

move -12345 to D

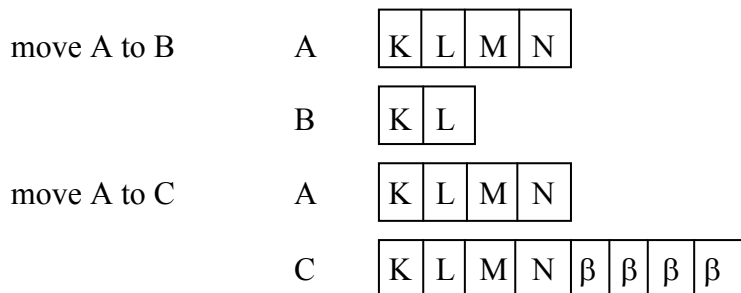
-	2	3
---	---	---

 rekenwaarde = -2300

➤ **Alfanumerieke overdracht**

De tekens worden van links naar rechts overgebracht. Als het ontvangende veld langer is dan het zendende veld, dan wordt het ontvangende veld aan de rechterkant aangevuld met spaties. Als het ontvangende veld korter is, dan worden de over te brengen waarden aan de rechterkant afgekapt. (uitz. JUSTIFIED-clausule)

01 A PIC X(4) value "KLMN".
 01 B PIC XX.
 01 C PIC X (8).

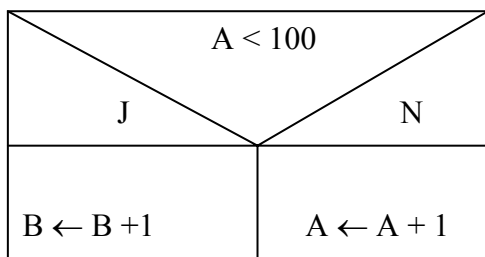


1.1.1 De IF-opdracht p 97 - 102

IF condition THEN { {statement-1} ...
 NEXT SENTENCE } { ELSE {statement-2}...[END-IF]
 ELSE NEXT SENTENCE
 END-IF }

Voorbeeld 1

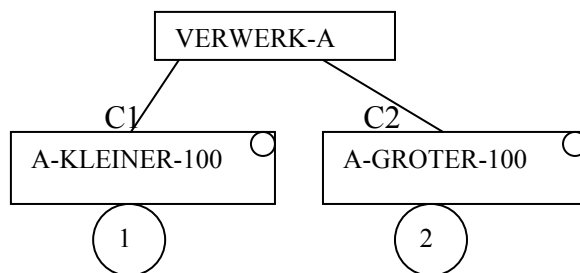
➤ NS-diagram:



➤ pseudo-code

ALS (A < 100)
 DAN B ← B + 1
 ANDERS A ← A + 1
 EINDE-ALS-DAN

Jackson:



Conditie en opdrachten:

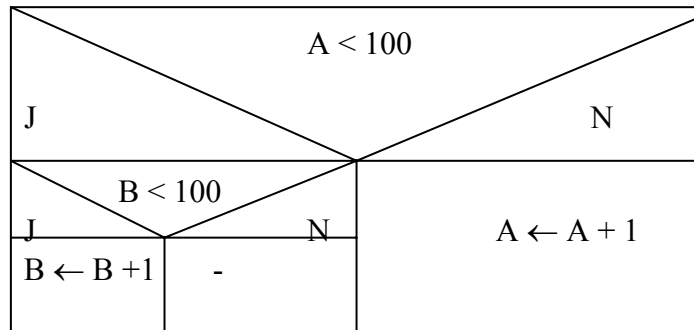
C1: ALS A < 100
 C2: ALS A >= 100
 1: B ← B + 1
 2: A ← A + 1

➤ COBOL-code:

```
IF A < 100
    THEN add 1 to B
    ELSE add 1 to A
END-IF
```

Voorbeeld 2

➤ NS-diagram



A	B	
97	84	B + 1
97	103	-
163	12	A + 1
163	106	A + 1

➤ pseudo-code

```
ALS (A < 100)
    DAN ALS (B < 100)
        DAN B ← B + 1
    EINDE ALS-DAN
    ANDERS A ← A + 1
EINDE-ALS-DAN
```

➤ COBOL code

```
IF A < 100
    THEN IF B < 100
        THEN add 1 to B
        END-IF
    ELSE add 1 to A
END-IF
```


➤ COBOL code:

```
IF A < 100
  THEN add 1 to B
    IF C < 100
      THEN add 1 to A
      ELSE add 1 to C
    END-IF
  ELSE IF B < 100
    THEN add 10 to A
    ELSE add 20 to C
  END-IF
  add 7 to B
  DISPLAY "GEDAAN"
END-IF
```

