

บทที่ 7 : Calculation Specification Form

IBM

International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic
Punch	

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Comments
			And	Or	Not						
1	C	7									
2	C	7									

C-Form เป็นฟอร์มที่อยู่ในลำดับก่อน Output Form แต่ที่นำมากล่าวที่หลังเพราะเนื้อหามากกว่า จุดประสงค์ของแบบฟอร์มนี้ก็เพื่อที่จะเป็นแบบที่ใช้เขียนคำสั่งในการประมวลผลที่เกี่ยวกับการคำนวณทางคณิตศาสตร์ การใช้หลักทางตรรกวิทยาและอื่น ๆ ที่ต้องการให้ประมวลผลจากข้อมูลใน

สิ่งที่ควรระวังในการใช้เรื่องที่เกี่ยวข้องกับการดำเนินการแบบต่าง ๆ ก็คือ ในกรณีที่ใช้การคำนวณทางคณิตศาสตร์ เช่นใช้เครื่องหมาย $+$, $-$, \times , \div นั้นจะต้องมีการตรวจสอบว่า data field ที่จะนำมาปฏิบัติการนั้นเป็นชนิด Numeric Data มิฉะนั้นจะทำให้เกิดความผิดพลาดขึ้นได้

แบบฟอร์มของ C-Form นี้แบ่งออกเป็นพื้นที่ย่อย ๆ ได้ตามหน้าที่ดังนี้คือ

Form Type: (สดมภ์ที่ 6) ใช้อักษร C เพื่อบ่งว่าเป็น Calculation Type (ดังนั้นจึงต้องมีตัว C ในทุกคำสั่งที่อยู่ใน C-Form)

Control Level: (สดมภ์ที่ 7-8) ใช้เพื่อเป็น Indicator ชนิดหนึ่งใช้ในการควบคุมการทำงานในการปฏิบัติการต่าง ๆ ในคำสั่งนั้น ๆ

Factor 1-and Factor 2: (สดมภ์ที่ 18-27 และสดมภ์ที่ 33-42)

ใช้เป็นชื่อของข้อมูล (data name) ที่จะนำมาประมวลผลภายใต้เงื่อนไขการปฏิบัติการต่าง ๆ โดยที่ข้อมูลที่นำมาใช้นั้นจะเป็นข้อมูลที่ได้มาจาก I-Form หรือได้จากการดำเนินงานใน C-Form

Result Field: (สดมภ์ที่ 43-48)

เป็นชื่อของข้อมูลที่เกิดขึ้นจากการปฏิบัติการกับ Factor 1 และ (หรือ) Factor 2

Field Length: (สดมภ์ที่ 49-51) คือขนาดความยาวของ Result Field

Decimal Positions: (สคมภที่ 52) เพื่อป้องกันข้อมูลใน Result Field มีจำนวนทศนิยมที่ตำแหน่ง (จำนวนสูงสุดคือ 9 ตำแหน่งที่สามารถใช้ได้) ในสคมภนี้

Resulting Indicator: (สคมภที่ 54-59) แบ่งออกเป็น 3 ส่วนย่อย ๆ คือ สคมภที่ 54-55 สคมภที่ 56-57 สคมภที่ 58-59 พื้นที่ย่อยดังกล่าวมีความหมายคือใช้ในพื้นที่ในการสร้าง Indicator (เป็นตัวเลข ตั้งแต่ 01-99) ซึ่งสามารถสร้างได้ถึง 3 ตัวภายใต้เงื่อนไขของการปฏิบัติการแต่ละแบบ หลักการสร้าง Indicators ที่เกิดขึ้นในส่วนนี้ไม่ควรจะไปซ้ำซ้อนกับตัวเลขที่เป็น Indicators ที่เกิดมาก่อนหน้านั้นทั้งนี้เพราะอาจจะทำให้การใช้ Indicators เกิดความสับสนขึ้นได้ การสร้างให้ Indicators เพื่อให้ On นั้นสามารถสร้างได้ในหลายสถานะการณ์ตามเงื่อนไขของการปฏิบัติการ เช่น

1. การใช้ Arithmetic โดยที่ค่าของ Result Field จะถูกตรวจสอบว่าเป็นค่าบวก หรือค่าลบ หรือ ค่าเป็นศูนย์
2. การใช้ Compare จะใช้แค่ Factor 1 และ Factor 2 แต่ไม่มี Result Field เกิดขึ้น และค่าที่จะเปรียบเทียบนั้นจะเป็นไปตามเงื่อนไขดังนี้คือ High ($1 > 2$) หรือ low ($1 < 2$) หรือ Equal ($1 = 2$)
3. การใช้ Look up ใช้ในกรณีของ Table เพื่อที่จะตรวจสอบ Factor 2 โดยมีเงื่อนไขว่าสิ่งต้องการตรวจสอบนั้นตรงกับเงื่อนไขข้อใดคือ high, low หรือ equal

Comment: (สคมภที่ 60-74)

เป็นส่วนที่ไม่เกี่ยวข้องกับคำสั่งในการประมวลผล แต่เป็นส่วนที่จะช่วยให้ผู้อ่านคำสั่งเข้าใจในลำดับของการทำงานของแต่ละขั้นตอนในโปรแกรม

ข้อสังเกต

1. ในกรณีที่ Result Field เป็นชื่อของข้อมูลที่เกิดขึ้นใหม่ใน C-Form ให้ยึดหลักการตั้งชื่อเช่นเดียวกับการตั้งชื่อใน Field Name ใน I-Form ทุกประการ
2. เมื่อมีการใช้ Result Field ซ้ำกันมากกว่า 1 ครั้ง เราต้องยึดขนาดความยาวและจำนวนทศนิยมของ result Field ที่เกิดขึ้นครั้งแรกเป็นเกณฑ์ จะไปกำหนดขนาดของ Result Field ใหม่ในครั้งที่ 2 หรือครั้งต่อ ๆ ไปไม่ได้
3. Result Field ที่ได้มาจากการคำนวณทางคณิตศาสตร์นั้น จะต้องเป็น Numeric เสมอ นั่นหมายความว่าในส่วนของ Decimal Positions ต้องกำหนดตัวเลข 0-9 ตามที่ตรงกับวัตถุประสงค์ที่ต้องการลงไป (ถึงจะเป็นจำนวนเต็มก็ต้องใส่ ศูนย์ลงไป) ในสคมภที่ 52

ตัวอย่างของการใช้ค่าคงที่ในการคำนวณ

	Factor 1	Operation	Factor 2	Result field	Field Length
FLDA	ADD	FLDB	FLDE	60	
FLDD	ADD	FLLB	FLDE	61	
FLDA	ADD	FLDC	FLDE	71H	
FLDA	ADD	FLDD	FLDE	51H	
FLDB	ADD	FLDA	FLDE	60	
1	ADD	FLDG	FLDG	82H	
1	ADD	FLDC	FLDF	32	

การใช้ Half Adjust (สคมภ์ที่ 53)

เมื่อต้องการการปรับพิเศษใน Result Field อันเกิดจากผลของการปฏิบัติการของ Factor 1 และ Factor 2 เราทำได้โดยการใช้ Half Adjust เข้าช่วยดังนี้ เช่นกำหนดให้

$$FLDA = 137.35$$

$$FLDB = 467.60$$

ดังนั้นโดยคำสั่งต่อไปนี้

$$C \quad FLAD \quad ADD \quad FLDB \quad FLDC \quad 61$$

ดังนั้นค่าของ FLDC = 604.95 แต่ใน format ของ Result Field กำหนดให้ FLDC ยาว 6 ตำแหน่ง มีทศนิยม 1 ตำแหน่ง ดังนั้น FLDC จะมีค่าเท่ากับ 604.9 แต่ถ้าใช้คำสั่ง

$$FLAD \quad ADD \quad FLDB \quad FLDC \quad 61H$$

ผลที่ได้จะเป็น

$$\begin{array}{r}
 FLDA \quad 134.35 \\
 FLDB \quad 467.60 \\
 \hline
 FLDC \quad 604.95
 \end{array}$$

5 บวกที่ตำแหน่งทศนิยม โดยคำสั่งของ Half Adjust

$$605.00$$

ดังนั้นค่าใน Result Field ที่ได้คือ 605.0

เพื่อความเข้าใจยิ่งขึ้นในการใช้ half adjust จากตัวอย่างต่อไปนี้ (ใช้จำนวนทศนิยม 2 ตำแหน่ง)

A)	1.6234	B)	1.6243	C)	1.460
	<u>4.0252</u>		<u>4.0202</u>		<u>5.114</u>
	5.6486		5.6445		6.574
	<u>5</u>		<u>5</u>		<u>5</u>
	<u>5.6591</u>		<u>5.6450</u>		<u>6.579</u>
ผลที่ได้	<u><u>5.65</u></u>		<u><u>5.64</u></u>		<u><u>6.57</u></u>

ในตัวอย่าง B และ C ค่าที่ไม่ปัดเศษและค่าที่ได้จากการปัดเศษจะมีผลเท่ากัน

กฎเกณฑ์ของการใช้ Arithmetic Operation

Arithmetic Operation สามารถใช้ได้กับ field, arrays, array element, table element หรือ literal แต่ต้องเป็น Numeric ความยาวของ Factor 1 และ Factor 2 จะต้องไม่เกิน 15 ตำแหน่ง

Arithmetic Operation ต่าง ๆ ที่ใช้กันในภาษา RPG

Add (ADD): คำสั่งที่ใช้กับการบวก

ลักษณะของคำสั่งที่ใช้กับการบวก

Indicator	Factor 1	Operations	Factor 2	Result Field	Resulting Indicators
optional	required	ADD	required	required	optional

ความหมาย : optional = มีหรือไม่มีก็ได้

: required = จำเป็นจะต้องมี

ตัวอย่าง

FLDA ADD FLDB FLDC 61

แบบฝึกหัด

IBM

International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

Line	C	Form Type Control Level (L, B, L, R, S, K)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Resulting Indicators	Comments
			And	And	Not							
01	C					F T O A	T A B					
02	C					G R O S S	M U L T					
03	C					V A R I O A	A D D					
04	C					H O L D	C O M P					
05	C						S D T O					
06	C					V A S O O	S U B					
07	C					A D F T O A	T A B					
08	C					Y D F T O A	A D D					
09	C						(A D D I T I O N A L C A L C U L A T I O N)					
10	C					G R S P A Y	T A B					
11	C					H O U R S	C O M P					
12	C					H O U R S	S U B					
13	C					R A T E	M U L T					
14	C					S A V E	A D D					
15	C					R A T E	M U L T					
	C					S A V E	A D D					
	C						C O T O F I C H					
	C						(A D D I T I O N A L C A L C U L A T I O N)					
	C					M A T A X	T A B					
	C						(A D D I T I O N A L C A L C U L A T I O N)					

