

9295

RA

DUPLICAAT

stichting  
mathematisch  
centrum



LR 2.1

RA

APRIL 1971

P. BEERTEMA en G.J.R. FÖRCH  
HANDLEIDING BIJ HET MC-ENQUETEPROGRAMMA

**2e boerhaavestraat 49 amsterdam**

BIBLIOTHEEK MATHEMATISCH CENTRUM  
AMSTERDAM

### 0.1. Voorwoord

De grote hoeveelheid enquete materiaal, die de laatste jaren aan het Mathematisch Centrum ter verwerking wordt aangeboden, maakte het wenselijk over een algemeen enqueteverwerkend programma te beschikken, zodat niet voor elke enquete een nieuw programma behoeft te worden geschreven. Dit programma zou onder meer de gebruiker de mogelijkheid moeten bieden tot het op eenvoudige wijze aanbrengen van hercoderingen in het basismateriaal, terwijl daarnaast de mogelijkheid zou moeten bestaan bepaalde statistische toetsen te doen uitvoeren.

Op initiatief van Van der Houwen werd in 1968 aan dit project begonnen. Aan de verwezenlijking van een voorlopige versie van het programma werkten P. Beertema, G.J.R. Förch, P.J. van der Houwen en D.P. de Jong. Een beschrijving van deze eerste versie treft men aan in MR 106. Het hierin beschreven programma was niet in staat om statistische toetsen uit te voeren. Bovendien bleek ook aan andere extra faciliteiten behoefte te bestaan.

Uit dit eerste programma ontwikkelde P. Beertema een enqueteprogramma, dat tegemoet komt aan de eisen, die de praktijk aan zo'n programma stelt. Bij de inpassing van de statistische toetsen geschiedde het één en ander in nauwe samenwerking met W.N. van Nooten.

Het programma is geschreven in ALGOL-60 en wel zodanig, dat het voor iedereen, die enige kennis van deze taal bezit, mogelijk is een uitgebreide enquete te laten verwerken.

Bij deze verwerking bestaan de volgende mogelijkheden:

1. Het aanbrengen van hercoderingen.
2. Het vormen van één- en tweedimensionale frequentietabellen (al dan niet onder nevenvoorwaarden).
3. Het doen uitvoeren van de volgende statistische toetsen:

Chi-kwadraten

Kendall

Wilcoxon

Kruskal-Wallis.

Daarnaast laat het programma de mogelijkheid open tot het gebruik van

speciale faciliteiten, met name het gebruik van de trommel en van magneetbanden.

Het standaardgedeelte van het programma is vastgelegd in de procedure process inquiry. Deze procedure zit in de procedurebibliotheek van het MCALL-systeem en kan in dat systeem dus zonder verdere declaratie worden aangeroepen. In het MILLI-systeem kan het enqueteprogramma niet worden gebruikt. Het is echter wel mogelijk een gebruikersprogramma in het MILLI-systeem te laten testen (zie 16).

B. en F.

0.2. Lijst van pagina's

Om na te gaan of een exemplaar al dan niet volledig is, kan gebruik worden gemaakt van onderstaande tabel.

Hoofdstuk 0	bestaat uit	5 blz.	
Hoofdstuk 1	bestaat uit	1 blz.	
Hoofdstuk 2	bestaat uit	2 blz.	
Hoofdstuk 3	bestaat uit	1 blz.	blz. 3-1 gecorrigeerd aug. 1972.
Hoofdstuk 4	bestaat uit	4 blz.	
Hoofdstuk 5	bestaat uit	2 blz.	
Hoofdstuk 6	bestaat uit	1 blz.	
Hoofdstuk 7	bestaat uit	8 blz.	blz.7-2 en 7-8 gecorrigeerd sept.1971.
Hoofdstuk 8	bestaat uit	6 blz.	
Hoofdstuk 9	bestaat uit	1 blz.	
Hoofdstuk 10	bestaat uit	2 blz.	
Hoofdstuk 11	bestaat uit	2 blz.	
Hoofdstuk 12	bestaat uit	2 blz.	
Hoofdstuk 13	bestaat uit	1 blz.	
Hoofdstuk 14	bestaat uit	2 blz.	
Hoofdstuk 15	bestaat uit	2 blz.	
Hoofdstuk 16	is vervallen.		
Hoofdstuk 17	bestaat uit	1 blz.	
Hoofdstuk 18	bestaat uit	3 blz.	
Hoofdstuk 19	bestaat uit	5 blz.	
			+ 10 blz. machineuitvoer.

Deze blz. zal bij elke aanvulling worden vervangen.

### 0.3. Inhoudsopgave

0. Algemeen
  - 0.1. Voorwoord
  - 0.2. Lijst van pagina's
  - 0.3. Inhoudsopgave
1. De verwerking van een enquête
2. Structuur van het gebruikersprogramma
3. general data
4. codelist
  - 4.1. Algemeen
  - 4.2. Het combineren van variabelen met eenzelfde waardebereik
5. ordinal en nominal
6. irrelevant
7. Typologieënprogramma
  - 7.1. Algemeen
  - 7.2. Controlemogelijkheden door het typologieënprogramma
  - 7.3. Hercoderingen
8. tablelist
  - 8.1. Algemeen
  - 8.2. Het samentrekken van tabelspecificaties
  - 8.3. Nevenvoorwaarden
9. Enkele opmerkingen met betrekking tot de procedures: general data, codelist, ordinal, nominal, irrelevant en tablelist
10. Het sommeren van frequentietabellen
11. De output van het programma
12. Samenwerking tussen hoofdprogramma en gebruikersprogramma
  - 12.1. Algemeen
  - 12.2. De decodeerfase
  - 12.3. De executiefase
13. Trommel en magneetband gebruik

14. Het getallenmateriaal
15. Foutmeldingen
  - 15.1. Algemeen
  - 15.2. Een speciale foutmelding
16. Testprocedure in het MILLI-systeem
17. Enkele algemene gegevens
18. Statistische toetsen
19. Een uitgewerkt voorbeeld.

## 1. De verwerking van een enquête

Voor het verwerken van een enquête dient de gebruiker over de volgende onderdelen te beschikken.

a. Een monitorkopje.

Hiervoor zij verwezen naar LR 1.1.

b. Een gebruikersprogramma.

Dit is een, door de gebruiker te schrijven, ALGOL-60 programma.

Het hoofddoel van deze handleiding is een beschrijving van de voorschriften waaraan dit gebruikersprogramma moet voldoen en de mogelijkheden, die via het gebruikersprogramma te realiseren zijn.

c. De door het programma te verwerken gegevens. (Het getalmateriaal.)

Genoemde onderdelen dienen op ponskaarten te staan.

N.B.

De verwerking van een enquête kan alleen geschieden in het MCALL systeem.

In het MILLI systeem kan wel het onder b genoemde gebruikersprogramma getest worden (zie 16).

## 2. Structuur van het gebruikersprogramma

Het gebruikersprogramma bestaat uit een vast gedeelte en een variabel gebruikersgedeelte. Het moet de volgende gedaante hebben:

```
'begin' 'procedure' users program (general data, codelist, ordinal, nominal,
  irrelevant, tablelist, summon, finish, report, resp, respondent);
'begin'
```

gebruikersgedeelte

```
'end' users program;
process inquiry (users program)
'end'
```

Het gebruikersgedeelte bestaat uit:

- a. eventuele declaraties van bij hercodering benodigde identificers;
- b. een aanroep van de procedure general data;
- c. één of meer eenheden, bestaande uit
  1. een aanroep van de procedure codelist;
  2. eventuele aanroepen van de procedures ordinal, nominal en irrelevant;
  3. eventueel een typologieënprogramma;
  4. één of meer aanroepen van de procedure tablelist met eventuele sommeringsprogramma's.

Men kan in één gebruikersprogramma desgewenst de procedure process inquiry meer dan één keer aanroepen. Dit kan van belang zijn als men verschillende delen van het getallen materiaal op verschillende manieren wil verwerken. Het gebruikersprogramma zou er dan b.v. als volgt gaan uitzien:

```
'begin' 'procedure' users program 1 (general data, codelist, ordinal,
  nominal, irrelevant, tablelist, summon, finish, report, resp,
  respondent);
'begin'
```

gebruikersgedeelte

```
'end' usersprogram 1;
```



```
'procedure' users program 2 (general data, codelist, ordinal, nominal,  
irrelevant, tablelist, summon, finish, report, resp, respondent);  
'begin'
```

```
gebruikersgedeelte
```

```
'end' usersprogram 2;  
processinquiry (usersprogram 1);  
processinquiry (usersprogram 2)  
'end'
```

### 3. general data

Een aanroep van de procedure general data heeft tot doel het verstrekken van basisgegevens, die voor het enqueteprogramma van belang zijn. general data heeft de volgende vorm:

```
general data ("x resp, x card, x col 1, maxtime");
```

Hierin moeten x resp, x card, x col 1 en maxtime gehele getallen zijn, die de volgende betekenis hebben:

x resp: het aantal te verwerken respondenten. Men kan desgewenst het programma zelf het aantal respondenten laten vaststellen. Daartoe geeft men voor x resp de waarde 0 mee.

x card: het aantal kaarten per respondent.

x col 1: het aantal, voor hercoderingen te gebruiken, variabelen, waarvoor een nummer gekozen wordt, dat groter is dan x card \* 80.

maxtime: de maximale tijd (in milli-uren), die het programma mag besteden aan het opbouwen van frequentietabellen. Bij overschrijding van deze tijd gaat het programma over op de uitvoer van de tabellen. Aangezien het programma het getallenmateriaal respondent voor respondent afwerkt, zullen deze frequentietabellen slechts gebaseerd zijn op de respondenten, die op het moment van afbreken verwerkt zijn.

#### Opmerking 1

$x \text{ card} * 80 + x \text{ col } 1$  moet kleiner zijn dan 1000.

#### Opmerking 2

Het dubbelaccent is één symbool.

#### 4. Codelist

##### 4.1. Algemeen

Een aanroep van de procedure codelist heeft tot doel het programma nadere gegevens te verstrekken omtrent de te gebruiken variabelen. Deze nadere gegevens zijn: de plaats op de ponskaarten, het waardebereik en eventueel in de output te vermelden namen. codelist heeft de volgende vorm:

```
codelist ("k1: waardebereik; k2 [naam]: waardebereik; .....");
```

Hierin zijn k1 en k2 karakterisering van variabelen. Hierbij doen zich twee mogelijkheden voor:

- a. Indien een variabele slechts één kolom op een ponskaart beslaat, dan wordt de variabele gekarakteriseerd door het nummer van die kolom. Hierbij zij opgemerkt, dat kolommen van volgkaarten doorgenummerd worden. Met 87 wordt de variabele aangeduid, die in kolom 87 staat, dus in kolom 7 van de eerste volgkaart.
- b. Indien een variabele meer dan één kolom in beslag neemt (natuurlijk wel opeenvolgende kolommen), dan wordt hij gekarakteriseerd door het nummer van de kolom, waarin hij begint en het aantal kolommen, dat hij beslaat. Deze twee grootheden worden door een punt gescheiden. Met 18.3 wordt dus de variabele aangeduid, die begint in kolom 18 en die 3 kolommen beslaat.

Achter de karakterisering van een variabele mag tussen teksthaken een naam worden meegegeven. Van zo'n naam zijn slechts de eerste 15 symbolen significant. Teksthaken, onderstrepingen, doorbalkingen, accenten en dubbele accenten zijn binnen een naam niet toegestaan.