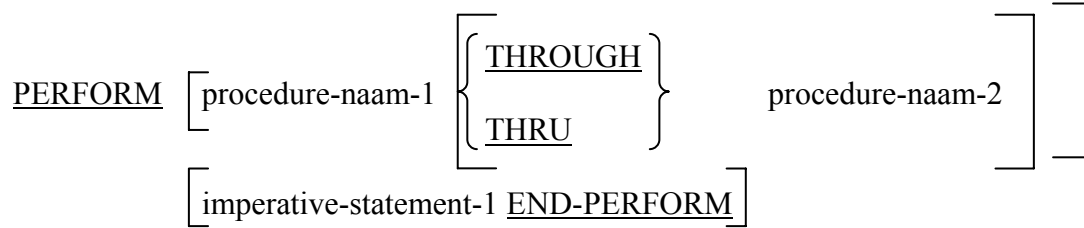
Opgave IF

Bereken van 4 ingevoerde toetsen het gemiddelde. Elke toets staat op 10 punten. Toon op basis van het gemiddelde hetvolgende op scherm:

0 – 4	:	slecht
5	:	voldoende
6 – 9	:	goed
10	:	uitstekend

1.1.2 Het PERFORM-statement p 84 – 85 en 104 - 117

➤ Eerste formaat



Vb.:

```

(1) ...
    ...
    PROCEDURE DIVISION.
        HOOFDPROGRAMMA.
    ...
    ...
        PERFORM
            add aantal to subtotaal           weinig gebruikt
            move subtotaal to totaal
        END-PERFORM
    ...
    ...
    
```

Opgave: Klant, artikel, aankoop en verkoop zijn invoerschermen.

Toon bij resultaten hetvolgende: voor aan- en verkoopbedrag: positief, negatief of nul.
 Toon welk artikel het is en of het met winst, verlies of geen van beide verkocht werd.

Bereken en toon de winst. Toon het juiste BTWpercentage.

HOGESCHOOL GENT		15/10/00
MUZIEK		
KLANTGEGEVENS		ARTIKEL
KLNK: NAAM: VOORNAAM:	MUZIEKCODE 1=CD 2=PLAAT 3=DVD	
AANKOOP - VERKOOP		
AANKOOPBEDRAG: BTWcode : . (1=6% 2=21%)		VERKOOPBEDRAG:
RESULTATEN		
AANKOOPBEDRAG: Positief Het artikel is een CD en wordt verkocht met WINST De winst is		VERKOOPBEDRAG: Positief Het BTWpercentage is ...
<ENTER> voor verder		
DEMOL PETER		OEFENING 4

PROCEDURE DIVISION.
HOOFDPROGRAMMA.

DISPLAY scherm-1	}	uitvoer-scherm-1
ACCEPT datum FROM date		
DISPLAY datum		
ACCEPT klnr...	}	inlezen-muziekgegevens
ACCEPT familienaam...		
...		
IF ... END-IF	}	uitvoer-resultaten
IF ... END-IF		
...		
ACCEPT TOETS.		
STOP RUN.		

Opgave : Herschrijven van vorige opgave, gebruik makend van het perform-statement, 1^{ste} formaat

...
...

PROCEDURE DIVISION.
HOOFDPROGRAMMA.

PERFORM uitvoer-scherm-1	}	→ oppassen met volgorde
PERFORM inlezen-muziekgegevens		
PERFORM uitvoer-resultaten		
STOP RUN.		
		OF
		PERFORM uitvoer-scherm-1
		THRU uitvoer-resultaten

Uitvoer-scherm-1.

DISPLAY scherm-1
ACCEPT datum from date
DISPLAY datum.

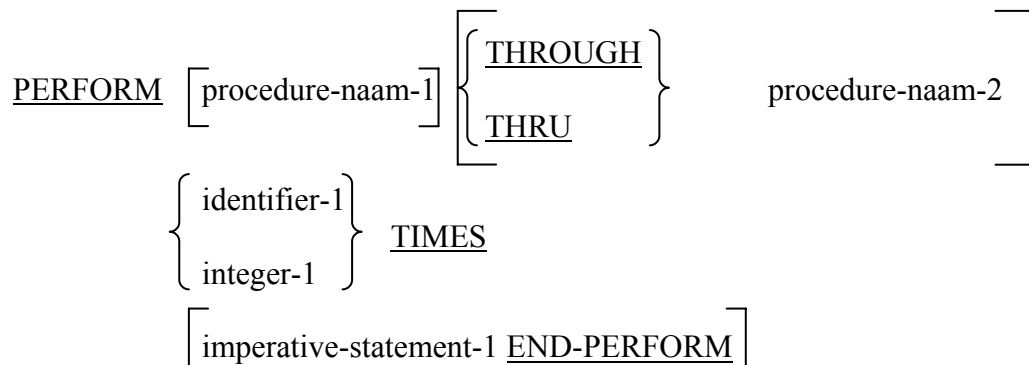
Inlezen-muziekgegevens.