จากคำสั่งต่อไปนี้ใน C-Form จงตรวจดูคำสั่งว่า คำสั่งใดไม่ถูกต้องหรือถูกต้อง ถ้าไม่ถูกต้องให้แก้ไขให้ถูกต้อง

Zero and ADD (Z-ADD)

ลักษณะการเขียนคำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
optional	blank	Z-ADD	required	required	optional

กฎเกณฑ์การใช้คำสั่ง SUB

- Factor 2 จะถูกลบออกจาก Factor 1
- ผลที่ได้จาก 1. จะถูกเก็บไว้ใน Result Field
- คำสั่งปฏิบัติการที่ใช้คือ SUB

ตัวอย่างของการเขียนคำสั่ง

Net Income = Income - Tax

(NET) = (IN) - (TAX)

เขียนเป็นคำสั่งในภาษา RPG ได้ดังนี้

1 01 IN SUB TAX NET 52

อาจใช้ Factor 2 เป็น literal ก็ได้

IBM

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 2

Program Identification 75 76 77 78 79 80

Line	From To Type Control (L, O, B, S, R)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Position Half Adjust (H)	Resulting Indicators Arithmetic Plus Minus Zero Compare High Low Equal 1 > 2 1 < 2 1 = 2 Lookup Table (Factor 2) is High Low Equal	Comments
		And	And	And								
01	C				A	Sub	1.5	5	3			

ความหมายของ B ก็คือ เหนือค่าคงที่ 1.5 ไปลบออกจากข้อมูลใน A ผลที่ได้นำไปเก็บที่ B ซึ่งมีถึง 3 ตำแหน่ง (เป็นทศนิยม 1 ตำแหน่ง)

Z-SUB (Zero and Sub)

ความหมายของคำสั่งนี้ก็คือวิธีการที่คล้ายคลึงกันกับ Z-ADD เพียงแต่ว่า Z-SUB จะทำให้ Result Field ว่างเสียก่อนแล้วจึงนำค่าใน Factor 2 ไปลบออกจาก Result Field ซึ่งความหมายของการใช้คำสั่งนี้ก็เหมือนกับการให้ Result Field มีค่าเท่ากับค่าใน Factor 2 แต่เครื่องหมายจะตรงกันข้ามกันเท่านั้น เราสามารถใช้ literal แทนใน Factor 2 ได้ ดังลักษณะของ Z-ADD ลักษณะการใช้คำสั่ง

Indicator Factor 1 Operation Factor 2 Result Resulting
Field Indicators

optional blank Z-SUB required requird optional

ตัวอย่างเช่น

C	01	Z-SUBA	B	52H	(A)
C	01	Z-SUB-1	C	52H	(B)
C	01	Z-SUB1.25	D	52H	(C)
C	01	Z-SUB-12.498	E	52H	(D)

คำสั่ง ผลที่ได้ (A) คือ B ค่าเท่ากับ A แต่มีเครื่องหมายตรงกันข้าม โดยที่ A มีความยาว 5 ตำแหน่ง เป็นทศนิยมเสี้ยว 2 ตำแหน่ง โดยมีการปัดเศษด้วย

ผลที่ได้ (B) คือ C จะมีค่าเป็น 001.00⁺

ผลที่ได้ (C) คือ D จะมีค่าเป็น 001.25⁺

ผลที่ได้ (D) คือ E จะมีค่าเป็น 012.50⁺

MULT: (Multiplication)

ลักษณะของการใช้คำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
optional	required	MULT	required	required	optional

กฎเกณฑ์ในการใช้คำสั่งคูณ

1. คำสั่งที่ใช้ในการคูณคือ MULT
2. เวลาใส่ Factor 1 และ Factor 2 เพื่อความรวดเร็วในการทำงานควรเลือก Factor ที่มีความยาวสั้นที่สุด ระหว่าง 2 Factor มาเป็น Factor 1
3. ผลที่ได้จาก 2. จะถูกเก็บไว้ใน Result Field

ตัวอย่างการใช้คำสั่งใน C-Form

C	10	UNIT	MULTQUAN	SALE	122 H
---	----	------	----------	------	-------

DIV : (Division)

ลักษณะการเขียนคำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
optional	required	DIV	required	required	optional

กฎเกณฑ์ของการใช้คำสั่ง DIV

1. คำสั่งที่ใช้ในการหาร DIV
2. Factor 1 จะเป็นตัวตั้งและ Factor 2 จะเป็นตัวหาร
3. ผลหารที่ได้จาก 2 จะถูกเก็บไว้ใน Result Field

ตัวอย่างของการใช้คำสั่งในการหารเช่น

$$\text{Average} = \text{Amount} \div \text{Volume}$$

$$(AV) = (AMO) (VOL)$$

ซึ่งจะเขียนได้ใน Coding Form ได้ดังนี้

01	AMO	DIV	VOL	AV	43H
----	-----	-----	-----	----	-----

Factor 1 หรือ Factor 2 เราอาจใช้เป็น literal ก็ได้เช่น

01	INC	DIV	12	FOOT	40H	ⓑ
----	-----	-----	----	------	-----	---

คำสั่งⓑก็เปรียบเสมือนกับการเปลี่ยนหน่วยจากนิ้วมาเป็นฟุต โดยการนำเลข 12 ไปหารจำนวนนิ้ว เพื่อความเข้าใจในการใช้คำสั่งนี้ในการคำนวณ จะขอยกตัวอย่างซึ่งเกี่ยวข้องกับการใช้คำสั่งหลาย ๆ คำสั่งรวมกันในการทำงาน ซึ่งประกอบด้วยหลายขั้นตอน ตัวอย่างเช่นการหา average-cost ซึ่งสามารถหาได้จากสูตรดังนี้คือ

$$\text{average cost} = ((\text{total material} + \text{total labor}) + (\text{total material} + \text{total labor}) \times \text{factory over head} + (\text{total material} + \text{total labor}) \times \text{office over head}) \div \text{quantity product}$$

ถ้าหากค่า factory over head = 23 %

และ office over head = 11 %

สูตรที่จะใช้ในการหาค่า average cost สามารถแยกเป็นคำสั่งย่อย ๆ เพื่อการเขียนโปรแกรมได้ดังนี้

(1) total material + total labor

(2) นำคำตอบจาก 1. คูณด้วย 23% factory over head

(3) นำคำตอบจาก 1. คูณด้วย 11% office overhead

(4) นำ (1) + (2) + (3)

(5) นำคำตอบจาก (4) ÷ quantity product = average cost

จากคำสั่ง (1) ถึง (5) นำมาเขียนคำสั่งใน C-Form ได้ดังนี้คือ

C	MATER	ADD LABOR	SUBT1	62
C	SUBT1	MULT.23	FACOH	62H
C	SUBT1	MULT.11	OFFOH	62H
C	SUBT1	MULTFACOH	SUBT2	82
C	SUBT2	ADD OFFOH	TOTAL	82
C	TOTAL	ADD QUANT	AVGCST	62H

ข้อสังเกต ในการใช้คำสั่ง DIV ในกรณีที่มี Factor 1 มีค่าเป็นศูนย์จะทำให้ Result Field มีค่าเป็นศูนย์ แต่ถ้าหาก factor 2 มีค่าเป็นศูนย์จะเกิด error ขึ้น และในกรณีที่ Factor 1 ทหารด้วย Factor 2 ไม่ลงตัว เศษที่เหลือจากการหารจะไปอยู่ใน Result Field (Result Field จะมีค่าตามจำนวนที่บ่งไว้ เช่นมีทศนิยม 1 ตัว หรือ 2 ตัวเป็นต้น) และถ้ามีการใช้ half adjust ด้วยเศษที่เหลือจะเป็นไปตามกฎที่ได้กล่าวมาแล้ว

การคิดจำนวนตำแหน่งของตัวตั้ง Factor 1 ตัวหาร Factor 2 และ Result Field จะต้องเป็นไปตามกฎดังนี้คือ

$$L_1 + (D_2 - D + D_r) \leq 15 \quad \dots(1)$$

$$L_2 - (D_2 - D_r + D_r) \leq 15 \quad \dots(2)$$

โดยที่

L = ขนาดความยาวของ Factor 1

L_2 = ขนาดความยาวของ Factor 2

D_1 = จำนวนทศนิยมใน Factor 1

D_2 = จำนวนทศนิยมใน Factor 2

D_r = จำนวนทศนิยมใน Result Field

แต่ถ้ามีการใช้ half adjust ด้วย เงื่อนไขที่ 1 จะเปลี่ยนไปเป็น

$$L_1 + (D_2 - D_1 + D_r) \leq 14$$

ดังที่ได้กล่าวมาแล้วว่า ถ้าหาก Factor 1 ทหารด้วย Factor 2 ไม่ลงตัว (ตามทศนิยมที่ปัดไว้แล้ว) ถ้าหากว่าเราต้องการนำเศษเหลือไปใช้งานอีก เราสามารถจะนำเศษเหลือไปเก็บไว้ในที่ที่ต้องการได้ โดยวิธีการใช้คำสั่งต่อไปนี้เข้าช่วย คำสั่งดังกล่าวคือ MVR

MVR : (Move Remainder)

มีลักษณะของการใช้คำสั่งดังนี้คือ

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
optional	blank	MVR	blank	required	optional

คำสั่ง MVR จะทำให้เศษเหลือที่อยู่ใน work area ที่ได้จากคำสั่งในการหารย้ายไปยัง Result Field ที่ปัดไว้ ดังนั้นคำสั่ง MVR ต้องเป็นคำสั่งในลำดับที่ถัดมาจากคำสั่งที่มีการหารเกิดขึ้นก่อนแล้ว

ตัวอย่างเช่น

C	10	AB	DIV F	LM	50
C			MVR	HOL	30

จะเห็นว่าในคำสั่งที่ 10 เราใช้ field AB แล้วหารด้วย F โดยที่ผลหารจะเกิดเป็น LM (ผลหารมี 5 ตำแหน่ง ไม่มีทศนิยม) คำสั่งในคำสั่งที่ 20 ถัดมามีความหมายว่าถ้าในคำสั่งที่ 10 มีเศษเหลือก็ให้นำเศษเหลือนั้นมาเก็บไว้ใน Result Field ที่มีชื่อว่า HOL

การกำหนดความยาวของ Result Field ที่จะใช้ในคำสั่ง MVR ให้ยึดหลักว่า การกำหนด

จำนวนทศนิยมที่จะใช้ใน Result Field เท่ากับผลบวกของ $(D_2 + D_r)$ หรือ D_1 ichever ค่าใดใหญ่กว่าให้ใช้
 ค่านั้น อาจจะสรุปเป็นหลักเกณฑ์ได้ดังนี้

$$\begin{aligned} &\text{จำนวนทศนิยมใน Result Field ที่เกิดจาก MVR} \\ &= \text{MAX} \{ (D_2 + D_r), D_1 \} \end{aligned}$$

