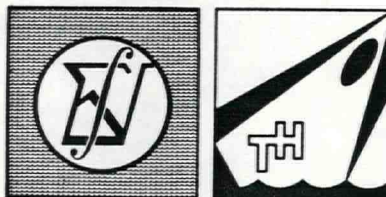


TECHNISCHE HOGESCHOOL DELFT
AFDELING DER SCHEEPSBOUW- EN SCHEEPVAARTKUNDE
CENTRALE WERKGROEP WISKUNDE

Rapport No. 4



- UNIVERSEEL PLOTPROGRAMMA

- Ing. A.P.de Zwaan en ing. J.G.L.Pijfers
Herzien door ing. W.B.Tinbergen

- november 1982

Delft University of Technology
Ship Hydromechanics Laboratory
Mekelweg 2
2628 CD DELFT
The Netherlands
Phone 015 -786882

INHOUD

blz.

1.	ALGEMEEN	
1.1.	Programma gegevens	1
1.2.	Doel	1
1.3.	Opzet	1
2.	ORGANISATIE VAN HET PROGRAMMA	
2.1.	Assen - schalen en raster	2
2.2.	Lijnen en punten in de grafiek	2
2.3.	Tekst onder de grafiek	2
2.4.	Benodigde randapparatuur	2
3.	BESTANDSORGANISATIE	
3.1.	Gebruik plotprogramma	3
3.2.	Betekenis en invoer van de formele parameters	3
3.2.1.	Stroomdiagram	6
3.3.	Invoer van tekst	9
3.3.1.	Algemeen	9
3.3.2.	Mogelijkheden van tekstinvoer	9
3.3.3.	Betekenis van verschillende symbolen	9
3.3.4.	Volgorde van opgegeven tekst bij grafiek	9
I.	BESCHRIJVING VAN DE GEBRUIKTE PROCEDURES IN HET PROGRAMMA CWWPLT	
I.1.	Procedure voor het plotten van tekst	11
	T code List	43
I.1.1.	Doel	11
I.1.2.	Algemeen	11
I.1.3.	Gebruik van de procedure	11
I.1.4.	Stroomdiagram	12
I.2.	Procedure voor het bepalen van minima en maxima	18
	MINMAX List	47
I.2.1.	Doel	18
I.2.2.	Algemeen	18
I.2.3.	Gebruik van de procedure	18
I.2.4.	Stroomdiagram	19
I.3.	Procedure voor Lineaire schaalverdeling van de as//x-as plotter	20
	SCAL XI List	48
I.3.1.	Doel	20
I.3.2.	Algemeen	20
I.3.3.	Gebruik van de procedure	20
I.3.4.	Stroomdiagram	21
I.4.	Procedure voor lineaire schaalverdeling van de as//y-as plotter	23
	SCAL YI List	49
I.4.1.	Doel	23
I.4.2.	Algemeen	23
I.4.3.	Gebruik van de procedure	23
I.4.4.	Stroomdiagram	24
I.5.	Procedure voor een lineaire gradenverdeling van de as//x-as plotter	26
	SCAL X2 List	50
I.5.1.	Doel	26
I.5.2.	Algemeen	26
I.5.3.	Gebruik van de procedure	26
I.5.4.	Stroomdiagram	27
I.6.	Procedure voor een lieaire gradenverdeling van de as//y-as plotter	28
	SCAL Y2 List	51
I.6.1.	Doel	28
I.6.2.	Algemeen	28
I.6.3.	Gebruik van de procedure	28
I.6.4.	Stroomdiagram	29

INHOUD

blz.

I.7.	Procedure voor het bepalen van de schaal voor x- of y-as	30
	TRANS List	30
	I.7.1. Doel	30
	I.7.2. Algemeen	30
	I.7.3. Gebruik van de procedure	30
	I.7.4. Stroomdiagram	31
I.8.	Procedure voor het plotten van de waarden langs de x-as	32
	XAS List	53
	I.8.1. Doel	32
	I.8.2. Algemeen	32
	I.8.3. Gebruik van de procedure	32
	I.8.4. Stroomdiagram	33
I.9.	Procedure voor het plotten van de waarden langs de y-as	34
	YAS List	54
	I.9.1. Doel	34
	I.9.2. Algemeen	34
	I.9.3. Gebruik van de procedure	34
	I.9.4. Stroomdiagram	35
I.10.	Procedure voor plaats van schaalwaarden	36
	SCHUIF List	55
	I.10.1. Doel	36
	I.10.2. Algemeen	36
	I.10.3. Gebruik van de procedure	36
	I.10.4. Stroomdiagram	37
I.11.	Procedure voor bepaling van aantal decimalen achter de komma	38
	NDC List	56
	I.11.1. Doel	38
	I.11.2. Algemeen	38
	I.11.3. Gebruik van de procedure	38
	I.11.4. Stroomdiagram	39
I.12.	Procedure voor het plotten van lijnen	40
	LINPLT List	57
	I.12.1. Doel	40
	I.12.2. Algemeen	40
	I.12.3. Gebruik van de procedure	40
	I.12.4. Stroomdiagram	41
I.13.	Gebruikers symbolenlijst	42
	LISTINGS VAN DE GEBRUIKTE <u>PROCEDURES</u> IN HET PROGRAMMA CWWPLT	43
	LISTINGS VAN HET TOTALE <u>PROGRAMMA CWWPLT</u>	58

1. ALGEMEEN.

1.1 Programma gegevens

- a. Taal : Algol 60
- b. Geheugen : 160 K
- c. Rekening : n.v.t.
- d. Naam procedure : CWWPLT

1.2 Doel.

Het plotten van grafieken op een door de gebruiker op te geven formaat. (Standaardformaat is A4). Er wordt zodanig rekening gehouden met de inbindrand en andere vrije randen, dat de grafiek direct te gebruiken is om te copieëren en in te binden in een rapport of publicatie.

Alle in een grafiek voorkomende gevallen, zoals bij-schriften bij de assen, verklaring van lijnen en symbolen, figurnummers met tekst onder de grafiek, gestrookte lijnen, stippellijnen enz. kunnen geplot worden, zodat zonder latere aanvullingen met de hand of met lettermallen een complete grafiek verkregen kan worden. Een beperking is echter, dat voor bij-schriften bij de assen en tekst onder de grafiek alleen gebruik kan worden gemaakt van hoofdletters (zie bijlage: symbolen-tafel). In verband met deze beperking wordt verwezen naar een rapport van A.P. de Zwaan:

"Proposal of Symbols and Quantities in Shipbuilding"
Based on the recommendations of the I.T.T.C., the I.S.S.C. and the International System of Units.

Hierin is in opdracht van de commissie van wetenschaps-beoefening speciaal voor de computer een symbolenlijst opgenomen, waarin alleen hoofdletters voorkomen. Bij gebruik van de plotter is deze symbolenlijst zeer aan te bevelen.

1.3 Opzet.

De procedure "CWWPLT" is opgebouwd uit verschillende andere procedures, die elk een specifieke handeling verrichten. Voor de beschrijving van deze procedures, zie de bijlagen.

Procedure "CWWPLT" maakt tevens gebruik van procedures uit de Fortranbibliotheek.

Voor het gebruik van procedure "CWWPLT" moeten de volgende bibliotheken worden aangesloten:

- a. SBAL.LIBCWW
- b. SYS1.FORTLIB
- c. SYS2.PLOTLIBF

2. ORGANISATIE VAN HET PROGRAMMA.

2.1 Assen, schalen en rasters.

De gebruiker heeft de volgende keuzemogelijkheden:

- a) De x-as op de lange of korte kant van het A4-formaat.
 - b) Voor de x-as en y-as onafhankelijk van elkaar verkrijgen van:
 1. lineaire schaalverdeling
 2. schaalverdeling in graden (fasehoeken)
- In voorbereiding zijn nog:
3. reciproke schaalverdeling
 4. logaritmische schaalverdeling
 5. polaire schaalverdeling (x- en y-as niet onafhankelijk).
- c) Men kan zelf de schaal opgeven of het programma de schaal laten bepalen.
 - d) Zowel bij de x-as als bij de y-as kan een bijschrift geplaatst worden van één regel, waarbij men alle symbolen uit de symbolenlijst (zie bijlage) kan gebruiken.
 - e) Men kan het programma wel of geen raster binnen een grafiek laten plotten.

2.2 Lijnen en punten in de grafiek.

De gebruiker heeft de volgende keuzemogelijkheden:

- a) Het plotten van punten met gecentreerde symbolen (codenummers 1 t/m 13 uit de symbolentafel) zonder deze te verbinden.
- b) Het plotten van punten met gecentreerde symbolen en deze punten verbinden met rechte lijnstukjes.
- c) Het plotten van punten met gecentreerde symbolen en deze punten door een strokende lijn met elkaar verbinden.
- d) Het plotten van niveaulijnen.
- e) Een lijn plotten, bestaande uit rechte lijnstukjes tussen de punten, zonder plotten van symbolen op die punten.
- f) Een strokende lijn plotten door een aantal punten, zonder plotten van symbolen op die punten.
- g) Op een van de bovengenoemde manieren een stippelijntje plotten.

2.3 Tekst onder de grafiek.

Indien de x-as op de lange kant van het A4-formaat gekozen wordt, dan is het mogelijk om onder de x-as maximaal 5 regels tekst te laten schrijven van elk maximaal 100 karakters.

Indien de x-as op de korte kant van het A4-formaat gekozen wordt, dan is het mogelijk om onder de x-as maximaal 9 regels tekst te laten schrijven van elk maximaal 75 karakters.

Men kan deze ruimte gebruiken voor het verklaren van lijnen en symbolen in de grafiek, voor figuurnummers met bijbehorende tekst, enz. Alle in de symbolentafel voorkomende symbolen zijn hiervoor te gebruiken met uitzondering van de codenummers 62(), 63() en 125('). Deze drie tekens hebben voor dit programma een andere

betekenis gekregen. Voor de invoer van tekst zie hoofdstuk 3.

2.4 Benodigde randapparatuur.

a) Plotter De plotter wordt geactiveerd met de stuurkaart: //GO.PLTDS DD SYSOUT=(I,, \$SPL).

3. BESTANDSORGANISATIE.

3.1 Gebruik plotprogramma.

De kop van de procedure ziet er als volgt uit:

```
'PROCEDURE' CWWPLT (I, J, K, X, Y, PAXY, PBX, PBY, PCX, PDX, PCY,
PDY, PEX, PXMIN, PXMAX, PEY, PYMIN, PYMAX, PPP, FAC);
```

```
'INTEGER' 'ARRAY' I, J, PAXY, PBX, PBY, PDX, PDY, PEX, PEY, PPP;
```

```
'REAL' FAC; 'INTEGER' K;
```

```
'REAL' 'ARRAY' X, Y, PCX, PCY, PXMIN, PXMAX, PYMIN, PYMAX;
```

Aan de declaraties van het programma moet worden toegevoegd: 'PROCEDURE' CWWPLT; 'CODE';

De procedure staat in de standaardbibliotheek van de "Centrale Werkgroep Wiskunde", genaamd:

SBAL.LIBCWW.

De volgende twee bibliotheken moeten ook worden aangesloten: SYS2.VFORTLIB en SYS2.PLOTLIBF.

3.2 Betekenis en invoer van de formele parameters.

Alle parameters zijn invoerparameters

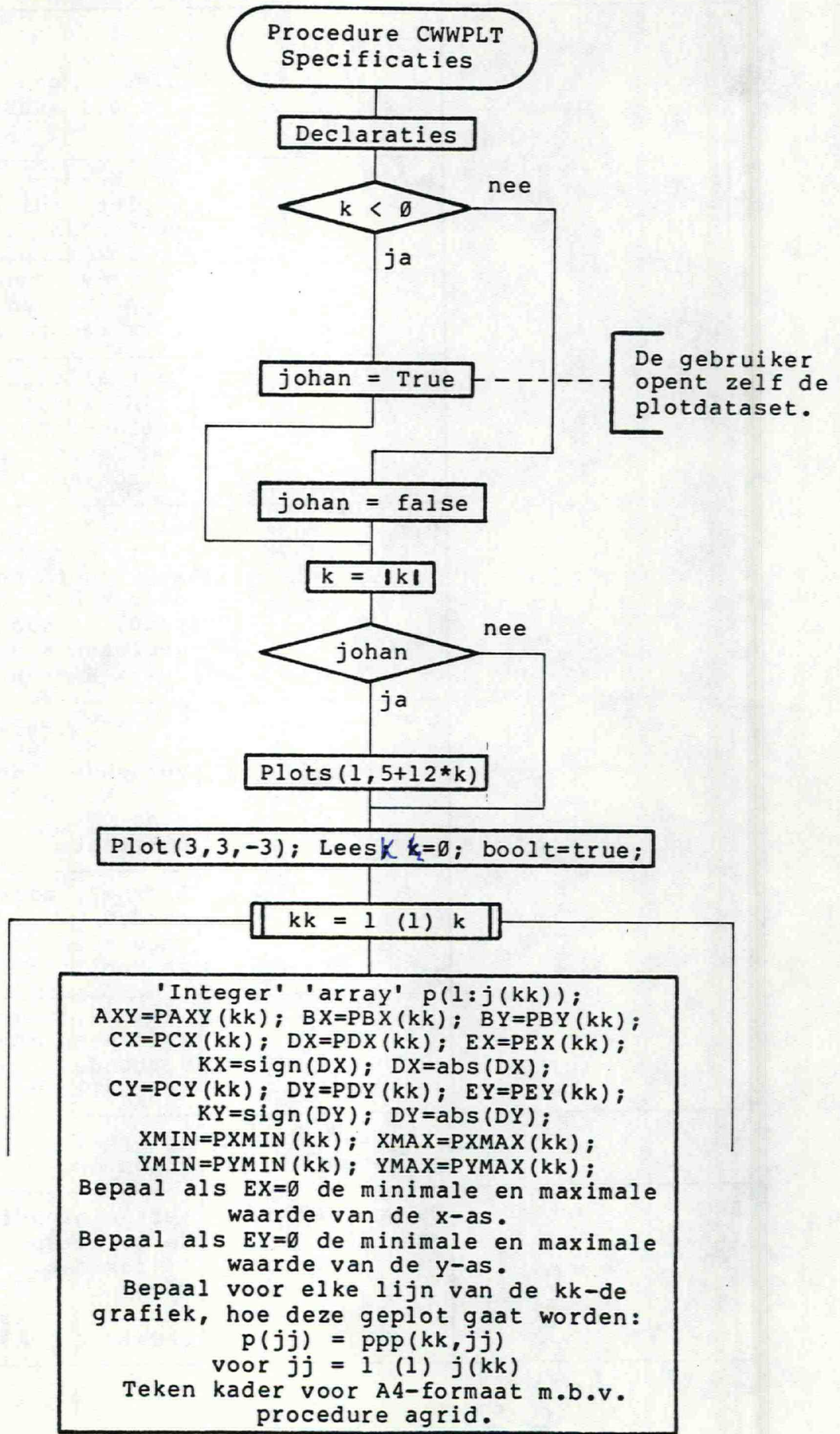
Parameterdeclaratie	Input	Omschrijving
I(/1:k,i:jm/) Integer array	I(/kk,jj/) met kk = 1,2,...k jj = 1,2,...j(/kk/)	jm is het grootste element uit het array J. I(/kk,jj/) is het aantal x-y coördinaten van de jj-de lijn in de kk-de grafiek.
J(/1:k/) Integer array	J(/kk/) met kk = 1,2,...k	J(/kk/) is het aantal lijnen in de kk-de grafiek.
K Integer	k	Aantal te plotten grafieken.
X(/1:k,l:jm,l:im/) Real array	X(/kk,jj,ii/) met kk = 1,2,...k jj = 1,2,...j(/kk/) ii = 1,2, ...I(/kk,jj/)	im is het grootste element uit het array I. X(/kk,jj,ii/) is de x-waarde van het ii-de punt in de jj-de lijn van de kk-de grafiek.
Y(/1:k,l:jm,l:im/) Real array	Y(/kk,jj,ii/) met kk = 1,2,...k jj = 1,2,...j(/kk/) ii = 1,2, ...I(/kk,jj/)	Y(/kk,jj,ii/) is de y-waarde van het ii-de punt in de jj-de lijn van de kk-de grafiek.

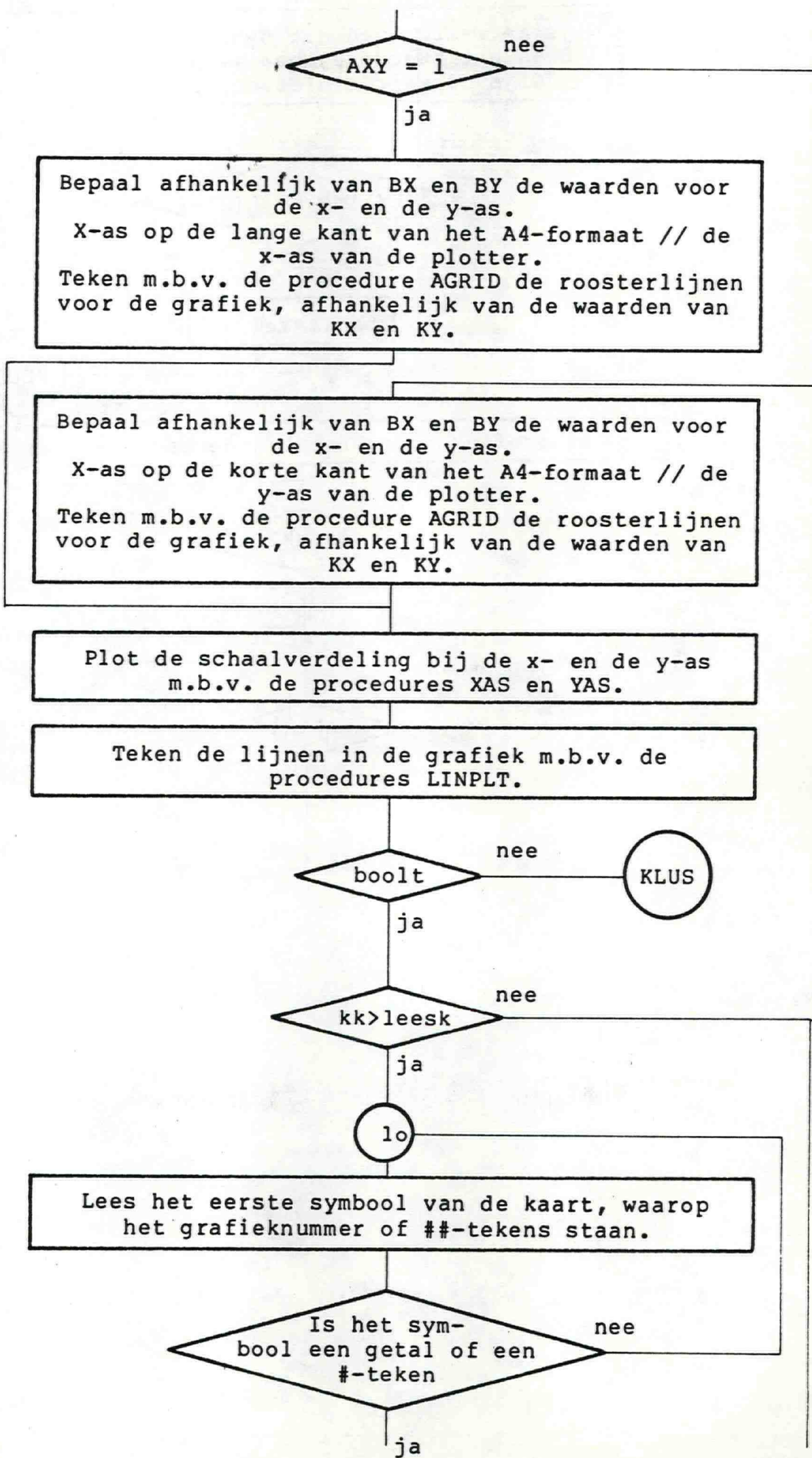
Parameterdeclaratie	Input	Omschrijving
PAXY(/1:k/) Integer array	1	X-as van de grafiek op de lange kant van het A4-formaat.
	2	X-as van de grafiek op de korte kant van het A4-formaat.
PBX(/1:k/) Integer array	1	X-as op lineaire schaal.
	2	X-as met schaalverdeling in graden.
PBY(/1:k/) Integer array	1	Y-as op lineaire schaal.
	2	Y-as met schaalverdeling in graden.
PCX(/1:k/) Real array	0	Het programma bepaalt zelf de schaal van de x-as.
	≠ 0	De ruimte tussen de roosterlijnen in de x-richting in cm.
PDX(/1:k/) Integer array	IPDX(/kk/)! ≠ 0	Getal dat het maximale aantal vakjes van de roosterlijnen in x-richting aangeeft voor de kk-de grafiek. Bij een negatieve waarde worden de roosterlijnen // x-as weggelaten. Alleen toegestaan als PCX(/kk/) = 0.
	0	
PCY(/1:k/) Real array	0	Het programma bepaalt zelf de schaal van de y-as.
	≠ 0	De ruimte tussen de roosterlijnen in de y-richting in cm.
PDY(/1:k/) Integer array	IPDY(/kk/)! ≠ 0	Getal dat het maximale aantal vakjes van de roosterlijnen in y-richting aangeeft voor de kk-de grafiek. Bij een negatieve waarde worden de roosterlijnen // y-as weggelaten. Alleen toegestaan als PCY(/kk/) = 0.
	0	

Parameterdeclaratie	Input	Omschrijving
PEX(/1:k/) Integer array	0	Programma bepaalt zelf de minimum en maximum x-waarde van de grafiek.
	1	Minimum en maximum x-waarde moeten door de gebruiker worden opgegeven.
PXMIN(/1:k/) Real array	PXMIN(/kk/) met kk = 1, 2, ...k	Minimum x-waarde van de x-as in de kk-de grafiek. Als PEX(/kk/)=0, dan PXMIN(/kk/)=0.
PXMAX(/1:k/) Real array	PXMAX(/kk/) met kk = 1, 2, ...k	Maximum x-waarde van de x-as in de kk-de grafiek. Als PEX(/kk/)=0, dan PXMAX(/kk/)=0.
PEY(/1:k/) Integer array	0	Programma bepaalt zelf de minimum en maximum y-waarde van de grafiek.
	1	Minimum en maximum y-waarde moeten door de gebruiker worden opgegeven.
PYMIN(/1:k/)	PYMIN(/kk/) met kk=1, 2, ...k	Minimum y-waarde van de y-as in de kk-de grafiek. Als PEY(/kk/)=0, dan PYMIN(/kk/)=0.
PYMAX(/1:k/)	PYMAX(/kk/) met kk=1, 2, ...k	Maximum y-waarde van de y-as in de kk-de grafiek. Als PEY(/kk/)=0, dan PYMAX(/kk/)=0.
PPP(/1:k,1:jm/) Integer array	PPP(/kk,jj/)= l0*m+n met kk=1, 2, ...k jj=1, 2, ..j(kk)	De waarde voor de jj-de lijn in de kk-de grafiek.
	m = 0	Er worden geen symbolen in de x-y punten geplot.
	m = 1	Elk x-y punt wordt geplot door een gecentreerd symbool. Voor jj=1 symbool code nr. 1. Voor jj=2 symbool code nr. 2 enz. Er zijn maximaal 12 gecentreerde symbolen.