ACCEPT klnr
...
...

Uitvoer-resultaten.

IF ... END-IF.
...
ACCEPT TOETS.

➤ Tweede formaat: de perform-opdracht met TIMES



...

...

PROCEDURE DIVISION.
HOOFDPROGRAMMA.

...

```

PERFORM 10 TIMES
  add ...
  move...
END-PERFORM

```

Opgave: Breid vorige opgave uit zodat er 5 muziekgegevens worden ingegeven

...

PROCEDURE DIVISION.
HOOFDPROGRAMMA.

```

PERFORM uitvoer-scherm-1 THRU uitvoer-resultaten 5 TIMES
STOP RUN.

```

...

Opgave: Breid vorige opgave uit zodat er verschillende muziekgegevens worden ingelezen.
Er wordt eerst aan de gebruiker gevraagd hoeveel er moeten worden ingelezen.

```

DATA DIVISION.
WORKING-STORAGE SECTION.
01 teller PIC 99.

```

PROCEDURE DIVISION.
HOOFDPROGRAMMA.

```

PERFORM invoer-teller
PERFORM uitvoer-scherm-1 THRU uitvoer-resultaten teller TIMES
STOP RUN.

```

INVOER-TELLER.

```

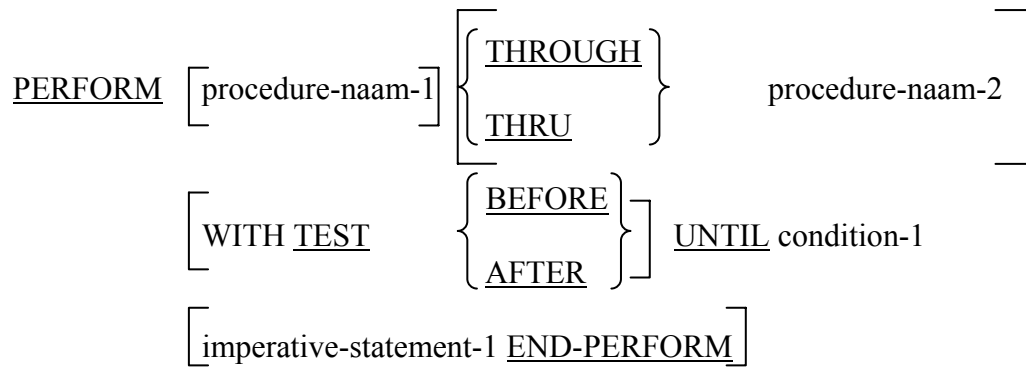
DISPLAY "Hoeveel muziekgegevens wil je inlezen ?"...
ACCEPT teller... .

```

UITVOER-SCHERM-1.

...

➤ Derde formaat: de perform-opdracht met UNTIL



Opgave : breid vorige opgave uit. Er worden muziekgegevens ingelezen tot het antwoord op de vraag "Wil je nog muziekgegevens inlezen (J/N)?" "N" of "n" is.

DATA DIVISION.

...

01 antwoord PIC X value 'N'.

...

PROCEDURE DIVISION.

HOOFDPROGRAMMA.

PERFORM uitvoer-scherm-1 THRU invoer-antwoord WITH TEST AFTER
UNTIL antwoord = "N" or antwoord = 'n'

STOP RUN.

uitvoer-scherm-1.

...

inlezen-muziekgegevens.

...

uitvoer-conclusies.

...

invoer-antwoord.

DISPLAY "Wil je nog muziekgegevens inlezen (J/N)?" ...

ACCEPT antwoord

Opgave : Er wordt eerst aan de gebruiker gevraagd of hij muziekgegevens wil inlezen – principe van vooruitlezen.

...

PROCEDURE DIVISION.

HOOFDPROGRAMMA.

PERFORM Invoer-Antwoord THRU
Uitvoer-resultaten WITH TEST BEFORE
UNTIL ANWOORD = "N" or "n"

STOP RUN.

INVOER-ANTWOORD.

UITVOER-SCHERM-1.

INLEZEN-MUZIEKGEGEVENS

UITVOER-RESULTATEN.

PROCEDURE DIVISION.
HOOFDPROGRAMMA.
 PERFORM INVOER-ANTWOORD
 PERFORM UITVOER-SCHERM-1
 THRU INVOER-ANTWOORD
 [WITH TEST BEFORE]
 UNTIL ANTWOORD = "N" or "n"
 STOP RUN.

UITVOER-SCHERM-1.

...

NOG BETER IS:

...

PROCEDURE DIVISION.
HOOFDPROGRAMMA.
 PERFORM INVOER-ANTWOORD
 PERFORM BODY WITH TEST BEFORE
 UNTIL ANTWOORD = "N" or "n"
 STOP RUN.