โดยที่ D_1 = จำนวนทศนิยมใน Factor 1 (เกิดจากคำสั่ง DIV ในลำดับก่อนคำสั่ง MVR)

และ D_r = จำนวนทศนิยมใน Result Field ที่เกิดจากคำสั่ง DIV
 ตัวอย่างการใช้ MVR ในการคำนวณ

กำหนดให้ Factor 1 = 11 ($\therefore D_1 = 0$)

Factor 2 = .76 ($\therefore D_2 = 2$)

Factor 1 \div Factor 2

$$11 \div .76$$

$$.76 \overline{) 11.0000} = 14.47... \text{ Result Field } (\therefore D_r = 2)$$

76

340

304

360

304

560

532

28

= Remainder

จำนวนทศนิยมใน Result Field ในคำสั่ง MVR

$$\text{Max} \{ (2 + 2), 0 \} = 4$$

สำหรับค่าของเศษเหลือที่จะนำไปเก็บที่ Result Field ภายใต้อำสั่ง MVR มีวิธีคิดดังนี้คือ (ใช้ตัวอย่างข้างบนเป็นหลัก)

$$\begin{array}{r} \text{Result} \quad 14.47 \\ \text{Factor 2} \quad \underline{.76} \times \\ \hline 10.9972 \end{array}$$

$$\text{Factor 1} \quad 11.0000 - 10.9972 = .0028 = \text{Remainder}$$

การคิดหาเศษเหลือใช้หลักการคำนวณที่ว่า เมื่อนำผลหารไปคูณกับตัวหารควรจะได้ค่าเท่ากับ Factor 1 ดังนั้นค่าที่หายไปจากการคำนวณจริงเพื่อให้มีค่าเท่ากับ Factor 1 จึงเป็นเศษเหลือ

จากตัวอย่างที่ยกมาคือ $11 \div 76$ เมื่อกำหนดจำนวนทศนิยมใน MVR เท่ากับ 4 ตำแหน่ง ดังนั้น ค่าใน Result Field ในคำสั่ง MVR จึงมีค่าเท่ากับ .0028 ถ้ากำหนด Field Length ของ Result Field ในคำสั่ง MVR แตกต่างไปจากเงื่อนไขที่บ่งไว้แล้วค่าของเศษเหลือใน Result Field ก็จะแตกต่างกันไป เช่น

Field Length	Decimal Positions	Result Value
5	0	00000
4	1	000.0
3	2	0.00
4	3	0.002
5	5	.00280

ข้อควรระวังในการใช้ MVR

1. เครื่องหมายของเศษเหลือ (Remainder) ในคำสั่ง จะเหมือนกันกับเครื่องหมายของ Result Field ในคำสั่ง DIV (ลำดับก่อนคำสั่งที่จะเกิด MVR)

2. ถ้ามีคำสั่ง MVR ตามมาหลังจากคำสั่งใน DIV แล้ว ในคำสั่ง DIV แล้วในคำสั่ง DIV จะใช้ Half adjust ไม่ได้ เช่น

C	01	A	DIV	B	C	42 H
C			MVR		D	52

3. จะใช้คำสั่ง MVR ควบคู่กับการใช้ half adjust ไม่ได้ เช่น

C	01	A	DIV	B	C	42
C	01		MVR		D	52 H

SQRT : (Square Root)

ลักษณะของการเขียนคำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
-----------	----------	-----------	----------	-----------------	-------------------------

optional	blank	SQRT	required	required	blank
-----------------	-------	------	----------	----------	-------

คำสั่ง SQRT คือการหาค่ารากที่สองของ Factor 2 แล้วค่าที่ได้นำไปเก็บไว้ใน Result Field ถ้าค่าใน Factor 2 เป็นศูนย์ ผลที่ได้ใน Result Field เป็นศูนย์ด้วย แต่ถ้าค่าใน Factor 2 เป็นค่าติดลบ ซึ่งไม่สามารถหาค่ารากที่สองได้ จะทำให้เกิด error ขึ้น 3 ประการคือ

1. HO Indicator จะเกิดขึ้น (HO set On)
2. * ERROR byte ใน RPGII จะ set error code E(X'C5')
3. โปรแกรมจะหยุดทำงานโดยการถูก cancel

หมายเหตุ : เราสามารถใช้ Half Adjust กับคำสั่ง SQRT ได้แต่จะใช้ Resulting Indicators ไม่ได้

: นอกจากคำสั่ง ADD,SUB,MULT,DIV,Z-ADD,Z-SUB ฯลฯ ที่ใช้กันบ่อย ๆ เป็นปกติแล้วยังมีคำสั่งอื่น ๆ อีกที่สามารถนำมาใช้เพื่อเป็นประโยชน์ในการประมวลผลเช่น

TESTZ : (Testing and Branching)

ลักษณะของการใช้คำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
-----------	----------	-----------	----------	-----------------	-------------------------

optional	blank	TESTZ	blank	required	required
----------	-------	-------	-------	----------	----------

คำสั่งดังกล่าวคือ TESTZ คำสั่งนี้ใช้กับการพิจารณาเงื่อนไขใน zone area ของตำแหน่งซ้ายสุด (left most = high order) ของตัวเลขใน data field ถ้าตรวจพบว่าตำแหน่งซ้ายสุดประกอบด้วย

zone area ก็จะได้ถือว่า field ดังกล่าวเป็น Alpha-numeric การตรวจสอบโดยคำสั่งจะส่งผลให้ Indicator ที่ตั้งไว้เกิดขึ้นเพื่อที่จะนำไปสู่การปฏิบัติตามคำสั่งอื่น ๆ ที่เกี่ยวข้องกับ Indicator ดังกล่าวจะเกิดขึ้น

การตรวจสอบโดยคำสั่ง TESTZ จะเป็นไปตามเงื่อนไขดังนี้

column	meaning	acceptable character
54-55	positive	A-I,&,0 ⁺
56-57	negative	J-R, -,0 ⁻
58-59	other	S-Z,0-9, Special Character

ตัวอย่างการใช้คำสั่ง TESTZ

C	01	TESTZ ALPHA	Resulting Indicators
			111213

จากคำสั่งที่ยกตัวอย่างมาจะเป็นการตรวจสอบเงื่อนไขของ ALPHA ถ้า character ในตำแหน่งซ้ายสุดของ field ALPHA เป็นบวก แล้ว Indicator 11 จะเกิดขึ้น แต่ถ้าค่าเป็นลบ Indicator 12 จะเกิดขึ้นและถ้าเป็นอย่างอื่น Indicator 13 จะเกิดขึ้น (การที่ตำแหน่งซ้ายสุดจะเป็นบวก ลบ หรืออื่น ๆ ให้ดูจากตารางที่กำหนดไว้ว่า character ใดจะมีเครื่องหมายเป็นอย่างไร หรือจะยึดหลักว่า character ตัวใดมีการเจาะที่ zone 12 ถือว่าเป็นบวก และถ้าเจาะ zone 11 ถือว่าเป็นลบ นอกนั้นเป็นอย่างอื่น) การใช้คำสั่ง TESTZ ในการตรวจสอบ zone ของตัวซ้ายสุดใน data field นั้นอาจจะหลีกเลี่ยงไปใช้วิธีการอื่น ๆ ซึ่งจะให้ผลเหมือนกับการใช้คำสั่ง TESTZ วิธีการนั้นก็คือให้ใช้ Indicator (ใน I-Form ณ.ที่ สดมภ์ที่ 65-70) ให้เป็นประโยชน์ ซึ่งใน field ดังกล่าวจะใช้สำหรับการตรวจสอบเครื่องหมายบวก ลบ และ ศูนย์ (หรือ blank) ใน data field ได้ ให้พิจารณาวิธีการที่ใช้ดังนี้คือ

Field Indicator

IFILE1	BB	01	
I		1 5	ALPHA
I		1 10	COL1 111213

ซึ่งการตรวจสอบเช่นนี้ใช้เป็นการสั่งใน I-Form โดยการใช้ Field Indicator จะให้ผลที่ได้ในการตรวจสอบเช่นเดียวกันกับการใช้คำสั่ง TESTZ ใน C-Form ทุกประการ

TESTN : (Test Numeric)

ลักษณะของการใช้คำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicator
optional	blank	TESTN	required	required	required

คำสั่ง TESTN ใช้สำหรับการตรวจสอบ Alphameric Field ว่า field ดังกล่าวนั้นมี character ที่เป็นตัวเลขหรือ blank ปรากฏอยู่หรือไม่ field ที่ถูกใช้ในการตรวจสอบด้วยคำสั่งนี้ก็คือ Result Field ผลที่เกิดจากการตรวจสอบจะปรากฏที่ Resulting Indicators ตามเงื่อนไขต่อไปนี้

1. ถ้า character ทุกตัวใน Result Field เป็นเลขล้วน ๆ แล้วจะส่งผลให้ Indicator ในสดมภ์ที่ 54-55 เกิดขึ้น แต่ถ้าใน Result Field มีทั้งตัวเลขและ blank ก็ส่งผลให้ Indicator ในสดมภ์ที่ 56-57 เกิดขึ้น และประการสุดท้าย ถ้า Result Field ดังกล่าว characters ทุกตัวเป็น blank หมดก็จะส่งผลให้ Indicator ในสดมภ์ที่ 58-59 เกิดขึ้น การกำหนดตัวเลขใน Indicators ทั้ง 3 ตัวคือสดมภ์ที่ 54-55 สดมภ์ที่ 56-57 และสดมภ์ที่ 58-59 อาจจะซ้ำหรือไม่ซ้ำกันก็ได้ขึ้นอยู่กับเงื่อนไขของการนำ Indicator เหล่านั้นไปใช้งาน

จุดประสงค์ของการใช้ TESTN ก็เพื่อที่จะตรวจสอบว่าข้อมูลใน field ใด field หนึ่งซึ่งเราจะนำไปใช้ในการคำนวณทางคณิตศาสตร์นั้นเป็น Numeric จริงหรือไม่ เพราะถ้าหากว่า field ดังกล่าวเป็น Alphameric การใช้วิธีการคำนวณทางคณิตศาสตร์จะเกิดความผิดพลาดขึ้นเพราะไม่สามารถจะกระทำได้ด้วยข้อมูลที่มิใช่ตัวเลข

Branching and Exit Operation

การดำเนินงานตามคำสั่งในโปรแกรมนั้นโดยปกติจะดำเนินการแต่ละคำสั่งนั้นเป็นลำดับตามที่เรียงลำดับไว้ในโปรแกรม แต่ในบางครั้งผู้เขียนคำสั่งมีความจำเป็นที่จะต้องให้มีการเปลี่ยนแปลงการ