Na de karakterisering van een variabele (al dan niet met naam) volgt een dubbele punt met daarachter het bij de variabele behorende waardebereik. Dit waardebereik kan op de volgende manieren worden gegeven:

1. De mogelijke waarden worden afzonderlijk genoemd, gescheiden door komma's. Bij een bepaalde waarde (bepaalde waarden) mag tussen tekst-

haken een naam worden meegegeven. Hiervan zijn slechts de eerste 6 symbolen significant.

2. Indien het waardebereik een aaneengesloten rij getallen is, kan de notatie laagste waarde / hoogste waarde gebruikt worden.
3. Indien het waardebereik afzonderlijke getallen en aaneengesloten rijen getallen bevat, kunnen beide notaties gecombineerd worden.

Het waardebereik wordt afgesloten door een puntkomma.

Voorbeelden van waardebereiknotaties zijn:

1. 1, 2, 4[naam], 10;
2. 3/9;
3. 1/3, 5/9;
4. 1/3, 5, 8/10, 11[tekst];

#### 4.2. Het combineren van variabelen met eenzelfde waardebereik

Wanneer een aantal variabelen eenzelfde waardebereik heeft, dan is het mogelijk door combinatie van deze variabelen in codelist een verkorte schrijfwijze te bereiken. Dit geschiedt dan door de karakterisering van de betreffende variabelen samen te trekken en daarachter de beschrijving van het (gezamenlijke) waardebereik te geven. De karakterisering van de variabelen en de beschrijving van het waardebereik worden natuurlijk weer door een dubbele punt gescheiden.

Het samentrekken van de karakterisering van zulke variabelen kan op één van de volgende manieren gebeuren:

- a. Indien de betreffende variabelen opeenvolgend zijn en bovendien elk één kolom beslaan, wordt de notatie:

(eerste variabele / laatste variabele)

gebruikt.

- b. Indien de betreffende variabelen opeenvolgend zijn en bovendien eenzelfde aantal kolommen beslaan, wordt de notatie:

(eerste variabele / laatste variabele) . aantal kolommen

gebruikt.

- c. Indien de betreffende variabelen niet opeenvolgend zijn, worden de

karacteriseringen van de variabelen afzonderlijk gebruikt, onderling gescheiden door een komma.

- d. Indien de betreffende variabelen gedeeltelijk opeenvolgend zijn kunnen de afzonderlijke notaties en de groepsnotaties door middel van komma's aaneengeschakeld worden.

Bij groepsnotaties mag aan de verschillende variabelen binnen de groep een gemeenschappelijke naam worden meegegeven.

### Voorbeelden

1. Met de notatie

$(17/21): 0/9;$

wordt aangegeven dat de variabelen 17, 18, 19, 20 en 21 alle een waardebereik van 0 t/m 9 hebben.

2. Met de notatie

$(5/8) [\text{tekst 1}] : 1, 2, 4, 5;$

wordt aangegeven dat de variabelen 5, 6, 7 en 8 alle de naam tekst 1 hebben en dat hun waardebereik is 1, 2, 4 en 5.

3. Met de notatie

$(1/7).3 : -75/+75;$

wordt aangegeven dat de variabelen 1.3, 4.3 en 7.3 een waardebereik  $-75/+75$  hebben.

4. Met de notatie

$8.2, 12.2, 27.3 : 1/5, 7/11, 13, 15;$

wordt aangegeven dat de variabelen 8.2, 12.2 en 27.3 als waardebereik hebben 1/5, 7/11, 13 en 15.

5. Met de notatie

$1, 3[\text{tekst 1}], (4/8).2[\text{tekst 2}], (17/22): 1, 3[+], 5[-], 7/9, 0;$

wordt aangegeven, dat de variabelen 1, 3 (met tekst 1 als naam), 4.2, 6.2 en 8.2 (met tekst 2 als gemeenschappelijke naam) en 17/22 alle als waardebereik hebben 1, 3 (met + als naam), 5 (met - als naam), 7 t/m 9 en 0.

Opmerking 1

Indien een variabele gehercodeerd (zie 7.3.) wordt, moet in de codelist, die aan de hercodering voorafgaat, het nieuwe waardebereik worden vermeld. Als hierbij een variabele met een meer-cijfercode overgaat in een één-cijfercode, moet voor de karakterisering van de variabele toch het oorspronkelijke voorschrift worden aangehouden. De karakterisering geeft namelijk aan, welke kolommen op de ponskaarten voor één variabele moeten worden samengenomen.

Opmerking 2

Indien voor hercodering een extra variabele wordt toegevoegd, dan hoeft bij de karakterisering van deze variabele niet het aantal kolommen te worden aangegeven, aangezien zo'n nieuwe variabele niet correspondeert met kolommen op de ponskaarten.

## 5. ordinal en nominal

Een aanroep van de procedure ordinal heeft tot doel aan het programma kenbaar te maken, dat een aantal waarden binnen het waardebereik van de in de aanroep genoemde variabelen een ordening hebben, die als zinvol wordt beschouwd. De opgegeven waarden vormen een "ordinale schaal".

Een aanroep van de procedure nominal heeft tot doel aan het programma kenbaar te maken, dat een aantal waarden binnen het waardebereik van de in de aanroep genoemde variabelen in zekere zin willekeurig gekozen is. Dat wil zeggen, dat de rekenkundige ordening van de codes niet samenhangt met een ordening in de verschijnselen, die de codes representeren.

ordinal en nominal hebben de volgende vorm:

```
ordinal ("k; k.n; ...");
```

```
nominal ("k; k.n; ...");
```

Hierin is k het nummer van de kolom, waarin de betreffende variabele begint. Indien alleen k wordt opgegeven, wordt het hele waardebereik van deze variabele als ordinaal (nominaal) beschouwd. Indien men achter het kolomnummer k een getal n toevoegt (door een punt van k gescheiden), dan worden slechts de eerste n waarden van het waardebereik als ordinaal (nominaal) beschouwd.

Ook bij ordinal en nominal is een samentrekking van opeenvolgende variabelen mogelijk en wel in de volgende vorm.

```
(eerste variabele / laatste variabele)
```

als de aanduiding op het gehele waardebereik van de variabelen slaat.

```
(eerste variabele / laatste variabele) . n
```

als de aanduiding alleen op de eerste n waarden slaat.

### Voorbeeld

In codelist staat:

```
16[leeftijd]: 18/30, 0[geen antwoord];
```

Dan zou een aanroep van ordinal kunnen zijn:

ordinal ("16.13");

Dit zou dan betekenen, dat de eerste 13 waarden van de variabele leeftijd als ordinaal worden opgevat. Dat zijn in dit geval dus de codes 18/30. code 0 wordt uitgesloten, want dat is de veertiende code.

Funktie van ordinal en nominal.

Door het gebruik van de procedures ordinal en nominal kan men aangeven welke statistische toetsen uitgevoerd moeten worden (zie 18). Is men niet in statistische toetsen geïnteresseerd, dan laat men het aanroepen van deze procedures achterwege.



## 6. irrelevant

Een aanroep van de procedure irrelevant heeft tot doel het programma mede te delen, dat de in de aanroep genoemde antwoordmogelijkheden van de genoemde variabelen als irrelevant beschouwd dienen te worden.

irrelevant heeft de volgende vorm:

```
irrelevant ("k:c; ...");
```

Hierin is k het nummer van de kolom waarin de variabele begint en c de code die als irrelevant beschouwd dient te worden.

Voor opeenvolgende variabelen met eenzelfde irrelevante code mag de notatie:

```
(eerste variabele / laatste variabele): c;
```

gebruikt worden.

De irrelevante antwoorden komen bij de tabel in een aparte categorie terecht. Deze speelt bij eventuele verdere bewerkingen geen rol.

### Opmerking 1

Alle variabelen die in een procedure ergens in het gebruikersprogramma worden genoemd, dienen in een voorafgaande aanroep van codelist gespecificeerd te zijn. Bij irrelevant moet echter de irrelevant verklaarde code niet in het gespecificeerde waardebereik van de betreffende variabele voorkomen.

Per variabele kan slechts één code irrelevant verklaard worden.

### Opmerking 2

Indien een variabele in een codelist opnieuw wordt gespecificeerd, dan wordt daarmee het effect van eventuele voorafgaande aanroepen van ordinal of nominal voor die variabele ongedaan gemaakt. Dit geldt niet voor irrelevant.

## 7. Typologieënprogramma

### 7.1. Algemeen

Het typologieënprogramma in de eerste eenheid kan twee functies vervullen, een controlerende functie en een hercoderende functie. De typologieënprogramma's in de latere eenheden kunnen alleen nog een hercoderende functie hebben.

De volgende elementen zijn voor de typologieënprogramma's van belang.

- a. De procedure report (text, skip), voor het geven van foutmeldingen. Hierin is text een string, die in de output afgedrukt wordt en skip is een boolean, die bepaalt of de betreffende respondent al dan niet geskipt wordt. Als voor skip de waarde 'true' wordt meegegeven, wordt de respondent waarin de fout gesignaleerd is wel geskipt en als voor skip de waarde 'false' wordt meegegeven, wordt hij niet geskipt. Het al dan niet skippen van de betreffende respondent wordt in de output vermeld.
- b. De procedure finish. Deze procedure draagt het programma op te stoppen met zijn opbouwwerkzaamheden en over te gaan tot het geven van de output. Het aantal op dat moment verwerkte respondenten wordt uitgeprint.
- c. Het integer array resp. De lengte van dit array is  $x \text{ card} * 80 + x \text{ col} + 1$ .

Verder worden in resp de gegevens opgeslagen van de respondent, waarmee het programma bezig is. Als in kolom k van de ponskaart(en) een nieuwe variabele begint, dan is de waarde van deze variabele voor de betreffende respondent opgeslagen in resp [k]. Hierbij dient nog wel onderscheid te worden gemaakt tussen variabelen, die slechts 1 kolom beslaan en variabelen, die meer dan 1 kolom in beslag nemen. Voor een 1 kolomsvariabele komt in resp de resymwaarde (zie LR 1.1 tabel 6.4) van de antwoordcode (voor zover deze binnen het gespecificeerde waardebereik ligt, zie 5). Voor een meerkolomsvariabele komt in resp de getalwaarde van de antwoordcode.

Indien kolom k op de ponskaart niet gebruikt wordt door een variabele, die in een codelist gespecificeerd is, dan krijgt resp[k] de resymwaarde van het symbool, dat in deze kolom op de ponskaart staat.

Indien men door hercodering een nieuwe variabele creëert, dan komt de waarde van deze variabele (voor de respondent waarmee het programma bezig is) in resp te staan op de plaats, die correspondeert met de karakterisering van de betreffende variabele. Dus als een nieuwe variabele gevormd wordt, die als karakterisering het getal k heeft en, die voor de respondent onder behandeling de waarde a heeft (a zal via hercodering voortkomen uit waarden op de ponskaarten), dan geldt dus bij deze respondent:  $\text{resp}[k] = a$ .

- d. De integer respondent. respondent geeft aan met de hoeveelste respondent het programma bezig is.

## 7.2. Controle mogelijkheden door het typologieënprogramma

In het typologieënprogramma van de eerste eenheid kan men laten nagaan, of een respondent correct is ingelezen. Mocht er iets mis blijken te zijn, dan kan men maatregelen laten nemen, om de invoer toch in de gewenste vorm te krijgen.