BODY.

 PERFORM UITVOER-SCHERM-1.
 PERFORM INLEZEN MUZIEKGEGEVENS
 PERFORM UITVOER-RESULTATEN.
 PERFORM INVOER-ANTWOORD.

UITVOER-SCHERM-1.

...

INLEZEN MUZIEKGEGEVENS...

...

UITVOER-RESULTATEN.

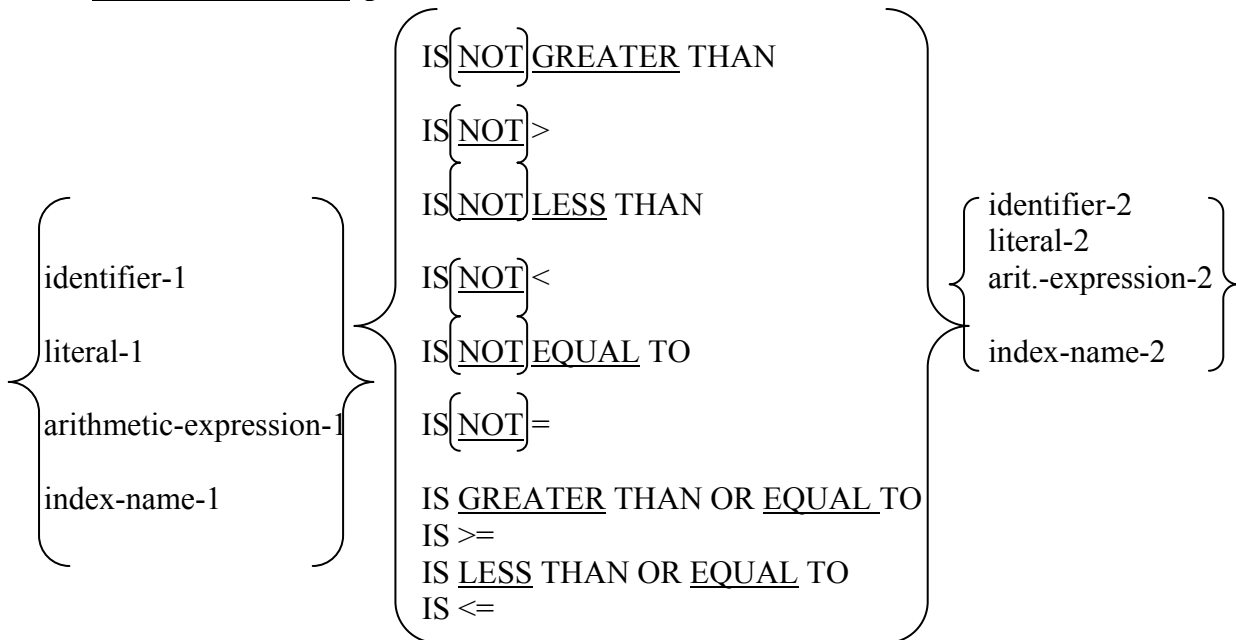
...

INVOER-ANTWOORD.

...

1.1.3 Soorten condities -voorwaarden

➤ De relatieconditie p 74



◆ numerieke vergelijking

01 A PIC 99V99 VALUE 12.35.
 01 B PIC 999V99 VALUE 12.35.
 01 C PIC S9 VALUE -1.
 01 D PIC 9PP VALUE 800.

if A = B...	waar
if A NOT < B...	waar
if A >= B...	waar
if A < C...	onwaar
if A - B >= C...	waar
if A - B <= zero	waar
if D = 8...	onwaar
if D = 800	waar
if D > 799	waar
if A > D - 750...	onwaar

◆ alfanumerieke vergelijking

01 A PIC XXX value "KLM".
 01 B PIC XXX value "KLP".

if A = "KLM"...	waar
if A = B...	onwaar

if A <= "KLZ"... waar
 if A > "abc"... onwaar
 if A < "KL4"... onwaar

- ◆ Operanden van ongelijke lengte: de kortste wordt denkbeeldig aangevuld met spaties

01 A PIC XXX value "ABC".

if A = "AB"... onwaar
 if A > "AB" waar
 if A <= "ABCD" onwaar
 if A <= "ABCD" waar

- ◆ alfanumerieke vergelijking met een numerieke operand: is toegestaan maar geeft veel fouten → afgeraden

➤ **De klasseconditie p 75**

om te testen of een veld een specifiek soort tekens bevat

Identifier IS (NOT) {
 NUMERIC
 ALPHABETIC
 ALPHABETIC-UPPER (hoofdletters en spaties)
 ALPHABETIC-LOWER (kleine letters en spaties)
 CLASS-NAME-1
 }

- ◆ numeric test

vb. (1)

01 artikelgegevens.