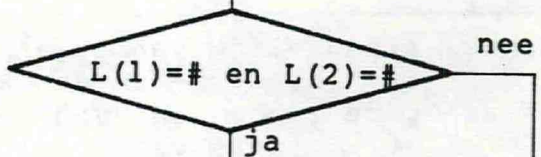
Parameterdeclaratie	Input	Omschrijving
	$m = i$	Elk i -de punt van het x - y array wordt geplot d.m.v. een gecentreerd symbool (verder als bij $m=1$).
	$n = 0$	Er worden geen lijnen geplot tussen de x - y punten in de grafiek.
	$n = 1$	De x - y punten van de lijn worden verbonden door rechte lijnstukjes.
	$n = 2$	Door alle punten van de lijn wordt d.m.v. een spline-fitting methode een vloeiende kromme getrokken.
	$n = 3$	Als $n=2$, maar nu wordt de lijn van het laatste x - y punt doorgetrokken naar het eerste x - y punt, zodat er een gesloten kromme ontstaat (b.v. niveaulijnen).
	$n = 6$	De x - y punten van de lijn worden om en om verbonden door rechte lijnstukjes, zodat er een stippellijn ontstaat.
	$n = 7$	Als $n=2$, maar de punten worden om en om verbonden, zodat er een stippellijn ontstaat.
	$n = 8$	Als $n=3$, maar de punten worden om en om verbonden, zodat er een stippellijn ontstaat.
Geef voor een stippellijn altijd een even aantal punten op.		
FAC real	De waarde van FAC	Verkleiningsfactor van de totale grafiek t.o.v. A4-formaat. FAC=1 A4-formaat. FAC= 2 A5-formaat. FAC=1/ 2 A3-formaat.

3.2.1 Stroomdiagram.





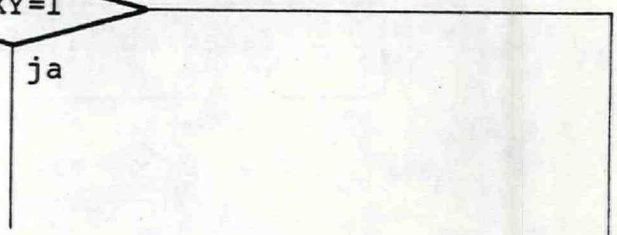
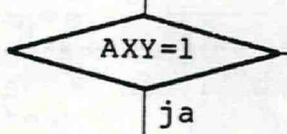
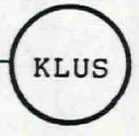
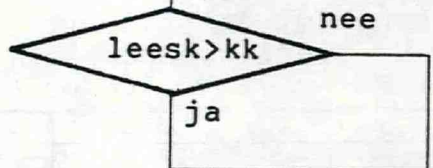
Wijs de waarde van dit symbool toe aan L(1)
Lees het tweede symbool van de kaart en
wijs de waarde hiervan toe aan L(2).



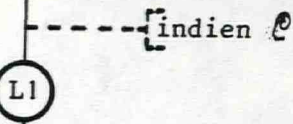
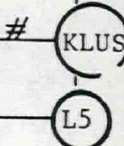
boolt=false;



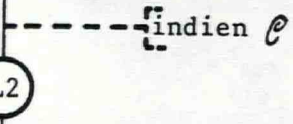
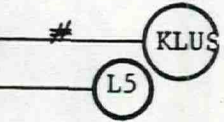
Bepaal grafieknummer waarbij de tekst komt: leesk



ALFX = 0; ALFY = 90; REG = 0
 Plot pijlen bij de x- en y-as m.b.v. de
 procedure AAROHD
 Tekst bij pijl x-as: TCODE(5.5,-.9,.3, ALFX,KLUS,L1,L5)

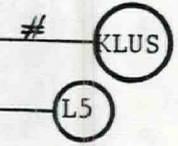


Tekst bij pijl y-as:
 TCODE(-1.6,5.5,.3, ALFY,KLUS,L2,L5)

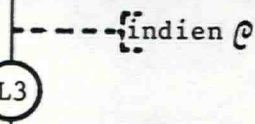
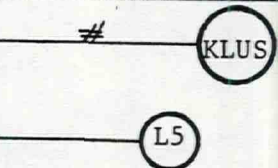


REG = REG + 1
 Tekst onder de x-as:
 TCODE(0,-1.6-(REG-1)*.47,.22,ALFX,KLUS,L2,L5)

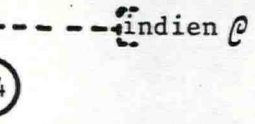
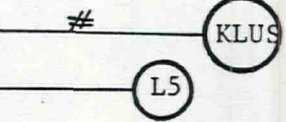
indien 0



ALFX=90, ALFY = 180, REG = 0
 Plot pijlen bij x- en y-as
 Tekst bij pijl x-as:
 TCODE(.9,5.5,.3,ALFX,KLUS,L3,L5)

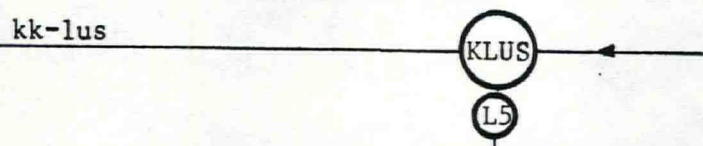
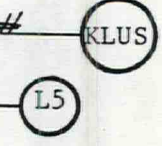


Tekst bij pijl y-as:
 TCODE(-5.5,-1.6,.3,ALFY,KLUS,L4,L5)



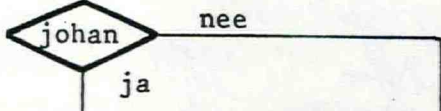
REG = REG + 1
 Tekst onder de x-as:
 TCODE(1.6+(REG-1)*.47,0,.22,ALFX,KLUS,L4,L5)

indien 0



Fout in tekst. Herstel deze fout.

einde kk-lus



Gebruiker sluit
 zelf plot dataset

Sluit plot dataset m.b.v. LASPLO.



3.3 Invoer van tekst.

3.3.1 Algemeen.

De tekst moet op kaart worden ingevoerd. Elke nieuwe regel moet op een nieuwe kaart beginnen.

Uit de symbolentafel (zie bijlage P42) blijkt, dat aan elk tekstsymbool een code nummer is toegekend. De symbolen die in de laatste vier kolommen staan komen voor op de ponsmachine. Het programma is zo ingericht, dat deze tekens niet in code behoeven te worden ingevuld, maar gewoon als tekst op kaart.

Deze leesbare tekst moet dan wel tussen quotes (') staan (zie hoofdstuk 2.3).

De andere niet op de ponsmachine voorkomende tekens moeten in code getallen worden opgegeven (de bovenste getallen uit de symbolenlijst), met elk teken gescheiden door een spatie of komma.

Elke regel moet, of deze nu in code of tekst is opgegeven, worden afgesloten met # of @.

3.3.2 Mogelijkheden van tekst invoer.

a) Tekst in code cijfers op kaart met elk symbool gescheiden door een komma of een spatie.

b) Tekst als leesbare tekst op kaart (tussen '')

c) Tekst met een enkele of dubbele index.

De index moet tussen de code cijfers 63 gezet worden.

Voorbeeld.

Enkele index: ALFA_{EFF} ⇒ 'ALFA' 63 'EFF' 63

Dubbele index: ALFA_{ωEFF}T ⇒ 'ALFA' *40,63 'EFF' 63,63 'T' 63

d) Tekst met een exponent.

De exponent moet tussen de code cijfers 62 gezet worden.

Voorbeeld. ALFA² ⇒ 'ALFA' 62 '2' 62

e) Een combinatie van a, b, c en d.

Voorbeeld.

$\omega \phi * ALFA_{EFF}^{20} : \sqrt{P*20}$

40, 63, 36, 63 '* ALFA' 62 '20' 62, 63 'EFF' 63, 63 'T' 63 ':'
57 'P*20' of

40, 63, 36, 63 '*ALFA' 63, 'EFF' 63, 63 'T' 63, 62 '20' 62 ':'
57 'P*20'

3.3.3 Betekenis van verschillende symbolen.

- @ - Nieuwe regel
- # - Einde tekst bij grafiek (afsluiting laatste regel)
- # # - Bij volgende grafieken c.q. bij geen enkele grafiek tekst.
- ' - Begin c.q. einde leesbare tekst voorkomend op de ponsmachine.
- 62⇌↑ - Begin c.q. einde exponent
- 63⇌↓ - Begin c.q. einde INDEX.

3.3.4 Volgorde van opgegeven tekst bij grafiek.

Kaart 1) Grafiek volgnummer > 1 bestaande uit tenminste 2 cijfers.

Volgnummer < 10: 01, 02, 03, 09.

Volgnummer > 10: 10, 11, 12,

Kaart 2) Tekst behorende bij de x-as)

Indien geen tekst: @

Kaart 3) Tekst behorende bij de y-as)

Indien geen tekst: @

Ruimte benodigd voor 1 tekst
symbool is 3 mm.

Kaart 4) Tekst onder x-as

Indien geen tekst dan moet de laatste regel afgesloten worden met #

Ruimte benodigd voor 1 tekst symbool is 2.2 mm.

5) Indien bij geen enkele grafiek meer tekst geef # # , vervolg anders met punt 1 enz.

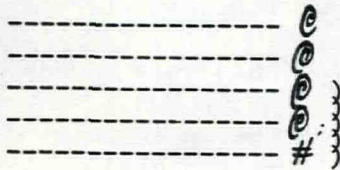
Opmerking

Alleen de invoer van de tekst behoeft te worden opgegeven voor de grafieken met bijbehorende volgnummer.

Voorbeeld. Stel van de 15 grafieken zijn er 3 voorzien van tekst.
Grafiek volgnummer 1, 13 en 14.

De invoer is dan als volgt:

01

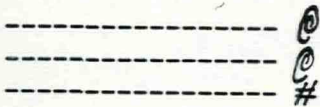


x-as

y-as

Tekst onder de x-as

13



x-as

y-as

Tekst onder de x-as

14



x-as

y-as

Tekst onder de x-as.

I. BESCHRIJVING VAN DE GEBRUIKTE PROCEDURES IN HET PROGRAMMA CWWPLT - T CODE

I.1 Procedure voor het plotten van tekst.

I.1.1 Doel.

Het inlezen en plotten van één of meer regels tekst.

I.1.2 Algemeen.

De naam van de procedure is TCODE.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de "Centrale Werkgroep Wiskunde".

Voor de mogelijkheden van tekst zie hoofdstuk 3.

I.1.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

```
'PROCEDURE' TCODE (X,Y,Q,ALFA,KLUS,TEST,SPRING);
```

```
'VALUE' Q; 'REAL' X,Y,Q; 'INTEGER' ALFA;
```

```
'LABEL' KLUS,TEST,SPRING;
```

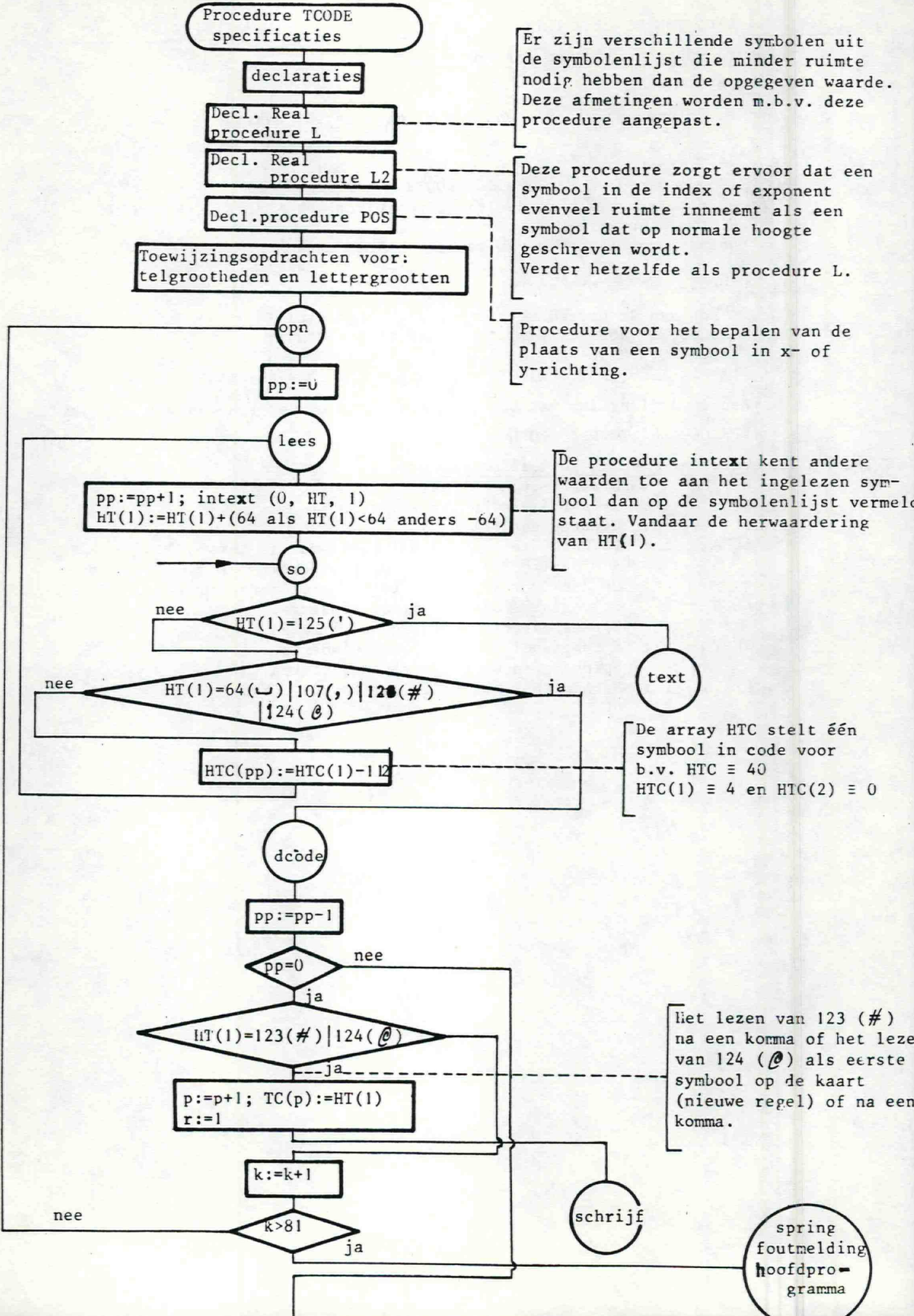
Aan de declaraties van het programma moet worden toegevoegd:

```
'PROCEDURE' TCODE; 'CODE';
```

Betekenis van de formele parameters:

X	i	:	X-waarde van het eerste symbool
Y	i	:	Y-waarde van het eerste symbool
Q	i	:	Grootte van het symbool in cm.
ALFA	i	:	De hoek die het symbool maakt met de x-as (de horizontale as van de plotter!) De hoek wordt gemeten tegen de wijzers van de klok in.
KLUS	i	:	Spronglabel als tekst ten einde is. Sprongopdracht reageert op # teken.
TEST	i	:	Spronglabel als regel ten einde is. Sprongopdracht reageert op @ teken.
SPRING	i	:	Spronglabel als de invoer van de tekst fout is.

I.1.4 Stroomdiagram.



Er zijn verschillende symbolen uit de symbolenlijst die minder ruimte nodig hebben dan de opgegeven waarde. Deze afmetingen worden m.b.v. deze procedure aangepast.

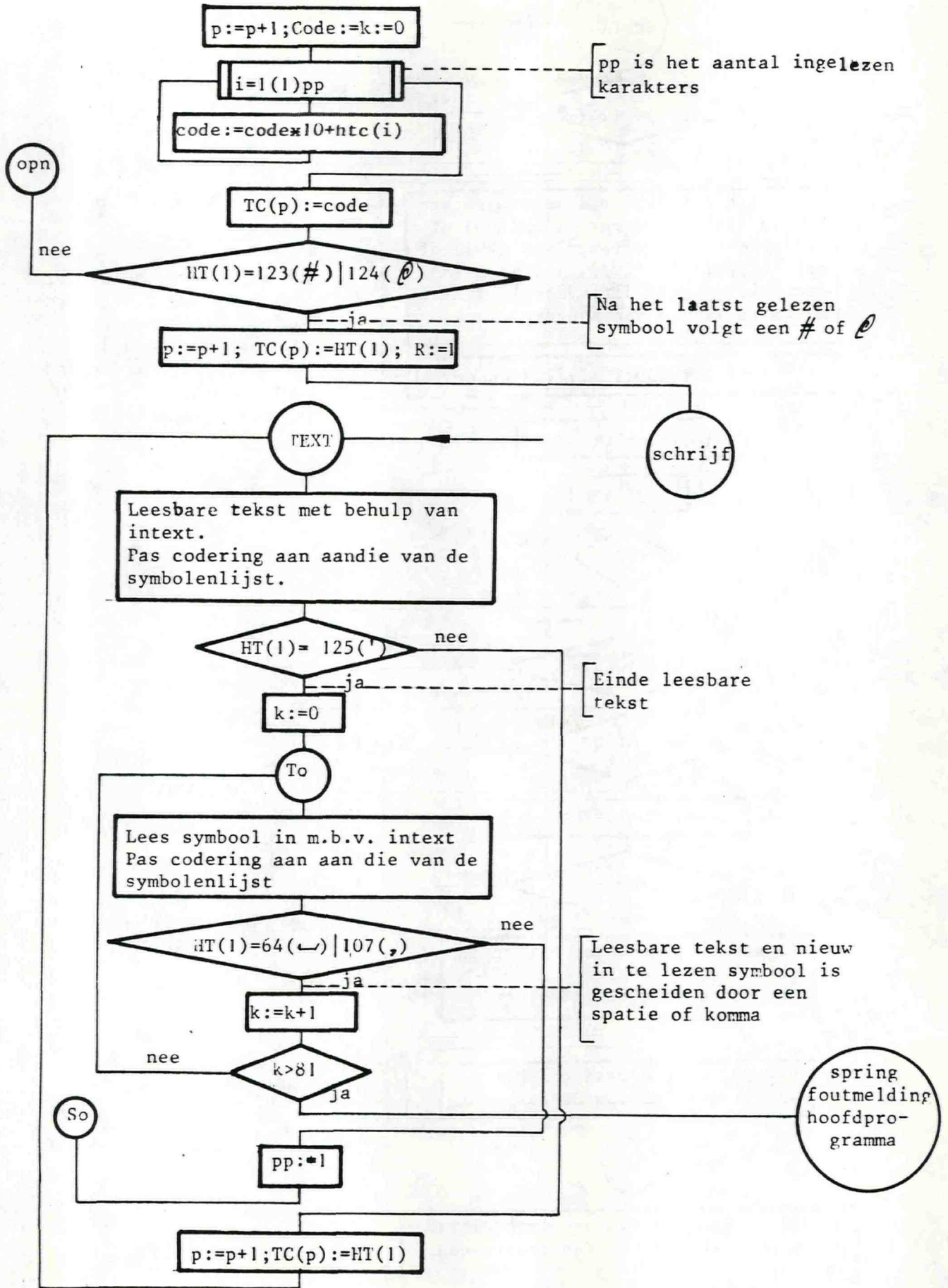
Deze procedure zorgt ervoor dat een symbool in de index of exponent evenveel ruimte inneemt als een symbool dat op normale hoogte geschreven wordt. Verder hetzelfde als procedure L.

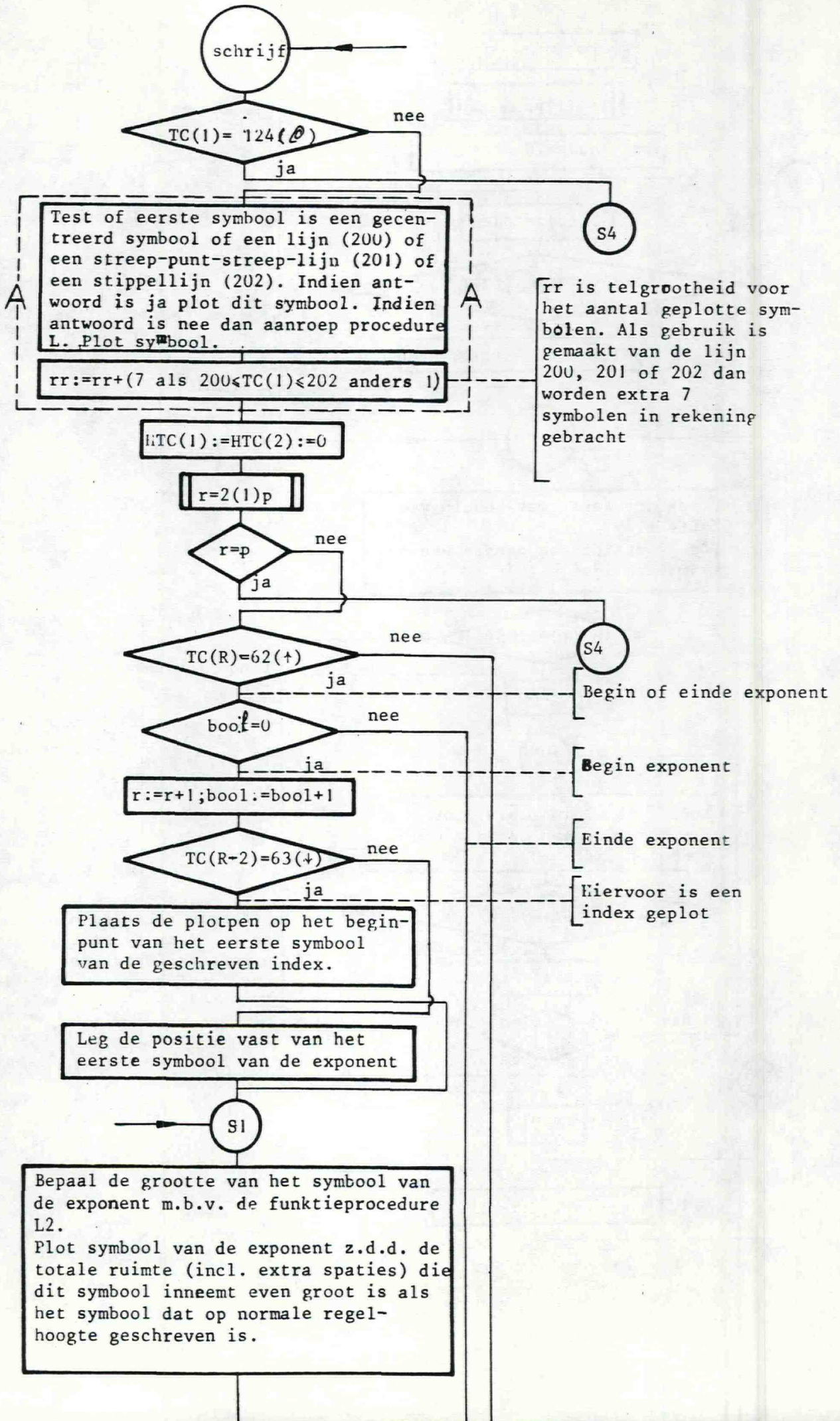
Procedure voor het bepalen van de plaats van een symbool in x- of y-richting.