### Voorbeeld

In kolom 1 van elke respondent dient een 9 te staan. Is dat niet het geval, dan mag de betreffende respondent niet gebruikt worden. In kolom 2 van elke respondent dient ook een 9 te staan. Staat er in plaats van deze 9 een spatie (resymwaarde 93), dan mag de betreffende respondent wel verder gebruikt worden, maar de afwijking moet gerapporteerd worden. Staat er in kolom 2 noch een 9, noch een spatie, dan mag de betreffende respondent niet gebruikt worden.

Deze controle kan men laten uitvoeren, door in het typologieënprogramma van de eerste eenheid het volgende stukje programma op te nemen:

```
'if' resp[1] ≠ 9 'then' report ("error column 1", 'true');  
'if' resp[2] = 93 'then' report ("harmless error", 'false');  
'else' 'if' resp[2] ≠ 9 'then' report ("error column 2", 'true');
```

Dit kan op de output tot de volgende meldingen leiden:

Indien men door hercodering een nieuwe variabele creëert, dan komt de waarde van deze variabele (voor de respondent waarmee het programma bezig is) in resp te staan op de plaats, die correspondeert met de karakterisering van de betreffende variabele. Dus als een nieuwe variabele gevormd wordt, die als karakterisering het getal k heeft en, die voor de respondent onder behandeling de waarde a heeft (a zal via hercodering voortkomen uit waarden op de ponskaarten), dan geldt dus bij deze respondent:  $\text{resp}[k] = a$ .

- d. De integer respondent. respondent geeft aan met de hoeveelste respondent het programma bezig is.

## 7.2. Controle mogelijkheden door het typologieënprogramma

In het typologieënprogramma van de eerste eenheid kan men laten nagaan, of een respondent aan bepaalde eisen voldoet. Mocht er iets mis blijken te zijn, dan kan men maatregelen laten nemen, om de invoer toch in de gewenste vorm te krijgen.

### Voorbeeld

In kolom 1 van elke respondent dient een 9 te staan. Is dat niet het geval, dan mag de betreffende respondent niet gebruikt worden. In kolom 2 van elke respondent dient ook een 9 te staan. Staat er in plaats van deze 9 een spatie (resymwaarde 93), dan mag de betreffende respondent wel verder gebruikt worden, maar de afwijking moet gerapporteerd worden. Staat er in kolom 2 noch een 9, noch een spatie, dan mag de betreffende respondent niet gebruikt worden.

Deze controle kan men laten uitvoeren, door in het typologieënprogramma van de eerste eenheid het volgende stukje programma op te nemen:

```
'if' resp[1] ≠ 9 'then' report ("error column 1", 'true');  
'if' resp[2] = 93 'then' report ("harmless error", 'false');  
'else' 'if' resp[2] ≠ 9 'then' report ("error column 2", 'true');
```

Dit kan op de output tot de volgende meldingen leiden (we gaan er van uit, dat in de voorgaande codelist variabele 2 niet is gespecificeerd. Zou variabele 2 wel gespecificeerd zijn, dan heeft het stukje programma niet de gewenste uitwerking, omdat de spatie dan in  $\text{resp}[2]$  niet als 93 zou verschijnen, maar als  $-93 * 10^4$  (zie 14)).

errors in input:

```
resp 16 : error column 1  skipped
resp 121 : error column 2  skipped
resp 140 : harmless error  not skipped
resp 201 : error column 1  skipped.
```

Dit zou dan betekenen dat respondenten 16 en 201 geskipt zijn omdat ze geen 9 hadden in kolom 1. Respondent 121 is geskipt omdat hij nog een 9 noch een spatie had in kolom 2. Respondent 140 is gewoon verwerkt, maar hij had een spatie in kolom 2.

Het kan ook voorkomen, dat men een controle op een bepaalde fout wil inbouwen en dat men, als die fout optreedt, het verder opbouwen van de frequentietabellen wil beëindigen. Dit kan bij constatering van de betreffende fout, bereikt worden door de aanroep finish.

De procedures report en finish zijn natuurlijk alleen gestandaardiseerde hulpmiddelen bij de controle van het getallenmateriaal. In het typologieënprogramma van de eerste eenheid van het gebruikersprogramma kan men ook nog allerlei controles inbouwen, die zich niet direkt met report en finish laten beschrijven. In zulke gevallen zal de gebruiker zelf een stukje programma moeten schrijven, dan aan zijn controle eisen tegemoet komt.

### 7.3. Hercoderingen

Dikwijls zal men bij de verwerking van een enquête door middel van hercoderingen wijzigingen in het basismateriaal willen aanbrengen, omdat het gewijzigde materiaal zich er beter toe leent de gewenste informatie te verstrekken (uiteindelijk is het doel van de enquêteverwerking om uit de onoverzichtelijke hoeveelheid informatie, die het basismateriaal bevat, zo overzichtelijk mogelijk de voor een bepaald doel relevante, informatie te destilleren). Deze hercoderingen zullen dan neerkomen op het volgens bepaalde regels veranderen van het codepatroon van een bepaalde variabele (bepaalde variabelen), of op het samenstellen van nieuwe variabele(n) uit de op de ponskaarten beschikbare gegevens.

In het gebruikersprogramma brengt men deze hercoderingen aan door de

juiste bewerkingen uit te voeren op het array resp.

Immers, de antwoordcode, die de respondent, waarmee het programma bezig is, heeft voor de variabele, die in kolom k op de ponskaarten begint, is gelijk aan de waarde van resp[k]. Indien men nu een verandering wil aanbrengen in deze antwoordcode, kan dit bereikt worden door resp[k] te veranderen.

Wil men een nieuwe variabele k3 vormen, die de som is van de variabele, die op de ponskaarten in kolom k1 begint en de variabele, die in kolom k2 begint, dan kan dit bereikt worden door resp[k3] gelijk te maken aan resp[k1] + resp[k2].

Hoe het één en ander in zijn werk gaat, kunnen we wellicht het beste bekijken aan de hand van een voorbeeld.

#### Voorbeeld

Stel dat we een enquête hebben met 7 vragen, dus 7 variabelen. De eerste 3 variabelen beslaan ieder 2 kolommen, de vierde variabele beslaat 3 kolommen, de laatste 3 variabelen beslaan ieder 1 kolom. Als het programma met een respondent bezig is, die voor deze 7 variabelen de waarden 2, 15, 4, 107, 4, 8 en 2 als antwoordcodes heeft, dan ziet de inhoud van het array resp er als volgt uit:

```
resp[1] = 2, resp[3] = 15, resp[5] = 4,  
resp[7] = 107, resp[10] = 4, resp[11] = 8,  
resp[12] = 2.
```

De waarden van resp[2], resp[4], resp[6], resp[8] en resp[9] doen niet ter zake, want in kolom 2, 4, 6, 8 en 9 begint geen variabele,

Als men nu hercoderingen wil aanbrengen, kunnen deze van tweeërlei aard zijn:

1. Eén of meer informatiedragende elementen van resp worden volgens een bepaald voorschrift gewijzigd. Dit voorschrift mag afhankelijk zijn van andere elementen van resp.
2. Er worden nieuwe elementen aan resp toegevoegd, die uit de reeds bestaande elementen worden geconstrueerd. Aan welk(e) element(en) van resp we de te construeren waarde(n) toekennen, zal afhangen van de

karakterisering die we de betreffende variabele in codelist gegeven hebben. De elementen van resp, die ons ter beschikking staan zijn resp[2], resp[4], resp[6], resp[8], resp[9], resp[13] t/m resp[80], dus de elementen van resp, die nog niet in gebruik zijn.

Deze resp elementen staan ons in dit geval ter beschikking, als we in general data xcol 1 de waarde 0 hebben gegeven. Zouden we xcol 1 echter de waarde 3 hebben gegeven, dan kunnen ook nog resp[81], resp[82] en resp[83] voor nieuwe variabelen worden gebruikt.

De gewenste hercoderingen, dus de gewenste veranderingen in resp, worden tot stand gebracht door middel van een stukje ALGOL-60 programma. De identifiërs, die in zo'n stuk programma nodig zijn, moeten door de gebruiker zelf gedeclareerd worden vóór de aanroep van de procedure general data (zie 1).

Hoe de gewenste hercoderingen kunnen worden aangebracht, zullen we toelichten met een paar voorbeelden.

#### Voorbeeld 1

Variabele 31 heeft als waardebereik 0/9. Men wil een hercodering aanbrengen volgens het voorschrift ( $\rightarrow$  staat voor "moet worden")

```

0 → 0
1, 2 → 1
3, 4, 5 → 2
6, 7, 8, 9 → 3

```

Het programma, dat deze hercodering uitvoert, zou er als volgt uit kunnen zien:

```

i:= resp[31]; resp[31]:= 'if' i = 0 'then' 0 'else'
'if' i = 1 v i = 2 'then' 1 'else'
'if' i > 2 ^ i < 6 'then' 2 'else' 3;

```

#### Voorbeeld 2

De variabelen 10.2, 12.2, 14.2, 16.2 en 18.2 hebben alle als waardebereik 1/8, 11/18, 21/28. Op al deze variabelen moet de volgende hercodering wor-

den aangebracht:

1/3, 11/13, 21/23 → 1

4/6, 14/16, 24/26 → 2

7, 8, 17, 18, 27, 28 → 3.

Het programma, dat deze hercodering uitvoert zou er als volgt uit kunnen zien:

```
'for' i:= 10 'step' 2 'until' 18 'do'
'begin' i1:= resp[i]; i1:= i1 - (i1:10) * 10;
      resp[i]:= 'if' i1 < 4 'then' 1
                'else' 'if' i1 < 7 'then' 2 'else' 3
'end' ;
```

### Voorbeeld 3

De variabele 10.2 wil men vervangen door een variabele, die de som is van de variabelen 10.2 en 12.2.

Het programma, dat deze hercodering uitvoert, ziet er als volgt uit:

```
resp[10]:= resp[10] + resp[12];
```

### Voorbeeld 4

De laatste gebruikte variabele op een kaart is 50.2. We willen nu 3 variabelen toevoegen, die de nummers 52, 53 en 81 moeten krijgen (om een variabele met nummer 81 te kunnen introduceren, moet in general data xcol 1 tenminste de waarde 1 hebben gehad).

De waarden van de nieuwe variabelen moeten als volgt gevormd worden:

52 het gemiddelde van 40.2 en 42.2

53 het gemiddelde van 44.2 en 46.2

81 het gemiddelde van 48.2 en 50.2.

Het programma, dat deze hercodering uitvoert, zou er als volgt uit kunnen zien:



```
resp[52]:= (resp[40] + resp[42])/2;  
resp[53]:= (resp[44] + resp[46])/2;  
resp[81]:= (resp[48] + resp[50])/2;
```

Bij dit voorbeeld is een opmerking op zijn plaats. resp is een integer array en dus wordt een waarde, die aan een element van resp wordt toegekend, herleid tot een integer. Dit houdt in, dat een gebroken waarde op het dichtsbijzijnde gehele getal wordt afgerond. In twijfelgevallen wordt naar boven afgerond. Dus als een getal, dat in resp[i] zou moeten komen, de waarde  $1\frac{1}{2}$  heeft, dan wordt aan resp[i] de waarde 2 toegekend.