 05 kleurnummer PIC 99.

 ...

ACCEPT kleurnummer ← controle invoer

1. PERFORM with test after until kleurnummer numeric
 ACCEPT kleurnummer
 END-PERFORM
2. PERFORM with test after until kleurnummer numeric
 ACCEPT kleurnummer
 If kleurnummer not numeric
 THEN DISPLAY "FOUT"
 END-IF
 END-PERFORM

```

ACCEPT kleurnummer
PERFORM with test after until kleurnummer numeric
  DISPLAY "FOUT"
ACCEPT kleurnummer
END-PERFORM

```

vb. (2)

01 bestelrecord.

```

03 artikelcode      PIC X(25).
03 besteld-aantal  PIC 999.

```

If besteld-aantal numeric

Then ...

Else ...

End-if

PIC	INHOUD	IF ,, UERIC
999	538	WAAR
999	53H	ONWAAR
999	XYZ	ONWAAR
S999	538	ONWAAR
S999	XYZ	ONWAAR
S999 sign trailing	53H	WAAR
S999 sign trailing separate	538+	WAAR
S999 sign trailing separate	538	ONWAAR
XXX	538	WAAR
XXX	53H	ONWAAR
XXX	XYZ	ONWAAR

- ◆ alphabetic test : Onderzoekt of een veld uitsluitend alfabetische tekens bevat, dat betekent hoofdletters, kleine letters en spaties.
- ◆ Alphabetic-lower test : Onderzoekt of een veld uitsluitend kleine letters en spaties bevat.
- ◆ Alphabetic-upper test : Onderzoekt of een veld uitsluitend hoofdletters en spaties bevat.

Vb.

```

01 A  PIC XXX  value "AB ".
01 B  PIC XXX  value "ab ".
01 C  PIC XXX  value "AB4".

```

```

If A alphabetic ...           waar
If A not alphabetic-lower    waar
If A alphabetic-upper        waar
If B alphabetic               waar
If B alphabetic-upper        onwaar
If C alphabetic               onwaar
If C not alphabetic -upper   waar

```

Controle invoer:

Perform with test after until naam is alphabetic

Accept naam ...

End-perform

➤ **De waarde- of tekenconditie** p 75

Bepaalt of de rekenkundige waarde van de operand kleiner dan (<), gelijk aan (=) of groter dan (>) nul (0) is.

Arithmetic-expression-1 IS $\left(\underline{\text{NOT}} \right) \left\{ \begin{array}{l} \underline{\text{POSITIVE}} \\ \underline{\text{NEGATIVE}} \\ \underline{\text{ZERO}} \end{array} \right\}$

Vb. (1)

01 A PIC S999

if A positive if A > 0

Vb. (2)

01 A PIC 999V99 value 137.35.

01 B PIC 99 value 13.

01 C PIC 999 value 300.

If A positive waar

If A negative onwaar

If A not negative waar

If A not zero waar

If B – C positive onwaar

➤ **De voorwaardenaamconditie** p 76

Is een zeer goed hulpmiddel om de duidelijkheid te verhogen; zoveel mogelijk toe te passen

- invoeren voorwaardenaamen

88 condition-name-1 $\left\{ \begin{array}{l} \underline{\text{VALUE IS}} \\ \underline{\text{VALUES ARE}} \end{array} \right\} \left\{ \text{literal-1} \left[\begin{array}{l} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{array} \right] \text{literal-2} \dots \right\}$

Vb 1.

01 landcode PIC 999.

88 België value 13.

88 Benelux values 13 thru 15.

88 EG values 13 thru 17, 18, 21 thru 29.

88 Europa values 1 thru 49.

88 Nato values 11 thru 29, 51.

ACCEPT landcode ← invoer-controle uit EG

```
PERFORM with test before until EG
  DISPLAY "FOUT"
  ACCEPT landcode
END-PERFORM
```

Vb 2.

01 artikelnummer

```
88 schrijfwaarden values "000" thru "149".
88 meubilair       values "251" thru "799".
88 verlichting     values "800" thru "999".
88 diversen        values "150" thru "250", "950" thru "999".
```

03 nummer PIC 99.

03 kleur PIC 9.

```
88 wit           value 0.
88 grijs         value 1.
88 zwart         value 2.
88 bruin         value 3
88 oranje        value 9.
```

401 → grijs meubilair

Vb 3.