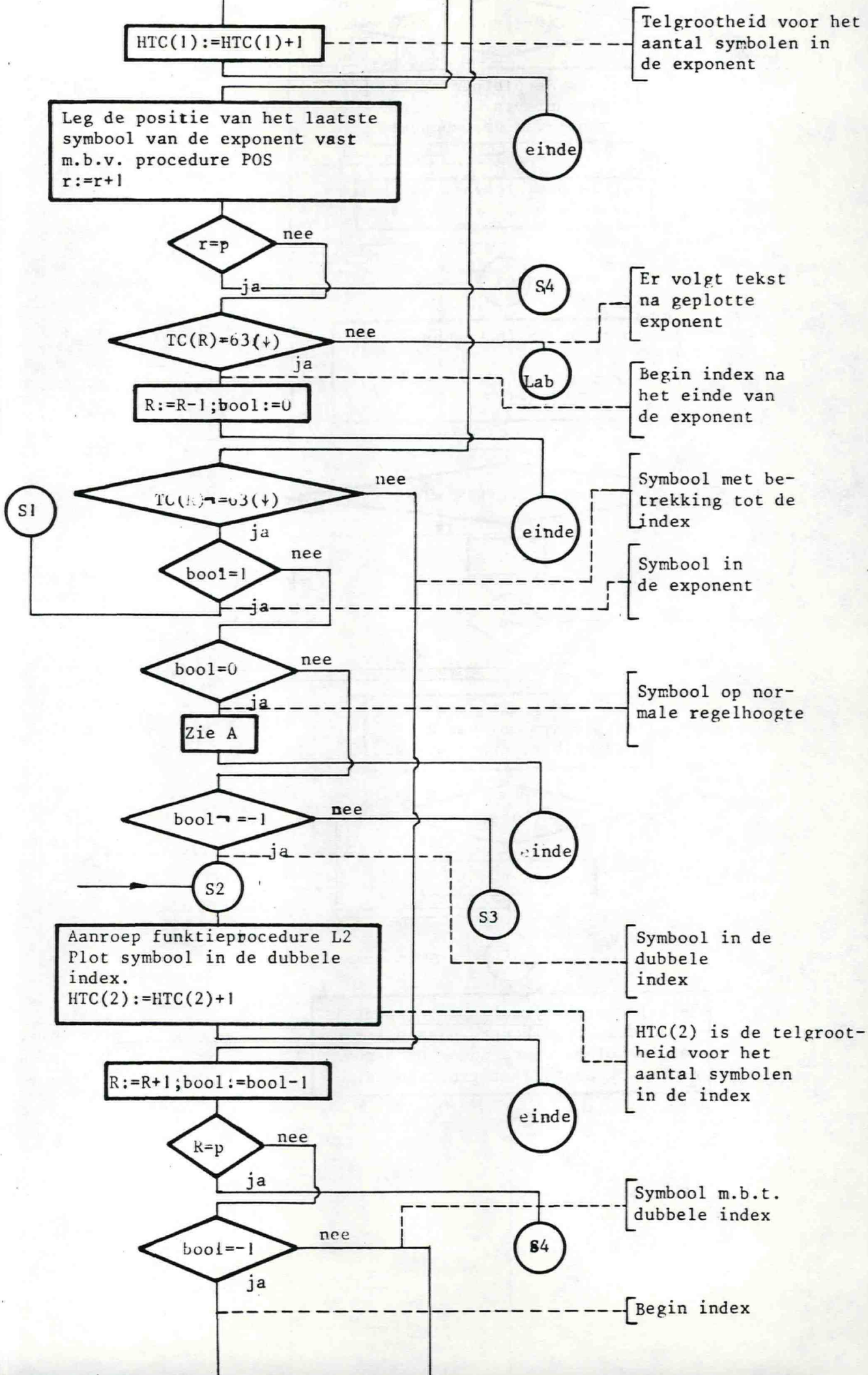
De procedure intext kent andere waarden toe aan het ingelezen symbool dan op de symbolenlijst vermeld staat. Vandaar de herwaardering van HT(1).

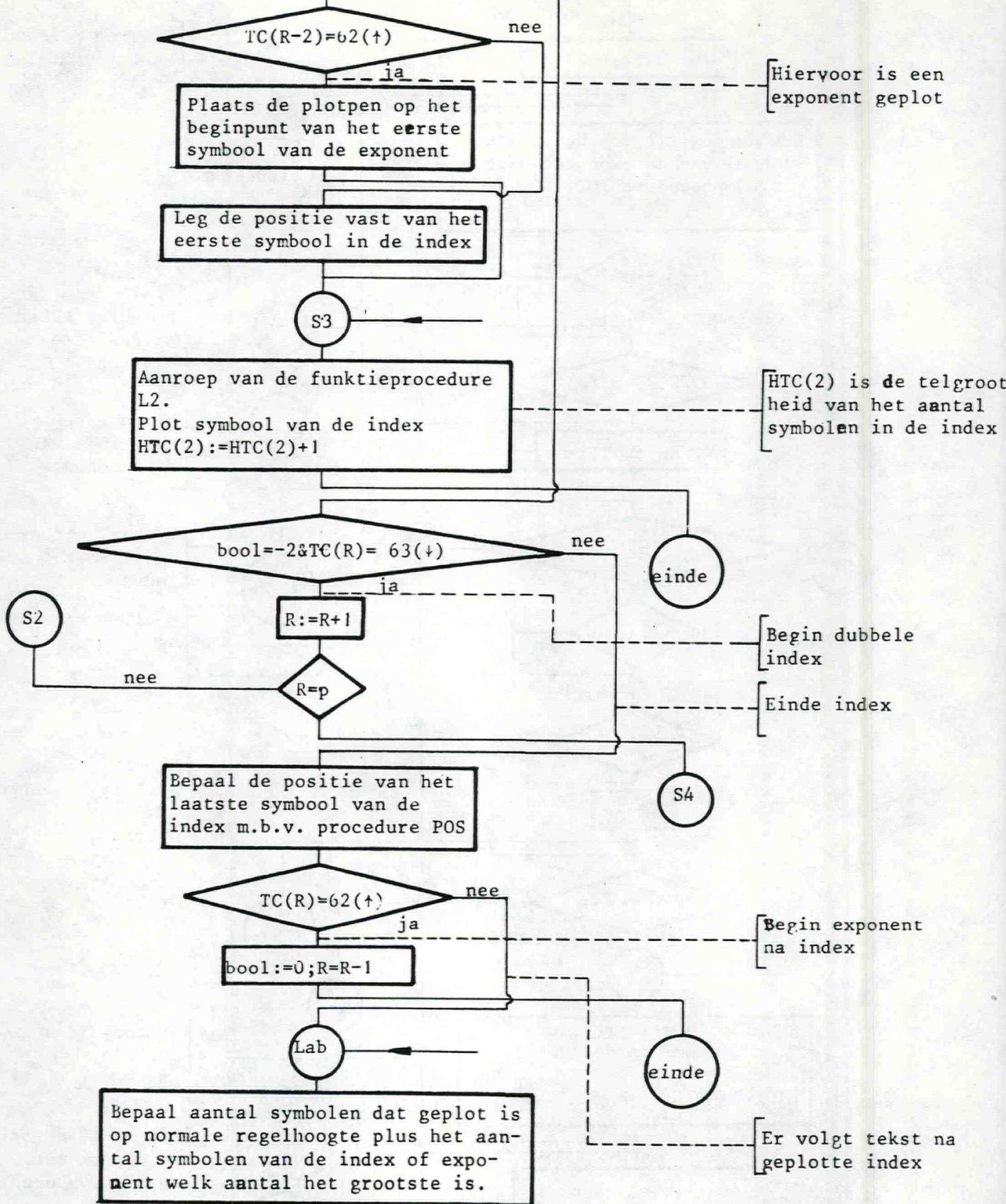
De array HTC stelt één symbool in code voor b.v. HTC ≡ 40
HTC(1) ≡ 4 en HTC(2) ≡ 0

liet lezen van 123 (#) na een komma of het lezen van 124 (@) als eerste symbool op de kaart (nieuwe regel) of na een komma.

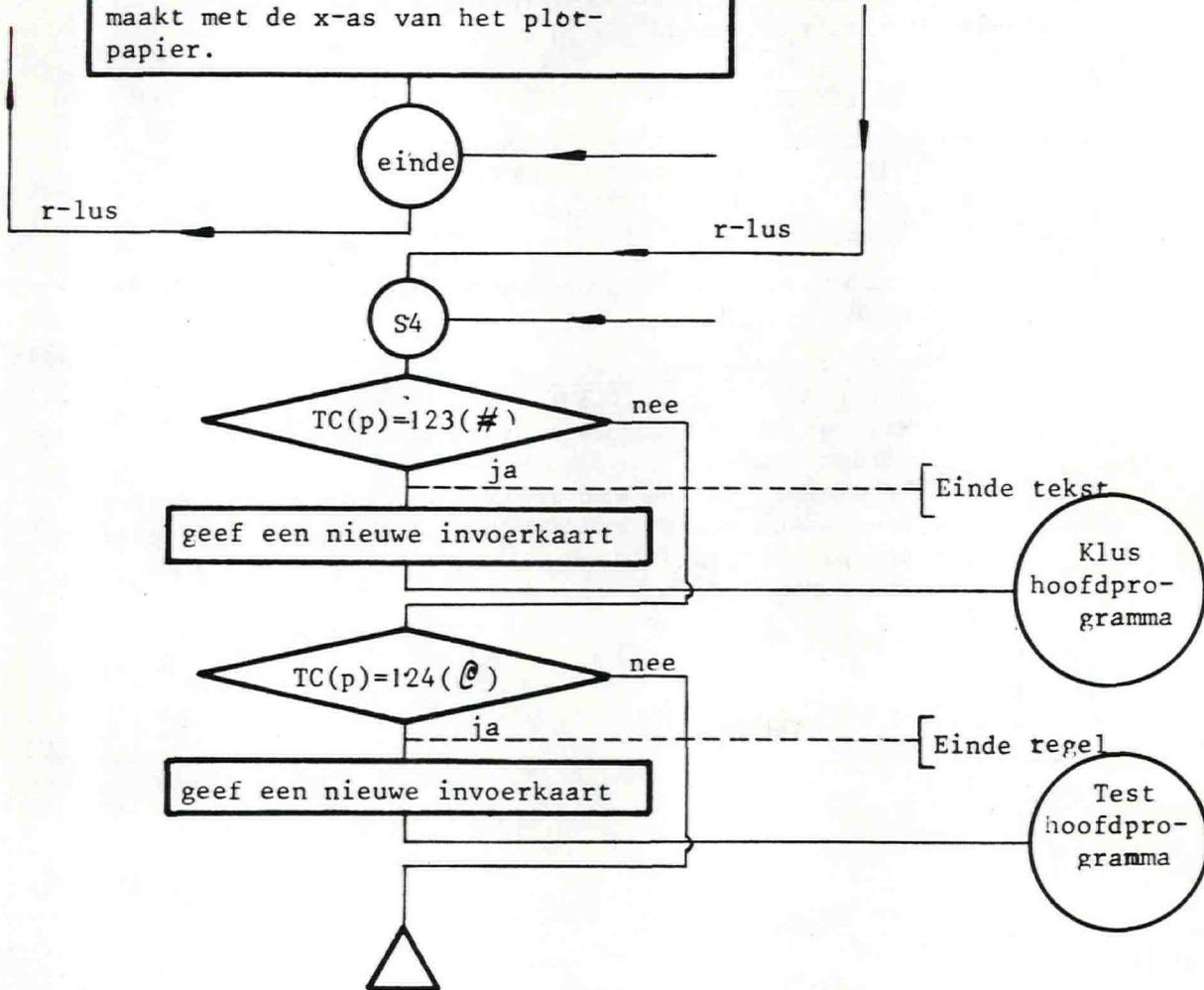








R:=R-1; bool:=0;
Plaats de plotpen op de positie van het geplote symbool met de grootste x-waarde als tekst onder een hoek van 0° met de x-as staat of met de grootste y-waarde als tekst onder een hoek van 90° met de x-as staat of de kleinste x-waarde als de tekst een hoek van 180° maakt met de x-as van het plot-papier.



I.2 Procedure voor het bepalen van minima en maxima - Minmax.

I.2.1 Doel.

Het bepalen van de hoogste en laagste waarde uit een array van getallen.

I.2.2 Algemeen.

De naam van de procedure is MINMAX.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de "Centrale Werkgroep Wiskunde".

I.2.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

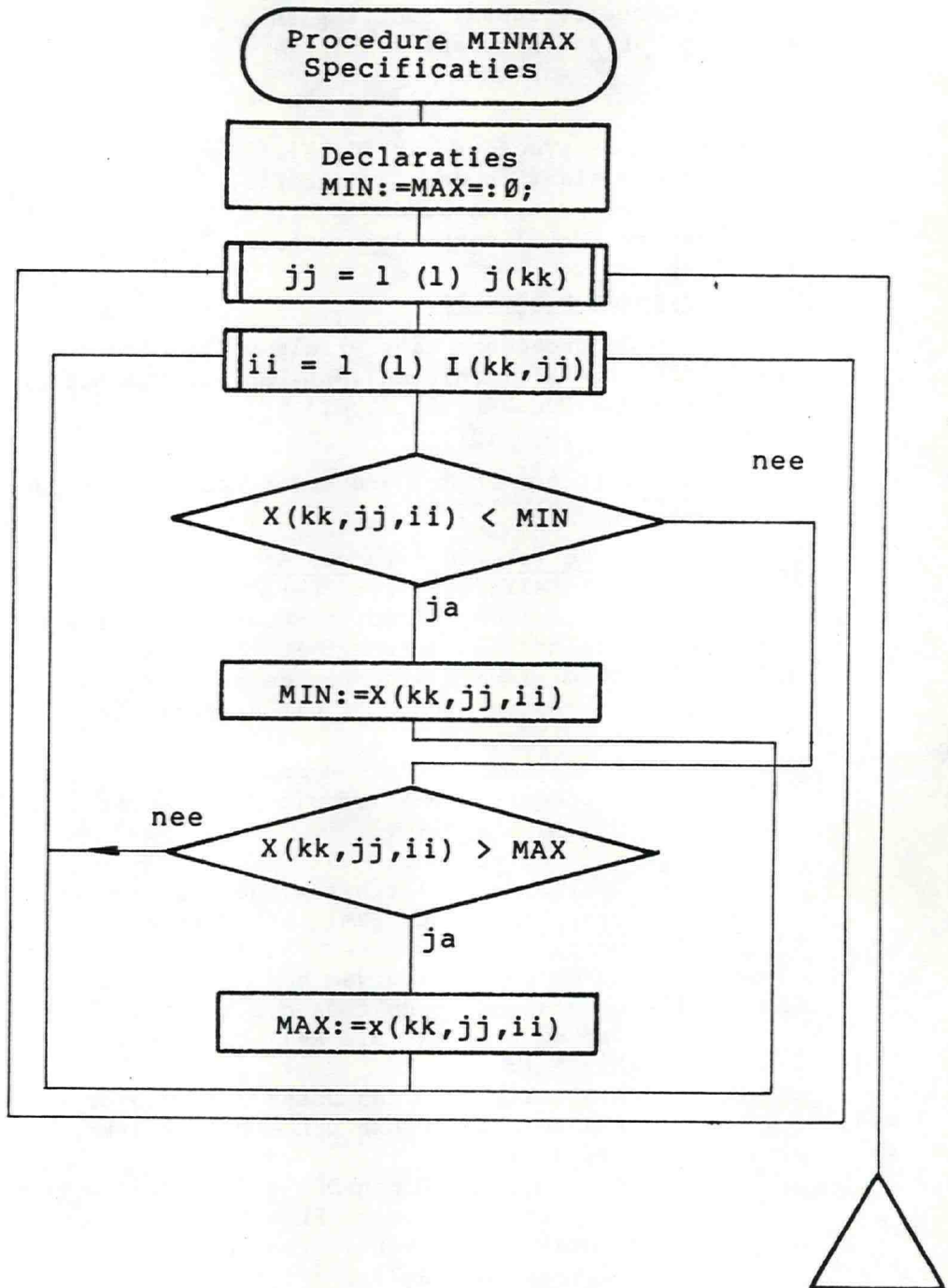
```
'PROCEDURE' MINMAX (X,MIN,MAX,I,J ,KK);  
'VALUE' X; "REAL' MIN,MAX; 'INTEGER' KK;  
'ARRAY' X; 'INTEGER' 'ARRAY' I,J;
```

Aan de declaraties van het programma moet worden toegevoegd:
'PROCEDURE' MINMAX; 'CODE';

Betekenis van de formele parameters:

X	i	: Array X(/1:K,1:JM,1:IM/)
		waarvan de minimum en maximum waarde bepaald moet worden voor X(/KK,1:JM,IM/)
MIN	o	: De minimum waarde van array X voor de kk-de grafiek.
MAX	o	: De maximum waarde van de array X voor de kk-de grafiek.
I	i	: Zie parameterlijst van CWWPLT
J	i	: Zie parameterlijst van CWWPLT
KK	i.	: Het grafieknnummer waarvoor min en max gelden.

I.2.4 Stroomdiagram.



I.3 Procedure voor lineaire schaalverdeling van de as // x-as plotter - SCAL X1.

I.3.1 Doel.

De procedure verzorgt een as evenwijdig aan de x-as van het plotpapier met een lineaire schaalverdeling en bepaalt tevens de positie van de as//y-as van het plotpapier.

I.3.2 Algemeen.

De naam van de procedure is SCALX1.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de "Centrale Werkgroep Wiskunde".

De procedure SCALX1 roept de procedure TRANS op.

I.3.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

```
'PROCEDURE' SCALX1 (A,MIN,MAX,NUL,DEL,NSP,SCL,N,CX,DX,CY,DY);
```

```
'REAL' MIN,MAX,NUL,DEL,SCL,CX,CY;
```

```
'INTEGER' NSP,A,N,DX,DY;
```

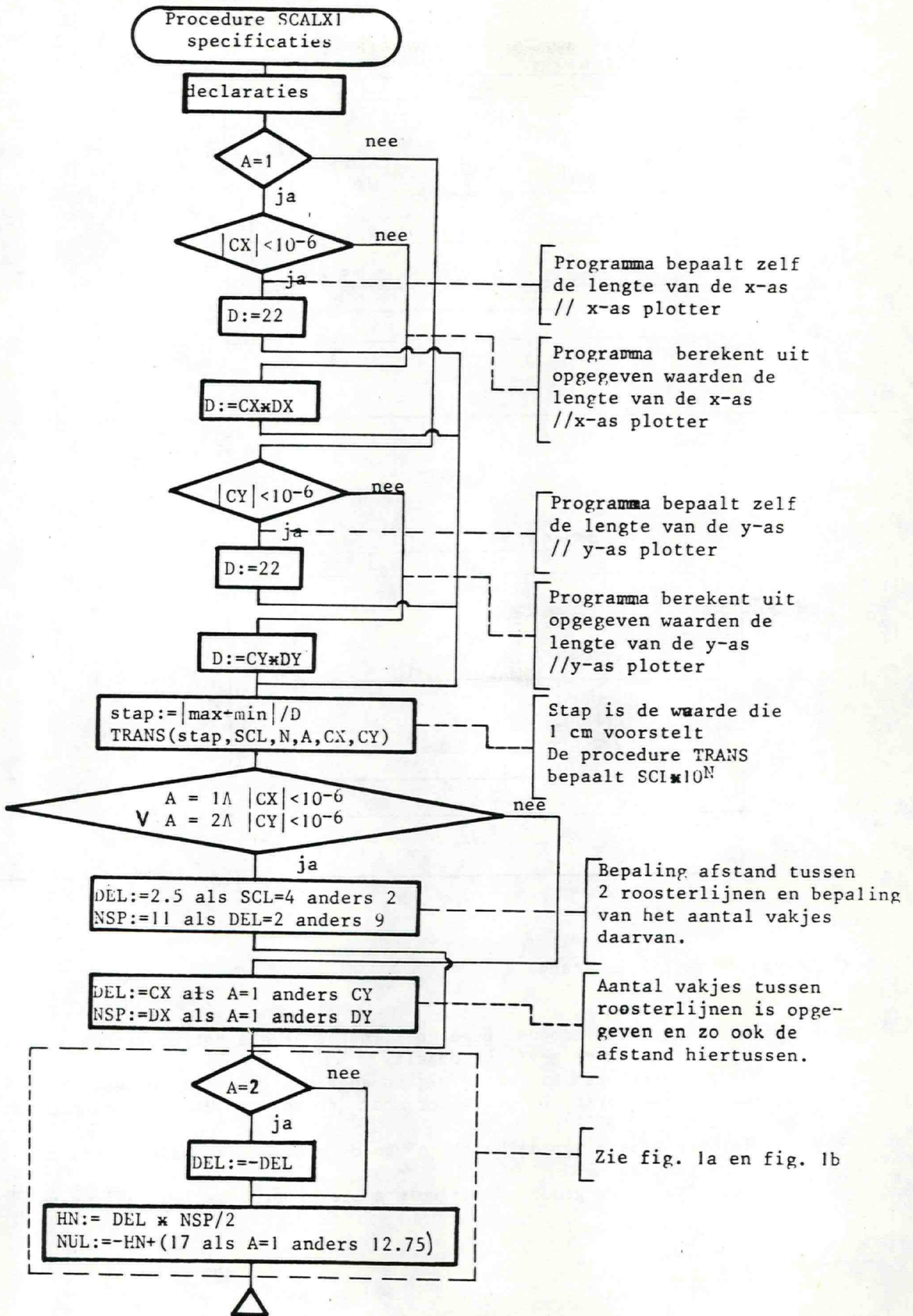
Aan de declaraties van het programma moet worden toegevoegd:

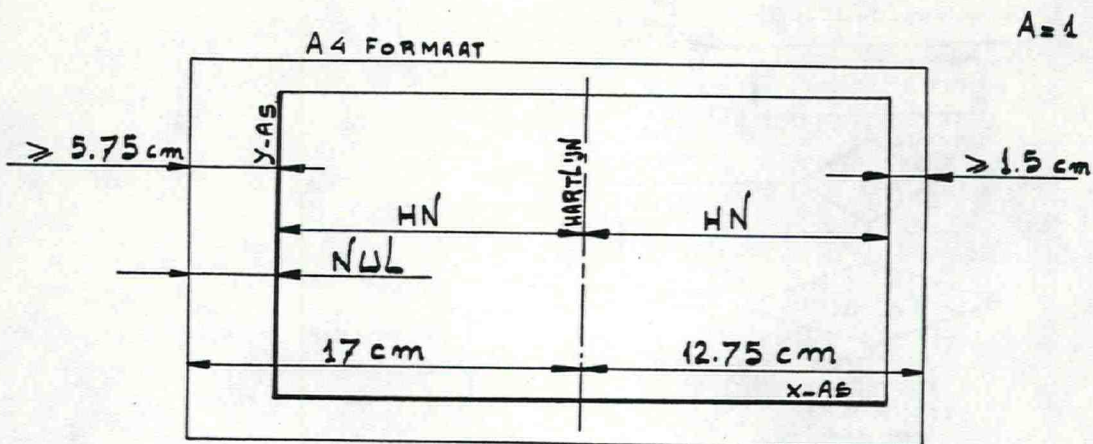
```
'PROCEDURE' SCALX1; 'CODE';
```

Betekenis van de formele parameters:

- A i : A = PAXY(/k/) (zie CWWPLT)
 Als A=1 dan verzorgt SCLAX1 de x-as van de grafiek.
 Als A=2 dan verzorgt SCALX1 de y-as van de grafiek.
- MIN i : De minimum waarde van de array X voortkomend uit de
 procedure MINMAX of uit de parameter PXMIN(/k/) van
 CWWPLT
- MAX i : De maximumwaarde van de array X voortkomend uit de
 procedure MINMAX of uit de parameter PXMAX(/k/) van
 CWWPLT.
 (Dit alles voor A=1).
 Indien A=2 geldt hetzelfde maar dan voor de array Y
 resp. parameter PYMIN of PYMAX.
- NUL o : De ruimte tussen de y-as (als A=1) of de x-as (als A=2)
 en de papierrand van het A4-formaat.
- DEL o : De afstand in cm tussen 2 roosterlijnen van de x-as
 van de grafiek (als A=1) of y-as van de grafiek (als A=2).
 Als A=1:
 Als PCX(/k/)=0 dan DEL = 2 of 2.5 cm, voor de x-as.
 Als PCX(/k/)≠0 dan DEL = PCX(/k/) voor de x-as.
 Als A=2:
 Als PCY(/k/)=0 dan DEL = 2 of 2.5 cm voor de y-as.
 Als PCY(/k/)≠0 dan DEL = PCY(/k/) voor de y-as.
- NSP o : Aantal vakjes van de roosterlijnen op de x-as (als A=1)
 of y-as (Als A=2).
 NSP = 11 als DEL = 2 } Als PCX(/k/) of
 NSP = 9 als DEL = 2.5 } PCY(/k/)=0
 NSP = PDX(/k/) (als A=1) of
 NSP = PDY(/k/) (als A=2).
- SCL o : Zie procedure TRANS
- N o : Zie procedure TRANS
- CX i : = PCX(/k/) zie CWWPLT
- DX i : = PDX(/k/) zie CWWPLT
- CY i : = PCY(/k/) zie CWWPLT
- DY i : = PDY(/k/) zie CWWPLT

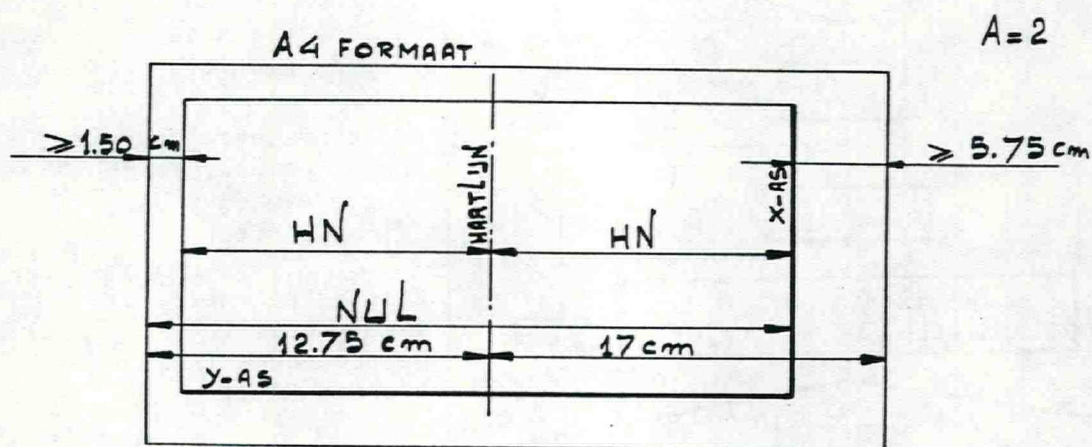
I.3.4 Stroomdiagram.