#### Voorbeeld 5

Variabele 27.2 heeft als waardebereik  $-2/2$ . Men wil dit als volgt wijzigen

```
-2 → 1  
-1 → 2  
0 → at random 2 of 3  
1 → 3  
2 → 4.
```

Het programma, dat deze hercodering uitvoert, zou er als volgt uit kunnen zien:

```
i:= resp[27]; resp[27]:= 'if' i < 0  
'then' i + 3 'else' 'if' i > 0 'then' i + 2  
'else' random + 2;
```

#### Opmerking 1

De hier gegeven voorbeelden hebben uiteraard niet tot doel, om een overzicht te geven van alle hercoderingen, die in de praktijk kunnen optreden. Ze dienen alleen om u een idee te geven, hoe u hercoderingen tot stand kunt brengen. Door middel van zo'n stukje ALGOL-60 programma kan redelijkerwijs elke gewenste hercodering uitgevoerd worden.

Opmerking 2

De in hercoderingen gebruikte variabelen moeten in een voorgaande aanroep van codelist gespecificeerd zijn. Hierbij moet de laatste specificatie overeenkomen met het waardebereik na hercodering.

Opmerking 3

Indien men in een typologieënprogramma een variabele gebruikt, die in een vorig typologieënprogramma is gehercodeerd, dan dient men er rekening mee te houden, dat de hercoderingen ongedaan zijn gemaakt voor de variabelen, die een nummer hebben dat kleiner is dan, of gelijk aan  $x \text{ card} * 80$ . Voor deze variabelen gelden weer de waarden, die oorspronkelijk op de ponskaarten stonden. Voor variabelen met een nummer groter dan  $x \text{ card} * 80$  gelden de waarden die zij in de laatste hercodering gekregen hebben (zie 12.3).

## 8. tablelist

### 8.1. Algemeen

Een aanroep van de procedure tablelist heeft tot doel, aan het programma kenbaar te maken, welke tabellen men uitgevoerd wenst te krijgen. Aan de specificatie van een uit te voeren tabel kunnen eventueel nog nevenvoorwaarden worden toegevoegd.

tablelist heeft de volgende vorm:

```
tablelist ("v1; v2 : c (nevenvoorwaarden);  
          v3-v4; v5-v6 : c (nevenvoorwaarden);  
          . . . . .");
```

Hierin zijn v1, v2, v3, v4, v5 en v6 karakterisering van variabelen. Een variabele wordt in tablelist gekarakteriseerd door de kolom, waarin hij begint. Het aantal kolommen, dat hij beslaat moet niet vermeld worden. Een ééndimensionale frequentietabel (rechte telling) wordt in tablelist gespecificeerd door de variabele te noemen, waarvan men de tabel wil hebben.

Een tweedimensionale frequentietabel (kruistabel) wordt in tablelist gespecificeerd door de variabelen, die tegen elkaar moeten worden uitgezet, beiden te noemen, onderling gescheiden door een minteken. Dus de notatie v1 - v2 geeft aan, dat er een tweedimensionale frequentietabel moet worden gevormd, waarin v1 tegen v2 wordt uitgezet.

Wil men behalve de uitvoer over de regeldrukker een frequentietabel ook op ponsband krijgen, dan kan men die wens aan het programma kenbaar maken door achter de specificatie van de betreffende frequentietabel een plusteken te plaatsen. Als men de ééndimensionale frequentietabel van variabele v1 ook op ponsband wil hebben, geeft men dat dus aan door in tablelist de specificatie v1 + te geven en als men de kruistabel van variabele v1 tegen variabele v2 ook op ponsband uitgevoerd wil krijgen, geeft men dat aan door in tablelist de specificatie v1 - v2 + te geven.

Na de tabelspecificatie volgt eventueel : c (nevenvoorwaarden). De specificatie van de tabel wordt daarna afgesloten door een puntkomma.

## 8.2. Het samentrekken van tabelspecificaties

Wanneer in een aantal tabellen de eerste of tweede variabele steeds dezelfde is, dan kan men de specificaties van deze tabellen in een compacte notatie weergeven door b.v.

$$v1 - (v2, v4/v6);$$

Deze specificatie komt dan overeen met de afzonderlijke specificaties

$$v1 - v2; v1 - v4; v1 - v5; v1 - v6;$$

Indien zo'n serie tabellen ook nog onder dezelfde nevenvoorwaarden moet worden uitgevoerd en men van al deze tabellen ponsuitvoer wil hebben, wordt de notatie

$$v1 - (v2, v4/v6) + : c \text{ (nevenvoorwaarden)};$$

Bij één-dimensionale tabellen is een soortgelijke samentrekking toegestaan wanneer de betreffende tabellen onder dezelfde nevenvoorwaarden gevormd en/of op ponsband uitgevoerd moeten worden. Bv.

$$(v7, v8, v10/v16) + : c \text{ (nevenvoorwaarden)};$$

We zullen dit nog illustreren met een paar voorbeelden.

### Voorbeeld 1

Gevraagd worden de tabellen

17-19, 17-21, 17-22, 17-30, 17-34, 17-35, 17-40, 28-33,

21-23, 22-23, 30-23, 32-23, 33-23, 37,23.

Hierbij beslaan de variabelen 19 en 30 elk twee kolommen.

De aanroep van tablelist ziet er dan als volgt uit:

tablelist ("17 - (19/22, 30, 34, 35, 40); 28-33;

(21, 22, 30/33, 37) - 23;");

Voorbeeld 2

Gevraagd worden de tabellen

17, 18, 19, 20, 23, 25, 26, 27, 28, 17-18, 17-19, 17-20,  
17-25, 17-27, 17-28, 17-29, 18-21, 19-21, 20-21.

Al deze tabellen moeten tevens worden uitgeponst. De aanroep van tablelist ziet er dan als volgt uit:

```
tablelist ("(17/20, 23, 25/28) + ;
           17 - (18/20, 25, 27/29) + ;
           (18/20) - 21 + ;");
```

Voorbeeld 3

Gevraagd worden de tabellen

20-21, 20-22, 20-23, 20-24, 20-27, 16, 17, 18, 19, 30.

Al deze tabellen moeten worden uitgeponst. Bovendien wordt bij al deze tabellen als nevenvoorwaarde gesteld:  $28 = 1 \wedge 29 \neq 2$ . De aanroep van tablelist ziet er dan als volgt uit

```
tablelist ("20 - (21/24, 27) + : c(28 = 1 \wedge 29 \neq 2);
           (16/19, 30) + : c(28 = 1 \wedge 29 \neq 1)");
```

Dit laatste voorbeeld attendeert ons op het volgende punt:

Het is niet mogelijk één- en tweedimensionale tabellen in één notatie samen te trekken.

8.3. Nevenvoorwaarden

Als men een frequentietabel niet wil betrekken op alle respondenten, maar slechts op die respondenten, die aan bepaalde voorwaarden voldoen, dan kan dit bewerkstelligd worden, door bij de tabelspecificatie nevenvoorwaarden op te nemen.

Deze nevenvoorwaarden worden opgegeven door achter de gewone tabelspecificatie (als er ponsuitvoer wordt gevraagd achter het + teken, dat

de ponsuitvoer activeert) een dubbele punt te plaatsen gevolgd door een c, met daarachter tussen haakjes de specificatie van de nevenvoorwaarde(n).

In deze nevenvoorwaarden wordt een variabele weer gekarakteriseerd door de kolom waarin hij begint.

Wanneer we de karakterisering van een variabele voorstellen door v en een bepaalde antwoordcode, die voor die variabele kan optreden, door m, dan kunnen de nevenvoorwaarden uit de volgende elementen worden opgebouwd:

1.  $v = m$ .

Dit betekent, dat alleen die respondenten, die voor variabele v een antwoordcode m hebben, voor de vorming van de betreffende frequentietabel gebruikt worden.

2.  $v < m$ .

Dit betekent, dat alleen die respondenten, die voor variabele v een antwoordcode kleiner dan m hebben, voor de vorming van de betreffende frequentietabel gebruikt worden.

3.  $v > m$ .

Dit betekent, dat alleen die respondenten, die voor variabele v een antwoordcode groter dan m hebben, voor de vorming van de betreffende frequentietabel gebruikt worden.

4.  $v \neq m$ .

Dit betekent, dat alleen die respondenten, die voor variabele v een antwoordcode ongelijk aan m hebben, voor de vorming van de betreffende frequentietabel gebruikt worden.

De elementen, die onder 1, 2, 3 en 4 zijn genoemd, kunnen tot ingewikkelder nevenvoorwaarden samengesteld worden door gebruik te maken van de tekens v (of) en ^ (en), waarbij ^ prioriteit heeft boven v.

De nevenvoorwaarde

voorwaarde 1 v voorwaarde 2

heeft het effect, dat die respondenten voor de vorming van de frequentietabel gebruikt worden, die hetzij aan voorwaarde 1, hetzij aan voorwaarde 2, hetzij aan beide voorwaarden voldoen.

De nevenvoorwaarde

voorwaarde 1  $\wedge$  voorwaarde 2

heeft het effect, dat alleen die respondenten voor de vorming van de frequentietabel gebruikt worden, die zowel aan voorwaarde 1 als aan voorwaarde 2 voldoen.

De nevenvoorwaarde

voorwaarde 1  $\vee$  voorwaarde 2  $\wedge$  voorwaarde 3

heeft het effect, dat voor de vorming van de frequentietabel de volgende categorieën respondenten gebruikt worden.

1. die aan voorwaarde 1 voldoen
2. die zowel aan voorwaarde 2 als aan voorwaarde 3 voldoen
3. die aan alle voorwaarden voldoen.

Het maximum aantal nevenvoorwaarden, dat door middel van  $\wedge$  en  $\vee$  tekens mag worden gecombineerd, is 15. Mocht men nog meer nevenvoorwaarden simultaan nodig hebben, dan zal er teruggegrepen moeten worden op hercodering.

#### Voorbeeld

Stel dat in tabel 1 de volgende tabel specificatie voorkomt:

6-7 : c (18 < 5  $\vee$  19 < 5  $\vee$  20  $\neq$  3  $\wedge$  21  $\neq$  3  $\vee$  22 = 7);

Voor de vorming van deze frequentietabel worden nu alleen de respondenten gebruikt, die aan één of meer van de volgende eisen voldoen:

- a. voor variabele 18 een antwoordcode kleiner dan 5
- b. voor variabele 19 een antwoordcode kleiner dan 5
- c. zowel voor variabele 20 als voor variabele 21 een antwoordcode ongelijk aan 3
- d. voor variabele 22 een antwoordcode 7.

#### Opmerking 1

De symbolen  $\leq$  en  $\geq$  zijn in nevenvoorwaarden niet toegestaan.

Opmerking 2

De in een aanroep van tabelist voorkomende variabelen, ook als ze alleen in nevenvoorwaarden genoemd worden, moeten in een voorgaande codelist gespecificeerd zijn.

Opmerking 3

Van de in 2 genoemde mogelijkheid om meer dan één usersprogram op te nemen in één programma, kan gebruik worden gemaakt als door een te groot aantal tabellen in een gebruikersprogramma de geheugencapaciteit zou worden overschreden (een schatting van de benodigde geheugenruimte kan worden verkregen via de testprocedure in het MILLI-Systeem, zie 16).

Men gaat dan als volgt te werk.

Het gebruikersprogramma wordt opgesplitst in twee of meer usersprograms en even zovele aanroepen van process inquiry. Het eerste usersprogram zet de invoer, d.w.z. het array resp (eventueel na hercodering) op de trommel of magneetband en de volgende usersprograms halen de invoer van dit medium.