01 antwoord PIC X.

```
88 goed-antwoord values "J", "j", "N", "n".
```

invoer-controle.

```
ACCEPT antwoord...
PERFORM with test before
  Until goed-antwoord
  DISPLAY "Foutieve invoer".
ACCEPT antwoord.
END-PERFORM.
```

- De **SET**-opdracht p. 61

```
SET {condition-name-1} ... TO TRUE
```

03 studiejaar PIC 9.

```
88 goed-studiejaar values 1, 2, 3.
88 eerste value 1.
88 tweede value 2.
88 derde value 3.
```

...

move 2 to studiejaar \Leftrightarrow set tweede to true
set goed-studiejaar to true \Leftrightarrow move 1 to studiejaar (1 is de eerste in de rij)

➤ **De negatieve conditie**

if NOT A=2

➤ **De samengestelde conditie** AND OR

if A = B AND C > 2

➤ **De afgekorte relatieconditie**

Vb1: if A = B AND A > C
=> if A = B AND > C

Vb2: if A = B AND A = C
=> if A = B OR C

Vb3: if A > B OR A = C OR A = D
=> if A > B OR = C OR D

1.1.4 De rekenkundige opdrachten p 67 - 73

- ◆ Prioriteitsregels p 81 zoals in de wiskunde: eerst enkelvoudige operatoren (= hoogste prioriteit), dan machtsverheffing, dan vermenigvuldiging en deling, dan tenslotte optelling en aftrekking (= laagste prioriteit).
- ◆ Om af te wijken van deze regels kan je haakjes gebruiken. Bij geneste haakjes worden de binnenste haakjes eerst uitgewerkt.
- ◆ Spatieregels bij rekenkundige operatoren: literals en identifiers moeten worden **voorafgegaan en gevolgd** door een spatie of een haakje (een constante zelf kan een + of een – bevatten ; dan mag er geen spatie tussen).
- ◆ Spatieregels voor haakjes: haakjes mogen worden gevolgd en worden voorafgegaan door een spatie.

1.1.4.1 COMPUTE – instructie p 67

- ◆ Deze instructie geeft de mogelijkheid om de waarde van een rekenkundige uitdrukking aan één of meer velden toe te kennen.
- ◆ Formaat :

COMPUTE {identificer-1 [ROUNDED]}...

$\left\{ \begin{array}{l} \text{EQUAL} \\ = \end{array} \right\}$ arithmetic-expression

[ON SIZE ERROR imperative-statement-1]

[NOT ON SIZE ERROR imperative-statement-2]

[END-COMPUTE]

Vb.

COMPUTE A = 2 * I		2 * I → A
COMPUTE A B = I ** 2		I ** 2 → A en B
COMPUTE A = 1	zelfde als	MOVE 1 TO A

- ◆ Nadeel van COMPUTE: de programmeur heeft onvoldoende controle op de tussenresultaten zowel wat betreft de grootte daarvan als wat betreft de afkapping.

Vb.

A = 7.6752 B = 3.12 C = 22.25

En het resultaatveld (D) kan alleen een geheel getal bevatten.

COMPUTE D = (C * (A / B)) / 2 A / B = 7.6752 / 3.12 = 2.46

- Indien 2.46 volledig wordt meegenomen dan $C * 2.46 = 22.25 * 2.46 = 54.735$
 $54.735 / 2 = 27.3675 \rightarrow \underline{27}$
- Indien 2.46 wordt afgerond tot 2.5 dan $C * 2.5 = 22.25 * 2.5 = 55.625$
 $55.625 / 2 = 27.8125 \rightarrow \underline{\text{afgerond 28 anders 27}}$

- Indien 2.46 wordt afgekapt tot 2.4 dan $C * 2.4 = 22.25 * 2.4 = 53.4$
 $53.4 / 2 = 26.7 \rightarrow$ **afgerond 27 anders 26**
- Indien 2.46 wordt afgekapt tot 2 dan $C * 2 = 22.25 * 2 = 44.5$
 $44.5 / 2 = 22.25 \rightarrow$ **22**
- ◆ Voordeel: ingewikkelde formules worden in één maal genoteerd, anders gebeurt dit d.m.v. een lange reeks instructies

1.1.4.2 ADD – instructie p 68

- ◆ Enkele algemene voorschriften voor **alle** rekenkundige bewerkingen:
 - ◆ Alle velden waarmee rekenkundige bewerkingen gebeuren moeten elementaire velden zijn.
 - ◆ Alle gebruikte velden en constanten moeten van het numerieke type zijn.
 - ◆ Het resultaatveld kan nooit een constante waarde zijn.
 - ◆ Geen enkel getal, resultaat of tussenresultaat mag meer dan 18 cijfers bevatten
- ◆ Formaat 1

$$\text{ADD } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \dots \text{ TO } \{\text{identifier-2}[\text{ROUNDED}]\} \dots$$

[ON SIZE ERROR imperative statement-1]

[NOT ON SIZE ERROR imperative statement-2]

[END-ADD]

- ◆ Formaat 2

$$\text{ADD } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \dots \text{ TO}$$

$$\left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\} \text{ GIVING } \{\text{identifier-3} [\text{ROUNDED}]\} \dots$$

[ON SIZE ERROR imperative statement-1]

[NOT ON SIZE ERROR imperative statement-2]

[END-ADD]

- ◆ Bij gebruik van het eerste formaat worden alle velden en/of constanten vóór het woord TO bij elkaar opgeteld. De aldus verkregen som wordt opgeteld bij elk van de velden na het woord TO.