ทำงานไม่ให้เป็นไปตามลำดับ ทั้งนี้ทั้งนั้นขึ้นอยู่กับเงื่อนไขหรือกฎเกณฑ์ที่ผู้เขียนคำสั่งต้องการ ซึ่งเงื่อนไขในการที่จะให้การประมวลผลเปลี่ยนแปลงไปสู่ลำดับ (คำสั่ง) ที่ผู้เขียนคำสั่งต้องการ ก็จะสามารถทำได้โดยการใช้คำสั่ง GOTO คำสั่งนี้จะช่วยให้มีการข้ามคำสั่งบางคำสั่งไปสู่คำสั่งที่ต้องการได้โดยมีการกำหนดด้วยเงื่อนไขบางอย่างที่ตั้งไว้เป็นตัวควบคุม

ลักษณะของการใช้คำสั่ง GOTO

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicator
optional	blank	GOTO	required	blank	blank

คำสั่ง GOTO จะทำให้มีการออกจากคำสั่งนั้นไปยังส่วนอื่นของโปรแกรมที่บ่งไว้ใน Factor 2 ดังนั้น จึงจำเป็นจะต้องบ่งชี้ที่จะไปคือ Factor 2 ไว้ด้วยโดยที่ Factor 2 คือชื่อของ routine ชื่อที่ใช้ไม่ได้ใช้ Indicator แต่เป็น Alphabetic Name ซึ่งตั้งขึ้นโดยผู้เขียนโปรแกรม (เป็นชื่อของ routine ที่จะไป)

กฎเกณฑ์ในการตั้งชื่อก็เช่นเดียวกันกับการตั้งชื่อ data field ทุกประการ การใช้ชื่อ routine ที่จะให้มีการข้ามไปสู่ส่วนนั้นได้ จะต้องมีการชี้แจงว่าคำสั่งนั้นเป็นคำสั่งที่ต้องการให้มา โดยการใช้คำสั่งว่า TAG เข้าช่วย โดยที่ลักษณะของการเขียนคำสั่งการใช้ TAG ก็คือ

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicator
blank	required	TAG	blank	blank	blank

สิ่งที่ต้องระมัดระวังก็คือชื่อของ routine ที่จะข้ามไปสู่คำสั่งนั้น กับชื่อของ Factor 1 ในคำสั่งจะต้อง

1. routine = ส่วนหนึ่งของโปรแกรม หรือคำสั่งย่อย ๆ หลาย ๆ คำสั่งประกอบกัน

เป็นชื่อเดียวกัน เพื่อความเข้าใจในการใช้คำสั่งดังกล่าวจะขอยกตัวอย่างดังนี้

Line	Code	Indicator	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Resulting Indicators				Comments	
			And	Or	Xor						Compare		High	Low		Equal
											>	<				
100					GRSPAY	TAG										
110					HOURS	COMP	40						2022			
120		20			HOURS	SUB	40	OVERHR	31							
130		20			RATE	MULT	OVERHR	SAVE	62H							
140		20			SAVE	MULT	1.5	SAVE	H							
150		N22			RATE	MULT	40	GROSS	62H							
160		N22			SAVE	ADD	GROSS	GROSS								
170		N22			GOTO	FICA										
180					(ADDITIONAL CALCULATIONS)											
190					FICA	TAG		DFICA	62H							
200					GROSS	MULT	.03	HOLD	62H							
210					YDFICA	ADD	DFICA									
220					HOLD	COMP	144.00						2121			
230		21			GOTO	ADFICA										
240					144.00	SUB	YDFICA	DFICA								
250					ADFICA	TAG										
260					YDFICA	ADD	DFICA	YDFICA								
270					(ADDITIONAL CALCULATIONS)											
280					WHTAX	TAG										
290					(ADDITIONAL CALCULATIONS)											

จากตัวอย่างนี้จะเห็นได้ว่าเมื่อมีการใช้คำสั่งเพื่อให้ข้าม (branch) ไปยัง routine ที่ต้องการภายใน routine ที่ให้ข้ามไปนั้นยังมีสิทธิที่จะ branch ออกจาก routine นั้น โดยการใช้คำสั่งในลักษณะเดียวกันได้

ลักษณะการทำงานของคำสั่งที่ยกตัวอย่างมาให้ดู

- คำสั่งที่ 10 โปรแกรมจะข้ามไปยัง GRSPAY จากคำสั่งอื่นๆ ที่มีมาก่อนหน้านั้น (ไม่ได้แสดงไว้ในโปรแกรม)
- คำสั่งที่ 20 นำจำนวนชั่วโมงทำงาน (HOURS) มาเปรียบเทียบกับค่าคงที่ 40 ถ้าหากมีค่ามากกว่า 40 ชั่วโมง Indicator 20 จะเกิดขึ้น แต่ถ้า HOURS น้อยกว่า 40 Indicator-22 จะเกิดขึ้น
- คำสั่งที่ 30 คำสั่งนี้จะเกิดขึ้นเมื่อ Indicator 20 เกิดขึ้น ความหมายของคำสั่งนี้ก็คือ ให้นำ HOURS ลบออกด้วย 40 แล้วนำผลที่ได้จากการลบไปเก็บไว้ที่ OVERHR (ซึ่งมีขนาดความยาว 3 ตำแหน่งและมีทศนิยม 1 ตำแหน่ง)
- คำสั่งที่ 40 คำสั่งนี้จะเกิดขึ้นเมื่อ Indicator 20 เกิดขึ้น คำสั่งนี้จะทำต่อจากคำสั่งที่ 30 ความหมายของคำสั่งนี้ก็คือ ให้นำ RATE มาคูณด้วย OVERHR แล้วนำผลที่ได้จากการคูณมาเก็บไว้ที่ SAVE

- คำสั่งที่ 50 คำสั่งนี้จะเกิดขึ้นเมื่อ Indicator 20 เกิดขึ้น และคำสั่งนี้จะทำต่อจากคำสั่งที่ 40 ความหมายของคำสั่งนี้ก็คือให้นำ SAVE มาคูณกับ 1.5 แล้วนำผลที่ได้ไปเก็บไว้ที่ SAVE
- คำสั่งที่ 60 คำสั่งนี้จะเกิดขึ้นเมื่อ HOURS มีค่าใด ๆ ยกเว้น HOURS มีค่าน้อยกว่า 40 (หรืออีกนัยหนึ่ง คำสั่งนี้จะเกิดขึ้นก็ต่อเมื่อ Indicator 22 ไม่เกิดขึ้น) ความหมายของคำสั่งนี้ก็คือให้นำ RATE ไปคูณด้วย 40 แล้วนำผลที่ได้ไว้ไปเก็บไว้ที่ GROSS
- คำสั่งที่ 70 จะเกิดขึ้นต่อจากคำสั่งที่ 60 และคำสั่งที่ 70 นี้จะเกิดขึ้นโดยลักษณะเช่นเดียวกันกับคำสั่งที่ 60 ความหมายของคำสั่งนี้ก็คือให้นำ SAVE ไปบวกเข้ากับ GROSS แล้วนำค่าที่ได้ไปเก็บไว้ที่ GROSS
- คำสั่งที่ 80 จะเกิดขึ้นในลำดับที่ถัดมาจากคำสั่งที่ 70 การเกิดคำสั่งนี้ก็คือเกิดขึ้นตามเงื่อนไขเช่นเดียวกันกับคำสั่งที่ 60 และคำสั่งที่ 70 ความหมายของคำสั่งนี้ก็คือให้ข้ามไปยัง routine ที่มีชื่อว่า FICA ซึ่งอยู่ที่คำสั่งที่ 110 นั่นก็คือ ให้ข้ามคำสั่งทั้งหลายที่อยู่ต่อจากคำสั่งที่ 80 ไปยังคำสั่งที่ 110 เลย
- คำสั่งที่ 110 คำสั่งนี้ไม่จำเป็นต้องมี Indicator ควบคุม เพราะเป็นผลเนื่องมาจากคำสั่งที่ 80 ความหมายของคำสั่งนี้ก็คือ ให้นำค่า GROSS ไปคูณด้วยค่าคงที่ .03 แล้วไปเก็บไว้ที่ DFICA

ส่วนคำสั่งอื่น ๆ ก็จะทำตามเงื่อนไขของแต่ละคำสั่ง

ข้อควรระวัง ในการใช้ GOTO ก็คือระวังอย่าให้เกิดการวน (loop) ในโปรแกรมมิฉะนั้นคอมพิวเตอร์จะวนทำงานอยู่ภายใน loop โดยไม่ออกจาก loop ซึ่งทำให้เกิดความผิดพลาด

ตัวอย่างการใช้ loop วน (Infinite Loop)

Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
FICA	TAG			
GROSS	MULT	.03	DFICA	62H
YDFICA	ADD	DFICA	HOLD	62H
HOLD	COMP	144.00		

20		GOTO	ADFICA		
	144.00	SUB	YDFICA	DFICA	
	ADFICA	ADD	DFICA	YDFICA	
		(ADDITION CALCULATION)			
20	GROSSA	TAG			
20	GOURS	COMP	40		1112
		(ADDITION CALCULATION)			
		GOTO	FICA		
		ADDITIONAL CALCULATION			

Move Operation

คำสั่งที่ใช้ในการ Move แบ่งออกเป็น 2 พวกคือ พวกแรกที่ใช้เกี่ยวกับการ Move Character และหรือ Move Field ส่วนอีกพวกหนึ่งคือคำสั่ง Move ที่เกี่ยวกับการ Move Zone ของ Character ใด Character หนึ่ง

พวกแรกมีอยู่ 2 คำสั่งคือ MOVE และ MOVE L (Move Left) ซึ่ง 2 คำสั่งนี้ใช้กับการสั่งให้ Factor 2 นั้น move ไปยัง Result Field ดังนั้นในคำสั่งที่ใช้เกี่ยวกับการปฏิบัติการเช่นนี้ไม่ต้องใส่ Factor 1, Resulting Indicators ส่วน Half Adjust นั้นไม่ต้องใช้ให้ว่างไว้ในส่วนดังกล่าว ในกรณีที่มีความยาวของ Factor 2 และ Result Field ไม่เท่ากัน ซึ่งแยกได้เป็น 2 กรณีคือ

1. Factor 2 มีขนาดความยาวมากกว่า Result Field ซึ่งถ้าเป็นเช่นนี้จะ Move เฉพาะส่วนที่เท่ากับของ Factor 2 ไปยัง Result Field เท่านั้น ส่วนที่เหลือจะไม่ Move

2. ถ้า Factor 2 มีขนาดความยาวสั้นกว่า Result Field ข้อมูลใน Factor 2 จะ Move ไปยัง Result Field ทั้งหมดส่วนตำแหน่งที่เหลือใน Result Field จะยังคงเป็นข้อมูลที่เดิม

ปัญหาที่เกิดขึ้นก็คือในกรณีที่ 1 และที่ 2 ส่วนที่ไม่ Move ไปยัง Result Field ของกรณีแรกคือส่วนใดของ Result Field ทางซ้ายหรือทางขวา ซึ่งการที่จะเป็นเช่นใดก็ขึ้นอยู่กับคำสั่งในการว่าป็นชนิดใด MOVE หรือ MOVE L