9. Enkele opmerkingen met betrekking tot de procedures: general data, codelist, ordinal, nominal, irrelevant en tablelist.

In een aanroep van de procedures general data, codelist, ordinal, nominal, irrelevant en tablelist zijn de layout symbolen spatie (= blanke kolom) en twnr (= einde kaart) toegestaan. Wel wordt erop gewezen, dat overtollige layout symbolen onnodig geheugenruimte in beslag nemen.

In alle genoemde procedures mag de punt komma, die de laatste specificatie afsluit, dus de punt komma direkt voor het afsluitende dubbelaccent, weggelaten worden.

## 10. Het sommeren van frequentietabellen

Het is mogelijk het programma frequentietabellen bij elkaar op te laten tellen. Dit kan van groot belang zijn, als de enquête, die verwerkt moet worden, vragen bevat, waarop meer dan één antwoord tegelijk kan worden gegeven.

Stel dat voor de vraag, die in de 10de kolom begint, de antwoorden gecodeerd worden met de cijfers 1 t/m 9, maar dat het ook mogelijk is, dat een respondent twee van deze antwoorden geeft. Men kan dan wel voor elke combinatie van twee antwoorden een aparte code invoeren, maar daardoor zou men 36 nieuwe codes moeten toevoegen, met alle nadelen van dien. Het is veel praktischer voor deze vraag twee variabelen te reserveren, de variabelen 10 en 11, die elk 0/9 als waardebereik hebben. Indien een respondent slechts één antwoord heeft gegeven, wordt dit gecodeerd door in één van beide variabelen de betreffende antwoordcode te zetten en in de andere de code 0 (= geen antwoord). Indien een respondent twee antwoorden heeft gegeven, zet men in kolom 10 het ene antwoord en in kolom 11 het andere.

Wil men nu een frequentietabel maken, waarin de betreffende vraag voorkomt, dan maakt men die tabel eerst met variabele 10, daarna met variabele 11 en vervolgens telt men de twee tabellen bij elkaar op.

Van de mogelijkheid tabellen bij elkaar op te tellen kan ook nog wel eens op andere manieren voordelig gebruik worden gemaakt, maar de bovenbeschreven toepassing is vermoedelijk de meest voorkomende.

Het optellen van frequentietabellen wordt gerealiseerd door gebruik van de procedure `summon`.

Een aanroep van de procedure `summon` heeft tot gevolg dat de laatste tablelist, die aan de aanroep vooraf gaat, nogmaals wordt doorlopen en dat de nieuw gevormde tabellen bij de oude worden opgeteld. Wil men hiermee het gewenste effect van het optellen van twee tabellen bereiken, dan dienen tussen de betreffende tablelist en de aanroep van `summon` natuurlijk enkele hercoderingen te worden aangebracht.

We kunnen de gang van zaken het beste aan de hand van een voorbeeld bekijken.

Voorbeeld

Men wil de volgende tabellen sommeren:

- a. De tabel van variabele 16 en die van variabele 17.
- b. De tabel van variabele 20 tegen 21 en die van variabele 20 tegen 22.

Dit kan in het gebruikers gedeelte als volgt bereikt worden

```
tablelist ("16; 20-21");  
resp[16]:= resp[17]; resp[21]:= resp[22]; summon;
```

Zou men niet alleen 16 en 17 bij elkaar willen optellen, maar 16, 17 en 18 en zouden niet alleen 20 tegen 21 en 20 tegen 22 bij elkaar moeten worden opgeteld, maar 20 tegen 21, 20 tegen 22 en 20 tegen 23, dan zou het stukje programma, dat aan deze wensen tegemoet komt, er als volgt uitzien:

```
tablelist ("16; 20-21");  
resp[16]:= resp[17]; resp[21]:= resp[22]; summon;  
resp[16]:= resp[18]; resp[21]:= resp[23]; summon;
```

In de output verschijnen de gesommeerde tabellen gewoon onder de naam van de oorspronkelijke specificaties, dus in dit geval onder 16 en 20-21.

### 11. De output van het programma

Alle in de verschillende aanroepen van tablelist gespecificeerde tabellen worden over de regeldrukker uitgeprint.

De printoutput van een frequentietabel bevat de volgende gegevens.

1. De karakterisering(en) van de variabele(n) waarop de tabel betrekking heeft. Deze karakterisering geschiedt door het nummer van de kolom, waarin de variabele begint. Indien een variabele een naam heeft meegereggen, wordt die naam ook vermeld.
2. De verschillende antwoordcodes, die in de op deze tabel betrekking hebbende codelist gespecificeerd zijn. Indien aan een antwoordcode een naam is meegegeven, wordt deze naam afgedrukt in plaats van de betreffende code.  
Bij een eendimensionale frequentietabel worden de antwoordcodes horizontaal uitgezet.  
Bij een tweedimensionale frequentietabel wordt de variabele, die in de specificatie van de tabel als eerste werd genoemd vertikaal uitgezet en de tweede horizontaal.
3. Uiteraard worden ook de frequenties van alle codes (combinaties van codes) gegeven.
4. In tweedimensionale frequentietabellen worden de frequenties horizontaal gepercenteerd.
5. Zowel horizontaal als vertikaal worden de frequenties gesommeerd. Er wordt ook gegeven hoeveel respondenten er in de hele frequentietabel zijn verwerkt. (Indien de frequentietabel uit optelling van twee tabellen voortkomt, dient daar natuurlijk rekening mee te worden gehouden.)
6. In horizontale richting worden de gemiddelden van de frequenties gegeven.
7. In horizontale richting worden de standaarddeviaties gegeven.
8. Onder de tabel wordt onder de categorie 'irrelevant answers' vermeld hoeveel niet relevante antwoorden het programma is tegengekomen bij de vorming van deze tabel (zie 6).
9. Onder de tabel wordt onder de categorie 'wrong answers' vermeld hoe-

veel foute antwoordcodes het programma is tegengekomen bij de vorming van deze tabel. Een foute antwoordcode is een antwoordcode, die niet gespecificeerd is in een codelist vóór de tablelist, die aanleiding gaf tot de vorming van de betreffende tabel en die ook niet als irrelevant is gespecificeerd.

10. Onder de tabel worden de resultaten gegeven van eventuele statistische toetsen (zie 18).

Indien men van een frequentietabel ponsuitvoer heeft gevraagd, is de structuur van deze ponsuitvoer als volgt.

1. Tussen accenten worden de nummer(s) gegeven van de variabele(n), waarop de tabel betrekking heeft. Deze nummers zijn onderling door een minteken gescheiden.
2. Daarna worden de afmetingen van de tabel gegeven en wel als volgt: eerst het aantal antwoordcodes dat vertikaal is uitgezet en daarna het aantal dat horizontaal is uitgezet.  
Indien het een ééndimensionale frequentietabel betreft, wordt voor het aantal verticale antwoordcodes het getal 1 gegeven.
3. Daarna worden de frequenties gegeven. Deze worden horizontaal rij voor rij afgewerkt.

Na elke tabel volgt een stuk blanke band.

## 12. Samenwerking tussen hoofdprogramma en gebruikersprogramma

### 12.1. Algemeen

Het hoofdprogramma, dat geactiveerd wordt door de aanroep process inquiry (users program), zorgt ervoor, dat de procedure users program, m.a.w. het gebruikers gedeelte van het programma, een aantal malen wordt doorlopen. (Uiteraard wordt er hierbij vanuit gegaan, dat het gebruikersprogramma door de ALGOL-vertaler syntactisch correct is bevonden.) Men kan hierbij twee fasen onderscheiden, de decodeerfase en de executiefase.

### 12.2. De decodeerfase

Het hoofdprogramma begint met het users program een keer te doorlopen, voordat er een respondent is ingelezen. In deze decodeerfase heeft de integer respondent de waarde 0. Het users program wordt deze eerste keer, dat het doorlopen wordt, gedecodeerd en er wordt nagegaan of het aan alle toelaatbaarheidseisen voldoet.

We wijzen er nog op, dat in de decodeerfase ook eventuele typologieënprogramma's doorlopen worden. Alle elementen van resp hebben dan de waarde 1. Dit zou wel eens tot moeilijkheden kunnen leiden, b.v. als er ergens in een typologieënprogramma door  $\text{resp}[k] - 1$  gedeeld wordt. Deze deling door 0 geeft dan een foutmelding voor de overschrijding van de integercapaciteit. Dergelijke moeilijkheden kunnen eventueel voorkomen worden door gebruik te maken van het feit, dat de waarde van respondent 0 is in de decodeerfase. Het stukje programma

```
'if' respondent  $\neq$  0 'then' a1:= a2/(resp[1] - 1);
```

leidt niet tot foutmeldingen in de decodeerfase.

Slechts indien in de decodeerfase geen fouten zijn gevonden, gaat het programma over tot de executiefase.

### 12.3. De executiefase

In deze fase wordt het gebruikersgedeelte voor elke respondent eenmaal doorlopen. De procedures general data, codelist en tablelist hebben in deze fase de volgende functie.

a. general data zorgt ervoor, dat de volgende respondent in het interne

array resp 0 wordt ingelezen.

- b. Iedere aanroep van codelist zorgt ervoor, dat het array resp 0 wordt overgenomen in het array resp, dat voor de gebruiker beschikbaar is. Het aantal elementen, dat overgenomen wordt, bedraagt  $x \text{ card} * 80$ . Nieuwe variabelen met nummers hoger dan  $x \text{ card} * 80$  blijven bij elke volgende aanroep van codelist ongewijzigd. Dit houdt dus in, dat men in elke eenheid van het gebruikersgedeelte voor variabelen met een nummer  $\leq x \text{ card} * 80$  bij hercoderingen de oorspronkelijke waarden als basis dient te nemen. Voor variabelen met een nummer groter dan  $x \text{ card} * 80$  moet de bij de laatste hercodering toegekende waarde als uitgangspunt worden genomen.
- c. Iedere aanroep van tablelist zorgt ervoor dat de respondent, die in behandeling is, in de gespecificeerde frequentietabellen wordt verwerkt.

### 13. Trommel- en magneetband gebruik

Het is mogelijk het gebruik van de trommel of van magneetbanden in het enqueteprogramma in te passen. Indien men van deze faciliteit gebruik wenst te maken, zal men er meestal goed aan doen op het Mathematisch Centrum hieromtrent advies in te winnen.

Om enig idee te geven van de ingrepen, die trommelgebruik mogelijk maken, zullen we hier een voorbeeld geven.

#### Voorbeeld

Men heeft een gebruikersprogramma, dat zeer veel hercoderingen bevat en dat bovendien veel tabellen wil laten uitvoeren. Het totale programma is hierdoor te groot geworden voor het beschikbare geheugen. Als we nu aannemen, dat er 3 kaarten per respondent zijn en dat er 3 extra variabelen nodig zijn, zou men dit probleem als volgt kunnen oplossen.

