00	75
16	.00	80
17	.00	85
18	.00	90
19	.00	95
20	.00	100

Project/Version/Nr. : HYDROLIN/test/1.00
Date (jjmmdd) : 880216

SEQ.NR	PHI-INDEX LOSS Iphi mm/hour	TIME tIphi hour
1	0.000	5
2	0.000	10
3	0.000	15
4	2.480	20
5	2.480	25
6	2.480	30
7	2.480	35
8	2.480	40
9	0.000	45
10	0.000	50
11	0.000	55
12	0.000	60
13	0.000	65
14	0.000	70
15	0.000	75
16	0.000	80
17	0.000	85
18	0.000	90
19	0.000	95
20	0.000	100

VOORBEELD / NIET "WIPED"

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	PRECIPITATION Ibr mm/hr	TIME tIbr hour
01	.00	5
02	.00	10
03	.00	15
04	5.00	20
05	.00	25
06	.00	30
07	.00	35
08	.00	40
09	.00	45
10	.00	50
11	.00	55
12	.00	60
13	.00	65
14	.00	70
15	.00	75
16	.00	80
17	.00	85
18	.00	90
19	.00	95
20	.00	100

VOORBEELD / NIET "WIPED"

Project/Version/Nr. : HYDROLIN/test/1.00
Date (jjmmdd) : 880212

SEQ.NR	PRECIPITATION Ibr mm/hour	TIME tIbr hour
1	9.350	5
2	26.530	10
3	17.620	15
4	13.170	20
5	5.530	25

VOORBEELD / "WIPED"

Project/Version/Nr. : HYDROLIN/test/1.00
Date (jjmmdd) : 880216

SEQ.NR	USERDEF LOSS Iusr mm/hour	TIME tIusr hour
1	0.000	5
2	0.000	10
3	0.000	15
4	0.000	20
5	0.000	25
6	0.000	30
7	0.000	35
8	0.000	40
9	0.000	45
10	0.000	50
11	0.000	55
12	0.000	60
13	0.000	65
14	0.000	70
15	0.000	75
16	0.000	80
17	0.000	85
18	0.000	90
19	0.000	95
20	0.000	100

VOORBEELD / INITIALISATIE

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	DISCHARGE Qbr cubic m/s	TIME tQbr hour
01	93.737	5
02	92.199	10
03	92.646	15
04	258.017	20
05	1045.862	25
06	2139.401	30
07	2412.913	35
08	2169.489	40
09	1206.810	45
10	575.472	50
11	215.298	55
12	91.482	60
13	91.696	65
14	92.932	70
15	92.555	75
16	91.555	80
17	93.829	85
18	93.708	90
19	92.512	95
20	91.283	100

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	DISCHARGE Qbr cubic m/s	TIME tQbr hour
01	93.737	5
02	92.199	10
03	92.646	15
04	98.017	20
05	145.862	25
06	339.401	30
07	512.913	35
08	669.489	40
09	406.810	45
10	175.472	50
11	115.298	55
12	91.482	60
13	91.696	65
14	92.932	70
15	92.555	75
16	91.555	80
17	93.829	85
18	93.708	90
19	92.512	95
20	91.283	100

VOORBEELD / NIET DIRECTE AFVOER

Project/version/Nr. : TU DELFT/test/1.00
Date (jjmmdd) : 880101

SEQ.NR	UNIT HYDROGRAPH QUH	TIME tUH hour
01	0.0000	0
02	0.1650	5
03	0.3735	10
04	0.3714	15
05	0.0000	20

Menu	:
Optie	:
Procedure	:

Omschrijving :

```
PROGRAM Directory;

(* HYDROLIN 1.00 extension *)

VAR
  Path: STRING[64];
  Ch  : Char;

BEGIN
  Ch := '1';    { initialize loop variable }
  Repeat
    IF Upcase(Ch) IN ['1', 'M', '2', 'R', '3', 'C', '0', 'Ø'] THEN
      BEGIN
        ClrScr;
        GetDir(0,Path);
        WriteLn('Current directory is ',Path);
        Writeln;
        WriteLn('Choose option: ');
        WriteLn('  1: Make a directory');
        WriteLn('  2: Remove a directory');
        WriteLn('  3: Change the current directory');
        WriteLn('  0: Quit');
        Writeln;
        Write('Option: ');
        Read(Kbd,Ch);

        {$I-}
        Case Upcase(Ch) OF
          '1','M': BEGIN
                     WriteLn('Make');
                     Write('Make what directory? ');
                     Readln(path);
                     MkDir(Path);
                   END;
          '2','R': BEGIN
                     WriteLn('Remove');
                     Write('Remove what directory? ');
                     Readln(path);
                     RmDir(Path);
                   END;
          '3','C': BEGIN
                     WriteLn('Change');
                     Writeln;
                     Write('Change to what directory? ');
                     Readln(path);
                     ChDir(Path);
                   END;
          '0','Ø': WriteLn('Quit');
         ELSE
        END; { case }

        {$I+}
        IF IOResult<>0 THEN
           BEGIN
             Write('*** Error: ', path);
             Delay(3000);
           END;
      END { if }
```

```
  ELSE
    Read(kbd, ch)
  UNTIL Upcase(Ch) IN ['Ø','0', #27];
END.
```

```
echo off
dir %1 | sort > lpt1:
```

```
{=============================================================}
PROGRAM QuickVideo; (* 870901 TU/D/HYDROLOGY  CP/M-80 *)
{=============================================================}

VAR Video           : Integer Absolute $F000;
    TPA             : Integer Absolute $BF00;
    R,S,T           : Integer;

{$U+}

BEGIN
R:=0;
 REPEAT
  FOR S:=1 TO 32 DO
  BEGIN
   FOR T:=1 TO 80 DO
   Write('*');
  END;
  Move(Video,TPA,4000);
  ClrScr;
  Delay(500);
  Move(TPA,Video,4000);
 UNTIL R=1;
END.
```

```
PROGRAM UsePointers; (* Example program M.J. Vos *)

(* HYDROLIN 1.00 extension *)

CONST N=2000;

TYPE OneDim     = ARRAY[1..N] OF Real;
     PtrOneDim  = ^OneDim;
     TwoDim     = ARRAY[1..N] OF PtrOneDim;
     PtrTwoDim  = ^TwoDim;

VAR  A          : PtrTwoDim;
     I,J        : Integer;

BEGIN
 New(A);
 FOR I:=1 TO N DO New(A^[I]);
 FOR I:=1 TO N
 BEGIN
  FOR J:=1 TO N DO
  BEGIN
   A^[I]^[J]:=1.0;
   WriteLn('i , j, A[I,J] = ',I:6,J:6,'':6,A^[I]^[J]:12);
  END;
 END;
END.
```

```
PROGRAM GetDate;

(* HYDROLIN 1.00 Extension *)

TYPE DateStr = STRING[10];

FUNCTION Date : DateStr;

TYPE
 RegPack = RECORD
   AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : Integer;
           END;
VAR
 RecPack       : RegPack;
 Month,Day     : STRING[2];
 Year          : STRING[4];
 Dx,Cx         : Integer;

BEGIN
 WITH RegPack DO
 BEGIN
  AX:=$2a SHL 8;
 END;
 MSDOS(RegPack);
 WITH RegPack DO
 BEGIN
  STR(CX,Year);
  STR(DX MOD 256,Day);
  STR(DX SHR 8,Month);
 END;
 Date:=Month+'/'+Day+'/'+Year;
END;  (* PROCEDURE Date *)

BEGIN(* Main *)
 WriteLn(Date);
END.
```

```
(*============================ MSDOS ==========================================*)
PROCEDURE M24(newmode:integer); (* newmode := 64 *)