หลักการใช้คำสั่ง MOVE:

ลักษณะของการเขียนคำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
optional	blank	MOVE	required	required	blank

คำสั่ง MOVE จะทำให้ character ที่อยู่ใน Factor 2 ในตำแหน่งขวาสุด (right most) ไปยังตำแหน่งขวาสุดของ Result Field

ดังนั้น ถ้า Factor 2 ยาวกว่า Result Field ตำแหน่งซ้ายสุดและถัดมาของ Factor 2 จะไม่ MOVE ไปยัง Result Field และในทำนองเดียวกัน ถ้า Result Field ยาวกว่า Factor 2 จะยังผลให้ตำแหน่งซ้ายสุดและถัดมาของ Result Field ไม่เปลี่ยนแปลง พิจารณาข้อมูลที่ได้จากการใช้คำสั่งดังกล่าว

Factor 2	Operation	Result Field	Result
3591	MOVE	12122	13597
1699999	MOVE	321321	699999
75.1	MOVE	123.45	127.51
5924. 11	MOVE	666.822	592.411
	MOVE	25	21

MOVE: (Move Left)

ลักษณะของการใช้คำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
optional	blank	MOVE	required	required	blank

คำสั่ง MOVEL จะทำให้ character (เริ่มจากตำแหน่งซ้ายสุดของ field) move จาก Factor 2 ไปยัง Result Field ถ้าหากว่า Factor 2 ยาวกว่า Result Field ตำแหน่งที่เหลือ (move มาหมดแล้ว) ก็คือตำแหน่งที่อยู่ขวาสุดของ field ใน Factor 2 จะเหลืออยู่แต่ถ้า Result Field ยาวกว่า Factor 2 ตำแหน่งของ Factor 2 ที่อยู่ทางขวาสุดจะยังคงเดิมไม่เปลี่ยนแปลง

ตัวอย่างเช่น

Factor 2	Operation	Result Field	Result
2555	MOVE:L	133	255
1.25	MOVE:L	876.38	125.38
879217.7	MOVE:L	499.668	879.217
336	MOVE:L	1244532	3364532
71.2	MOVE:L	69.44	71.24

MOVEA : (Move Array)

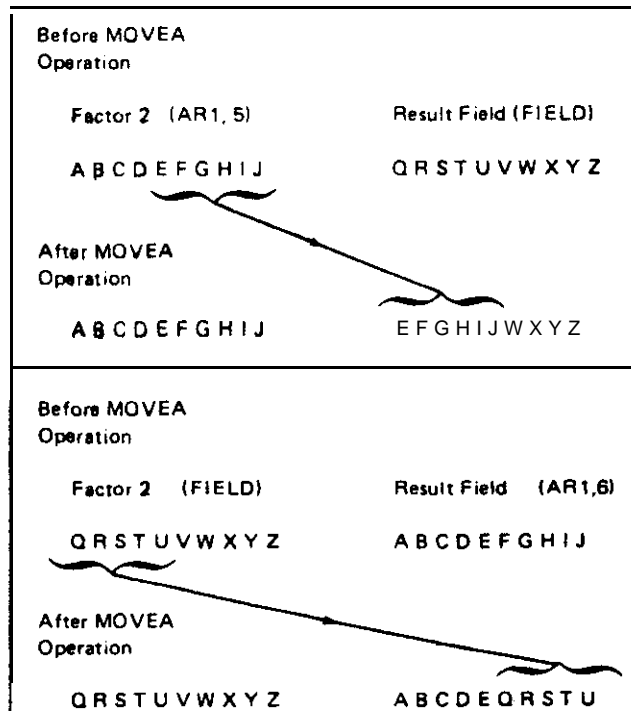
ลักษณะของการใช้คำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
optional	blank	MOVEA	required	required	blank

คำสั่ง MOVEA จะทำให้ character ใน Factor 2 เริ่มตั้งแต่ตำแหน่งซ้ายสุดของ field นั้น move จาก Factor 2 ไปยัง Result Field การใช้คำสั่ง MOVEA เพราะ Factor 2 หรือ Result Field อาจจะเป็นกลุ่ม (array) หรือ Alphameric ก็ได้

การย้ายข้อมูลจะเริ่มจากสมาชิกตัวแรกของกลุ่ม (array) ถ้าเราบ่งเฉพาะชื่อของกลุ่ม ใน Factor 2 หรือจะเริ่มจากสมาชิกของกลุ่ม (array) ที่เราเจาะจงก็ได้ โดยพิจารณาจากตัวอย่างต่อไปนี้

Factor 2	Result Field	Content of Result Field after MOVEA Operation
AR1	FIELD	ABCDEFGHIJ
FIELD	AR1	QRSTUVWXYZ
AR1,5	FIELD	EFGHIJWXYZ
FIELD	AR1,6	ABCDEQRSTU
AR2	FIELD	KLMNOPWXYZ
FIELD	AR2	QRSTUW
FIELD	AR2,2	KLQRST
AR1,X	AR2,Y	KLHIJP
AR2,Y	AR1,X	ABCDEFGHIMNO



MHLZO : (Move Zone)

คำสั่งที่เกี่ยวกับการ move พวกที่ 2 คือ คำสั่งที่เกี่ยวกับการ move zone high order หรือ low order ใน Factor 2 ไปยัง high order หรือ low order ใน Result Field โดยสามารถแยก move zone ได้ 4 แบบคือ

Move High-to-Low-Zone (MHLZO)

ลักษณะของคำสั่ง

Indicator	Factor I	Operation	Factor 2	Result Field	Resulting Indicator
optional	blank	MHLZO	required	required	blank

คำสั่ง MHLZO ทำให้มีการย้าย zone ของ character จากลำดับซ้ายสุดของ Factor 2 ไปยังตำแหน่งของ character ตัวขวาสุดของ Result Field โดยที่ Factor 2 จะต้องเป็น Alphameric Result Field จะเป็น Numeric หรือ Alphameric ก็ได้

Move Low to-High Zone : (MHHZO)

ลักษณะของคำสั่ง

Indicator	Factor I	Operation	Factor 2	Result Field	Resulting Indicator
optional	blank	MHHZO	required	required	blank

คำสั่ง MLHZO ก็คือคำสั่งที่ให้ย้าย zone ของ Factor 2 ตัวที่อยู่ในตำแหน่งซ้ายสุดของ Result Field ไปยังตำแหน่งซ้ายที่สุดของ Result Field ดังนั้น Factor 2 และ Result Field จะต้องเป็น Alphameric

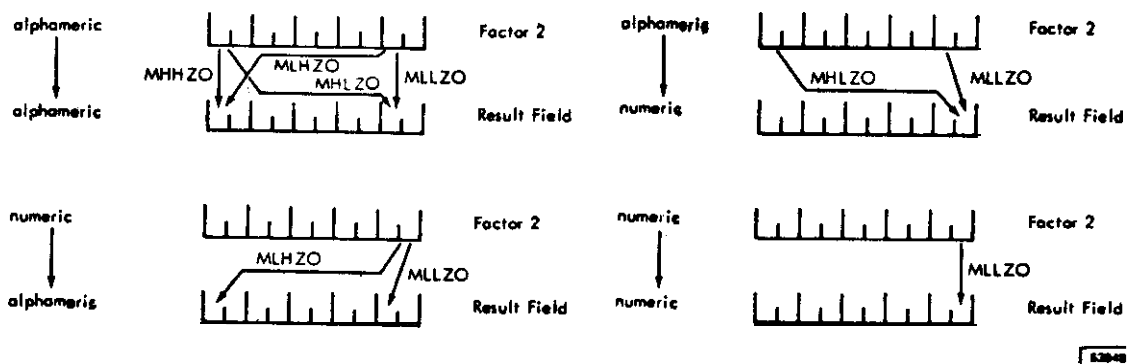
Move Low-to-Low **Zone** : (MLLZO)

ลักษณะของคำสั่ง

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
optional	blank	MLLZO	required	required	blank

คำสั่ง MLLZO ก็คือคำสั่งที่ให้ย้าย zone ของ character ที่อยู่ในตำแหน่งที่ขวาสุดของ Factor 2 ไปยังตำแหน่งที่อยู่ขวาสุดของ Result Field โดยที่ Factor 2 และ Result Field จะเป็น Alphameric หรือ Numeric ก็ได้

พิจารณาการ move zone แต่ละชนิดจากแผนผังดังต่อไปนี้



ข้อควรระวังในการใช้คำสั่งเกี่ยวกับการ Move Zone :

1. ถ้าย้าย zone จาก Alphameric field ไปยัง Numeric Field จะยังผลให้ Result Field กลายเป็นจำนวนบวก ยกเว้นในกรณีที่ Factor 2 มี zone เป็นลบ ในกรณีที่ เป็น zone 0 จะถือว่าเป็น zone ที่ทำให้เกิดค่าเป็นบวก

ตัวอย่างในการใช้คำสั่ง

Indicator	Factor 2	Result Field	Result
MHLZO	Alphameric	Numeric	
	ABC	359	351
	NOP	453	45L
MHLZO	Alphameric	Numeric	
	TAB	123	12c
MLHZO	Alphameric		
	123	N35	E3E

SETON and SETOF

SETON เป็นคำสั่งที่ใช้เพื่อสั่งให้ Indicator ใน Resulting Indicators เกิดขึ้น เพื่อประโยชน์ในการประมวลผล ในคำสั่งที่ต้องการ โดยมีลักษณะของคำสั่งดังนี้

Move Zone Operations

Factor 2 Result Field	Alphabetic Alphabetic	Bits 0-3 of rightmost byte of Factor 2 are moved to bits 0-3 of rightmost byte of Result Field
Factor 2 Result Field	Alphabetic Numeric	Bits 0-3 of rightmost byte of Factor 2 are moved to bits 4-7 of the rightmost byte of the Result Field
Factor 2 Result Field	Numeric Alphabetic	bits 4-7 of rightmost byte of Factor 2 are moved to bits 0-3 of rightmost byte of the Result Field
Factor 2 Result Field	Numeric Numeric	Bits 4-7 of rightmost byte of Factor 2 are moved to bits 4-7 of rightmost byte of the Result Field
MNLZO		
Factor 2 Result Field	Alphabetic Numeric	Bits 0-3 of leftmost byte of Factor 2 are moved to bits 4-7 of rightmost byte of Result Field
Factor 2 Result Field	Alphabetic Alphabetic	Bits 0-3 of leftmost byte of Factor 2 are moved to bits 0-3 of rightmost byte of the Result Field
MLHZO		
Factor 2 Result Field	Alphabetic Alphabetic	Bits 0-3 of rightmost byte of Factor 2 are moved to bits 0-3 of leftmost byte of Result Field
Factor 2 Result Field	Numeric Alphabetic	Bits 4-7 of rightmost byte of Factor 2 are moved to bits 0-3 of leftmost byte of the Result Field
MNHZO		
Factor 2 Result Field	Alphabetic Alphabetic	Bits 0-3 of leftmost byte of Factor 2 are moved to bits 0-3 of leftmost byte of Result Field