Vb.

ADD A TO B	$A + B \rightarrow B \quad (B = A + B)$
ADD A B TO C	$A + B + C \rightarrow C$
ADD A B TO C D	$A + B + C \rightarrow C$ en $A + B + D \rightarrow D$
ADD A TO B C	$A + B \rightarrow B$ en $A + C \rightarrow C$
ADD A C TO C	$A + C + C \rightarrow C$

- ◆ Bij gebruik van het tweede formaat worden alle velden en/of constanten vóór het woord GIVING bij elkaar geteld. De som komt in het (de) veld(en) na het woordje GIVING.

Vb.

```
ADD A B GIVING C           A + B → C
ADD A 2 B GIVING C D      A + 2 + B → C   A + 2 + B → D
```

- ◆ ROUNDED-optie p 70

Wanneer na het resultaatveld ROUNDED is beschreven, dan wordt dit veld — als het resultaatveld meer cijfers bevat dan het resultaatveld mag bevatten — afgerond. Zonder ROUNDED wordt afgekapt

Vb.

```
Als  02 A  PIC 99V99  VALUE 12.68.
      02 B  PIC 99V99  VALUE 23.18.
      02 C  PIC 99V9   VALUE 0.
      02 D  PIC 99V9   VALUE 0.
```

```
ADD A B TO C           C BEVAT 35.8
ADD A B TO C ROUNDED  C BEVAT 35.9
ADD A B TO C D ROUNDED C BEVAT 35.8 D BEVAT 35.9
```

- ◆ ON SIZE ERROR – optie p 72

- ◆ Indien het resultaat meer cijfers voor de komma bevat dan het resultaatveld kan bevatten, noemt men dit overloop of ‘overflow’ of ‘size error’. Bij ON SIZE ERROR-optie wordt bij overloop het imperatieve statement uitgevoerd.

- ◆ Wanneer er meer dan 1 resultaatveld is genoemd, dan worden eerst de andere resultaten afgewerkt voordat het imperatieve statement wordt uitgevoerd.
- ◆ Indien er overloop is en de ON SIZE ERROR – optie is niet gespecificeerd, dan is de inhoud van elk veld waar overloop voorkomt onvoorspelbaar
- ◆ Indien er overloop is en de ON SIZE ERROR is wel gespecificeerd, dan blijft het veld ongewijzigd.

Vb.

```
02 A  PIC 99V99  VALUE 88.16.
02 B  PIC 99V99  VALUE 12.38.
02 C  PIC 99V99.
```

```
ADD A B GIVING C ON SIZE ERROR PERFORM FOUT
```

De som van A en B (100.54) is te groot voor veld C en het programma zal procedure fout uitvoeren.

```
02 A  PIC 999    VALUE 876.
02 B  PIC 999    VALUE 150.
02 C  PIC 999    VALUE 0.
02 D  PIC 9999   VALUE 0.
```

```
ADD A B TO C           C = onbepaald
ADD A B TO C ON SIZE ERROR PERFORM FOUT C = 000
```

ADD A B TO C D SIZE ERROR PERFORM FOUT

C = 000 D = 1026

1.1.4.3 SUBTRACT – instructie p 68

◆ Formaat 1:

SUBTRACT $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \dots \text{FROM } \{ \text{identifier-2 [ROUNDED]} \} \dots$

[ON SIZE ERROR imperative-statement-1]

[NOT ON SIZE ERROR imperative-statement-2]

[END-SUBTRACT]

Alle velden en/of constanten vóór FROM worden bij elkaar opgeteld; deze som wordt afgetrokken van het veld na FROM en het resultaat wordt overgebracht naar elk veld na GIVING.

Vb.

SUBTRACT A FROM B	$B - A \rightarrow B$	
SUBTRACT A B FROM C	$C - (A + B) \rightarrow C$	
SUBTRACT A B FROM C D	$C - (A + B) \rightarrow C$	$D - (A + B) \rightarrow D$
SUBTRACT A FROM B C	$B - C \rightarrow B$	$C - A \rightarrow C$

◆ Formaat 2

SUBTRACT $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \dots \text{FROM } \left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\}$

GIVING $\{ \text{identifier-3 [ROUNDED]} \} \dots$

[ON SIZE ERROR imperative-statement-1]

[NOT ON SIZE ERROR imperative-statement-2]

[END-SUBTRACT]

Alle velden en/of constanten vóór FROM worden bij elkaar opgeteld; deze som wordt afgetrokken van het veld na FROM en het resultaat wordt overgebracht naar elk veld na GIVING.