- a. Er wordt een apart programma geschreven, dat de hercoderingen uitvoert en de resultaten op de trommel zet. Voor de  $i^{\text{de}}$  respondent gebeurt dit dan door de aanroep:

```
to drum (resp, 40000 + i * 243);
```

- b. Direct daarna wordt het enqueteprogramma gedraaid, waarbij het gebruikersgedeelte er als volgt uit kan zien:

```
-----  
'begin' 'if' respondent = 0 'then'  
    'begin' general data ("...");  
        codelist ("...");  
    'end' 'else'  
    from drum (resp, 40000 + respondent * 243);  
    tablelist ("...")  
'end'  
-----
```



#### 14. Het getallenmateriaal

Bij de codering van het getallenmateriaal zal meestal alleen gebruik worden gemaakt van cijfercodes. Het is echter ook mogelijk om enquetes te verwerken, waarbij lettercodes zijn toegepast. Men dient dan echter wel rekening te houden met de volgende punten:

1. Het programma zet een lettercode direkt om in zijn resymwaarde. Dit houdt in dat een a wordt omgezet in een 10, een b in 11, een c in 12 enz.
2. In codelist, ordinal, nominal, irrelevant en tablelist mogen lettercodes worden opgenomen en men kan ze zelfs in samentrekkende notaties opnemen. Zo zal in codelist de notatie 122 : 1/f worden geïnterpreteerd in die zin, dat variabele 122 de codes 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e en f kan hebben. Hierbij worden a, b, c, d, e en f vervangen door 10, 11, 12, 13, 14 en 15.

In dit verband dient men natuurlijk voorzichtig te zijn met de code 0 (nul) omdat de code o (letter o) nu eenmaal ook is toegestaan.

3. Een lettercode wordt in de output weergegeven door zijn resymwaarde. Wil men nu in de output toch de letter krijgen, dan dient deze letter als naam te worden meegegeven. Als men in codelist de specificatie geeft:

122 : 1/9, a[a], b[b], c[c], d[d], e[e], f[f];

dan zullen in de output de letters onder hun naam verschijnen.

4. In hercoderingen dient men er rekening mee te houden, dat lettercodes door hun resymwaarde worden gerepresenteerd, en men zal dan ook de gewenste bewerkingen op deze getallen moeten uitvoeren.
5. Indien een variabele meer dan 1 kolom beslaat moeten lettercoderingen vermeden worden.

Indien een variabele op de kaartinvoer een waarde heeft, die niet is gespecificeerd in codelist (dit kan bijvoorbeeld het gevolg zijn van het voorkomen van een niet genoemd symbool), dan wordt de betreffende respondent niet gebruikt bij de opbouw van die frequentietabellen, waarin de betreffende variabele een rol speelt. In de output wordt deze respondent dan

in de categorie wrong answers opgenomen. In het array resp verschijnt het antwoord wel. Als het betreffende symbool geen cijfer of letter is, komt in resp de waarde:

$$-10^4 * (\text{resymwaarde van het symbool}).$$

Voor een overzicht van de resymwaarden zij verwezen naar LR 1.1. tabel 6.4.

Voor die variabelen, die niet in een codelist zijn gespecificeerd, geldt, dat voor elk symbool op de kaart rechtstreeks de resymwaarden wordt afgeleverd in het element van resp, dat correspondeert met de betreffende kolom op de kaart. Hiervan kan soms gebruik worden gemaakt om controles op het materiaal uit te voeren.

#### Opmerking 1

Indien voor een variabele de antwoordcode van een bepaalde respondent uit minder cijfers bestaat dan er kolommen gereserveerd zijn voor die variabele, dan worden de overtollige kolommen gevuld met spaties (blank). Het doet er niet toe of deze blanke kolommen vóór, na, of binnen de antwoordcode komen.

Stel dat voor de variabele, die in kolom 9 begint, 3 kolommen gereserveerd zijn en een respondent heeft een 1 als antwoordcode, dan zijn er 3 patronen toegestaan:

- a. 1 in kolom 9, blank in kolom 10 en kolom 11
- b. blank in kolom 9, 1 in kolom 10, blank in kolom 11
- c. blank in kolom 9 en kolom 10, 1 in kolom 11.

#### Opmerking 2

Blanke kaarten in de invoer worden geskipt. Dit houdt in, dat men er goed aan doet elke kaart een identificatienummer te geven. Het toekennen van respondentnummers - en voor kaarten binnen een respondent het toekennen van volgnummers - is toch zeer aan te bevelen, omdat men daardoor de mogelijkheid krijgt om eventuele verstoringen in de kaartvolgorde op te vangen in het gebruikersgedeelte.

## 15. Foutmeldingen

### 15.1. Algemeen

Foutmeldingen in de decodeerfase worden gegeven, indien men zich binnen een aanroep van een van de procedures general data, codelist, ordinal, nominal, irrelevant of tablelist niet houdt aan de voorschriften van dit enqueteprogramma. De algemene vorm van zo'n foutmelding is:

error/foutnummer/volgnummer procedureaanroep/regelnummer binnen deze procedure.

Wanneer achter het laatste getal het symbool \* wordt uitgeprint, betekent dit, dat de gesignaleerde fout het onmogelijk maakt, het programma verder voort te zetten. Het programma wordt in dat geval direkt afgebroken, zonder de decodeerfase eerst te voltooien.

De betekenis van de verschillende foutnummers is

<u>foutnummer</u>	<u>betekenis</u>
1000	) ontbreekt
1001	: ontbreekt
1002	; ontbreekt
1003	: of c of ( ontbreekt bij de opgave van nevenvoorwaarden.
1004	Voor karakterisering van een variabele een getal gebruikt, dat groter is dan $x \text{ card} * 80 + x \text{ col } 1$ .
1005	Karakterisering van variabele niet in voorgaande codelist gespecificeerd.
1006	Onjuiste nevenvoorwaarde gebruikt.
1007	Meer nevenvoorwaarden dan toegestaan.
1008	In general data is de waarde van $x \text{ card} * 80 + x \text{ col } 1$ te groot (d.w.z. groter dan 999).
1009	Geheugenruimte in decodeerfase uitgeput door te grote lengte van codelist en tablelist.

#### Opmerking

Het kan natuurlijk ook voorkomen, dat het gebruikersprogramma niet in correct ALGOL 60 gesteld is. In dat geval worden de foutmeldingen gegeven zoals ze beschreven zijn in LR 1.1.

## 15.2. Een speciale foutmelding

Wanneer tijdens de decodeerfase op een gegeven ogenblik blijkt, dat de geheugenruimte niet toereikend is voor verdere frequentietabellen, dan wordt hiervan op de volgende manier melding gemaakt:

\* available space exceeded \* number of tables accepted:

Hierachter wordt het aantal tabellen gegeven dat nog in de beschikbare geheugenruimte paste. Na deze melding wordt de decodeerfase afgebroken en gaat het programma, indien er verder geen fouten zijn geconstateerd over tot de executiefase.

### Opmerking

De gebruiker kan er zelf voor zorgen, dat er op fouten wordt gecontroleerd tijdens de executiefase. Hierbij kan hij gebruik maken van de procedure report (zie 7).

## 16. Testprocedure in het MILLI-systeem

In het MILLI-systeem is het enqueteprogramma niet beschikbaar. Wel kan een gebruikersprogramma in het MILLI-systeem worden aangeboden om het te laten testen. In de procedure bibliotheek van het MILLI-systeem is namelijk de voorziening getroffen dat alleen de decodeerfase doorlopen wordt. In deze decodeerfase wordt onderzocht of het programma syntactisch correct is volgens de normen, die het enqueteprogramma stelt. Indien syntactische correctheid wordt geconstateerd, wordt tegelijk een schatting van de benodigde geheugenruimte gegeven.

17. Enkele algemene gegevens

1. Het hoofdprogramma neemt aan geheugenruimte ongeveer 10k in beslag.
2. Daar de invoer met de procedure resym wordt gelezen, is het gebruik van dubbelposingen niet toegestaan. Indien men toch verschillende antwoorden op een vraag tegelijk wil kunnen verwerken, dan kan dit geschieden door gebruik te maken van de procedure summon (zie 10).
3. Het programma geeft pas output, nadat alle respondenten verwerkt zijn (althans de output van frequentietabellen. Foutmeldingen via de procedure report worden gegeven zodra de fout geconstateerd is.) Dit kan veel tijd in beslag nemen. Wordt in de opbouwfase het programma afgebroken wegens overschrijding van de in het monitorkopje opgegeven tijd (voor de gegevens hieromtrent zij verwezen naar LR 1.1), dan wordt er geen output gegeven en is de gebruikte machinetijd dus vruchteloos geweest. Om dit bezwaar te ondervangen is in general data maxtime opgenomen (zie 3). Om van deze faciliteit profijt te trekken, moet de aan maxtime meegegeven waarde uiteraard kleiner zijn dan de in het monitorkopje opgegeven tijd, aangezien de output ook nog enige tijd vergt.
4. Het aantal respondenten, dat verwerkt kan worden, is in principe onbeperkt.

18. Statistische toetsen

Op een frequentietabel wordt alleen dan een statistische toets uitgevoerd, als beide bij de tabel betrokken variabelen gespecificeerd zijn in een voorafgaande aanroep van ordinal of nominal. Wordt een variabele in een aanroep van codelist opnieuw gespecificeerd, dan vervallen daarmee eventuele eerdere specificaties van deze variabele in ordinal en nominal. De toetsen worden slechts uitgevoerd op dat gedeelte van de tabel, dat onder de voorgaande specificatie van ordinal en/of nominal valt (dit is van belang als slechts een gedeelte van het waardebereik van een variabele als ordinal of nominal is gespecificeerd).

Welke toetsen worden uitgevoerd kan worden afgelezen uit het volgende schema:

<u>type variabele 1</u>	<u>type variabele 2</u>	<u>uitgevoerde toets</u>
ordinaal	ordinaal	Kendall
nominaal	nominaal	$\chi^2$
nominaal (dichotoom)	ordinaal	Wilcoxon
nominaal (polytoom)	ordinaal	Kruskal-Wallis

Er zij met nadruk op gewezen, dat voor het verkrijgen van de toetsen van Wilcoxon of Kruskal-Wallis in de aanroep van tablelist de nominale variabele als eerste moet worden genoemd. Wordt de ordinale variabele het eerst genoemd, dan wordt geen toets uitgevoerd.

De keuze van toetsen volgens het bovenstaande schema is zo geconstrueerd dat, gezien het schaaltype (ordinaal of nominaal) van de betrokken variabelen zinvolle toetsen worden uitgevoerd. Noch zinvolheid noch bruikbaarheid van de toetsingsresultaten hangt echter van het schaaltype alleen af. Of de toets zinvol is, hangt in eerste instantie af van de in het kader van het onderzoek geformuleerde hypothesen. Of de toets bruikbaar is, hangt in belangrijke mate af van het aantal respondenten, dat in de gebruikte toets betrokken is, en mede van de verdeling van deze respondenten over de getoetste tabel. Voor nadere informatie hierover zij verwezen naar de statistische literatuur.

Op grond van bovenstaande overwegingen wordt, om de schijn te vermijden van bruikbaarheid zonder meer, bij geen van de toetsen een overschrijdingskans gegeven. Volstaan wordt met het geven van toetsingsgrootheden, waarbij, in het geval van zinvolheid en bruikbaarheid, de overschrijdingskansen in de standaard statistische tabellen kunnen worden opgezocht. In het geval van de toets van Kendall en van de  $\chi^2$ -toets worden associatiematen gegeven.

Verder wordt aan de gebruiker overgelaten, of hij een linksezijdige of een rechtsezijdige toets, dan wel een tweezijdige toets wil toepassen.

Het volgende schema geeft een overzicht van de beschikbare gegevens en van de wijze waarop daarvan gebruik kan worden gemaakt. Voor die grootheden, die in de output worden gegeven, zijn de namen waaronder ze daar verschijnen in dit schema tussen haakjes vermeld. (We gaan ervan uit, dat de afmetingen van de tabel waarop de toetsen worden uitgevoerd  $n$  en  $m$  zijn. De kleinste van deze afmetingen noemen we  $q$ .)