Examples of Move Zone Operations

Instruction	Factor 2	Result Field	Results
MLLZO	ALPHAMERIC A234A	ALPHAMERIC B395J	B395D
	ALPHAMERIC B12C	NUMERIC 1235	123E
	NUMERIC 1235	ALPHAMERIC 1A2T	1A2C
	NUMERIC 1235	NUMERIC 1357	135P
MNLZO	ALPHAMERIC L135	NUMERIC 1375	137N
	ALPHAMERIC AC12	ALPHAMERIC RA13	RA1C
MLHZO	NUMERIC 134	ALPHAMERIC RBA	IBA
	ALPHAMERIC R12	ALPHAMERIC C35	C3E
MNHZO	ALPHAMERIC TOOBIG	ALPHAMERIC MLITTLE	DLITTLE

Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicator
optional	blank	SETON	blank	blank	required

คำสั่งนี้จะใช้ใน C-Form ดังตัวอย่างต่อไปนี้

C	10	SETON	11
---	----	-------	----

SETOF: เป็นคำสั่งเพื่อเปิด Indicator ใน Resulting Indicators การใช้คำสั่ง SETOF จะเหมือนกับการใช้คำสั่ง SETON ทุกประการเพียงแต่ความหมายจะตรงกันข้ามเท่านั้น ตัวอย่างของการใช้คำสั่งทั้งสองนี้ในการประมวลผลมีดังนี้คือ สมมุติว่ามีบริษัทแห่งหนึ่งดำเนินธุรกิจเกี่ยวกับการรับประกันภัย โดยที่บริษัทต้องการที่จะทำสถิติของลูกค้าที่ได้รับอุบัติเหตุในช่วงปีที่ผ่านมา การคิดจำนวนสถิติอุบัติเหตุให้แยกเป็นบุคคล 3 ระดับอายุคือ

กลุ่มที่ 1 อายุต่ำกว่า 25 ปี

กลุ่มที่ 2 อายุ 25 ปี

กลุ่มที่ 3 อายุมากกว่า 25 ปี

ผลที่ต้องการก็คือยอดรวมของผู้ที่ได้รับอุบัติเหตุในแต่ละกลุ่มอายุ

กำหนด Input Record ดังนี้

col. 1-30	NAME
col. 40-50	CODE
col. 71-72	AGE
col. 73-73	SEX
col. 74-75	ACCID (จำนวนครั้งของอุบัติเหตุ)
col. 80	"A"

จากปัญหานี้สามารถนำมาเขียนเป็นคำสั่งได้ดังนี้คือ

						Resulting
						Indicators
I-Form :						+ - 0
	IINFILE	AA 12	80 CA			
	I	OR 13	80NCA			
	I			1 30		NAME
	I			40 50		CODE
	I			71 720		AGE
	I			73 73		SEX
	I			74 740		ACCID

C-Form :

	Indicator	Factor 1	Operation	Factor 2	Result Field	Resulting Indicators
10 c	12	AGE	COMP	25		203040
20 c	12	TOT	ADD	1	TOT	
30 c	20N15	ACCID	ADD	AC025	AC025	
40 c	30N15	ACCID	ADD	ACU25	ACU25	
50 c	40N15	ACCID	ADD	ACE 25	ACE25	

จากคำสั่งนี้ ถ้าหากว่า Record ที่กำลังเข้ามาเป็น Record ที่ไม่มี A ที่สคมภที่ 80 Indicator 12 จะเกิดขึ้น ซึ่งถ้าเป็นดังนี้จะไม่มีการทำตามคำสั่งที่ 10 ใน C-Form ดังนั้นถ้าหาก Record ก่อนหน้านั้นที่เข้ามา A ที่สคมภที่ 80 ส่งผลให้ Indicator ตัวใดตัวหนึ่ง ในกลุ่มของ Indicator 20,30,40 ซึ่งเกิดขึ้นก่อนหน้านั้นอยู่แล้วและยังคงส่งผลในการ On อีกใน Record ที่ถัดมา นั้นหมายความว่า จะเกิดความผิดพลาดขึ้นแล้วเพราะคำสั่งใน C-Form เราต้องการให้ทำงานเฉพาะ Record ที่มี A ที่สคมภที่ 80 หรือ จะกล่าวอีกนัยหนึ่งได้ว่า คำสั่งใน C-Form เราต้องการให้มีการคำนวณก็ต่อเมื่อ Indicator 12 เกิดขึ้นเท่านั้น ซึ่งถ้าหากเกิดกรณีเช่นนี้ขึ้น เราจะมีวิธีการป้องกันความผิดพลาดที่อาจจะเกิดขึ้นด้วยเหตุดังกล่าวได้โดยการเขียนคำสั่งใน C-Form ใหม่ดังนี้

10	C	12	AGE	COMP 25	203040
20	C	12	TOT	ADD 1	TOT 40
30	C	20N1512	ACCID	ADD ACO25	ACO25 40
40	C	30N1512	ACCID	ADD ACU25	ACU 40
50	C	40N1512	ACCID	ADD ACE25	ACE25 40

จากการคำสั่งที่ปรับปรุงใหม่โดยการเพิ่ม Indicator 12 เข้าไปในคำสั่งที่ 30, 40 และ 50 ก็จะแก้ความผิดพลาดอันอาจเกิดขึ้นด้วยสาเหตุของ Invalid Record ได้อีกทางหนึ่ง

หนทางที่เราสามารถแก้ไขความผิดพลาดชนิดนี้โดยการใช้คำสั่ง SETOF เข้าช่วยเพื่อที่จะปิด Indicators ที่เกิดขึ้นเนื่องจากผลของ Record ก่อนหน้านั้น (ก่อนหน้า Record ที่เรากำลังจะให้ประมวลผล) การที่จะใช้คำสั่ง SETOF เข้าช่วยจะทำให้คำสั่งใน C-Form เปลี่ยนแปลงไปบ้างแต่จะให้ผลตรงกันกับวิธีที่เราแก้ไขมาก่อนหน้านี้

10	c	13		SETOF	203040
20	c	12	AGE	COMP 25	203040
30	c	12	TOT	ADD 1	TOT 40
40	C	20N15	ACCID	ADD AC025	AC025 40
50	c	30N15	ACCID	ADD ACU25	ACU25 40
60	C	40N15	ACCID	ADD ACE25	ACE25 40

สำหรับคำสั่งของ SETON จะมีความหมายตรงกันข้ามกับคำสั่ง SETOF คำสั่ง SETON จะทำให้ Indicators ที่ต้องการเกิดขึ้น ดังนั้นจึงขึ้นอยู่กับวัตถุประสงค์ของผู้ใช้ว่าจะใช้คำสั่ง SETON นั้น สำหรับ Indicators ใดที่ต้องการ ตัวอย่างของการใช้คำสั่ง SETON

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (L, L, S, S, I)	Indicators					Factor 1	Operation	Factor 2	Result Field	Field Length	Operator Subroutine	Resulting Indicators			Comments		
			Not	And		Not	Plus							Minus	Zero				
				Not	Not											Not			
3																			
4																			
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			
28																			
29																			
30																			
31																			
32																			
33																			
34																			
35																			
36																			
37																			
38																			
39																			
40																			
41																			
42																			
43																			
44																			
45																			
46																			
47																			
48																			
49																			
50																			
51																			
52																			
53																			
54																			
55																			
56																			
57																			
58																			
59																			
60																			
61																			
62																			
63																			
64																			
65																			
66																			
67																			
68																			
69																			
70																			
71																			
72																			
73																			
74																			

สรุป การใช้คำสั่ง SETON และคำสั่ง SETOF ใน

1. ให้ใช้คำสั่งดังกล่าวใน Operation Field ใน C-Form
2. Indicator ที่ต้องการจะ set ให้เขียนไว้ที่ตำแหน่งในของ Resulting Indicators
3. สดมภ์ที่ 7-17 ซึ่งเป็น Level Indicator และ Indicator อื่น ๆ ก็สามารถใช้เป็นเงื่อนไขในการที่จะควบคุมคำสั่งนี้ได้

ลักษณะของการคำนวณใน C-Form :

คำสั่งแต่ละคำสั่งในโปรแกรมจะมีลำดับการทำงานตามลำดับที่ของคำสั่งที่ปรากฏอยู่ในโปรแกรม ยกเว้นในกรณีที่มีเงื่อนไขในการบังคับให้เปลี่ยนแปลงลำดับในการทำงาน เช่นการใช้-branch, subroutine การใช้ Indicators ต่าง ๆ บังคับ เป็นต้น

ในตอนทีกล่าวถึง C-Form รายละเอียดส่วนใหญ่ที่ปรากฏจะกล่าวถึงเรื่องเกี่ยวกับการใช้ปฏิบัติการต่าง ๆ ส่วน Indicators ที่ใช้ยังไม่ได้กล่าวถึงรายละเอียดมากนัก Indicator ที่กล่าวถึงในที่นี้ปรากฏ C-Form ก็คือ Indicator ที่ปรากฏในสดมภ์ที่ 9-17 ซึ่ง Indicators ดังกล่าว จะเป็นส่วนที่บ่งหรือบังคับการทำงานในแต่ละคำสั่ง

เรื่อง Field Indicator นี้จะแบ่งออกเป็น 3 Indicator โดยที่แต่ละ Indicator มีส่วนที่สามารถใช้ Not (N) รวมอยู่ด้วย ดังนั้นการใส่ Not ก็หมายถึง Indicator นั้นปิด (OF) ที่ใช้กันอยู่มีที่มากมาย

แหล่งด้วยกัน เช่นมาจาก I-Form ในส่วนที่เป็น Record Indicators (สดมภ์ที่ 19-20) หรือ Field Indicators ในสดมภ์ที่ 75-70 หรือมาจาก C-Form ในส่วนที่เป็น Resulting Indicator (สดมภ์ที่ 54-59) โดยที่ถ้าหาก Indicator ในแต่ละแบบฟอร์ม ถ้าทดสอบแล้วถูกต้องตรงกับเงื่อนไขนั้น-Indicators ที่กำหนดไว้ก็จะเกิดขึ้น คำสั่งในที่ถูกบังคับด้วยเงื่อนไขของ Indicator จะมีการทำงานตามคำสั่งนั้นเกิดขึ้น นอกจากนี้แล้ว Indicator ที่กล่าวถึงเหล่านี้จะสามารถโยงไปใช้เกี่ยวกับการพิมพ์ผลลัพธ์ใน O-Form ด้วย