Vb.

SUBTRACT A FROM B GIVING C	$B - A \rightarrow C$
SUBTRACT A BB FROM C GIVING D	$C - (A + B) \rightarrow D$

[NOT ON SIZE ERROR imperative-statement-2]

[END-DIVIDE]

De eerste operand wordt gedeeld op elk veld van de operands na INTO waarnaar het resultaat wordt overgebracht

Vb.

DIVIDE A INTO B C	$B / A \rightarrow B$	$C / A \rightarrow C$
DIVIDE 100 INTO A	$A / 100 \rightarrow A$	

◆ Formaat 2

DIVIDE { identifier-1 } { INTO } { identifier-2 }
 { literal-1 } { BY } { literal-2 }

GIVING identifier-3 [ROUNDED]...

[ON SIZE ERROR imperative-statement-1]

[NOT ON SIZE ERROR imperative-statement-2]

[END-DIVIDE]

De eerste operand wordt gedeeld door (BY) of gedeeld op (INTO) de tweede operand , waarbij beide operands ongewijzigd blijven; het resultaat wordt overgebracht naar elk veld na het woord GIVING.

Vb.

DIVIDE A BY B GIVING C D	$A / B \rightarrow C$	$A / B \rightarrow D$
DIVIDE A INTO B GIVING C	$B / A \rightarrow C$	
DIVIDE 20 BY A GIVING C	$20 / A \rightarrow C$	
DIVIDE 20 INTO A GIVING C	$A / 20 \rightarrow C$	

◆ Formaat 3

DIVIDE { identifier-1 } { INTO } { identifier-2 }
 { literal-1 } { BY } { literal-2 }

GIVING identifier-3 [ROUNDED]...

REMAINDER identifier-4

[ON SIZE ERROR imperative-statement-1]

[NOT ON SIZE ERROR imperative-statement-2]

[END-DIVIDE]

Dit derde formaat doet hetzelfde als het tweede formaat; bovendien wordt de rest van de deling overgebracht naar identifier-4 en dit als volgt:

Rest = A - Q * B waarbij A = deeltal B = deler Q = quotiënt voor ev. afronding

Na de restbepaling wordt het quotiënt (eventueel afgerond) overgebracht naar identifier-3.

Opm.

- Bij DIVIDE ... BY ... moet altijd GIVING gebruikt worden.
- Bij REMAINDER moet steeds GIVING aanwezig zijn.
- Bij 3^e formaat mag slechts 1 resultaatveld staan, want er is maar 1 restveld. Het restveld mag **niet** ROUNDED zijn.

Vb.

DIVIDE A BY B GIVING C REMAINDER R als A = 40 en B = 16

→ C = 2 en R = 8

DIVIDE A INTO B GIVING C REMAINDER R als A = 40 en B = 16

→ C = 0 en R = 16

Het resultaat van een deling zal naar een hulpveld gebracht worden (ontoegankelijk voor programmeur), van waaruit de resultaatvelden en het restveld wordt gevuld. De inhoud van dat hulpveld noemt men tussenresultaat.

Hulpveld is nodig:

- als er meer dan 1 resultaatveld is
- bij afronding
- bij restbepaling
- Specificatie van meer dan 1 resultaatveld

De lengte van het hulpveld telt voor de decimale komma zoveel tekens als het resultaatveld met de meeste posities; na de decimale komma zoveel als het resultaatveld met meeste posities.

Vb.

02 A PIC 9(6)V99 VALUE 12694.08.

02 B PIC 9(4)V9 VALUE 600.

02 C PIC 9(5).

02 D PIC 9(4)V99.

02 E PIC 9(3)V999.

02 F PIC 9V9.

DIVIDE A BY B GIVING C D E hulpveld 9(5)V9(5) → 00021.156

C → 00021 D → 0021.15 E → 021.156

- Afronding p 70

Opgave: Na invoer van twee willekeurige getallen wordt hierop een bewerking uitgevoerd en het resultaat hiervan op het scherm getoond. De mogelijke bewerkingen worden beschreven in de scherm-layout.

HOGESCHOOL GENT		<datum>
BASISBEWERKINGEN		
GETAL1 : ____ GETAL2 : ____	KEUZE: _	1. OPTELLEN 2. VERSCHIL 3. PRODUKT 4. DELING 5. DELING MET REST
RESULTAAT zonder COMPUTE: RESULTAAT met COMPUTE :		
VERDER (J / N): _		
ARENTS CHRIS		OEFENING 8

Experimenteer met dit programma als volgt:

- Gehele getallen
- Decimale getallen
- Te kleine resultaatvelden
- Rounded
- On size error
- CONTROLEER DELING DOOR NUL !!!!!