X-AS PLOTTER

Fig. 1a



X-AS PLOTTER

Fig. 1b.

Positie van de verticale assen.

- Opmerking: De maten die hierboven gegeven zijn gelden als het programma zelf de maten voor de grafiek bepaalt. De gebruiker kan hiervan afwijken door zelf op te geven hoe groot de grafiek wordt. De gebruiker moet dan wel op de volgende punten letten.
- 1e. De plaats van de hartlijn van de grafiek is zoals hierboven gegeven.
 - 2e. Blijft er genoeg vrije ruimte over voor tekst en inbindrand.

I.4 Procedure voor lineaire schaalverdeling van de as//y-as plotter - SCAL Y1.

I.4.1 Doel.

De procedure verzorgt een as evenwijdig aan de y-as van het plotpapier en bepaalt tevens de positie van de as//x-as van het plotpapier.

I.4.2 Algemeen.

De naam van de procedure is SCALY1.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de Centrale Werkgroep Wiskunde.

De procedure SCALY1 roept de procedure TRANS op.

I.4.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

'PROCEDURE' SCALY1 (A,MIN,MAX,NUL,DEL,NSP,SCL,N,CX,DX,CY,DY);

'REAL' MIN,MAX,NUL,DEL,SCL,CX,CY;

'INTEGER' NSP,A,N,DX,DY;

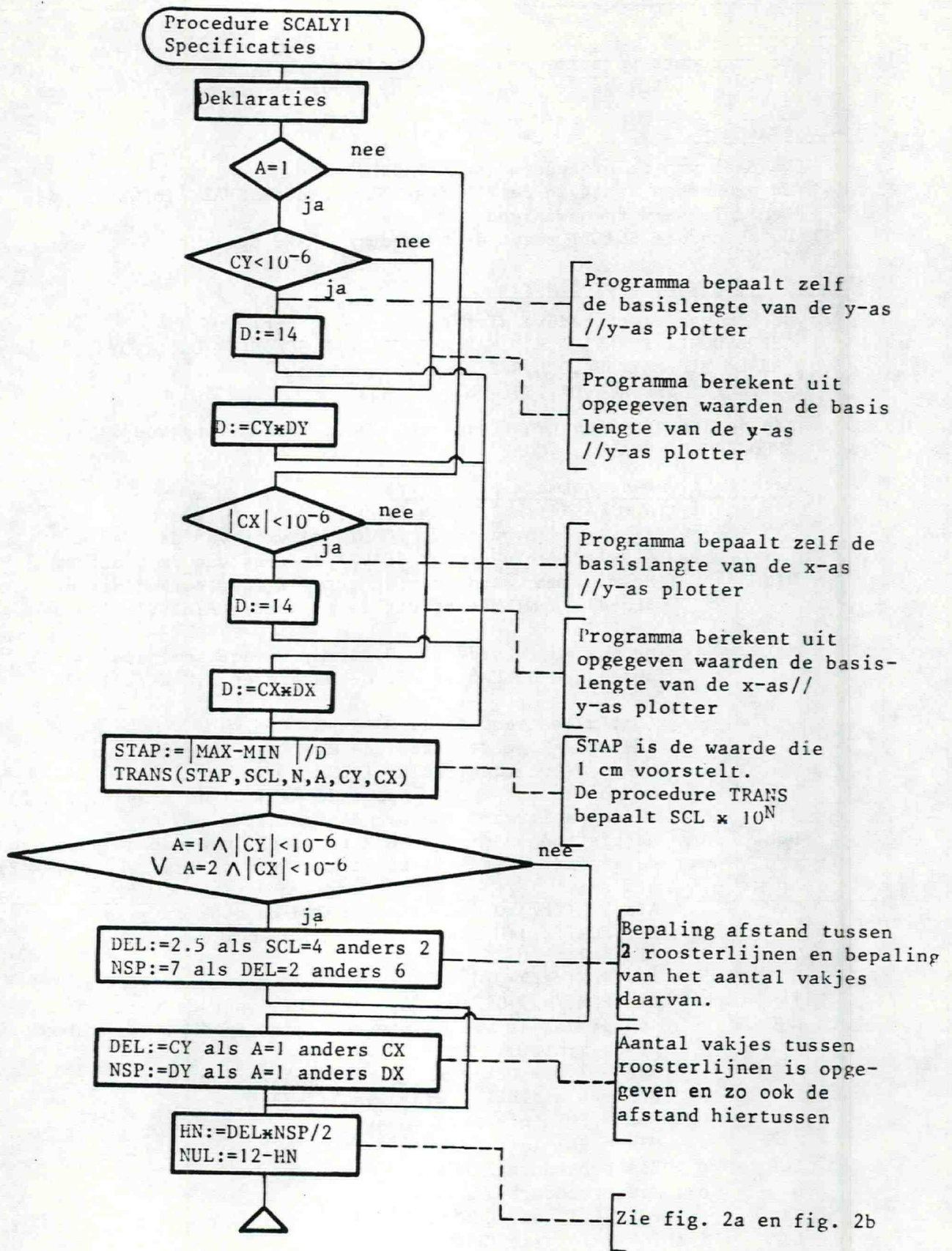
Aan de declaratie van het programma moet worden toegevoegd:

'PROCEDURE' SCALY1; 'CODE';

Betekenis van de formele parameters

- A i : $A = PAXY (/k/)$ (zie CWWPLT)
 Als A=1 dan verzorgt SCALY1 de y-as van de grafiek.
 Als A=2 dan verzorgt SCALY1 de x-as van de grafiek.
- MIN i : De minimum waarde van de array Y voortkomend uit de
 procedure MINMAX of uit de parameter PYMIN(/k/) van
 CWWPLT.
- MAX i : De maximum waarde van de array Y voortkomend uit de
 procedure MINMAX of uit de parameter PYMAX(/k/) van
 CWWPLT.
 (Dit alles voor A=1).
 Indien A=2 geldt hetzelfde maar dan voor de array X
 resp. de waarden PXMIN of PXMAX.
- NUL o : De ruimte tussen de x-as (als A=1) of de y-as (als A=2)
 en de papierrand van het A4-formaat.
- DEL o : De afstand in cm tussen 2 roosterlijnen van de y-as van
 de grafiek (als A=1) of de x-as van de grafiek (als A=2).
 Als A=1:
 Als PCY(/k/)=0 dan DEL=2 of 2.5 cm voor de y-as
 Als PCY(/k/)≠0 dan DEL= PCY(/k/) voor de x-as.
 Als A=2:
 Als PCX(/k/)=0 dan DEL=2 of 2.5 cm voor de y-as
 Als PCX(/k/)≠0 dan DEL = PCX(/k/) voor de x-as
- NSP o : Aantal vakjes van de roosterlijnen op de y-as (als A=1)
 of de x-as (als A=2).
 NSP = 7 als DEL = 2 } Als PCY(/k/) of
 NSP = 6 als DEL = 2.5 } PCX(/k/)=0
 NSP = PDY (als A=1)
 NSP = PDX (als A=2)
- SCL o : Zie procedure TRANS
- N o : Zie procedure TRANS
- CX i : PCX(/k/) zie CWWPLT
- DX i : PDX(/k/) zie CWWPLT
- CY i : PCY(/k/) zie CWWPLT
- DY i : PDY(/k/) zie CWWPLT

I.4.4



A=1

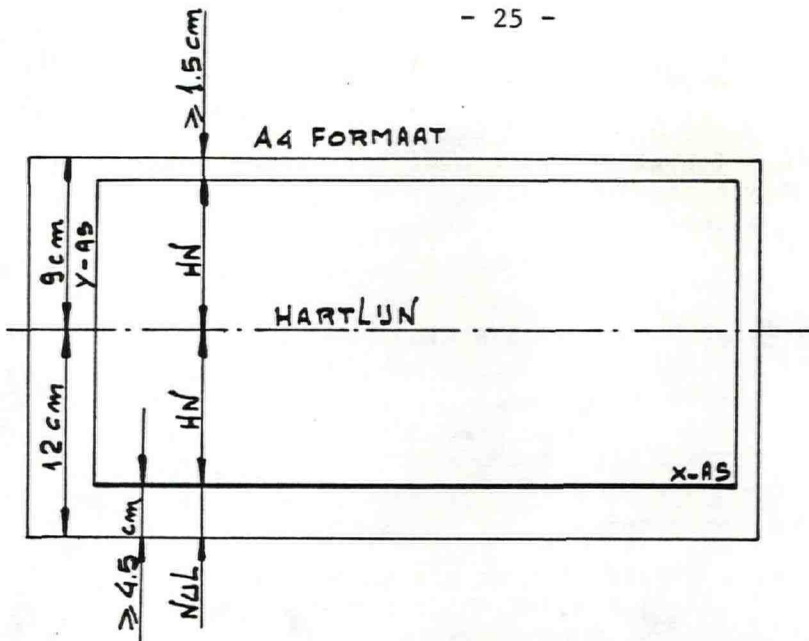


Fig. 2a.

A=2

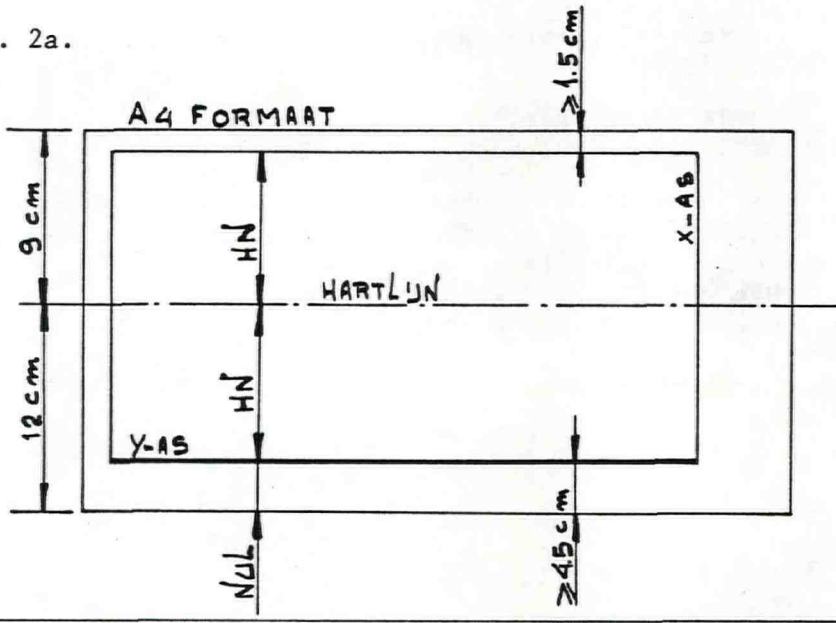


Fig. 2b. Positie van de horizontale assen.

Opmerking: De maten die hierboven gegeven zijn gelden als het programma zelf de maten voor de grafiek bepaalt.
 De gebruiker kan hiervan afwijken door zelf op te geven hoe groot de grafiek wordt.
 De gebruiker moet dan wel op de volgende punten letten
 1e. De plaats van de hartlijn is zoals hierboven gegeven.
 2e. Blijft er genoeg vrije ruimte over voor tekst en inbindrand.

I.5 Procedure voor een lineaire gradenverdeling van de as//x-as plotter - SCAL X2.

I.5.1 Doel.

De procedure verzorgt een as evenwijdig aan de x-as van het plotpapier met een lineaire gradenverdeling en bepaalt tevens de positie van de as//y-as van het plotpapier.

I.5.2 Algemeen.

De naam van de procedure is 'SXALX2.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de "Centrale Werkgroep Wiskunde".

De procedure SCALX2 roept de procedure TRANS en de procedure SCALX1 op.

I.5.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

'PROCEDURE' SCALX2 (A,MIN,MAX,NUL,DEL,NSP,SCL,N,CX,DX,CY,DY);

'REAL' MIN,MAX,NUL,DEL,SCL,CX,CY;

'INTEGER' NSP,A,N,DX,DY;

Aan de declaraties van het programma moet worden toegevoegd:

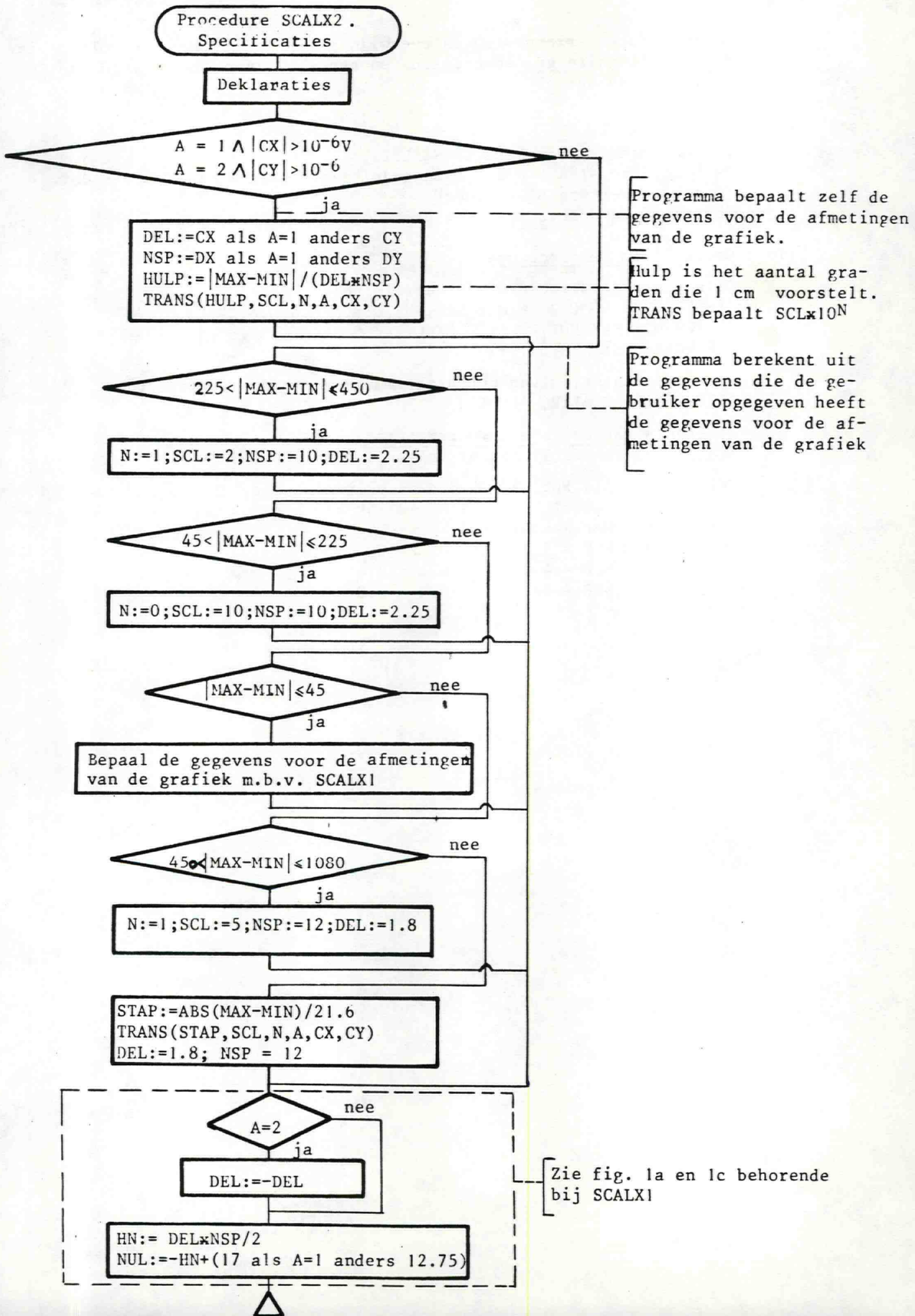
'PROCEDURE' SCALX2; 'CODE';

Betekenis der formele parameters.

Zie procedure SCALX1 met de volgende verschillen:

- DEL o : De waarde welke DEL kan aannemen als $PCX(/k/)=0$ zijn nu:
 DEL=2.25 of 1.8 cm afhankelijk van max. bereik van de as.
- NSP o : Het aantal vakjes is:
 Als DEL = 2.25 NSP = 10
 Als DEL = 1.8 NSP = 12

I.5.4 Stroomdiagram.



I.6 Procedure voor een lineaire gradenverdeling van de as//y-as plotter- SCALY2.

I.6.1 Doel.

De procedure verzorgt een as evenwijdig aan de y-as van het plotpapier met een lineaire gradenverdeling en bepaalt tevens de positie van de as//x-as van het plotpapier.

I.6.2 Algemeen.

De naam van de procedure is SCALY2.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de "Centrale Werkgroep Wiskunde".

De procedure SCALY2 roept de procedure TRANS en de procedure SCALY1 op.

I.6.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

```
'PROCEDURE' SCALY2 (A,MIN,MAX,NUL,DEL,NSP,SCL,N,CX,DX,CY,DY);
```

```
'REAL' MIN,MAX,NUL,DEL,SCL,CX,CY;
```

```
'INTEGER' NSP,A,N,DX,DY;
```

Aan de declaraties van het programma moet worden toegevoegd:

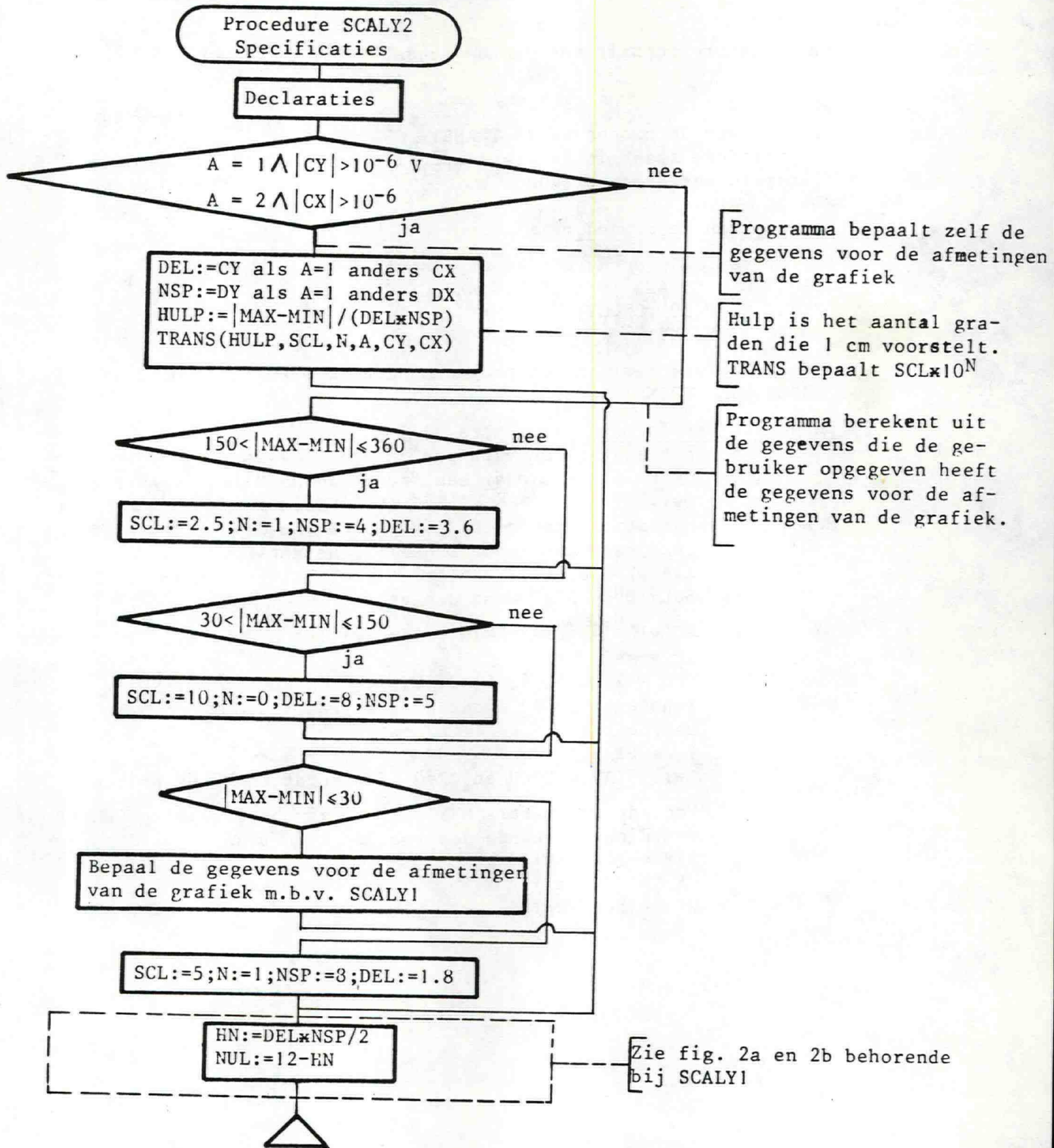
```
'PROCEDURE' SCALY2; 'CODE';
```

Betekenis van de formele parameters:

Zie procedure SCALY1 met de volgende verschillen:

- DEL o : Als $PCY(/k/)=0$ dan is de waarde die DEL kan aannemen 3, 3.6 of 1.8 cm afhankelijk van het max. bereik van de as.
- NSP o : Het aantal vakjes is:
- | | | |
|---------|-----|-----------|
| NSP = 5 | als | DEL = 3 |
| NSP = 4 | als | DEL = 3.6 |
| NSP = 8 | als | DEL = 1.8 |

I.6.4 Stroomdiagram.