ความสัมพันธ์ระหว่าง Indicator ใน I-Form, C-Form, และ O-Form

ความสัมพันธ์ระหว่าง C-Form กับ O-Form และการใช้ Blank After ใน O-Form

ใน C-Form นั้น Result Field ที่เกิดขึ้นนั้นเราสามารถนำผลที่ได้ไปพิมพ์ใน O-Form ได้ หรือในกรณีที่มีการคำนวณค่าของ Result Field ใน C-Form เมื่อผลที่ได้ถูกนำไปพิมพ์โดยคำสั่ง ใน O-Form เราต้องการให้ Result Field นั้นเป็นศูนย์ (ค่าที่ได้กลายเป็นศูนย์) เพื่อที่จะนำ Result Field เดิมนั้นไปทำการคำนวณใหม่ เราก็สามารถที่จะใช้คำสั่งใน O-Form เข้ามาช่วยได้โดยการใช้คำสั่งใน Blank After (สคมภ์ที่ 39) ซึ่งความหมายของ blank After (B) ก็คือการทำให้ค่าใน Result Field ที่นำไปพิมพ์แล้วนั้นว่างลง

จากตัวอย่างของโปรแกรมต่อไปนี้จะแสดงให้เห็นว่าเมื่อ Record แรก ถูกอ่านเข้ามาจะทำให้ REGH รวมกับ TOT ซึ่ง TOT จะมีค่าเป็นศูนย์เพราะยังไม่มีกำหนด TOT มาก่อนหน้านั้น หลังจากนั้นจึงนำ OTH มารวมกับ TOT เมื่อมีการนำ TOT ไปพิมพ์แล้วค่าของ TOT จะถูกทำให้เป็นศูนย์ ดังนั้นค่าของ TOT ที่จะนำไปคำนวณใน Record ที่ 2 จึงเป็นศูนย์ ซึ่งจะเป็นเช่นนี้เรื่อย ๆ ไป สำหรับ Record ต่อ ๆ ไป

BUS/367*PRG*CL 3-7		PAYROL		06/11/71		PAGE 0001	
021	02 020 H						PAYROL
022	10 030 EMPLE IP F 132 132						PAYROL
023	30 010 EMPLE IP F 132 132						PAYROL
024	30 020 I						PAYROL
025	30 030 I						PAYROL
026	30 040 I						PAYROL
027	30 050 I						PAYROL
028	30 060 I						PAYROL
029	30 070 I						PAYROL
030	30 080 I						PAYROL
031	30 090 I						PAYROL
032	40 010 C C9 REGH ADD TOT TOT 41						PAYROL
033	40 020 C C9 OTH ADD TOT TOT 41						PAYROL
034	50 010 PR-PRG H 33:1 IP						PAYROL
035	50 020 C LR						PAYROL
036	50 030 C						PAYROL
037	50 040 C						PAYROL
038	50 050 C						PAYROL
039	50 060 C						PAYROL
040	50 070 C						PAYROL
041	50 080 C						PAYROL
042	50 090 C						PAYROL
043	50 100 C						PAYROL
044	50 110 C						PAYROL
045	50 120 C						PAYROL
046	50 130 C						PAYROL
047	50 140 C						PAYROL
048	50 150 C						PAYROL
049	50 160 C						PAYROL
050	50 170 C						PAYROL
051	50 180 C						PAYROL
052	50 190 C						PAYROL
053	50 200 C						PAYROL
054	51 010 C						PAYROL
055	51 020 C						PAYROL
056	51 030 C						PAYROL
057	51 040 C						PAYROL
058	51 050 C						PAYROL
059	51 060 C						PAYROL
060	51 070 C						PAYROL
061	51 080 C						PAYROL
062	51 090 C						PAYROL
063	51 100 C						PAYROL
064	51 110 C						PAYROL
065	51 120 C						PAYROL
066	51 130 C						PAYROL

การหาค่า Grand Total

ดังที่ได้กล่าวมาแล้วว่าภาษา RPG เป็นภาษาที่เหมาะสมสำหรับงานรายงาน เช่นการ
ทำบัญชีเงินเดือน การทำรายงานของลูกค้า ฯลฯ เป็นต้น ซึ่งงานดังกล่าวมีลักษณะเด่นคือจะต้องมี
การยอดรวม (Grand Total) ของทุก record รวมกัน ปัญหาที่ตามมาก็คือ ถ้าเราต้องการลักษณะของ
รายงานแบบนี้เราจะมีวิธีการเขียนคำสั่งได้อย่างไรโดยที่จะให้มีการนำข้อมูลในแต่ละ Record
มารวมกันและพิมพ์ผลลัพธ์ที่ได้ออกมาตามวัตถุประสงค์ที่ต้องการเมื่อมีการประมวลผล Record
สุดท้ายเสร็จแล้ว (ค่าของ Record สุดท้ายได้ถูกนำเข้ามาเข้าไปสะสมแล้ว) การประมวลผลข้อมูล
เพื่อให้เป็นไปตามวัตถุประสงค์ที่ต้องการนี้ ส่วนที่เกี่ยวข้องกับงานนี้ที่สำคัญมีอยู่ 2 ส่วนคือ
ส่วนของการคำนวณใน C-Form ซึ่งใช้ในขั้นปฏิบัติการการสะสมข้อมูลของแต่ละ Record ในข้อมูล
ที่ต้องการ และส่วนที่ 2 คือส่วนของการพิมพ์ใน O-Form เพื่อที่จะพิมพ์ยอดรวมของทุก Record
แล้ว

ต่อไปนี้จะเป็นส่วนที่แสดงถึงคำสั่งในขั้นตอนการคำนวณสำหรับปัญหานี้ (หายอดรวม)

10	C	REGH	ADD	TOT	TOT	41
20	C	OTH	ADD	TOT	TOT	41
30	C	COUNT	ADD	1	COUNT	40
40	C	TOT	ADD	GRDTOT	GRDTOT	51

การพิมพ์ยอดรวมใน O-Form ดังที่ได้เคยกล่าวมาแล้วว่าลักษณะของรายงานที่พิมพ์นั้นโดย
ปกติจะประกอบด้วย 3 ส่วนคือ Headings, Detail และ Total โดยที่การพิมพ์ Headings มักจะถูกควบคุม
ด้วย Indicators 2 ตัวคือ IP และ (หรือ) Overflow Indicator ส่วนในการพิมพ์ Detail นั้นมักจะถูกควบคุม
ด้วย Record Indicator และการพิมพ์ Total ก็เช่นเดียวกัน ถึงแม้ว่าเราจะสามารถสั่งให้พิมพ์-
Total ได้ตามที่ต้องการในช่วงใดช่วงหนึ่ง แต่โดยปกติแล้วเรามักจะให้มีการพิมพ์ Total ก็ต่อ
เมื่อ Record สุดท้ายถูกประมวลผลไปแล้วเท่านั้น ซึ่งการพิมพ์ในลักษณะดังกล่าวนี้ ภาษา RPG
ได้กำหนด Indicator เพื่อประโยชน์ดังกล่าวในการพิมพ์อยู่แล้ว Indicator ที่ใช้นี้ก็คือ Last Record
Indicator (LR) ซึ่ง Indicator นี้จะทำหน้าที่ในการควบคุมการทำงานให้เป็นไปตามคำสั่งที่
ต้องการ โดยที่ LR จะเกิดขึ้น (set On) ก็ต่อเมื่อ Record สุดท้ายถูกอ่านเข้าไปประมวลผลแล้ว
หมายเหตุ LR Indicator จะเกิดขึ้นเทียบได้กับว่าเมื่อพบ control card ไปนี้ /* ตัวอย่างเพื่อแสดง
ถึงการทำงานซึ่งใช้ประโยชน์จาก Indicator LR

001	00 000 H								
002	10 020 FINFILE IP F 80 80				READ40			SYSROR	PAYROL
003	30 010 FREPORT U F 132 132			OF	PRINTER			SYSLS1	PAYROL
004	30 220 I				1			30DEP	PAYROL
005	30 030 I				4			28 EMPNAM	PAYROL
006	30 040 I				29			370EMPNO	PAYROL
007	30 050 I				38			401REGH	PAYROL
008	30 060 I				41			442REGH	PAYROL
009	30 070 I				45			471DTH	PAYROL
010	30 080 I				48			5120TR	PAYROL
011	30 090 I			BE 10 80NCP					PAYROL
012	40 010 C	09		REGH	ADD	TOT	TOT	41	
013	40 020 C	09		GTH	ADD	TOT	TOT	41	
014	40 030 C	09		CGUNT	ADD	1	CGUNT	40	PAYROL
015	40 040 C	09		TOT	ADD	GROTOT	GROTOT	51	PAYROL
016	50 010 DRPCAT H 3301 IP								PAYROL
017	50 020 D	OR		OF					PAYROL
018	50 030 D						45	'PAYROLL REPORT'	PAYROL
019	50 040 D						68	'PAGE'	PAYROL
020	50 050 D				PAGE 2		73		PAYROL
021	50 060 D		H 1	IP					PAYROL
022	50 070 D	OR		OF					PAYROL
023	50 080 D				5			'REP.'	PAYROL
024	50 090 D				21			'EMPLOYEE'	PAYROL
025	50 100 D				42			'EMPLOYEE'	PAYROL
026	50 110 D				50			'REG.'	PAYROL
027	50 120 D				58			'REG.'	PAYROL
028	50 130 D				65			'OVT.'	PAYROL
029	50 140 D				73			'OVT.'	PAYROL
030	50 150 D				81			'TGATL'	PAYROL
031	50 160 D		H 2	IP					PAYROL
032	50 170 D	OR		OF					PAYROL
033	50 180 D				19			'NAME'	PAYROL
034	50 190 D				40			'NUMBER'	PAYROL
035	50 200 D				51			'HOURS'	PAYROL
036	51 010 D				58			'RATE'	PAYROL
037	51 020 D				66			'HOURS'	PAYROL
038	51 030 D				73			'RATE'	PAYROL
039	51 040 D				81			'HOURS'	PAYROL
040	51 050 D		D 1	09					PAYROL
041	51 060 D				DEP 2		3		PAYROL
042	51 070 D				EMPNAM		31		PAYROL
043	51 080 D				EMPNO 2		43		PAYROL
044	51 090 D				REGH		50	'C . '	PAYROL
045	51 100 D				REGH		58	' . . '	PAYROL
046	51 110 D				DTH		65	' . . '	PAYROL
047	51 120 D				QTR		73	' . . '	PAYROL
048	51 130 D				TOT H		80	' . . '	PAYROL
049	52 010 D		T 3	LR					PAYROL
050	52 020 D						31	'TOTAL CARDS PROCESSED= '	PAYROL
051	52 030 D				COUNT 2		35		PAYROL
052	52 040 D						74	'TOTAL HOURS '	PAYROL
053	52 050 D				GROTOT		80	' O. '	PAYROL