(*======================= 640 X 400 pixels  ================================*)

TYPE
 Reg          = RECORD
                 CASE Boolean OF
                  True :(AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags:Integer);
                  False:(AL,AH,BL,BH,CL,CH,DL,DH          :Byte   );
                END;

VAR  regs    : Reg;

BEGIN
 regs.ax := newmode;
 Intr($10,regs);
END;

FUNCTION Baseaddress(pY : Integer) :Integer;
(*Baseaddress*)
(*Baseaddress = address of first pixel of Pixline pY*)

BEGIN
 Write('Olivetti M24 ?  ');LowVideo;Write('(J/N)  ');
 Read(Ch);WriteLn;IF Ja THEN
 BEGIN
  Baseaddress := 8192*(pY MOD 4) + 80*(pY DIV 4);
(*Baseaddress := (pY AND 3) SHL 13 + (pY AND -4) SHL 4 +
  (pY AND -4) SHL 2;*)
 END ELSE
 BEGIN
(*Baseaddress := (pY AND 1) SHL 13 + (pY AND -2) SHL 5 +
  (pY AND -2) SHL 3;*)
 Baseaddress := 8192*(pY MOD 2) + 80*(pY DIV 2);
 END;
END; (* Baseaddress *)


PROCEDURE SetPix1(pX,pY :Integer);

TYPE
 Reg          = RECORD
                 CASE Boolean OF
                  True :(AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags:Integer);
                  False:(AL,AH,BL,BH,CL,CH,DL,DH          :Byte   );
                END;

 VAR  Dummy : Integer;
      regs  : Reg;

BEGIN
(*Dummy := Baseaddress(pY)+pX SHR 3;
 Mem[ScreenBase:Dummy] := Mem[ScreenBase:Dummy]
 OR (128 SHR(pX AND 7));
 BEGIN*)
  regs.ax := $C01;
  regs.cx := pX;
  regs.dx := pY;
```

```
  Intr($10,regs);
(*END;*)
END; (* SetPix1 *)


PROCEDURE M24XY(X,Y : Integer);

TYPE
 Reg          = RECORD
                 CASE Boolean OF
                  True :(AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags:Integer);
                  False:(AL,AH,BL,BH,CL,CH,DL,DH          :Byte   );
                END;

VAR  regs : Reg;

BEGIN
 regs.ax := $200;
 regs.bx := 0;
 regs.dx := (Y-1)*256+X-1;
 Intr($10,regs);
END; (* M24XY *)


PROCEDURE M24draw(X1,Y1,X2,Y2 : Integer);

VAR  X,Y,deltaX,deltaY,Xstep,Ystep,direction : Integer;

BEGIN
 X := X1;
 Y := Y1;
 Xstep := 1;
 Ystep := 1;
 IF X1 > X2 THEN Xstep := -1 ELSE Xstep := 1;
 IF Y1 > Y2 THEN Ystep := -1 ELSE Ystep := 1;
 deltaX := Abs(X2-X1);
 deltaY := Abs(Y2-Y1);
 IF deltaX = 0 THEN direction := -1 ELSE direction := 0;
 WHILE NOT ((X = X2) AND (Y = Y2)) DO
 BEGIN
  SetPix1(X,Y);
  IF direction < 0 THEN
  BEGIN
   Y := Y + Ystep;
   direction := direction + deltaX
  END ELSE
  BEGIN
   X := X + Xstep;
   direction := direction - deltaY;
  END
 END
END; (* M24draw *)

(*END ============================== M24.INC ========================== *)
```

```
program SortExampleOne {Customer File};

type
  CustRec = record
              Number: integer;
              Name:   string[30];
              Addr:   string[20];
              City:   string[12];
              State:  string[3];
              Zip:    string[5];
            end;

var
  CustFile: file of CustRec;
  Customer: CustRec;

{$I SORT.BOX}

procedure Inp; {this procedure is forward declared in SORT.BOX}
begin
  repeat
    Read(CustFile,Customer);
    SortRelease(Customer);
  until EOF(CustFile);
end;

function Less; {this boolean function has two parameters, X and Y}
               {and is forward declared in SORT.BOX}
var
  FirstCust:  CustRec absolute X;
  SecondCust: CustRec absolute Y;
begin
  Less := FirstCust.Number < SecondCust.Number;
end;

procedure OutP;
var
  I:  Integer;
begin
  repeat
    SortReturn(Customer);
    with Customer do