I.7 Procedure voor het bepalen van de schaal voor x of y-as- TRANS.

I.7.1 Doel.

De procedure bepaalt een "ronde schaal" voor de assen.

I.7.2 Algemeen.

De naam van de procedure is TRANS.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de "Centrale Werkgroep Wiskunde".

I.7.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

```
'PROCEDURE' TRANS (STAP,SCL,N,AXY,CX,CY);
```

```
'REAL' STAP,SCL,CX,CY;
```

```
'INTEGER' N,AXY;
```

Aan de declaraties van het programma moet worden toegevoegd:

```
'PROCEDURE' TRANS; 'CODE';
```

Betekenis der formele parameters

STAP i : Het aantal eenheden per cm van de x-as of y-as voordat de x- of y-as naar een "ronde schaal" is getransformeerd.

Bv.: $STAP = \frac{|XMAX-XMIN|}{(DEL*NSP)}$

SCL o : Het aantal eenheden / 10^N per cm van de x- of y-as nadat deze naar een ronde schaal is getransformeerd.

Als $CX \neq 0$ of $CY \neq 0$ dan geldt voor de x-as of y-as.

$SCL * 10^N = STAP$ want dan geldt:

$SCL * 10^N = \frac{|XMAX-XMIN|}{(CX*NSP)}$ (voor de x-as) (Idem voor de y-as)

$SCL = 2, 2.5, 4, 5$ of 10 .

N o : Exponent van 10 nadat STAP is getransformeerd naar een waarde

$1 < STAP < 10$

Bv.: $STAP = 2000$ en $CX \neq 0$ dan wordt $SCL=2$ en $N=3$.

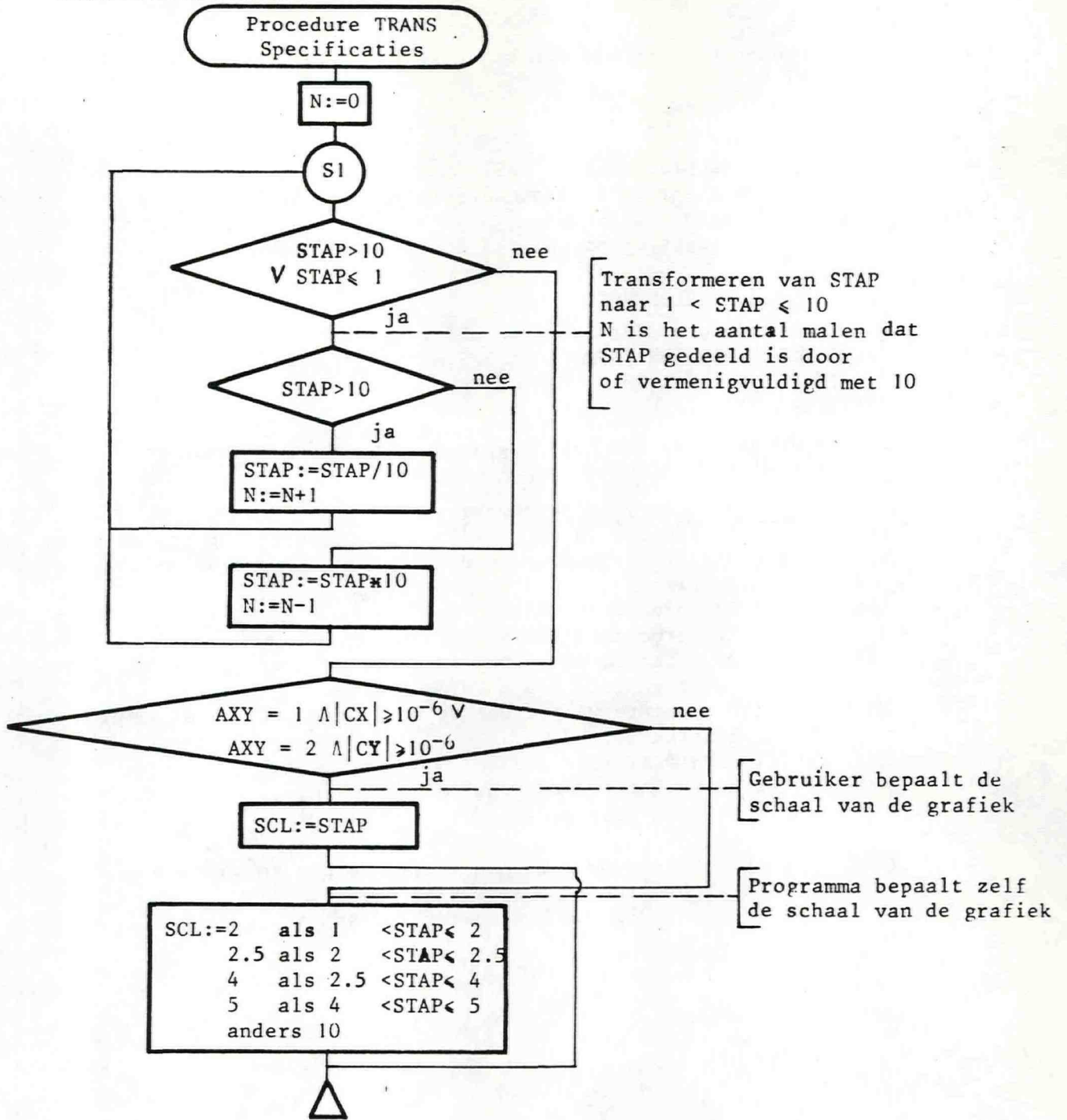
Voor de parameters AXY, CX, CY zie beschrijving CWWPLT met dien verstande dat voor de k-de grafiek geldt:

$AXY = PAXY(/k/)$

$CX = PCX(/k/)$

$CY = PCY(/k/)$

I.7.4 Stroomdiagram.



I.8 Procedure voor het plotten van de waarden langs de x-as- XAS.

I.8.1 Doel.

De procedure plot de waarden bij elke roosterlijn langs de x-as van de grafiek.

I.8.2 Algemeen.

De naam van de procedure is XAS.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de Centrale Werkgroep Wiskunde.

De procedure roept de procedures SCHUIF en NDC aan.

I.8.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

```
'PROCEDURE' XAS (DEL,NSP,SCL,N,XXS,XMIN,XMAX,AXY, CODE);
```

```
'REAL' DEL,SCL,XXS,XMIN,XMAX;
```

```
'INTEGER' NSP,N,AXY, CODE;
```

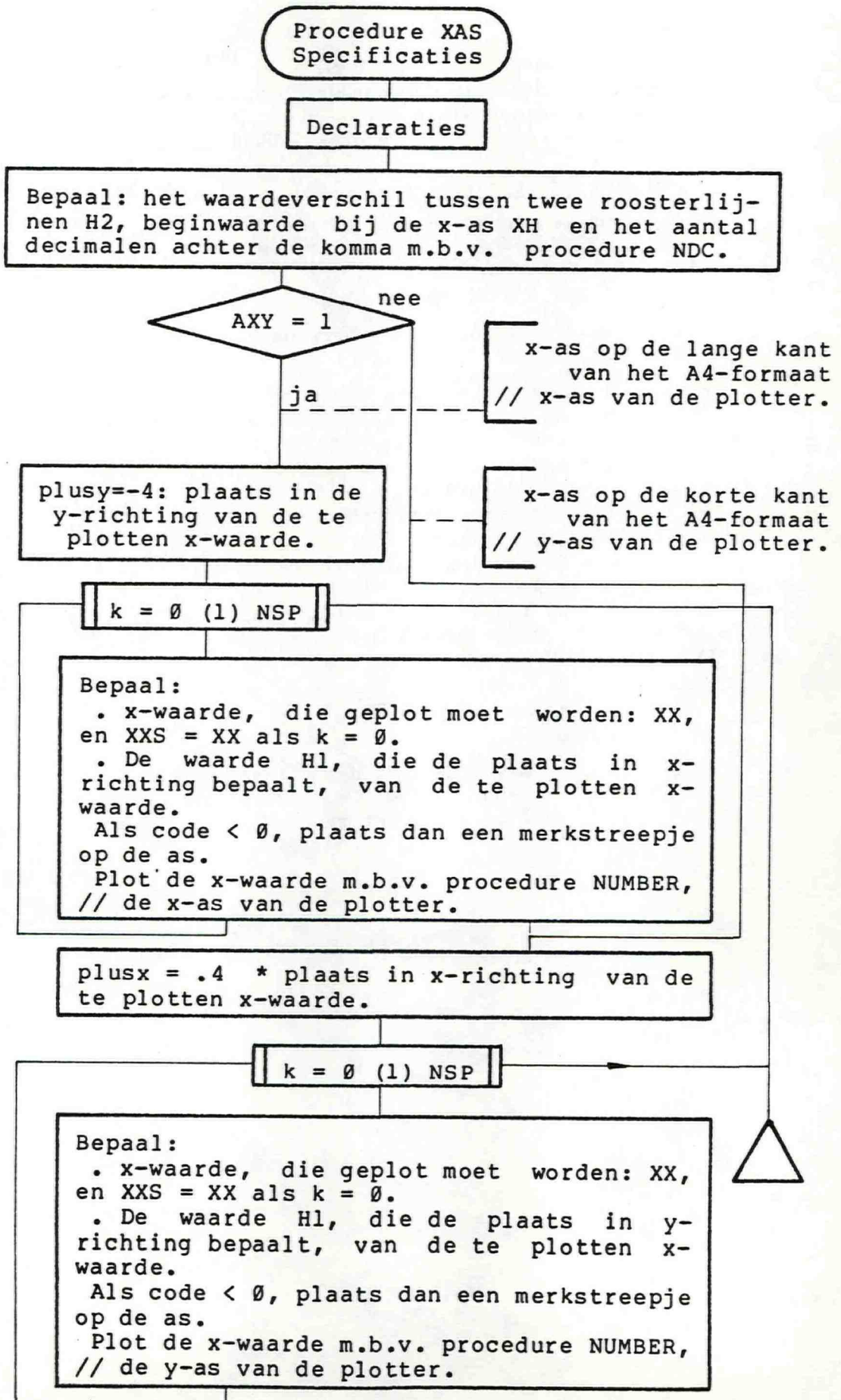
Aan de declaraties van het programma moet worden toegevoegd:

```
'PROCEDURE' XAS; 'CODE';
```

Betekenis van de formele parameters

DEL	i	: De afstand in cm tussen twee roosterlijnen van de x-as.
NSP	i	: Aantal vakjes van de roosterlijnen langs de x-as van de grafiek.
SCL	i	: Zie procedure TRANS
N	i	: Zie procedure TRANS
XXS	o	: De x-waarde welke bij de eerste (laagste) roosterlijn wordt geplot. $XXS \leq XMIN$
XMIN	i	: De laagste waarde van de array $X(K, J, I)$ van de k-de grafiek (zie 3.2).
AXY	i	: $AXY = PAXY$ (zie 3.2).
XMAX	i	: De hoogste waarde van de array $X(K, J, I)$ van de k-de grafiek (zie 3.2).
CODE	i	: Als code 0, dan worden de punten op de x-as gemarkeerd d.m.v. streepjes. (Nodig als het raster //Y-as is weggelaten).

I.8.4 Stroomdiagram.



I.9 Procedure voor het plotten van de waarden langs de y-as- YAS.

I.9.1 Doel.

De procedure plot de waarden bij elke roosterlijn loodrecht op de y-as van de grafiek.

I.9.2 Algemeen.

De naam van de procedure is YAS.

De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de Centrale Werkgroep Wiskunde.

De procedure roept de procedures SCHUIF en NDC aan.

I.9.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

```
'PROCEDURE' YAS (DEL,NSP,SCL,N,YYS,YMIN,YMAX,AXY , CODE);
```

```
'REAL' DEL,SCL,YYS,YMAX,YMIN;
```

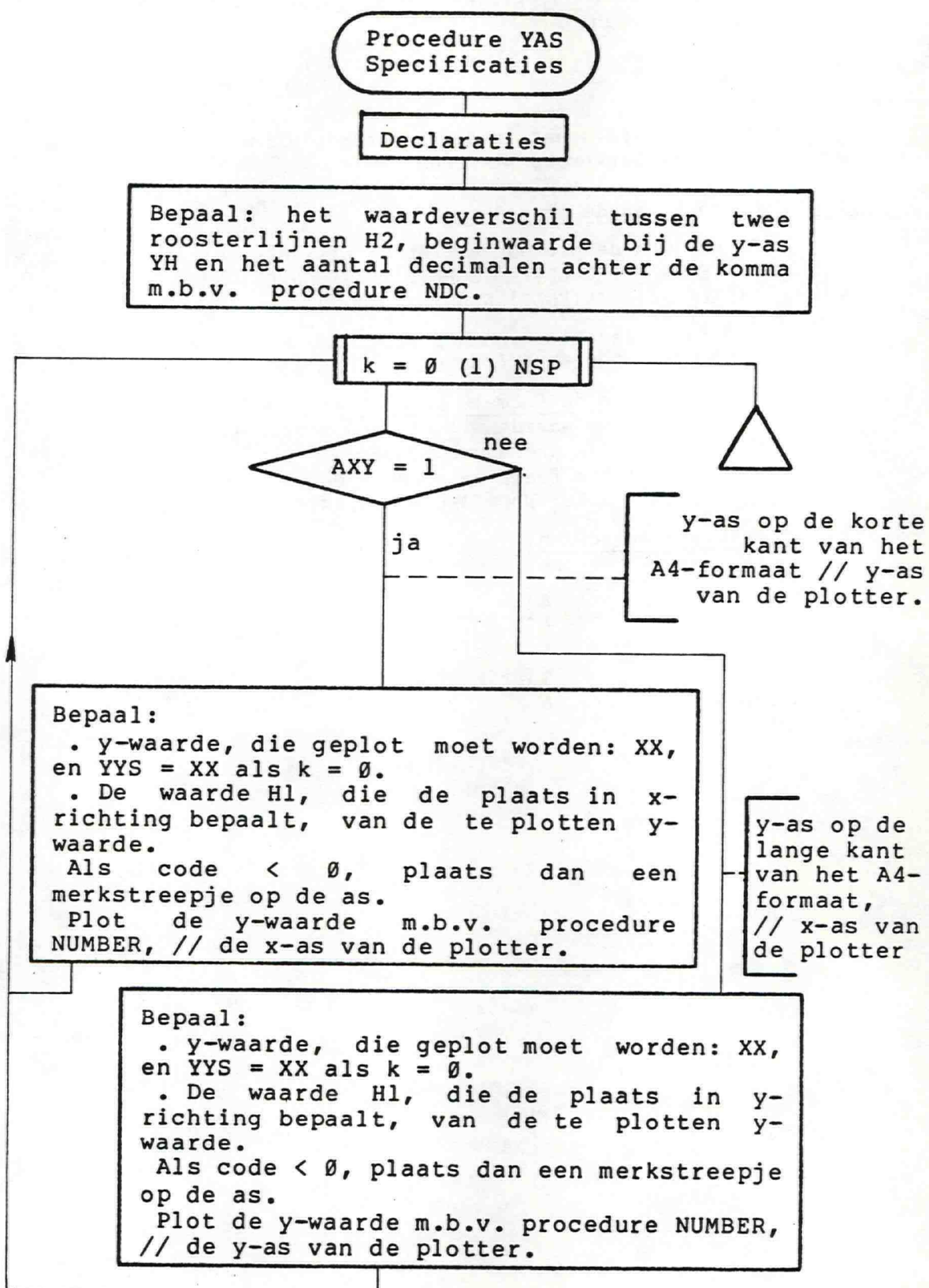
```
'INTEGER' NSP,N,AXY,.CODE;
```

Aan de declaraties van het programma moet worden toegevoegd:
'PROCEDURE' YAS; 'CODE';

Betekenis van de formele parameters

DEL	i	: De afstand in cm tussen twee roosterlijnen van de y-as.
NSP	i	: Aantal vakjes van de roosterlijnen langs de y-as van de grafiek.
SCL	i	: Zie procedure TRANS
N	i	: Zie procedure TRANS
YYS	o	: De y-waarde welke bij de laagste roosterlijn wordt geplott. $YYS \leq YMIN$
YMIN	i	: De laagste waarde van de array $Y(K, J, I)$ van de k-de grafiek (zie 3.2).
AXY	i	: $AXY:=PAXY(k)$ voor de k-de grafiek (zie 3.2)
YMAX	i	: De hoogste waarde van de array $Y(K, J, I)$ van de k-de grafiek (zie 3.2).
CODE	i	: Als code 0, dan worden de punten op de y-as gemarkeerd d.m.v. streepjes. (nodig als het raster //X-as is weggelaten.)

I.9.4 Stroomdiagram.



I.10 Procedure voor plaats van schaalwaarden - SCHUIF.

I.10.1 Doel.

De procedure zorgt er voor dat schaalwaarden bij x- en y-as onder of naast de roosterlijnen op de goede afstand van de x- of y-as geplot worden.

I.10.2 Algemeen.

De naam van de procedure is SCHUIF.
De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de Centrale Werkgroep Wiskunde.

I.10.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

'INTEGER' 'PROCEDURE' SCHUIF (XX,N,NDEC);

'REAL' XX; 'INTEGER' N,NDEC;

Aan de declaraties van het programma moet worden toegevoegd:
'INTEGER' 'PROCEDURE' SCHUIF; 'CODE';

Betekenis van de formele parameters:

XX i : De waarde die bij een roosterlijn geplot moet worden

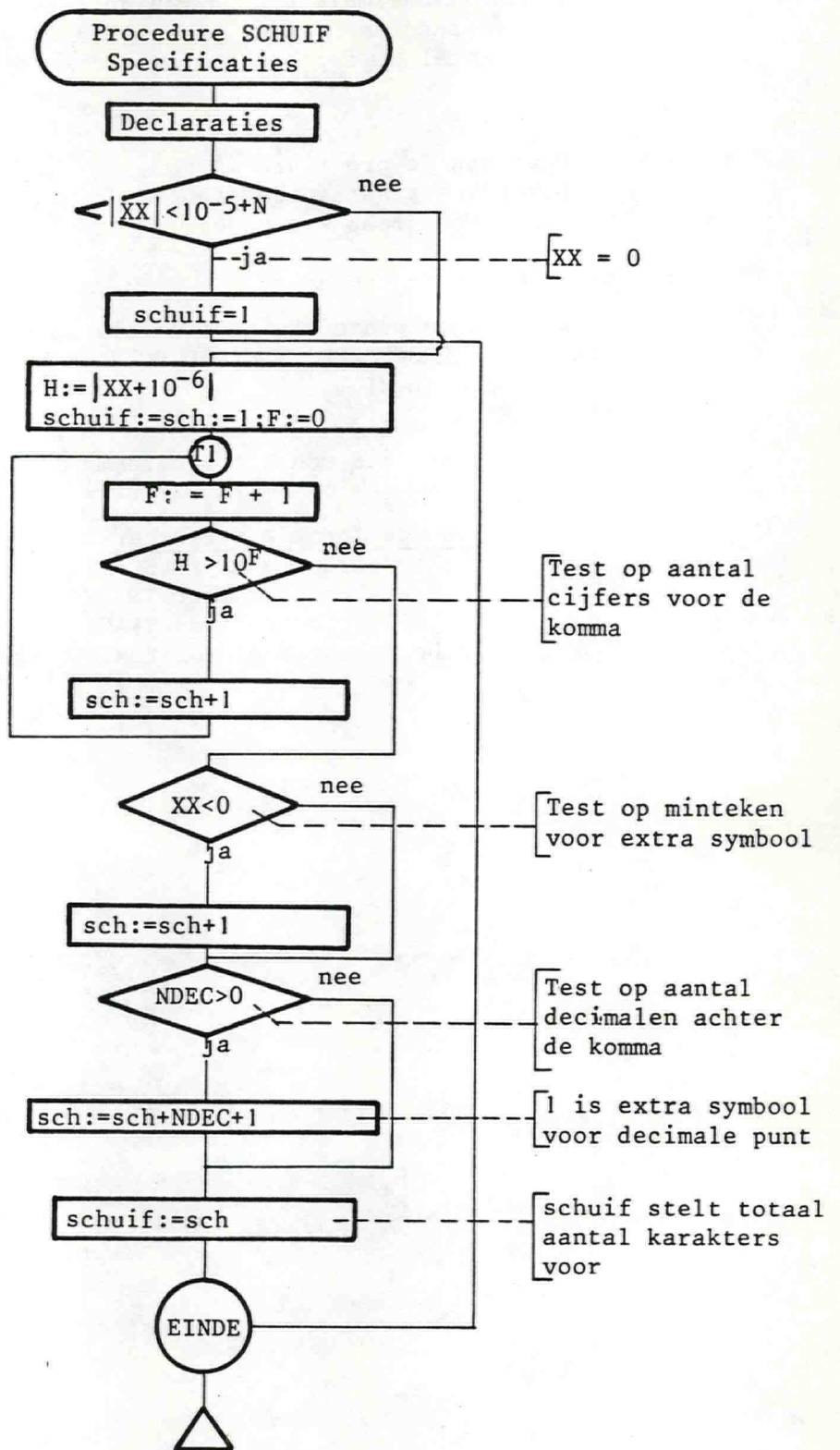
N i : Zie procedure TRANS.

NDEC i : Zie functieprocedure NDC
 NDEC:=NDC (DEL,SCL,N,XMIN);

Enkele voorbeelden:

<u>XX</u>	<u>schuif</u>
- 4.5	4
1000	4
-2000	5
.50	3
-.0008	6

I.10.4 Stroomdiagram.



I.11 Procedure voor bepaling van aantal decimalen achter de komma - NDC.

I.11.1 Doel.

De procedure is een hulpprocedure voor de procedure SCHUIF en zorgt er voor dat de decimale punten van de schaalwaarden recht onder elkaar komen te staan en, dat er per as (x- of y-as) voor alle schaalwaarden een gelijk aantal cijfers achter de decimale punt komen te staan.

I.11.2 Algemeen.

De naam van de procedure is NDC.
De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de Centrale Werkgroep Wiskunde.

I.11.3 Gebruik van de procedure.