Toets	basis-grootheid	spreiding van de basisgrootheid	toetings-grootheid	aantal graden van vrijheid	verdeling bij grote h <sup>1)</sup>	associatie maat
Kendall	Kendall's S  (S)	$\sigma_S$  (ST DEV)	gestandaardiseerde S (SL,SR en S2) <sup>3)</sup>	-	standaard-normaal	$\tau$ (KENDALL'S T)
$\chi^2$	-	-	$\chi^2$ (CHISQUARE)	(DF)	chi-kwadraat	$\chi^2/h(q-1)$ (DA) <sup>2)</sup>
Wilcoxon	Wilcoxon's W  (WILCOXON)	$\sigma_W$  (SW)	gestandaardiseerde W (WL,WR en W2) <sup>4)</sup>	-	standaard-normaal	-
Kruskal-Wallis	-	-	H <sup>5)</sup> (KRUSKAL-WALLIS)	(DF)	chi-kwadraat	-

1) Hierin is h het aantal respondenten, dat bijdraagt tot het getoetste deel van de tabel.

2) DA staat voor degree of association (zie voor deze maat bv. H. Cramér, Mathematical Methods of Statistics, 1961, Princeton University Press, p443).

3) De grootheden voor linkseenzijdige (SL), rechtseenzijdige (SR) en tweezijdige (S2) toetsing worden gegeven

4) De grootheden voor linkseenzijdige (WL), rechtseenzijdige (WR) en tweezijdige (W2) toetsing worden gegeven.

5) Zie W.H. Kruskal en W.A. Wallis, Use of ranks in one-criterion variance analysis, J. American Statistical Association, 47 (1952), 583-621.

19. Een uitgewerkt voorbeeld

Tot slot een uitgewerkt voorbeeld. We zullen hiervoor kiezen een enquête van 19 vragen, die in 1969 onder 30 studenten zou zijn gehouden. We willen wel opmerken, dat zowel de vragen als de antwoorden volledig gefingeerd zijn.

<u>vragenlijst</u>	<u>codering</u>	<u>kolomnummers</u>
<u>1</u> Volgnummer respondent	1 - 30	1, 2
<u>2</u> Leeftijd in jaren	15 - 35	4, 5
<u>3</u> Vooropleiding	1: HBS; 2: Gymnasium; 3: HTS; 4: andere; 12: HBS + Gymnasium; 13: HBS + HTS; ----- 1234: HBS + Gymnasium + HTS + andere;	6, 7, 8, 9
<u>4</u> Studierichting	1: wiskunde; 2: natuurkunde; 3: sterrekunde; 4: scheikunde;	10
<u>5</u> Fase van de studie	1: candidandus; 2: candidaat; 3: drs.	11
<u>6</u> Jaar van aankomst	1: 1955 of eerder; 2: 1956; 3: 1957; ----- 14: 1968;	12, 13
<u>7</u> Nog studerend in de oorspronkelijke studie- richting	1: neen; 2: ja;	14
<u>8</u> Volgt colleges buiten eigen faculteit	-1: nooit; 0: soms; +1: regelmatig;	15, 16
<u>9</u> Studieruimte	1: thuis; 2: op kamers;	17
<u>10</u> Mensa bezoek	1: nooit; 2: soms; 3: vaak; 4: altijd;	18
<u>11</u> Afstand kamer tot collegeruimten	1: minder dan 1 km; 2: 1-3 km; 3: 3-6 km; 4: meer dan 6 km;	19

<u>12</u> Middel van vervoer naar colleges	1: openbaar vervoer; 2: lopen; 3: fiets; 4: brom- fiets; 5: motor of scooter; 6: auto;	20
<u>13</u> Afstand studieplaats tot ouderlijk huis	1: minder dan 10 km; 2: 10-30 km; 3: 30-70 km; 4: meer dan 70 km;	21
<u>14</u> Frequentie bezoek ouderlijk huis	aantal keren per jaar 0-53 (53 = vaker dan eens per week)	22-23
<u>15</u> Burgerlijke staat	1: ongehuwd; 2: gehuwd; 3: gehuwd geweest;	24
<u>16</u> Sportbeoefening	1: denksporten; 2: roeien; 3: tennis; 4: zaalsporten; 5: andere sporten;	25, 26, 27, 28 en 29 (zie toelichting onder vragen- lijst)
<u>17</u> Gezondheidstoestand	1: zwak; 2: redelijk; 3: goed; 4: zeer goed;	30
<u>18</u> Voorziening levens- onderhoud	1: toelage ouders; 2: beurs of renteloos voorschot; 3: werkstudent;	31
<u>19</u> Alleen voor hen, die vraag 18 met 1 of 2 hebben beantwoord. Bijverdiensten	1: geen; 2: zo nu en dan; 3: regelmatig;	32

Toelichting bij vraag 16: Voor vraag 16 zijn vijf kolommen gereserveerd. Indien slechts één van de vijf genoemde sport-categorieën wordt bedreven, dan wordt in één van de kolommen de betreffende antwoordcode ingevuld en de andere kolommen worden blank gelaten. Indien meer dan één sport-categorie van toepassing is, dan kan dit worden aangegeven door in meer dan één kolom een antwoordcode in te vullen. Als een respondent bijv. roeit, tennist en zeilt, dan kan dit resulteren in het volgende antwoordpatroon:

kolom 25 een code 2 (roeien)  
kolom 26 een code 3 (tennis)  
kolom 27 een code 5 (andere sporten)

### Scorematrix

De nu volgende scorematrix zou kunnen corresponderen met bovenstaande vragenlijst.

01	18001211122+1242322612	43
02	16000421142-12423412134	323
03	20000443102 01134153124	411
04	18000211141-1242322611	123
05	30013443 12+11124153235	311
06	20000142102-124212261235	43
07	23001432 71+1223522623	212
08	17000121132-111251531	421
29	20000142102-111121531124	421
10	28013433 42-1112415311	421
11	18000131122-11223153134	421
12	20000232132 01135153112	31
13	22002422 81-12434230115	221
14	20000212112+124343261234	43
15	19000221112 01133153135	422
16	21000122101+123312401	23
17	23001321 82 02423448212345421	
18	25001343 52 0111215325	411
19	20000142112-12433326145	43
20	18000342122-1232421212	413
21	21000343 92+12346126123	43
22	24001221 92-12423226125	421
23	34123413 12-12424312312	311
24	20000311111-1222332615	422
25	20000212132-1134115312	421
26	23000341102 02235240234	421
27	21000431 91-12423336123	43
28	26001442 62-1112115325	321
29	20000321131-111241531	121
30	18000142122-122233121125	321

Frequentietabellen

- ééndimensionaal: vraag 8;  
vraag 15;  
vraag 15 voor de gezonde respondenten.  
(d.w.z. de respondenten die vraag 17 met 3 of 4  
beantwoord hebben.)
- tweedimensionaal: vraag 9 tegen vraag 10;  
vraag 12 tegen vraag 15;  
vraag 10 tegen vraag 15;  
vraag 5 tegen vraag 10 voor de respondenten die  
thuiswonen en geen werkstudent zijn;  
vraag 5 tegen vraag 19 voor de respondenten die  
vraag 18 met 1 of 2 beantwoord hebben;  
vraag 16 tegen vraag 6;  
vraag 16 tegen vraag 6 voor de respondenten die  
jonger zijn dan 25 jaar.

Bij deze laatste twee frequentietabellen is het de bedoeling dat voor vraag 16 achtereenvolgens de kolommen 25, 26, 27, 28 en 29 wordt genomen en dat de aldus ontstane tabellen bij elkaar worden opgeteld.

In de kolommen 25, 26, 27, 28 en 29 dient een blank als een irrelevante code te worden opgevat.

Hercoderingen

Van vraag 14 wordt een klasse-indeling gemaakt volgens het voorschrift:

- 0 - 6 wordt 1 (zelden),
- 7 - 12 wordt 2 (regelmatig),
- 13 - 26 wordt 3 (vaak),
- 27 - 53 wordt 4 (zeer vaak).

Van vraag 6 wordt een klassificatie gemaakt volgens het voorschrift:

- 1967, 1968 wordt 1 ('67-'68)
- 1964, 1965, 1966 wordt 2 ('64-'66)
- 1960, 1961, 1962, 1963 wordt 3 ('60-'63)

vóór 1960

wordt 4 (vóór '60)

Uit vraag 2 en vraag 6 wordt de leeftijd bij aanvang van de studie berekend en als nieuwe variabele toegevoegd ("vraag 20", "kolom" 33).

Van deze variabele "leeftijd in jaar van aankomst" wordt de volgende klassering gemaakt:

15, 16, 17	wordt 1
18, 19, 20	wordt 2
21, 22, 23, 24	wordt 3
25 en ouder	wordt 4.

Met het gehercodeerde materiaal worden de volgende frequentietabellen gemaakt.

ééndimensionaal:	vraag 14 voor de ongehuwde respondenten;
tweedimensionaal:	vraag 13 tegen vraag 14 voor de ongehuwde respondenten;
	vraag 2 tegen "vraag 20";
	vraag 14 tegen vraag 19.

### Statistische toetsen

Op de tabel van vraag 9 tegen vraag 10 moet de toets van Wilcoxon worden uitgevoerd.

Op de tabellen van vraag 5 tegen vraag 10 en vraag 5 tegen vraag 19 moet de toets van Kruskal Wallis worden uitgevoerd.

Op de tabel van vraag 12 tegen vraag 15 moet  $\chi^2$  worden berekend, waarbij het de bedoeling is, dat alleen de eerste twee antwoordcodes van vraag 15 in beschouwing worden genomen.

Op de tabel van vraag 14 tegen vraag 19 moet de toets van Kendall worden uitgevoerd.

Alle gewenste gegevens kunnen uit het basismateriaal (de scorematrix) worden gehaald door het volgende programma.