    begin
      Write(Number, ' ',Name,' ');
      for I := Length(Name) to 30 do Write(' ');
      Write(Addr);
      for I := Length(Addr) to 20 do Write(' ');
      Write(City);
      for I := Length(City) to 12 do Write(' ');
      WriteLn(State,' ',Zip);
    end;
  until SortEOS;
end;

begin {program SortExampleOne}
  Assign(CustFile,'CUSTOMER.DTA');
  Reset(Custfile);
  WriteLn(TurboSort(SizeOf(CustRec)));
```

```
end.
```

```
program SortExampleTwo { Customer File and Stock File };

type
  CustRec = record
              Number: integer;
              Name:   string[30];
              Addr:   string[20];
              City:   string[12];
              State:  string[3];
              Zip:    string[5];
            end;

  ItemRec = record
              Number:  integer;
              Descrip: string[30];
              InStock: integer;
              Price:   real;
            end;

var
  CustFile:   file of CustRec;
  Customer:   CustRec;
  StockFile:  file of ItemRec;
  Item:       ItemRec;
  Choice:     Char;

{$I SORT.BOX}

procedure Inp; {this procedure is forward declared in SORT.BOX}
begin
  case Choice of
    'C': begin
           repeat
             Read(CustFile,Customer);
             SortRelease(Customer);
           until EOF(CustFile);
         end;
    'S': begin
           repeat
             Read(StockFile,Item);
             SortRelease(Item);
           until EOF(StockFile);
         end;
  end; {case}
end;

function Less; {this boolean function has two parameters, X and Y}
               {and is forward declared in SORT.BOX}
var
  FirstCust:  CustRec absolute X;
  SecondCust: CustRec absolute Y;
  FirstItem:  ItemRec absolute X;
  SecondItem: ItemRec absolute Y;
begin
  case Choice of
    'C': Less := FirstCust.Number < SecondCust.Number;
    'S': Less := (FirstItem.InStock < SecondItem.InStock) or
                 ((FirstItem.InStock = SecondItem.InStock) and
```

```
                    (FirstItem.Price < SecondItem.Price));
  end;
end;

procedure OutP;
var
  I:  Integer;
begin
  case Choice of
    'C': begin
           repeat
             SortReturn(Customer);
             with Customer do
             begin
               Write(Number, ' ',Name,' ');
               for I := Length(Name) to 30 do Write(' ');
               Write(Addr);
               for I := Length(Addr) to 20 do Write(' ');
               Write(City);
               for I := Length(City) to 12 do Write(' ');
               WriteLn(State,' ',Zip);
             end;
           until SortEOS;
         end;
    'S': begin
           repeat
             SortReturn(Item);
             with Item do
             begin
               Write(Number, ' ',Descrip,' ');
               for I := Length(Descrip) to 30 do Write(' ');
               WriteLn(InStock:5,Price:8:2);
             end;
           until SortEOS;
         end;
  end; {case}
end;

begin {program SortExampleOne}
  Write('Sort Customers or Stock? (enter C or S): ');
  repeat
    read(Kbd,Choice);
    Choice := UpCase(Choice);
  until Choice in ['C','S'];
  WriteLn(Choice);
  case Choice of
    'C': begin
           Assign(CustFile,'CUSTOMER.DTA');
           Reset(CustFile);
           WriteLn(TurboSort(SizeOf(CustRec)));
         end;
    'S': begin
           Assign(StockFile,'STOCK.DTA');
           Reset(StockFile);
           WriteLn(TurboSort(SizeOf(ItemRec)));
         end;
  end; {case}
end.
```