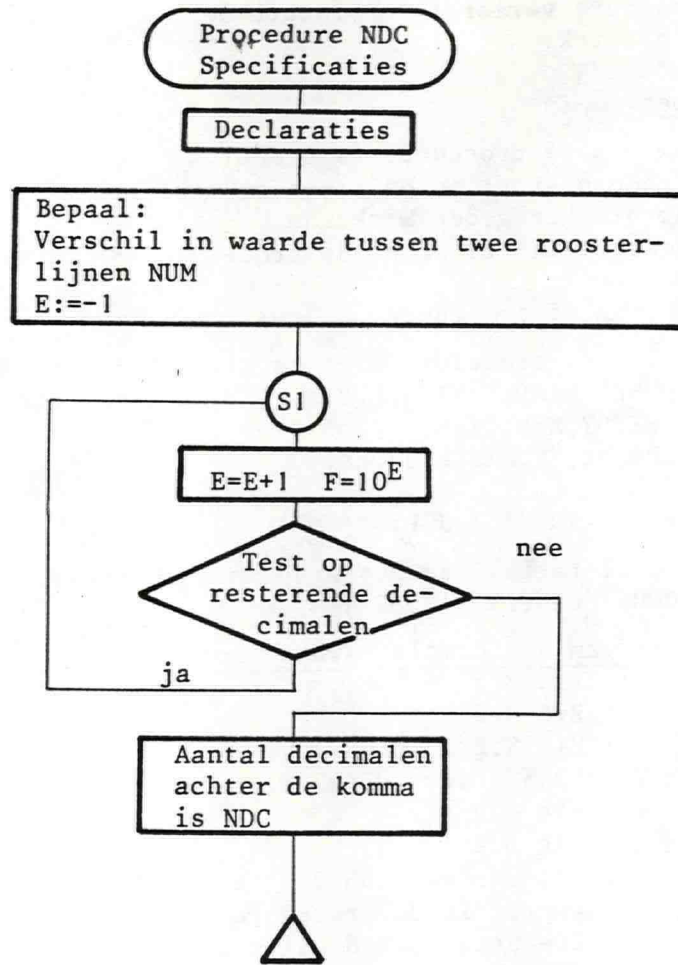
De kop van de procedure ziet er als volgt uit:
'INTEGER' 'PROCEDURE' NDC(DEL,SCL,N,MIN);
'REAL' DEL,SCL,MIN;
'INTEGER' N;

Aan de declaraties van het programma moet worden toegevoegd:
'INTEGER' 'PROCEDURE' NDC; 'CODE';

Betekenis van de formele parameters:

DEL	i	: Zie parameterlijst SCALX1
SCL	i	: Zie parameterlijst TRANS
N	i	: Zie parameterlijst TRANS
MIN	i	: De waarde X MIN of Y MIN afhankelijk van de x- of y-as.

I.11.4 Stroomdiagram.



I.12 Procedure voor het plotten van lijnen- LINPLT.

I.12.1 Doel.

De procedure verzorgt de plotuitvoer van de lijnen en/of punten in de grafiek.

I.12.2 Algemeen.

De naam van de procedure is LINPLT.
De procedure staat in de standaardbibliotheek SBAL.LIBCWW van de Centrale Werkgroep Wiskunde.
De procedure LINPLT roept de procedure ASMOT op.

I.12.3 Gebruik van de procedure.

De kop van de procedure ziet er als volgt uit:

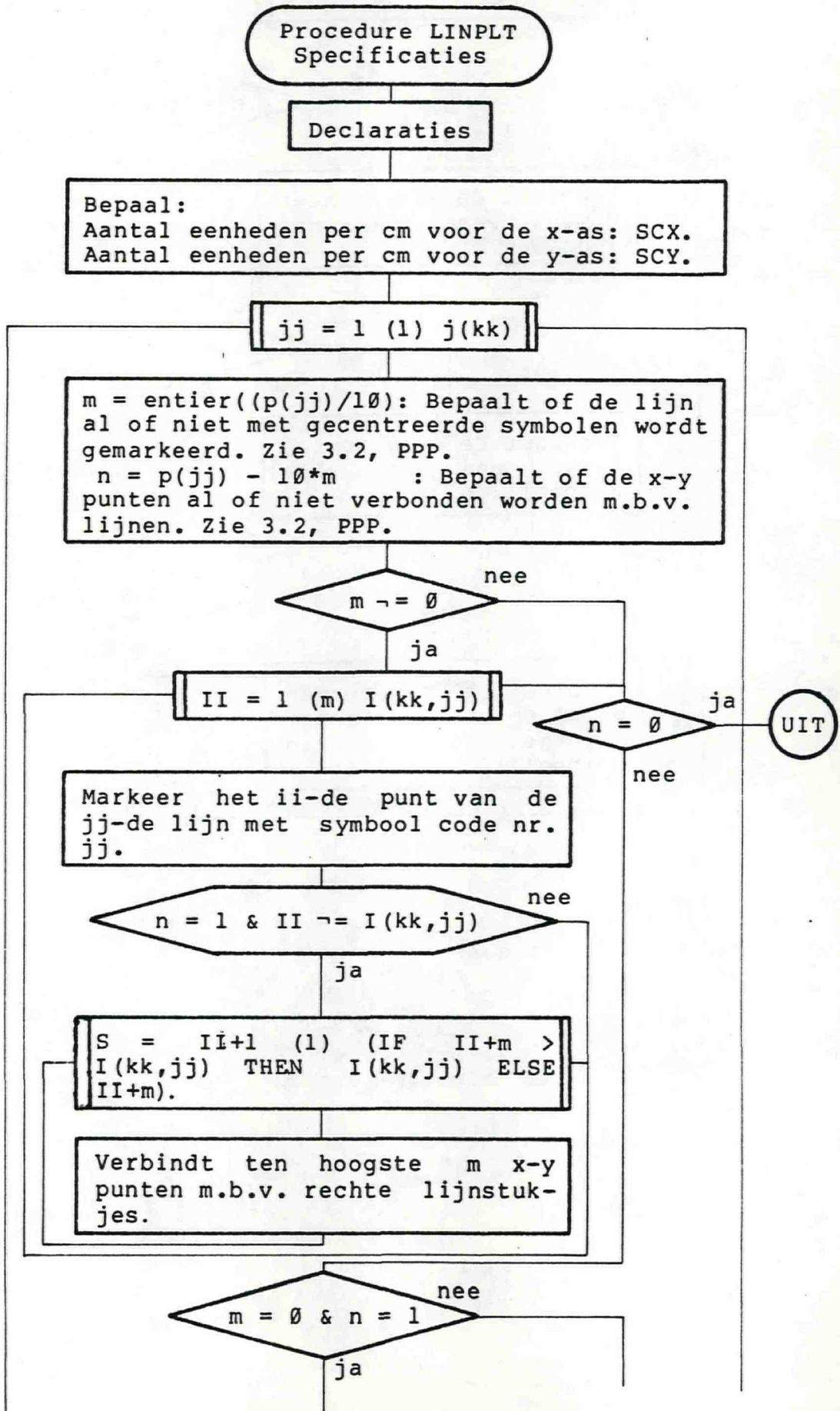
```
'PROCEDURE' LINPLT (I,J,K,P,X,Y,SX,NX,SY,NY, XXS,YYS,AXY,JJ,KK);  
'REAL' SX,SY,XXS, YYS;  
'INTEGER' NX,NY,AXY,K,JJ,KK;  
'ARRAY' X,Y;  
'INTEGER' 'ARRAY' I,J,P;
```

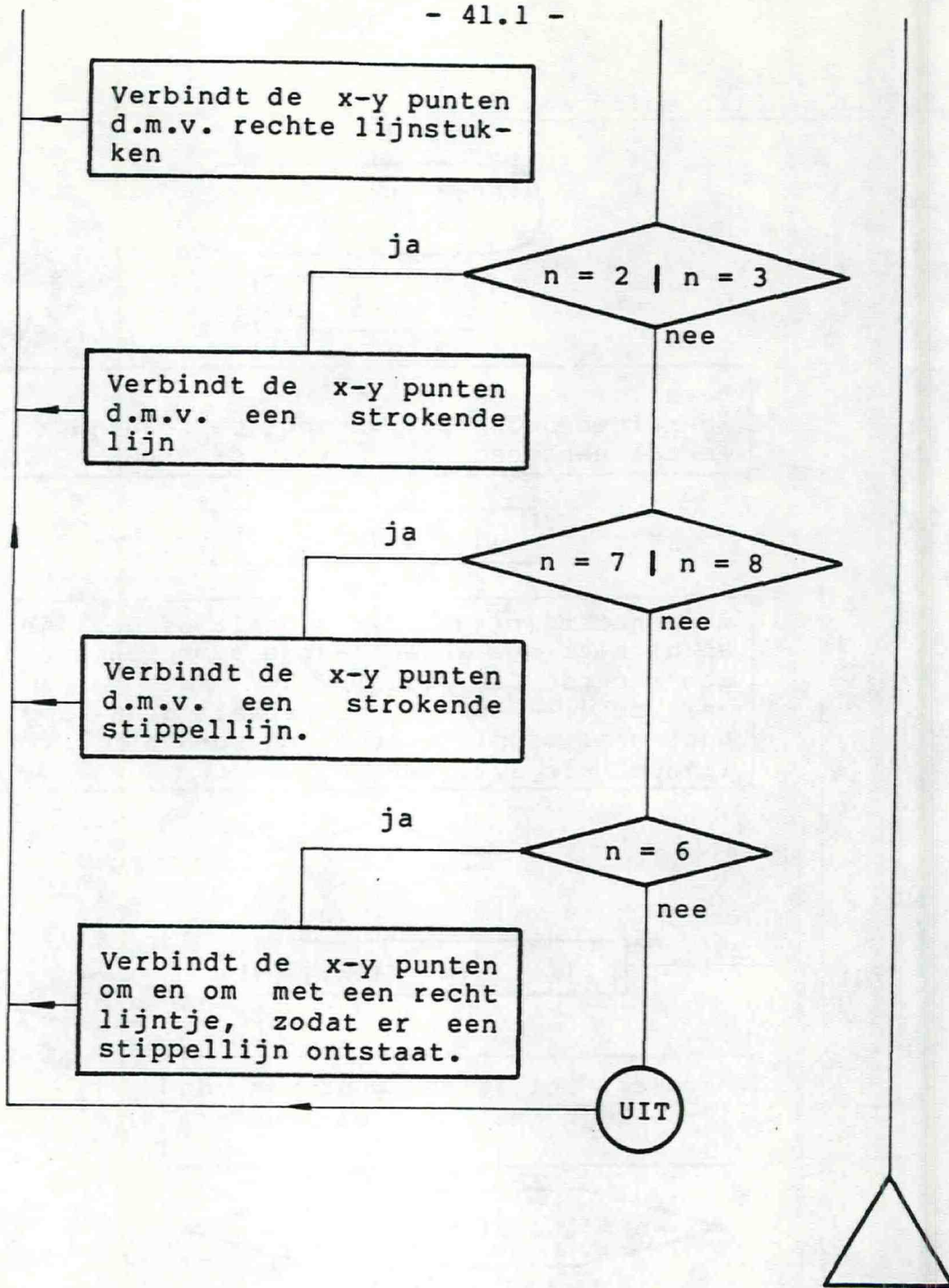
Aan de declaraties van het programma moet worden toegevoegd:
'PROCEDURE' LINPLT; 'CODE';

Betekenis van de formele parameters:

I	i : Zie 3.2
J	i : Zie 3.2
K	i : Zie 3.2
P(JJ)	= PPP(kk,jj) voor de kk-de grafiek, zie 3.2.
X	i : Zie 3.2
Y	i : Zie 3.2
SX	i : Het aantal eenheden per $\text{cm}/10^{NX}$ van de x-as. (wordt in de procedure TRANS bepaald door SCL).
NX	i : Zie parameter N bij procedure TRANS; NX = N voor de x-as.
SY	i : Het aantal eenheden per $\text{cm}/10^{NY}$ van de y-as. (wordt in de procedure TRANS bepaald door SCL).
NY	i : Zie parameter N bij procedure TRANS; NY = N voor de y-as.
XXS	i : Zie procedure XAS.
YYS	i : Zie procedure YAS.
AXY	i : $AXY := PAXY(kk)$ zie 3.2.
JJ	i : Lijnnummer van de kk-de grafiek.
KK	i : Grafieknummer.

I.12.4 Stroomdiagram.





CHARACTERS AVAILABLE IN THE IBM 360 SYMBOL ROUTINE

CODES NEXT TO EACH SYMBOL ARE: 1. INTEGER CODE USED IN SPECIAL SYMBOL CALL.
 2. INTERNAL HEXADECIMAL CODE (EBCDIC MODULO 128).
 SEE THE IBM 360 REFERENCE DATA CARD FOR CORRESPONDING PUNCHED CARD CODES.

0 00		16 10		32 20	}	48 30	Σ	64 40	80 50	&	96 60	-	112 70	0	
1 01		17 11	BS	33 21	{	49 31	\div	65 41	A 51	J 61	/	113 71	1		
2 02		18 12	^	34 22	μ	50 32	\leq	66 42	B 52	K 62	S	114 72	2		
3 03	+	19 13	\equiv	35 23	π	51 33	\geq	67 43	C 53	L 63	T	115 73	3		
4 04	X	20 14	\rightarrow	36 24	ϕ	52 34	Δ	68 44	D 54	M 64	U	116 74	4		
5 05		21 15	CR	37 25	\ominus	53 35	\square	69 45	E 55	N 65	V	117 75	5		
6 06		22 16	\neq	38 26	ψ	54 36	\square	70 46	F 56	O 66	W	118 76	6		
7 07		23 17	\pm	39 27	\times	55 37	\backslash	71 47	G 57	P 67	X	119 77	7		
8 08	Z	24 18	$_$	40 28	w	56 38	\Uparrow	72 48	H 58	Q 68	Y	120 78	8		
9 09	Y	25 19	NUL	41 29	λ	57 39	$\sqrt{\quad}$	73 49	I 59	R 69	Z	121 79	9		
10 0A		26 1A	$_$	42 2A	α	58 3A	\dagger	74 4A	ϕ 5A	\downarrow 6A	∞	122 7A	\circ		
11 0B		27 1B	\int	43 2B	δ	59 3B	\ddagger	75 4B	\circ 5B	\$ 6B	y	123 7B	#		
12 0C		28 1C	\supset	44 2C	ϵ	60 3C	\leftarrow	76 4C	< 5C	\times 6C	% 7C	124 7C	\odot		
13 0D		29 1D	\vee	45 2D	η	61 3D	\times	77 4D	(5D) 6D	$_$	125 7D	v		
14 0E		30 1E	\sim	46 2E	SUP	62 3E	\uparrow	78 4E	+	y 5E	y 6E	110 7E	>	126 7E	=
15 0F	$_$	31 1F	\approx	47 2F	SUB	63 3F	\downarrow	79 4F	 5F	$_$ 6F	?	111 7F	127 7F	v	

NOTES: 1. THE FIRST 14 SYMBOLS IN THE TABLE ARE CENTERED SYMBOLS.
 2. ANY INTEGER OR HEX CODE, OTHER THAN THOSE SHOWN, WILL BE CONSIDERED MODULO 128. THEREFORE EITHER OF TWO PUNCHED CARD CODES FROM THE 360 REFERENCE CARD MAY BE USED FOR EACH SYMBOL.

COMPILER VERSION : BATCH
OPTIONS IN EFFECT: SIZE(KBYTE(S))=154, IOL= 6, SEG=SEGM,
EBCDIC(E3), NDI(OLST,LOAD(L),LONG(LP),OPTO,NOPAG,SE,SOURCE(S),NST,SWAPO,TEST(T),M,NODUMP

```

0  *PROCEDURE *TCODE(X,Y,Q,ALFA,KLUS,TEST,SPRING); *VALUE Q; *REAL X,Y,Q; 0010
3  *INTEGER *ALFA,*LABEL *KLUS,TEST,SPRING; 0020
5  *BEGIN *REAL *X,*Y,*Q; *X:=X+Q; *Y:=Y+Q; *Q:=Q+Q; *ALFA:=ALFA+Q; *KLUS:=KLUS+Q; *TEST:=TEST+Q; *SPRING:=SPRING+Q; 0030
7  *A2:=A2+Q; *L1:=L1+Q; *ESP:=ESP+Q; *ESP2:=ESP2+Q; *ESP3:=ESP3+Q; 0040
9  *INTEGER *ARRAY *TC(1:200/),HTC(1:3/),HT(1:1/); *INTEGER *P,PP,CODE, 0050
9  I,R,K,N,SOUL,K; 0060
9  *REAL *PROCEDURE *L; 0070
10 *IF *TC(1:R/)>=32*TC(1:R/)<=45*TC(1:R/)=27*TC(1:R/)=46*TC(1:R/)=53*TC(1:R/)= 0080
11 54*THEN 0090
11 *BEGIN *L:=L+Q; *ESP:=ESP+Q; *IF *ALFA=90*THEN *BEGIN *A:=A1; *S:=Y; *END 0100
19 *ELSE *BEGIN *A:=X; *R:=Y+Q; *IF *ALFA=180*THEN *B1*ELSE *B1; *END; 0110
24 *END *ELSE 0120
26 *BEGIN *L:=Q; *A:=X; *R:=Y; *ESP:=Q; 0130
31 *END; 0140
32 *REAL *PROCEDURE *L2; 0150
33 *IF *TC(1:R/)>=32*TC(1:R/)<=45*TC(1:R/)=27*TC(1:R/)=48*TC(1:R/)=53*TC(1:R/)= 0160
34 54*THEN 0170
34 *BEGIN *L2:=L+Q; *ESP:=ESP+Q; *IF *ALFA=90*THEN *BEGIN *A:=A2; *B:=Y; *END 0180
42 *ELSE *BEGIN *A:=X; *B:=Y+Q; *IF *ALFA=180*THEN *B2*ELSE *B2; *END; 0190
47 *END *ELSE 0200
49 *BEGIN *L2:=Q; *A:=X; *B:=Y; *ESP:=ESP1; 0210
54 *END; 0220
55 *PROCEDURE *POS(A,B,C); 0222
55 *VALUE *C; *REAL *A,*B; *INTEGER *C; 00223000
59 *IF *ALFA=90*THEN *B:=B+C*Q*ELSE *A:=X+C*Q*(*IF *ALFA=180*THEN 00224000
63 -1*ELSE *L); 0230
63 *SOUL:=Q; *Q2:=Q*7; *P:=Q; *RXQ:=RX1:=X; *X2:=X; *RYQ:=RY1:=Y; *K:=Q; *L3:=.65 0240
70 *AL4:=Q2/3; *LK:=Q/5; *LK2:=Q2/6; *A1:=X+.1975*Q; *A2:=X+.1975*Q2; 0250
75 *B1:=Q*.1975; *B2:=Q2*.1975; *ESP1:=.15*Q; *ESP2:=.06*Q; *ESP3:=.3*Q; *RR:=Q; 0260
81 *OPN:=P:=Q; 0270
83 *LEES:=P:=PP+1; *INTEXT(0,HT,1); *HT(1/):=HT(1/)+( *IF *HT(1/)<64*THEN *64 0280
87 *ELSE *-64); 0290
97 *SO:=1; *HT(1/)=125; *P:=1; *THEN *GOTO *TEXT; *IF *HT(1/)=64 0300
91 *HT(1/)=107; *HT(1/)=123; *HT(1/)=124; *HT(1/)=125; *THEN *GOTO *DCODE; 0310
92 *HTC(1:PP/):=HT(1/)-1; *GOTO *LEES; 0320
94 *DCODE:=PP:=PP-1; *IF *PP=0*THEN 0330
97 *BEGIN *IF *HT(1/)=123; *HT(1/)=124*THEN 0340
97 * * * * *IN *P:=P+1; *TC(1:PP/):=HT(1/); *R:=1; *GOTO *SCHR1; *IF 0350
104 *END *ELSE *IF *HT(1/)=54; *HT(1/)=107*THEN 0360
107 *BEGIN *K:=K+1; *IF *K>91*THEN *GOTO *SPRING; *GOTO *OPN; 0370
112 *END; 0380
113 *END *P:=P+1; *CODE:=Q; *K:=Q; 0390
117 *OR *I:=1; *STEP *I; *UNTIL *PP=0; *CODE:=CODE+1; *HTC(1/); 0400
119 *TC(1:PP/):=CODE; *IF *HT(1/)=123; *HT(1/)=124*THEN 0410
121 *BEGIN *P:=P+1; *TC(1:PP/):=HT(1/); *R:=1; *GOTO *SCHR1; *IF 0420
124 *END; *IF *HT(1/)=125; *THEN *GOTO *TEXT; *GOTO *OPN; 0430
130 *TEXT:=INTEXT(0,HT,1); *HT(1/):=HT(1/)+( *IF *HT(1/)<64*THEN *64*ELSE *-64); 0440
134 *IF *HT(1/)=125; *THEN 0450
134 *BEGIN *K:=Q; 0460
135 *TO: *INTEXT(0,HT,1); *HT(1/):=HT(1/)+( *IF *HT(1/)<64*THEN *64*ELSE *-64);

```

```

139      *IF HT(/1/)=64 HT(/1/)=107 THEN*
140      *BEGIN *K:=K+1; *IF *K>81 *THE *GOTO *SPRING; *GOTO *T0;
141      *CW:=0; *IF *GOTO *S0;
142      *END; *P:=P+1; TC(/P/):=HT(/1/); *GOTO *TEXT;
143      *SCHEDULE: *IF *TC(/1/)=124 THEN *GOTO *S4; *IF *TC(/1/)<=13 THEN*
144      *BEGIN *IF *ALFA=0 THEN *MARK(X+0.5, Y, Q, TC(/1/), ALFA, -1) *ELSE *
145      *MARK(X, Y+0.5, Q, TC(/1/), ALFA, -1); *RR:=RR+1;
146      *END; *ELSE *IF *TC(/1/)=200 THEN*
147      *MARK(X, Y+1.5, Q, ALFA, -1) *ELSE *IF *TC(/1/)=201 THEN*
148      *BEGIN *MARK(X, Y+0.57, Q, ALFA, -1); *IF *ALFA=90 THEN*
149      *BEGIN *MARK(X, Y+0.57, Q, 2, 109, ALFA, -1); *MARK(X, Y+0.97, Q, 0.57, 109, ALFA,
150      * -1);
151      *END; *ELSE *
152      *BEGIN *MARK(X+0.67, Y, Q, 2, 109, ALFA, -1);
153      *MARK(X+0.97, Y, Q, 0.57, 109, ALFA, -1);
154      *END; *
155      *END; *ELSE *IF *TC(/1/)=202 THEN*
156      *BEGIN *MARK(X, Y, Q, 4, 109, ALFA, -1); *IF *ALFA=90 THEN*
157      *BEGIN *MARK(X, Y+0.57, Q, 4, 109, ALFA, -1);
158      *MARK(X, Y+1.14, Q, 4, 109, ALFA, -1);
159      *END; *ELSE *
160      *BEGIN *MARK(X+0.57, Y, Q, 4, 109, ALFA, -1);
161      *MARK(X+1.14, Y, Q, 4, 109, ALFA, -1);
162      *END; *
163      *END; *ELSE *
164      *BEGIN *LL:=1; *IF *ESP>.0001 THEN*
165      *BEGIN *MARK(A, B, ESP, 64, ALFA+180, -1);
166      *IF *ALFA=90 THEN*
167      *BEGIN *MARK(M, 999.0, LL, TC(/R/), ALFA, -1);
168      *MARK(A, 999.0, ESP, 64, ALFA+180, -1);
169      *END; *ELSE *
170      *BEGIN *MARK(999.0, B, ESP, 64, ALFA+180, -1);
171      *MARK(999.0, B, ESP, 64, ALFA+180, -1);
172      *END; *
173      *END; *ELSE *MARK(A, B, LL, TC(/R/), ALFA, -1);
174      *END; *RR:=RR+(*IF *TC(/R/)>=200 & TC(/R/)<=202 THEN *7 *ELSE *1);
175      *HTC(/1/):=HTC(/2/); *C;
176      *FOR *R:=2 *STEP *1 *UNTIL *P *DO*
177      *BEGIN *IF *R=0 THEN *GOTO *S4; *IF *TC(/R/)=62 THEN *
178      *BEGIN *R:=R+1; *B00L:=B00L+1; *IF *TC(/R-2/)=63 THEN *
179      *BEGIN *IF *ALFA=90 THEN *MARK(X, Y, Q, Q, 64, ALFA, -1)
180      *ELSE *MARK(X, Q, *IF *ALFA=180 THEN *Q *ELSE *-Q), Y, Q,
181      * 64, ALFA, -1);
182      *END; *ELSE *POS(R, X, Y, Q, RR);
183      *LL2:=LL2; *IF *ALFA=90 THEN*
184      *BEGIN *MARK(A, 999.0, ABS(ESP), 64, ALFA+(*IF *ESP<0 THEN *180
185      *ELSE *0), -1); *MARK(A-L3, 999.0, LL2, TC(/R/), ALFA, -1);
186      *MARK(A, 999.0, ABS(ESP), 64, ALFA+(*IF *ESP<0 THEN *180 *ELSE *0)
187      *-1);
188      *END; *ELSE *
189      *BEGIN *MARK(999.0, R, ABS(ESP), 64, ALFA+(*IF *ESP<0 THEN *180
190      *ELSE *0), -1); *MARK(999.0, 9+(*IF *ALFA=180 THEN *-L3 *ELSE *L3)
191      *LL2, TC(/R/), ALFA, -1); *MARK(999.0, B, ABS(ESP), 64, ALFA+(*IF *
192      *ESP<0 THEN *180 *ELSE *0), -1);