N.B. In het gegeven programma is kolom 4 (vraag 2) als ordinal gedeclareerd. Deze declaratie heeft op de output in dit geval geen enkele invloed. Hij is bedoeld als illustratie van het feit dat een nominal of ordinal declaratie op één van de variabelen van een tweedimensionale frequentietabel geen effect heeft.

```

1  'BEGIN' 'PROCEDURE' USERS PROGRAM(GENERAL DATA,CODELIST,ORDINAL,NOMINAL,
2      IRRELEVANT,TABLELIST,SUMMON,FINISH,REPORT,RESP,RESPONDENT);
3      'BEGIN' 'INTEGER' I)
4
5      GENERAL DATA("30,1,0,50");
6
7      CODELIST("
8          4,2{LEEFTIJD}: 15/35)
9          11{FASE STUDIE}: 1{KAND,S}, 2{KAND,}, 3{DRS,}
10         12,2{ 1/14}
11         15,2{KOLL BUITEN FAKULTEIT}: -1{NOOT}, 0{SOMS}, 1{REGELM}
12         17{STUDIERUIMTE}: 1{THUIS}, 2{KAMERS}
13         18{MENSABEZOEK}: 1{NOOT}, 2{SOMS}, 3{VAAK}, 4{ALTIJD}
14         20{VERVOERMIDDEL}: 1{OPENB,}, 2{LOPEN}, 3{FIETS}, 4{BROMF,},
15             5{MOT/SC}, 6{AUTO}
16         21{AFST OUDERL Hs}: 1{<10 KM}, 2{10-30}, 3{30-70}, 4{>70 KM}
17         22,2{BEZ/J OUDERL H}: 1{ZELDEN}, 2{REGELM}, 3{VAAK}, 4{Z VAAK}
18         24{BURGERL. STAAT}: 1{ONGEH}, 2{GEHUWD}, 3{G. GEW}
19         30{GEZONDHEID}: 1{ZWAK}, 2{REDEL,}, 3{GOED}, 4{Z GOED}
20         31{VOORZ LEVENSOND}: 1{OUDERS}, 2{BEURS}, 3{WERKST}
21         32{BIJVERDIENSTEN}: 1{GEEN}, 2{SOMS}, 3{REGELMATIG}
22         33{LEEFT AANV STUDIE}: 1{15-17}, 2{18-20}, 3{21-24}, 4{OUDER}");
23
24      ORDINAL("4; 18; 22; 32");
25      NOMINAL("11; 17; 20; 24,2");
26
27      'IF' RESP[1]*10+RESP[2]*RESPONDENT 'THEN'
28      REPORT("INCORRECT RESP NR",IFALSE);
29      'IF' RESP[22]
30      RESP[22]='IF' |<7 'THEN' 1 'ELSE' 'IF' |<13 'THEN' 2 'ELSE'
31          'IF' |<27 'THEN' 3 'ELSE' 4)
32      'IF' RESP[4]-(15-RESP[12])
33      RESP[33]='IF' |<15 'THEN' =1 'ELSE' 'IF' |<18 'THEN' 1 'ELSE'
34          'IF' |<21 'THEN' 2 'ELSE' 'IF' |<25 'THEN' 3 'ELSE' 4)
35
36      TABLELIST("15+; 24; 17-18; 22-32)
37          24{C(30=3 v 30=4); 22{C(24=1); 21-22{C(24=1);
38          (20,18)=24+; 11=18{C(17=1 ^ 31+3)} 11-32{C(31<3); 4-33"});
39
40
41      CODELIST("
42          12,2{JAAR AANKOMST}: 1{67-68}, 2{64-66}, 3{61-63}, 4{VOOR61}
43          (25/29){SPORTBEOEFENING}: 1{DENKSP}, 2{ROEIEN}, 3{TENNIS},
44              4{ZAALSP}, 5{ANDERE}");
45
46      IRRELEVANT("(25/29)=93,4");
47
48      'IF' RESP[12]
49      RESP[12]='IF' |<7 'THEN' 4 'ELSE' 'IF' |<10 'THEN' 3 'ELSE'
50          'IF' |<13 'THEN' 2 'ELSE' 1)
51
52      TABLELIST("25=12; 25=12{C(4<25)}");
53      'FOR' 'I'=26 'STEP' 1 'UNTIL' 29 'DO'
54      'BEGIN' RESP[25]=RESP[I]; SUMMON 'END'
55  'END' USERS PROGRAM)
56  PROCESS INQUIRY(USERS PROGRAM)
57  'END'

```

ERRORS IN INPUT

RESP 9 | INCORRECT RESP NR NOT SKIPPED

30 RESPONDENTS HAVE BEEN PROCESSED, 0 OF WHICH WERE SKIPPED

2 MH WERE NEEDED TO FORM THE TABLES



CROSSTABLES CORRESPONDING TO TABLELIST 1

19 [KOLL BUITEN FAK]

	NOOIT	SOMS	REGELM	TOTAL	MEAN	ST,DEV.
	18	6	6	30	-.40	.814
	60,0	20,0	20,0			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS: 0

24 [BURGERL, STAAT]

	ONGEH	GEHUWD	G, GEW	TOTAL	MEAN	ST,DEV.
	23	6	1	30	1,27	,921
	76,7	20,0	3,3			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS: 0

17 [STUDIERUIMTE] - 18 [MENSABEZOEK]

	NOOIT	SOMS	VAAK	ALTIJD	TOTAL	MEAN	ST,DEV.
THUIS	10	1	1	0	12	1,25	,622
KAMERS	0	4	3	11	18	3,39	,890
	0	22,2	16,7	61,1			
TOTAL	10	5	4	11	30	2,53	1,306
	33,3	16,7	13,3	36,7			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS: 0

WILCOXON: +15,0000                      SW: 45,0103  
 WL: -4,4434                              WR: -4,4879                      W2: +4,4434

22 (BEZ/J OUDERL H) = 32 (BIJVERD|ENSTEN)

	GEEN	SOMS	REGELM	TOTAL	MEAN	ST,DEV,
ZELDEN	0	0	0	0	,00	,000
REGELM	2	0	2	4	2,00	1,155
VAAK	1	2	1	4	2,00	,816
Z VAAK	13	1	0	14	1,07	,267
TOTAL	16	3	3	22	1,41	,734
	72,7	13,6	13,6			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS : 6

KENBALLS T1 =,5607 S1 =,65000n+2 ST DEVI +,23509n+2  
 S11 =2,7223 S11 =2,8074 S21 =2,7223

24 (BURGERL, STAAT) C( 30 = 3 v 30 = 4 )

	ONGEH	GEHUWD	G, GEW	TOTAL	MEAN	ST,DEV,
	19	5	1	25	1,28	,542
	76,0	20,0	4,0			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS : 0

22 (BEZ/J OUDERL H) C( 24 = 1 )

	ZELDEN	REGELM	VAAK	Z VAAK	TOTAL	MEAN	ST,DEV,
	0	3	8	12	23	3,39	,722
	,0	13,0	34,8	52,2			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS : 0

21 [AFST OUDERL MS] - 22 [BEZ/J OUDERL H] C( 24 = 1 )

	ZELDEN	REGELM	VAAK	Z VAAK	TOTAL	MEAN	ST,DEV.
<10 KM	0	0	1	9	10	3,90	,316
10-30	0	1	4	2	7	3,14	,690
30-70	0	1	3	1	5	3,00	,707
>70 KM	0	1	0	0	1	2,00	,000
TOTAL	0	3	8	12	23	3,39	,722
	,0	13,0	34,8	52,2			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS: 0

20 [VERVOERMIDDEL] - 24 [BURGERL, STAAT]

	ONGEH	GEHUWD	G, GEW	TOTAL	MEAN	ST,DEV.
OPENS.	3	1	0	4	1,25	,500
LOPEN	1	1	0	2	1,50	,707
FLETS	10	1	0	11	1,09	,302
BROMF.	6	1	1	8	1,37	,744
MOT/SC	2	2	0	4	1,50	,577
AUTO	1	0	0	1	1,00	,000
TOTAL	23	6	1	30	1,27	,521
	76,7	20,0	3,3			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS: 0

CHISQUARE: 4,52      DF: 5      DAI: ,1560

10 [MENSABEZOEK] = 24 [BURGERL, STAAT]

	ONGEH	GEHUWD	G, GEW	TOTAL	MEAN	ST,DEV.
NOOIT	7	3	0	10	1,30	,483
SOMS	20,0	30,0	0	5	1,40	,548
VAAK	60,0	40,0	0	4	1,00	,000
ALTIJD	100,0	0	0	11	1,27	,647
	81,8	2,1	2,1			
TOTAL	23	6	1	30	1,27	,521
	76,7	20,0	3,3			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS : 0

11 [FABE STUDIE] = 18 [MENSABEZOEK] C( 17 = 1 + 31 + 3 )

	NOOIT	SOMS	VAAK	ALTIJD	TOTAL	MEAN	ST,DEV.
KAND, S	3	1	0	0	4	1,25	,500
KAND,	75,0	25,0	0	0	4	1,50	1,000
DRS,	75,0	0	25,0	0	4	1,00	,000
	100,0	0	0	0			
TOTAL	10	1	1	0	12	1,25	,622
	83,3	8,3	8,3	,0			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS : 0

KRUSKAL WALLISI +1,1136 DFI 2

13 (PASE STUDIE) - 32 (BIJVERDIENSTEN) C( 31 &lt; 3 )

	GEEN	SOMS	REGELM	TOTAL	MEAN	ST,DEV,
KAND,S	6	2	2	10	1,60	,843
KAND,	5	1	1	7	1,43	,787
DRS,	5	0	0	5	1,00	,000
TOTAL	16	3	3	22	1,41	,734
	72,7	13,6	13,6			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS 1 1

KRUSKAL WALLISI +2,5108

DFI 2

4 [LEEF TIJD] = 33 [LEEF T AANV STUD]

	15-17	18-20	21-24	OUDER	TOTAL	MEAN	ST. DEV.
15	0	0	0	0	0	,00	,000
16	1	0	0	0	1	1,00	,000
17	1	0	0	0	1	1,00	,000
18	5	0	0	0	5	1,00	,000
19	1	0	0	0	1	1,00	,000
20	6	3	0	0	9	1,33	,500
21	3	0	0	0	3	1,00	,000
22	1	0	0	0	1	1,00	,000
23	2	1	0	0	3	1,33	,577
24	0	1	0	0	1	2,00	,000
25	1	0	0	0	1	1,00	,000
26	1	0	0	0	1	1,00	,000
27	0	0	0	0	0	,00	,000
28	1	0	0	0	1	1,00	,000
29	0	0	0	0	0	,00	,000
30	1	0	0	0	1	1,00	,000
31	0	0	0	0	0	,00	,000
32	0	0	0	0	0	,00	,000
33	0	0	0	0	0	,00	,000
34	0	1	0	0	1	2,00	,000
35	0	0	0	0	0	,00	,000
TOTAL	24	6	0	0	30	1,20	,407
	80,0	20,0	,0	,0			

IRRELEVANT ANSWERS: 0  
 WRONG ANSWERS 1 0

CROSSTABLES CORRESPONDING TO TABLE 2 (SUMMED)

25 [SPORTBEOEFENING] = 12 [JAAR AANKOMST]

	67-68	64-66	61-63	VOOR61	TOTAL	MEAN	ST. DEV.
DENKSP	2	2	2	2	8	2,50	1,195
ROEIEN	25,0	25,0	25,0	25,0	14	2,29	,825
TENNIS	14,3	50,0	28,6	7,1	11	2,45	,820
ZAALSP	9,1	45,5	36,4	9,1	8	2,00	,535
ANDERE	12,5	25,0	12,5	0	11	2,82	,874
TOTAL	6	25	14	7	52	2,42	,871
	11,9	48,1	26,9	13,5			

IRRELEVANT ANSWERS: 98  
 WRONG ANSWERS: 0

25 [SPORTBEOEFENING] = 12 [JAAR AANKOMST] C( 4 = 25 )

	67-68	64-66	61-63	VOOR61	TOTAL	MEAN	ST. DEV.
DENKSP	2	2	2	0	6	2,00	,894
ROEIEN	33,3	33,3	33,3	0	13	2,15	,689
TENNIS	18,8	53,8	30,8	0	10	2,30	,675
ZAALSP	10,0	50,0	40,0	0	8	2,00	,535
ANDERE	12,5	25,0	12,5	0	8	2,37	,518
TOTAL	6	25	14	0	45	2,18	,650
	13,3	55,6	31,1	0			

IRRELEVANT ANSWERS: 80  
 WRONG ANSWERS: 0

Ponsuitvoer

' 15 '

1 3

18 6 6

' 20 - 24 '

6 3

3 1 0

1 1 0

10 1 0

6 1 1

2 2 0

1 0 0

' 18 - 24 '

4 3

7 3 0

3 2 0

4 0 0

9 1 1