```

```

250      *END:HTC(/I/)=HTC(/I/)+1;GOTO E INDE;
253      *END:POS(RX,SY,RR+HTC(/I/));R:=R+1;IF R=P THEN *GOTO S4;
258      *IF TC(/R/)=63 THEN *
259      *BEGIN R:=R-1;R00L:=0;*GOTO E INDE;
263      *END;*S0TC LA3;
265      *END;*I:=TC(/R/)-63 THEN *
267      *BEGIN *
269      *IF BOOL=1 THEN *GOTO S1;*IF BOOL=0 THEN *
271      *BEGIN *IF TC(/R/)=13 THEN *
273      *BEGIN *IF ALFA=90 THEN *MARK(X-Q*.5,999.0,0,TC(/R/),ALFA,-1)
276      *ELSE *MARK(999.0,Y+Q*.5,Q,TC(/R/),ALFA,-1);
278      *END *
279      *ELSE *IF TC(/R/)=200 THEN *MARK(999.0,999.0,1.54,109,ALFA,-1)
282      *ELSE *IF TC(/R/)=201 THEN *
284      *BEGIN *WHERE(RX,RY,F);MARK(RX,RY,0.57,109,ALFA,-1);
287      *IF ALFA=90 THEN *
289      *BEGIN *MARK(RX,RY+0.67,0.2,109,ALFA,-1);
290      *MARK(RX,RY+0.97,0.57,109,ALFA,-1);
291      *END *ELSE *
293      *BEGIN *MARK(RX+0.67,RY,0.2,109,ALFA,-1);
295      *MARK(RX+0.97,RY,0.57,109,ALFA,-1);
296      *END *;
297      *END *ELSE *IF TC(/R/)=202 THEN *
300      *BEGIN *WHERE(RX,RY,F);MARK(RX,RY,0.4,109,ALFA,-1);
303      *IF ALFA=90 THEN *
304      *BEGIN *MARK(RX,RY+0.57,0.4,109,ALFA,-1);
306      *MARK(RX,RY+1.14,0.4,109,ALFA,-1);
307      *END *ELSE *
309      *BEGIN *MARK(RX+0.57,RY,0.4,109,ALFA,-1);
311      *MARK(RX+1.14,RY,0.4,109,ALFA,-1);
312      *END *;
313      *END *ELSE *
315      *BEGIN *LL:=L;*IF ALFA=90 THEN *
318      *BEGIN *IF ESP>.00001 THEN *
320      *MARK(A,999.0,ESP,64,ALFA+180,-1);
321      *MARK(A,999.0,LL,TC(/R/),ALFA,-1);
322      *IF ESP>.00001 THEN *
323      *MARK(A,999.0,ESP,64,ALFA+180,-1);
324      *END *ELSE *
326      *BEGIN *IF ESP>.00001 THEN *
329      *MARK(999.0,8,ESP,64,ALFA+180,-1);
329      *MARK(999.0,8,LL,TC(/R/),ALFA,-1);
330      *IF ESP>.00001 THEN *
331      *MARK(999.0,8,ESP,64,ALFA+180,-1);
332      *END *;
333      *END *;R:=R+1;*IF TC(/R/)>=200&TC(/R/)<=202 THEN *7*ELSE *1);
335      *GOTO E INDE;
336      *END *;IF BOOL=-1 THEN *
338      *BEGIN *
339      *LL2:=L2;*IF ALFA=90 THEN *
342      *BEGIN *MARK(A,999.0,ABS(ESP),64,ALFA+180,ALFA+180,ALFA,-1);
344      *ELSE *0,-1;MARK(A+02,999.0,LL2,TC(/R/),ALFA,-1);
345      *MARK(A,999.0,ABS(ESP),64,ALFA+180,ALFA+180,ALFA,-1);
346      *END *ELSE *
345

```

```

0838
0840
0850
0850
0860
0870
0880
0880
0890
0900
0910
0920
0930
0940
0950
0960
0970
0980
0990
1000
1010
1020
1030
1040
1050
1060
1070
1080
1090
1100
1110
1120
1130
1140
1150
1150
01159000
01160000
1161
01162000
01163000
01164000
01165000
01166000
01167000
01168000
01169000
01170000
01171000
1175
1180
1190
1200
1210
1220
1230
1231
1232

```

```

349 *BEGIN*MARK(999.0,B,ABS(ESP),64,ALFA+(.IF*ESP<0*THEN*180
350 *ELSE*0),-1);MARK(999.0,B+(.IF*ALFA=180*THEN*Q2*ELSE*-Q2)
351 *LL2,TC(R/),ALFA,-1);MARK(999.0,B,ABS(ESP),64,ALFA+
352 (.IF*ESP<0*THEN*180*ELSE*0),-1);
353 *END*;*HTC(/2/):=HTC(/2/)+1;*GOTO*EINDE;
354 *END*;*GOTO*S3;
355 *END*;*R:=K+1;*BOOL:=6*COL-1;*IF*R=P*THEN*GOTO*S4;
356 *IF*BOOL=-1*THEN*
357 *BEGIN*;*IF*TC(/2-2/)=62*THEN*
358 *BEGIN*;*IF*ALFA=90*THEN*MARK(X,R*Q-Q,64,ALFA,-1)*ELSE*
359 *MARK(R*X+(.IF*ALFA=180*THEN*Q*ELSE*-Q),Y,Q,64,ALFA,-1);
360 *END*;*ELSE*POS(R*X,Q,R);
361 *LL2:=L2;*IF*ALFA=90*THEN*
362 *BEGIN*MARK(A,999.0,ABS(ESP),64,ALFA+(.IF*ESP<0*THEN*180*ELSE*0
363 ),-1);MARK(A,4,999.0,LL2,TC(R/),ALFA,-1);MARK(A,999.0,
364 ABS(ESP),64,ALFA+(.IF*ESP<0*THEN*180*ELSE*0),-1);
365 *END*;*ELSE*
366 *5*GIN*MARK(999.0,B,ABS(ESP),64,ALFA+(.IF*ESP<0*THEN*180*ELSE*0
367 ),-1);MARK(999.0,B+(.IF*ALFA=180*THEN*Q4*ELSE*-L4),LL2,
368 TC(R/),ALFA,-1);MARK(999.0,B,ABS(ESP),64,ALFA+(.IF*ESP<0
369 *THEN*180*ELSE*0),-1);
370 *END*;*HTC(/2/):=HTC(/2/)+1;*GOTO*EINDE;
371 *END*;*IF*BOOL=-2*TC(/2/)=63*THEN*
372 *BEGIN*;*R:=R+1;*IF*R=P*THEN*GOTO*S4;*GOTO*S2;
373 *END*;*PJS(RX2,PY2,RR+HTC(/2/));*IF*TC(/K/)=62*THEN*
374 *BEGIN*BOOL:=0;*R:=R-1;*GOTO*EINDE;
375 *END*;*
376 *LAR:=RR*(.IF*HTC(/1/)>=HTC(/2/))*THEN*HTC(/1/)*ELSE*HTC(/2/));
377 *R:=R-1;*BOOL:=0;*IF*ALFA=90*THEN*MARK(X,.IF*R*Q>=R*Y1*R*Q>=R*Y2*THEN*
378 *RYC-Q*ELSE*(.IF*RY1>=R*Y0&RY1>=R*Y2*THEN*RY1-Q*ELSE*RY2-Q,Q,
379 *64,ALFA,-1)*ELSE*(.IF*ALFA=180*THEN*MARK(.IF*R*X0<=R*X1&R*X0<=R*X2*THEN
380 *R*X0+Q*ELSE*(.IF*R*X0&R*X1<=R*X2*THEN*R*X1+Q*ELSE*R*X2+Q,Y,
381 *Q,64,ALFA,-1)*ELSE*MARK(.IF*R*X0>=R*X1&R*X0>=R*X2*THEN*R*X0-Q*ELSE*
382 *IF*R*X1>=R*X0&R*X1>=R*X2*THEN*R*X1-Q*ELSE*R*X2-Q,Y,Q,64,ALFA,
383 *-1);*HTC(/1/):=HTC(/2/):=0;
384 *EINDE;
385 *END*;*
386 *S4*;*IF*TC(/P/)=122*THEN*
387 *BEGIN*SYSACT(0,1,4,1)*GOTO*KLUS;
388 *END*;*IF*TC(/P/)=124*THEN*
389 *REGI(*SYSACT(0,1,4,1)*GOTO*TEST;
390 *END*;*
391 *END*PROCEDURE TCUDE;

```

NO ERRORS FOUND

```

COMPIER VERSION : INCOME.
OPTIONS IN EFFECT: SIZE(KRYTHS)= 204,IDL= 6,
CPU(CSECS)= 250,CHD= 500,JCL=//,PGS= 10,
EBCDIC(FB),NOIDLST,LOAD(L),LONG(LP),OPTO,NOPAG,
SE, SOURCE(S),NST,SWAPO,TEST(T),W,NODUPP

0  *PROCEDURE * MINMAX(X,MIN,MAX,I,J,KK);
1  *VALUE * X;
2  *REAL * MIN,MAX;
3  *INTEGER * KK;
4  *REAL * AHAY, X;
5  *INTEGER * AHAY, I,J;
6  *BEGIN *
7  *   *INTEGER * JJ,II;
8  *   *REAL * D;
9  *   MIN:=MAX:=0;
10 *   *FOR * JJ:=J(/KK/) *STEP* -1 *UNTIL* 1 *DO*
11 *     *FOR * II:=I(/KK,JJ/) *STEP* -1 *UNTIL* 1 *DO*
12 *       *BEGIN *
13 *         *D:=X(/KK,JJ,II/);
14 *         *IF * D < MIN *THEN* MIN:=D *ELSE*
15 *           *IF * D > MAX *THEN* MAX:=D;
16 *       *END *;
17 *     *END *;
18 *   *END * PROCEDURE MINMAX;
19
20

```

```

00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190

```

COMPILER VERSION : INCOME

OPTIONS IN EFFECT: SIZE(KBYTES)= 204,IDL= 6,
CPU(CSECS)= 250,CRD= 500,JCL=//,PGS= 10,
EBCDIC(EH),NOIDLST,LOAD(L),LONG(LP),OPTO,NOPAG,
SE, SOURCE(S),NSI,SWAPO,TEST(T),W,NODUMP

```

0  *PROCEDURE* SCALX(A,MIN,MAX,NUL,DEL,NSP,SCL,N,CX,DX,CY,DY);
1  *HEAL* MIN,MAX,NUL,DEL,SCL,CX,CY;
2  *INTEGER* NSP,A,N,DX,DY;
3  *BEGIN*
4  *PROCEDURE* IRANS: 'CODE';
5  *HEAL* D,STAP,HN,SCLD;
6  *INTEGER* ABSD;
7  D:= 'IF' A = 1 'THEN'
8  (*'IF' ABS(CX) < '-6' THEN* 22 *ELSE* CX*DX)
9  *ELSE*
10 (*'IF' ABS(CY) < '-6' THEN* 22 *ELSE* CY*DY);
11 STAP:=ABS((MAX-MIN)/D);
12 IRANS(STAP,SCL,N,A,CX,CY);
13 *IF' A = 1 & ABS(CX) < '-6 | A = 2 & ABS(CY) < '-6 *THEN'
14 *DEL:='IF' ENTIER(SCL*.01) = 4 *THEN* 2.5 *ELSE* 2;
15 *NSP:='IF' ENTIER(DEL*.51) = 2 *THEN* 11 *ELSE* 9;
16 *ABSD:='IF' A = 1 *THEN* DX *ELSE* DY;
17 *ABSD:=ABS(ABSD);
18 *IF' ABSD < '-6 *THEN' *GO TO* WEG;
19 SCLD:=SCL*DEL;
20 *IF' SCLD > 9 *THEN*
21 *BEGIN* SCLD:=.1*SCLD;
22 *GO TO* TR;
23 *END*;
24 *IF' ABSD < SCLD*NSP *THEN* NSP:=ENTIER(ABSD/SCLD*.65);
25
26 WEG:
27
28 *COMMENT* SCL = EENHEDEN/CM
29 SCLD = EENHEDEN/VAKJE
30 NSP = AANTAL VAKJES;
31
32 *END*
33 *ELSE*
34 *BEGIN* DEL:='IF' A = 1 *THEN* CX *ELSE* CY;
35 *NSP:='IF' A = 1 *THEN* DX *ELSE* DY;
36 *END*;
37 *IF' A = 2 *THEN* DEL:=-DEL;
38 HN:=DEL*NSP/2;
39 NUL:=-HN*( 'IF' A = 1 *THEN* 17 *ELSE* 12.75);
40
41 *END* EINDE PROCEDURE SCALX1;

```

00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420

COMPILER VERSION : INCORE

OPTIONS IN EFFECT: SIZE(KBYTES)= 204,IDL= 6,
CPU(CSECS)= 250,CRD= 500,JCL=//,PGS= 10,
EBCDIC(FB),NOIDLST,LOAD(L),LONG(LP),OPTO,NOPAG,
SE, SOURCE(S),NSI,SWAPO,TEST(I),M,NODUMP

```

0  *PROCEDURE SCALY1(A,MIN,MAX,NUL,DEL,NSP,SCL,N,CX,DX,CY,DY);
1  *REAL MIN,MAX,NUL,DEL,SCL,CX,CY;
2  *INTEGER NSP,A,N,DX,DY;
3  *BEGIN
4  *PROCEDURE IRANS; *CODE;
5  *REAL D,STAP,HN,SCLD;
6  *INTEGER AMSD;
7  D:=IF A = 1 *THEN
8  (*IF ABS(CY) < *-6 *THEN* 14 *ELSE* CY*DY)
9  *ELSE;
10 (*IF ABS(CX) < *-6 *THEN* 14 *ELSE* CX*DX);
11 STAP:=ABS((MAX-MIN)/D);
12 IRANS(STAP,SCL,N,A,CY,CX);
13 *IF A = 1 & ABS(CY) < *-6 | A = 2 & ABS(CX) < *-6 *THEN*
14 *BEGIN DEL:=IF ENTIER(SCL+.01) = 4 *THEN* 2.5 *ELSE* 2;
15 NSP:=IF ENTIER(DEL+.51) = 2 *THEN* 7 *ELSE* 6;
16 AMSD:=IF A = 1 *THEN* DY *ELSE* DX;
17 AMSD:=AMS(AMSD);
18 *IF AMSD < *-6 *THEN* 'GO TO' WEG;
19 SCLD:=SCL*DEL;
20 *IF SCLD > 9 *THEN*
21 *BEGIN SCLD:=.1*SCLD;
22 *GO TO* TR;
23 *END;
24 *IF AMSD < SCLD*NSP *THEN* NSP:=ENTIER(AMSD/SCLD+.65);
25 WEG:
26
27
28 *COMMENT SCL = ENHEDEN/CM
29 SCLD = ENHEDEN/VAKJE
30 NSP = AANTAL VAKJES;
31
32 *END;
33 *ELSE;
34 *BEGIN DEL:=IF A = 1 *THEN* CY *ELSE* CX;
35 NSP:=IF A = 1 *THEN* DY *ELSE* DX;
36 *END;
37 HN:=DEL*NSP/2;
38 NUL:=12-HN;
39 *END EINDE PROCEDURE SCALY1;

```

```

00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410

```

```

COMPILER VERSION : PATCH
OPTIONS 1. EFFECT: SIZE(CY)C=15, IDLE = 6, SEG=SEGM,
        LACBIC(4), NOT(DEL, A(DA, DL))LONG(LP), OPTG, NOPAG, SE, SOURCE(S), NST, SWAPO, TEST(T), W, NODUMP
0  *PROCEDURE SCALX2(A, MIN, MAX, NUL, DEL, NSP, SCL, N, CX, DX, CY, DY);
1  *REAL N, MAX, NUL, DEL, SCL, CX, CY;
2  *INTEGER NSP, A, N, DX, DY;
3  *BEGIN;
4  *PROCEDURE TRANS;
5  *PROCEDURE SCALX1;
6  *REAL H;
7  *IF A=0 THEN H:=1;
8  *IF A=1 THEN H:=2;
9  *IF A=2 THEN H:=3;
10 *IF A=3 THEN H:=4;
11 *IF A=4 THEN H:=5;
12 *IF A=5 THEN H:=6;
13 *IF A=6 THEN H:=7;
14 *IF A=7 THEN H:=8;
15 *IF A=8 THEN H:=9;
16 *IF A=9 THEN H:=10;
17 *IF A=10 THEN H:=11;
18 *IF A=11 THEN H:=12;
19 *IF A=12 THEN H:=13;
20 *IF A=13 THEN H:=14;
21 *IF A=14 THEN H:=15;
22 *IF A=15 THEN H:=16;
23 *IF A=16 THEN H:=17;
24 *IF A=17 THEN H:=18;
25 *IF A=18 THEN H:=19;
26 *IF A=19 THEN H:=20;
27 *IF A=20 THEN H:=21;
28 *IF A=21 THEN H:=22;
29 *IF A=22 THEN H:=23;
30 *IF A=23 THEN H:=24;
31 *IF A=24 THEN H:=25;
32 *IF A=25 THEN H:=26;
33 *IF A=26 THEN H:=27;
34 *IF A=27 THEN H:=28;
35 *IF A=28 THEN H:=29;
36 *IF A=29 THEN H:=30;
37 *IF A=30 THEN H:=31;
38 *IF A=31 THEN H:=32;
39 *IF A=32 THEN H:=33;
40 *IF A=33 THEN H:=34;
41 *IF A=34 THEN H:=35;
42 *IF A=35 THEN H:=36;
43 *IF A=36 THEN H:=37;
44 *IF A=37 THEN H:=38;
45 *IF A=38 THEN H:=39;
46 *IF A=39 THEN H:=40;
47 *IF A=40 THEN H:=41;
48 *IF A=41 THEN H:=42;
49 *IF A=42 THEN H:=43;
50 *IF A=43 THEN H:=44;
51 *IF A=44 THEN H:=45;
52 *IF A=45 THEN H:=46;
53 *IF A=46 THEN H:=47;
54 *IF A=47 THEN H:=48;
55 *IF A=48 THEN H:=49;
56 *IF A=49 THEN H:=50;

```

DIAGNOSTICS
6 NS: CODE; TRANS WARNING 49 PROCEDURE NOT FOUND IN LIBRARY
8 XI: CODE; SCALX1 WARNING 49 PROCEDURE NOT FOUND IN LIBRARY
TOTAL NUMBER OF MESSAGES IS: 2

01080000
01090000
01100000
01110000
01120000
01130000
01140000
01150000
01160000
01170000
01180000
01190000
01200000
01210000
01220000

01320000
01330000
01340000
01360000

FAST ALGJL COMPILER OF DELFT, RELEASE OF 1/ 6/1976

INCR.NR.+SOURCE LISTING

```

COMPILER VERSION : BATCH
OPTIONS IN EFFECT: SIZE(KBYTES)=154,IDL= 6,SEG=SEGM,
EBCDIC(EB),WJDLST,LOAD(L),LUNG(LP),OPTJ,NOPAG,SE,SOURCE(S),NST,SWAPO,TEST(T),d,NJDUUMP
0 *PROCEDURE* SCALYZ(A,MIN,MAX,NUL,DEL,NSP,SCL,N,CX,DX,CY,DY);
1 *REAL* MIN,MAX,NUL,DEL,SCL,CX,CY;
2 *INTEGER* NSP,A,N,DX,DY;
3 *BEGIN*
4 *PROCEDURE* TRANS; *CODE*;
5 *PROCEDURE* SCALY1; *CODE*;
6
7 *REAL* HN;
8 *IF* A=1 *AND* ABS(CY)>=6 *AND* ABS(CX)>=6 *THEN*
9 *BEGIN* *REAL* HULP;
10 DEL:=*IF* A=1 *THEN* CY *ELSE* CX;
11 NSP:=*IF* A=1 *THEN* DY *ELSE* DX;
12 HULP:=ABS(MAX-MIN)/(DEL*NSP);
13 TRANS(HULP,SCL,N,A,CY,CX);
14 *END* *ELSE*
15 *IF* ABS(MAX-MIN)<=30 *THEN*
16 SCALY1(A,MIN,MAX,NUL,DEL,NSP,SCL,N,CX,DX,CY,DY) *ELSE*
17 *IF* ABS(MAX-MIN)>=30 *AND* ABS(CX)<=150 *THEN*
18 *BEGIN*
19 SCL:=10;N:=0;DEL:=3;NSP:=5;
20 *END* *ELSE*
21 *IF* ABS(MAX-MIN)>=150 *AND* ABS(MAX-MIN)<=360 *THEN*
22 *BEGIN*
23 SCL:=2.5;N:=1;NSP:=4;DEL:=3.6;
24 *END* *ELSE*
25 *IF* ABS(MAX-MIN)>=360 *THEN*
26 *BEGIN*
27 SCL:=5;N:=1;NSP:=2;DEL:=1.8;
28 *END*;
29 HN:=DEL*NSP/2;
30 NUL:=12*HN;
31 *END* *EINDE* PROCEDURE SCALYZ;
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

```

01370000
01380000
01390000
01400000
01410000
01420000
01430000
01440000
01450000
01460000
01470000
01480000
01490000
01500000
01510000
01520000
01530000
01540000
01550000
01620000
01630000
01640000

```

```

DIAGNOSTICS
6 NS:CODE TRANS WARNING 49 PROCEDURE NOT FOUND IN LIBRARY
8 Y1:CODE SCALY1 WARNING 49 PROCEDURE NOT FOUND IN LIBRARY
TOTAL NUMBER OF MESSAGES IS: 2

```

COMPILER VERSION : BATCH

```

OPTIONS IN EFFECT: SIZE(KBYTES)= 64, IOL= 6, SEG=SEGM,
                  EBCDIC(FB), IDLST, LOAD(L), LUNG(LP), OPTO, NOPAG, SE, SOURCE(S), NST, SWAP I, TEST(T), *
C * PROCEDURE TRANS (STAP, SCL, N, AX, CX, CY);
  1 * REAL STAP, SCL, CX, CY; * INTEGER N, AX;
  3 * BEGIN
  4   N:=0;
  5   S1: *IF STAP>10|STAP<=1 THEN*
  7 * BEGIN
  8 *IF STAP>10 THEN*
  9 * BEGIN
 10   STAP:=STAP/10; N:=N+1; GOTO S1;
 13 * END * ELSE
 15 * BEGIN
 16   STAP:=STAP*10; N:=N-1; GOTO S1;
 19 * END;
 20 * END;
 21 *IF AX=1&ABS(CX)>=-6|AXY=2&ABS(CY)>=-6 THEN*
 22 SCL:=STAP*ELSE*
 24 * BEGIN
 25   SCL:=*IF STAP>1&STAP<=2 THEN*2*ELSE*
 26 *IF STAP>2&STAP<=2.5 THEN*2.5*ELSE*
 26 *IF STAP>2.5&STAP<=4 THEN*4*ELSE*
 26 *IF STAP>4&STAP<=5 THEN*5*ELSE*10;
 26 * END;
 27 * END * EINDE PROCEDURE TRANS;

```

NO ERRORS FOUND

```

00190000
00200000
00210000
00220000
00230000
00240000
00250000
00260000
00270000
00280000
00290000
00300000
00310000
00320000
00330000
00340000
00350000
00360000
00370000

```

```

COMPILER VERSION : INCOMP
OPTIONS IN EFFECT: SIZE(KBYTES) = 204, IDL = 6,
CPU(CSECS) = 250, CRD = 500, JCL = //, PGS = 10,
EBCDIC(EB), MIDLST, LOAD(L), LONG(LP), OPTO, NOPAG,
SE, SOURCE(S), NSI, SWAPO, TESTI(T), W, MODUMP

0  *PROCEDURE XAS( DEL, NSP, SCL, N, XXS, XMIN, XMAX, XXY, CODE ):
1  *REAL DEL, SCL, XXS, XMIN, XMAX;
2  *INTEGER N, NSP, XXY, CODE;
3  *BEGIN
4  *INTEGER *PROCEDURE SCHUIF; *CODE*;
5  *INTEGER *PROCEDURE NDC; *CODE*;
6  *HEAL PLUSX, ELUSY, H1, H2, H3, XX, XH, DELK;
7  *INTEGER K, NDC;
8  H2 := SCL * DEL * 10 ** N;
9  XH := *IF ABS(XMIN) < 10 ** (-5 * N) * THEN XMIN * ELSE
10 *IF ABS(XMAX) < 10 ** (-5 * N) * THEN -ABS(H2) * HSP * ELSE
11 *ENTIER(AFS(-.99 * XMIN / H2) * 1) * AFS(H2) * SIGN(XMIN);
12 NDC := NDC(DEL, SCL, N, XH);
13 *IF XXY = 1 * THEN
14 *BEGIN * PLUSX := -.4;
15 *FOR K := 0 * STEP 1 * UNTIL NSP * DO
16 *BEGIN * XX := H * K * H2;
17 *IF K = 0 * THEN * XYS := XX;
18 H1 := .125 * SCHUIF(XX, N, NDEC);
19 DELK := DEL * K;
20 *IF * CODE < 0 * THEN
21 *BEGIN * PLOI(DELK, -.1, 3);
22 * PLOI(DELK, 0, 2);
23 * PLOI(DELK, -.1, 2);
24 *END*;
25 NUMBER(-H1 * DELK, PLUSX, .25, XX, 0,
26 *IF ABS(XX) < 10 ** (-5 * N) * THEN -1 * ELSE * NDEC);
27 *END*;
28 *END*;
29 *ELSE
30 *BEGIN * PLUSX := .4;
31 *FOR K := 0 * STEP 1 * UNTIL NSP * DO
32 *BEGIN * XX := H * K * H2;
33 *IF K = 0 * THEN * XYS := XX;
34 H1 := .125 * SCHUIF(XX, N, NDEC);
35 DELK := DEL * K;
36 *IF * CODE < 0 * THEN
37 *BEGIN * PLOI(.1, DELK, 3);
38 * PLOI(0, DELK, 2);
39 * PLOI(.1, DELK, 2);
40 *END*;
41 *END*;
42 *IF ABS(XX) < 10 ** (-5 * N) * THEN -1 * ELSE * NDEC);
43 *END*;
44 *END* EIHDE PROCEDURE XAS;
45
46
47
48
49
50
51

```

```

00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000361
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

```

COMPILER VERSION : INCORE

OPTIONS IN EFFECT: SIZE(KBYTES) = 204, IDL = 6,
CPU(CSECS) = 250, CRD = 500, JCL = //, PGS = 10,
EBCDIC(ER), NOIDLEST, LOAD(L), LONG(LP), OPTO, NOPAG,
SE, SOURCE(S), NST, SWAPO, TEST(T), W, NODUMP

```

0 *PROCEDURE YAS(DEL, NSP, SCL, N, YYS, YMIN, YMAX, AXY, CODE);
1 *REAL DEL, SCL, YYS, YMAX, YMIN;
2 *INTEGER AXY, CODE, N, NSP;
3 *BEGIN
4 *INTEG * * PROCEDURE SCHUIF; CODE;
5 *INTEG * * PROCEDURE NDC; CODE;
6 *REAL HI, H2, YH, H3, XX, PPLUS, DELK;
7 *INTEG *K, NDEC;
8 H2 = SCL * DEL * 10 ** N;
9 YH = * IF * ABS(YMIN) < 10 ** (-5 * N) * THEN * YMIN * ELSE *
10 * IF * ABS(YMAX) < 10 ** (-5 * N) * THEN * -ABS(H2) * NSP * ELSE *
11 * RTIER(A * S * (.99 * YMIN / H2) + 1) * ABS(H2) * SIGN(YMIN);
12 NDEC = NDC(DEL, SCL, N, YH);
13 *FOR *K := 0 * STEP * 1 * UNTIL * NSP * DO *
14 *BEGIN * * IF * AXY = 1 * THEN *
15 *BEGIN * XX := YH * K * H2;
16 DELK := DEL * K;
17 *IF * K = 0 * THEN * YYS := XX;
18 H1 := .25 * SCHUIF(XX, N, NDEC);
19 *IF * CODE < 0 * THEN *
20 *BEGIN * PLOI(-.1, DELK, 3);
21 PLOI(0, DELK, 2);
22 PLOI(-.1, DELK, 2);
23 *END *;
24 NUMBER(-.1 - H1, -.1 * DELK, .25, XX, 0,
25 *IF * ABS(XX) < 10 ** (-5 * N) * THEN * -1 * ELSE * NDEC);
26 *END *
27 *ELSE *
28 *BEGIN * XX := YH * K * H2;
29 DELK := DEL * K;
30 *IF * K = 0 * THEN * YYS := XX;
31 H1 := .25 * SCHUIF(XX, N, NDEC);
32 *IF * CODE < 0 * THEN *
33 *BEGIN * PLOI(DELK, -.1, 3);
34 PLOI(DELK, 0, 2);
35 PLOI(DELK, -.1, 2);
36 *END *;
37 NUMBER(.1 * DELK, -.1 - H1, .25, XX, 90,
38 *IF * ABS(XX) < 10 ** (-5 * N) * THEN * -1 * ELSE * NDEC);
39 *END *;
40 *END *;
41 *END * EIIDE PROCEDURE YAS;
42

```

00000040
00000050
00000060
00000070
00000080
00000090
00001000
0000110
0000120
0000130
0000140
0000150
0000160
0000170
0000180
0000190
0000200
0000210
0000220
0000230
0000240
0000250
0000260
0000270
0000280
0000290
0000300
0000310
0000320
0000330
0000340
0000350
0000360
0000370
0000380
0000390
0000400
0000410
0000420
0000430
0000440

```
COMPILER VERSION : INCORE
OPTIONS IN EFFECT: SIZE(KBYTES)= 204, IDL= 6,
CPU(CSECS)= 250, CRD= 500, JCL=//, PGS= 10,
EBCDIC(EH), NOIDLST, LOAD(L), LONG(LP), OPTO, NOPAG,
SE, SOURCE(S), NSI, SWAPU, TEST(T), M, NODUMP

0  *INTEGER* *PROCEDURE* SCHUIF(XX,N,NDEC);
1  *HEAL* XX;
2  *INTEGER* N,NDEC;
3  *BEGIN* *INTEGER* I,SCH,F;
5  *HEAL* H;
6  *IF* ABS(XX) < 10**(-5+N) *THEN*
7  *BEGIN* SCHUIF:=1;
9  *GO TO* EINDE;
10 *END*;
11 H:=ABS(XX)**-6;
12 SCHUIF:=SCH:=1;
13 F:=0;
14 T1: F:=F+1;
16 *IF* H >= 10**F *THEN*
17 *BEGIN* SCH:=SCH+1;
19 *GO TO* T1;
20 *END*;
21 *IF* XX < 0 *THEN* SCH:=SCH+1;
23 *IF* NDEC > 0 *THEN* SCH:=SCH+NDEC+1;
25 SCHUIF:=SCH;
26 *END* PROCEDURE SCHUIF;
```

```
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
```

COMPILER VERSION : INCONF
 OPTIONS IN EFFECT : SIZE(KBYTES) = 204,IDL = 6,
 CPU(CSECS) = 250,CHD = 500,JCL=//,PGS = 10,
 ERCDIC(FB),NCIDLST,LOAD(1),LONG(LP),OPTO,NOPAG,
 SE,SOURCE(S),NSI,SWAPO,TEST(T),N,NODUMP

```

0      *INTEGER* *PROCEDURE* NDC(DEL,SCL,N,MIN);
1      *REAL* DEL,SCL,MIN;
2      *INTEGER* N;
3      *BEGIN*
4      *REAL* NUM,MN,DELTA;
5      *INTEGER* E;
6      NUM:=ABS(DEL*SCL*10**N);
7      MN:=NUM-ENTIER(NUM);
8      MN:=ABS(MN);
9      MN:=MN-ENTIER(MN);
10     E:=-1;
11     DELTA:=*-6;
12     E:=E+1;
13     S1:
14     *IF* NUM - ENTIER(NUM*DELTA) < *-6 &
15     MN - ENTIER(MN*DELTA) < *-6 *THEN*
16     NDC:= *IF* E = 0 *THEN* -1 *ELSE* E *ELSE*
17     *BEGIN*
18     NUM:=NUM*10;
19     MN:=MN-ENTIER(NUM);
20     MN:=MN*10;
21     MN:=MN-ENTIER(MN);
22     DELTA:=DELTA*10;
23     *GO TO* S1;
24     *END*;
25     *END* NDC;
    
```


COMPILER VERSION : INCOME

OPTIONS IN EFFECT: SIZE(KBYTES) = 204, IDL = 6,
CPU(CSECS) = 250, CRD = 500, JCL = //, PGS = 10,
EBCDIC(FB), NOIDLST, LOAD(L), LONG(LP), OPTO, NOBAG,
SE, SOURCE(S), NST, SWAPO, TEST(T), M, NODUMP

```

0 *PROCEDURE* LINPLT(I,J,K,P,X,Y, SX,NX,SY, NY, XXS, YXS, AXY, JJ, KK);
1 *HEAL* SX,SY,XXS,YXS;
2 *INTEGEN* NX,NY,AXY,K, JJ, KK;
3 *HEAL* AHMAY, X,Y;
4 *INTEGEN* ARRAY I,J,P;
5 *BEGIN*
6 *PROCEDURE* ASHOOT; 'CODE';
7 *HEAL* SCX, SCY;
8 *INTEGEN* M,N, IKKJJ, S, II, JKK, STOP, IPEN;
9 SCX:=SX*10**NX;
10 SCY:=SY*10**NY;
11 JKK:=J/KK/;
12 *FOR* JJ:=1 *STEP* 1 *UNTIL* JKK *DO*
13 *M:=ENTLPH(P/JJ//10);
14 IKKJJ:=I/KK, JJ/;
15 *IF* M = 0 *THEN*
16 *MARK*((X/KK, JJ, 1/)-XXS)/SCX, (Y/KK, JJ, 1/)-YYS)/SCY,
17 *2, JJ, (AXY-1)*90, -1);
18 *IF* IKKJJ > M *THEN*
19 *FOR* II:=M+1 *STEP* M *UNTIL* IKKJJ *DO*
20 *MARK*((X/KK, JJ, II)-XXS)/SCX, (Y/KK, JJ, II/)-
21 *YYS)/SCY, *2, JJ, (AXY-1)*90, -1);
22 *END*;
23 *IF* N = 0 *THEN* 'GO TO' UIT;
24 PLOT((X/KK, JJ, 1/)-XXS)/SCX, (Y/KK, JJ, 1/)-YYS)/SCY, 3);
25 *IF* IKKJJ > 1 & N = 1 *THEN*
26 *FOR* S:=2 *STEP* 1 *UNTIL* IKKJJ *DO*
27 PLOT((X/KK, JJ, S/)-XXS)/SCX, (Y/KK, JJ, S/)-YYS)/SCY, 2);
28 *IF* N = 2 | N = 7 | N = 3 | N = 8 *THEN*
29 *BEGIN* II:=1;
30 *IF* N = 0 *THEN* II:=IKKJJ;
31 ASMOOT((X/KK, JJ, II)-XXS)/SCX,
32 (Y/KK, JJ, II/)-YYS)/SCY,
33 *IF* N = 3 *THEN* -1 *ELSE* 0);
34 IKKJJ:=IKKJJ-1;
35 *IF* N = 2 | N = 3 | IKKJJ < 5 *THEN*
36 *FOR* II:=2 *STEP* 1 *UNTIL* IKKJJ *DO*
37 ASMOOT((X/KK, JJ, II/)-XXS)/SCX,
38 (Y/KK, JJ, II/)-YYS)/SCY, -2);
39 II:=IKKJJ+1;
40 ASMOOT((X/KK, JJ, II/)-XXS)/SCX,
41 (Y/KK, JJ, II/)-YYS)/SCY, -24);
42 *END*;
43 IPEN:=-2;
44 *IF* N = 7 & IKKJJ > 4 *THEN*
45 *FOR* II:=2 *STEP* 1 *UNTIL* IKKJJ *DO*
46 *BEGIN* ASMOOT((X/KK, JJ, II/)-XXS)/SCX,
47 (Y/KK, JJ, II/)-YYS)/SCY, IPEN);
48
49
50
51

```

```

51      IPEN:=-IPEN-5;
52      *END*;
53      II:=IKKJJ+1;
54      ASMOOT((X/(KK,JJ,II))-XXS)/SCX,
55      (Y/(KK,JJ,II))-YYS)/SCY,IPEN);
56      ASMOOT((X/(KK,JJ,II))-XXS)/SCX,
57      (Y/(KK,JJ,II))-YYS)/SCY,-24);
58      *END*;
59      IPEN:=-3;
60      IKKJJ:=IKKJJ+1;
61      *IF* N = 6 IKKJJ > 5 *THEN*
62      *FOR* II:=1 *STEP* 1 *UNTIL* IKKJJ *DO*
63      *BEGIN* ASMOOT((X/(KK,JJ,II))-XXS)/SCX,
64      (Y/(KK,JJ,II))-YYS)/SCY,IPEN);
65      *END*;
66      ASMOOT((X/(KK,JJ,1))-XXS)/SCX,
67      (Y/(KK,JJ,1))-YYS)/SCY,-3);
68      ASMOOT((X/(KK,JJ,2))-XXS)/SCX,
69      (Y/(KK,JJ,2))-YYS)/SCY,-24);
70      *END*;
71      *IF* N = 6 *THEN*
72      *BEGIN* IPEN:=2;
73      *IF* IKKJJ > 1 *THEN*
74      *FOR* S:=2 *STEP* 1 *UNTIL* IKKJJ *DO*
75      *BEGIN* PLOT((X/(KK,JJ,S))-XXS)/SCX,
76      (Y/(KK,JJ,S))-YYS)/SCY,IPEN);
77      *END*;
78      *END*;
79      *END*;
80      *END*;
81      *END* PROCEDURE LINPLT;
82      *END* PROCEDURE LINPLT;

```

```

0000520
0000530
0000540
0000550
0000560
0000570
0000580
0000590
0000600
0000610
0000620
0000630
0000640
0000650
0000660
0000670
0000680
0000690
0000700
0000710
0000720
0000730
0000740
0000750
0000760
0000770
0000780
0000790
0000800
0000810
0000820
0000830
0000840
0000850

```



```

14  *ABTMIN*THOUELADETHUC(DEL,SCL,N,0,0);*CAL*IFI,SCL,MIN;
15  *ABTMIN*THOUELADETHUC(DEL,SCL,N,0,0);*CAL*IFI,SCL,MIN;
16  *CAL*IFI,SCL,MIN;
17  *CAL*IFI,SCL,MIN;
18  *CAL*IFI,SCL,MIN;
19  *CAL*IFI,SCL,MIN;
20  *CAL*IFI,SCL,MIN;
21  *CAL*IFI,SCL,MIN;
22  *CAL*IFI,SCL,MIN;
23  *CAL*IFI,SCL,MIN;
24  *CAL*IFI,SCL,MIN;
25  *CAL*IFI,SCL,MIN;
26  *CAL*IFI,SCL,MIN;
27  *CAL*IFI,SCL,MIN;
28  *CAL*IFI,SCL,MIN;
29  *CAL*IFI,SCL,MIN;
30  *CAL*IFI,SCL,MIN;
31  *CAL*IFI,SCL,MIN;
32  *CAL*IFI,SCL,MIN;
33  *CAL*IFI,SCL,MIN;
34  *CAL*IFI,SCL,MIN;
35  *CAL*IFI,SCL,MIN;
36  *CAL*IFI,SCL,MIN;
37  *CAL*IFI,SCL,MIN;
38  *CAL*IFI,SCL,MIN;
39  *CAL*IFI,SCL,MIN;
40  *CAL*IFI,SCL,MIN;
41  *CAL*IFI,SCL,MIN;
42  *CAL*IFI,SCL,MIN;
43  *CAL*IFI,SCL,MIN;
44  *CAL*IFI,SCL,MIN;
45  *CAL*IFI,SCL,MIN;
46  *CAL*IFI,SCL,MIN;
47  *CAL*IFI,SCL,MIN;
48  *CAL*IFI,SCL,MIN;
49  *CAL*IFI,SCL,MIN;
50  *CAL*IFI,SCL,MIN;
51  *CAL*IFI,SCL,MIN;
52  *CAL*IFI,SCL,MIN;
53  *CAL*IFI,SCL,MIN;
54  *CAL*IFI,SCL,MIN;
55  *CAL*IFI,SCL,MIN;
56  *CAL*IFI,SCL,MIN;
57  *CAL*IFI,SCL,MIN;
58  *CAL*IFI,SCL,MIN;
59  *CAL*IFI,SCL,MIN;
60  *CAL*IFI,SCL,MIN;
61  *CAL*IFI,SCL,MIN;
62  *CAL*IFI,SCL,MIN;
63  *CAL*IFI,SCL,MIN;
64  *CAL*IFI,SCL,MIN;
65  *CAL*IFI,SCL,MIN;
66  *CAL*IFI,SCL,MIN;
67  *CAL*IFI,SCL,MIN;
68  *CAL*IFI,SCL,MIN;
69  *CAL*IFI,SCL,MIN;
70  *CAL*IFI,SCL,MIN;
71  *CAL*IFI,SCL,MIN;
72  *CAL*IFI,SCL,MIN;
73  *CAL*IFI,SCL,MIN;
74  *CAL*IFI,SCL,MIN;
75  *CAL*IFI,SCL,MIN;
76  *CAL*IFI,SCL,MIN;
77  *CAL*IFI,SCL,MIN;
78  *CAL*IFI,SCL,MIN;
79  *CAL*IFI,SCL,MIN;
80  *CAL*IFI,SCL,MIN;
81  *CAL*IFI,SCL,MIN;
82  *CAL*IFI,SCL,MIN;
83  *CAL*IFI,SCL,MIN;
84  *CAL*IFI,SCL,MIN;
85  *CAL*IFI,SCL,MIN;
86  *CAL*IFI,SCL,MIN;
87  *CAL*IFI,SCL,MIN;
88  *CAL*IFI,SCL,MIN;
89  *CAL*IFI,SCL,MIN;
90  *CAL*IFI,SCL,MIN;
91  *CAL*IFI,SCL,MIN;
92  *CAL*IFI,SCL,MIN;
93  *CAL*IFI,SCL,MIN;
94  *CAL*IFI,SCL,MIN;
95  *CAL*IFI,SCL,MIN;
96  *CAL*IFI,SCL,MIN;
97  *CAL*IFI,SCL,MIN;
98  *CAL*IFI,SCL,MIN;
99  *CAL*IFI,SCL,MIN;
100 *CAL*IFI,SCL,MIN;
101 *CAL*IFI,SCL,MIN;
102 *CAL*IFI,SCL,MIN;
103 *CAL*IFI,SCL,MIN;
104 *CAL*IFI,SCL,MIN;
105 *CAL*IFI,SCL,MIN;
106 *CAL*IFI,SCL,MIN;
107 *CAL*IFI,SCL,MIN;
108 *CAL*IFI,SCL,MIN;
109 *CAL*IFI,SCL,MIN;
110 *CAL*IFI,SCL,MIN;
111 *CAL*IFI,SCL,MIN;
112 *CAL*IFI,SCL,MIN;
113 *CAL*IFI,SCL,MIN;
114 *CAL*IFI,SCL,MIN;
115 *CAL*IFI,SCL,MIN;
116 *CAL*IFI,SCL,MIN;
117 *CAL*IFI,SCL,MIN;
118 *CAL*IFI,SCL,MIN;
119 *CAL*IFI,SCL,MIN;
120 *CAL*IFI,SCL,MIN;
121 *CAL*IFI,SCL,MIN;
122 *CAL*IFI,SCL,MIN;
123 *CAL*IFI,SCL,MIN;
124 *CAL*IFI,SCL,MIN;
125 *CAL*IFI,SCL,MIN;
126 *CAL*IFI,SCL,MIN;
127 *CAL*IFI,SCL,MIN;
128 *CAL*IFI,SCL,MIN;
129 *CAL*IFI,SCL,MIN;
130 *CAL*IFI,SCL,MIN;
131 *CAL*IFI,SCL,MIN;
132 *CAL*IFI,SCL,MIN;
133 *CAL*IFI,SCL,MIN;
134 *CAL*IFI,SCL,MIN;
135 *CAL*IFI,SCL,MIN;
136 *CAL*IFI,SCL,MIN;
137 *CAL*IFI,SCL,MIN;
138 *CAL*IFI,SCL,MIN;
139 *CAL*IFI,SCL,MIN;
140 *CAL*IFI,SCL,MIN;
141 *CAL*IFI,SCL,MIN;
142 *CAL*IFI,SCL,MIN;
143 *CAL*IFI,SCL,MIN;
144 *CAL*IFI,SCL,MIN;
145 *CAL*IFI,SCL,MIN;
146 *CAL*IFI,SCL,MIN;
147 *CAL*IFI,SCL,MIN;
148 *CAL*IFI,SCL,MIN;
149 *CAL*IFI,SCL,MIN;
150 *CAL*IFI,SCL,MIN;
151 *CAL*IFI,SCL,MIN;

```


LAISSEZ-VOUS DE JERUSALEM, LE 17/02/1977

53.10.47 PAGE 4

```

232 43: 00002360
233 44: 00002370
234 45: 00002380
235 46: 00002390
236 47: 00002400
237 48: 00002410
238 49: 00002420
239 50: 00002430
240 51: 00002440

```

COUSINING(1,('FCOT' IN INVOLU VAN DE SEKSI HRFSTILL DEZE FCOT

'IF'~OJHAI~'IHEH'LASFLU;

00LEAHIB(1,(' FINDE CMAILI '));

LIN(1,1);

'END' FINDE PROCEEDURE CMAILI;