PAYROLL REPORT

DEP.	EMPLOYEE NAME	EMPLOYEE NUMBER	REG. HOURS	REG. RATE	OVT. HOURS	OVT. RATE	TOTAL HOURS
1	FRANK CAGGIANTO	111111111	35.0	5.00	5.0	7.00	40.0
2	ALBERTA SAPROFF	222111333	30.5	4.00	5.5	6.00	36.0
2	MICHAEL KUSHNER	333445566	35.4	7.00	2.0	9.00	37.5
3	DAVID ZUCARELLI	666788999	40.0	3.00	1.0	4.50	41.0
3	MERVIN DOUGLAS	555889910	30.4	5.00	5.0	7.00	35.5
3	JOHN SMITH	777000222	37.0	6.00	5.0	8.00	42.0
4	SAMUEL JACOBSON	333992227	40.0	5.00	7.0	7.00	47.0
4	MORRIS HACKERSMITH	666777821	40.0	3.00	5.0	5.00	45.0
4	ARTHUR CAGGENBURGER	111220809	41.0	4.00	7.0	6.00	48.0
4	ALVIN CHAMSDRE	776559987	39.0	4.50	1.0	5.50	40.0
4	MARK SAPROFSKY	225917937	35.0	5.50	10.0	7.00	45.0
5	AHE LEVINE	789878947	40.0	3.00	3.0	4.00	43.0
5	JAKE KELLY	457357357	37.5	4.00	5.0	5.00	42.5
5	EDD KLINE	123567794	35.0	3.00	2.0	4.00	37.0
5	MARTHA SMITH	855755655	32.0	5.00	1.0	6.50	33.0
8	DORIS AMSTERDAM	333250196	22.0	7.50	4.0	8.50	24.0
8	ERIC STEKLE	70809511	35.0	5.00	7.0	6.00	42.0
8	DONALD SCHWARTZ	357219655	40.0	4.00	8.0	5.50	48.0
8	STEVEN OSTES	111123456	40.0	4.00	2.0	5.50	42.0
8	PAULINE MACKLEY	778921789	32.0	3.00	0.0	4.00	32.0
10	PHYLLIS DAVIDOFF	135792512	31.0	5.00	1.0	6.50	32.0
10	JAMES KELLY	135892567	30.0	2.50	4.0	4.00	34.0
10	GLORIA CARANBERRY	836388833	25.0	3.00	7.0	4.50	27.0
10	SOPHIA LEVY	257153115	27.0	2.00	8.0	4.00	35.0
11	JOHN SOUTH	355532210	40.0	7.00	1.0	8.00	41.0
11	MICHAEL KLASSNER	782223140	40.0	5.00	3.0	6.00	43.0
11	ALVIN GLAZER	876521111	35.0	4.00	4.0	5.00	39.0
12	MARK TRAM	113112111	20.0	3.00	1.0	5.00	21.0
12	DENNIS COOPER	151617189	25.0	5.00	2.0	6.00	27.0
12	SHAN ALLEN	169189177	27.0	7.00	0.0	9.00	27.0
12	HERTHA STEIN	359210111	25.0	10.00	2.0	12.00	27.0
13	CAROL GOLDBERG	135137182	40.0	8.00	5.0	9.00	45.0
13	ROBERT ARANGOFF	252426277	37.0	4.00	3.0	5.00	40.0
13	MAVIS CARPENTER	778965170	36.0	3.00	3.0	6.00	45.0
13	BLAINE DANDELL	359789100	35.5	3.50	7.0	4.50	42.5
14	KATHERINE ZULLER	382510101	38.5	5.50	1.0	6.50	39.5
15	DONALD FRANKEL	898786854	35.0	6.50	10.0	7.50	45.0
15	MIKE DONNELL	433982155	37.5	8.00	2.0	9.00	39.5
16	STEVE WATSON	787654311	38.4	4.00	3.0	5.00	41.5
16	LARRY GIBBS	158173666	40.5	3.00	5.0	5.00	45.5
17	LOUIS GRACEN	636578913	40.0	7.00	5.0	8.00	45.0
17	DAVID CASTEN	259925988	40.0	5.00	6.0	6.00	46.0
19	ELLEN LAWRENCE	835708082	37.5	4.00	3.0	5.00	40.5
19	CATHY ANCHOR	359444467	35.5	5.00	1.0	7.00	36.5
19	MARY SUSSKIND	666783213	35.0	4.00	2.0	5.00	37.0
20	JANE GALLY	877787739	35.0	3.00	5.0	4.50	40.0
20	DORIS CARTER	251782359	37.0	4.50	7.0	6.00	44.0
20	FRANCIS JONES	621354922	36.0	5.00	2.0	7.00	38.0

TOTAL CARDS PROCESSED= 48

TOTAL HOURS 1864.0

หมายเหตุ เราสามารถใช้ LR ทั้งใน C-Form และ O-Form ได้ด้วย

การพิมพ์จำนวนเลขหน้าของรายงาน (Page Numbering)

ลักษณะของแบบรายงานที่พิมพ์ออกมานั้นนอกจากประกอบด้วยรายละเอียดของแหล่งที่ของรายงานเช่น บริษัท หน่วยงาน เป็นรายงานเรื่องอะไร แล้ว บางครั้งยังมีความจำเป็นที่จะต้องให้มีการพิมพ์รายละเอียดในเรื่อง เลขที่ของหน้าแบบรายงานนั้นว่าเป็นหน้าที่เท่าไร ทั้งนี้เพื่อความสะดวกในกรณีที่จำนวนรายงานมีมากทั้งนี้เพื่อเอื้อความสะดวกให้ในเวลาที่ต้องมีการค้นหารายละเอียดบางอย่าง

ตัวอย่างแบบรายงานที่แสดงเลขหน้าของกระดาษรายงานประกอบด้วย

3 F COMPANY

PAGE001

PAYROLL REPORT

JANUARY 1976

ซึ่งในส่วนที่เป็นเลขหน้าของรายงานนั้นเราสามารถจะสั่งให้พิมพ์ออกมาได้โดย 2 กระบวนการคือ

กระบวนการที่ 1. เขียนคำสั่งในการคำนวณเลขหน้าใน C-Form โดยเริ่มต้นจากหน้าที่ต้องการ เช่น 1 และเพิ่มขึ้นอีกทีละ 1 หน้าเมื่อมีการพิมพ์ไปแล้วหมดหน้ากระดาษ ทั้งนี้ในการคำนวณแบบนี้จะถูกคุมด้วย Indicator ที่จะโยงไปให้สัมพันธ์กับการพิมพ์รายงานด้วย

กระบวนการที่ 2. เป็นวิธีการที่ง่ายกว่าแบบที่ 1 มากเพราะได้มีการเจาะจงไว้ล่วงหน้าแล้ว ในภาษานี้ โดยที่มีการกำหนด field หนึ่งที่ชื่อว่า PAGE เพื่อทำหน้าที่สำหรับกำหนดเลขเพื่อวัตถุประสงค์ดังกล่าว ดังนั้น ถ้าผู้ใช้ต้องการจะใช้หมายเลขหน้าก็สามารถเรียก field ที่ชื่อว่า PAGE มาใช้ได้ทันทีเลย

ลักษณะของข้อมูลที่เก็บไว้ใน field จะเป็นข้อมูลที่เป็นเลขจำนวนเต็มมีจำนวน 4 หลัก ถ้าเป็นเลขหน้ารายงานหน้าแรกก็จะมีค่าเป็น 0001 ใน field เป็นต้น ถ้าเราไม่ต้องการเลขศูนย์ที่อยู่ข้างหน้า ก็สามารถทำได้โดยการใช้ Edit code

ตัวอย่างการใช้คำสั่ง PAGE

IBM

International Business Machines Corporation

RPG OUTPUT FORMAT SPECIFICATIONS

Date _____

Program _____

Punching Instruction	Graphic				
	Punch				

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Skip				Output Indicators				Field Name	Edit Codes				Starting Sign Position		
			Before	After	Before	After	And	And	Commas	Zero Balances to Print		No Sign	CR	-	K			
01	C	REPORT																
02	O																	
03	O									PAGE								
04	O																	
06	O																	
08	O																	
09	O																	
10	O																	
11	O																	
12	O																	
13	O																	
14	O																	
15	O																	

การพิมพ์วันที่แบบรายงาน (การใช้ UDATE)

ในการทำงานเดียวกันกับที่เราใช้ PAGE ในการพิมพ์เลขหน้า เราก็สามารถใช้ field name ที่ชื่อว่า UDATE ในการพิมพ์ข้อมูลเกี่ยวกับวันที่ทำรายงานอยู่นั้นได้ โดยอาศัยการใช้ประโยชน์จากสิ่งที่กำหนดไว้ในภาษานี้แล้ว โดยที่ UDATE คือ field name ซึ่งมี format ดังนี้คือ ประกอบด้วยเลขหน้า 6 ตำแหน่ง โดยที่ 2 ตำแหน่งแรกเป็นเลขที่หมายถึงวันที่ 2 ตำแหน่งถัดมาหมายถึงเดือน และ 2 ตำแหน่งสุดท้ายหมายถึงปี ดังนั้นการใช้คำสั่งใด ๆ ใน C-Form หรือ O-Form โดยอ้างถึง-UDATE ก็จะถือเสมือนว่า UDATE ดังกล่าวทำหน้าที่เป็น date field เช่นเดียวกับ data field อื่น ๆ เพียงแต่ขอให้เข้าใจถึงลักษณะของข้อมูลที่อยู่ใน UDATE นี้เพื่อที่จะได้นำไปใช้ได้ถูกต้อง นอกจากนี้เรายังสามารถใช้คำสั่งแยกย่อยใน data field นี้ได้โดยการใช้ UMONTH ซึ่งหมายถึงเดือน UDAY ซึ่งหมายถึง วันที่ และ UYEAR ซึ่งหมายถึง ปี โดยที่ field name พวกนี้ก็เหมือน reserve word นั่นเอง

ตัวอย่างการใช้ reserve word กลุ่มนี้ในการเขียนคำสั่ง



International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 of 2
 Program Identification 75 76 77 78 79 80

Line	Form Type	Control Point (LD, LR, LP, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions (H)	Resulting Indicators			Comments
			And	And	Not							Arithmetic	Plus	Minus	
01	C	*				NOTE -	UPDATE	BEING	USED	TO	COMPARE	TO	CARD	INPUT	DATE.
02	C					DATE	COMP	IN	DATE					10	DATE.
03	C	A				NOTE -	YEAR	BEING	USED	TO	COMPARE	TO	CARD	INPUT	YEAR.
04	C					YEAR	COMP	IN	YEAR						



International Business Machines Corporation

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 of 2
 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (M/D/E)	Space	Skip	Output Indicators			Field Name	Edit Codes	Storing Sign Position
						Before	After	Not			
01	O	REPORT	H						DATE		
02	O								DATE		
03	O								DATE		
04	O								DATE		
05	O								DATE		
06	O								DATE		
07	O								DATE		

แบบฝึกหัด

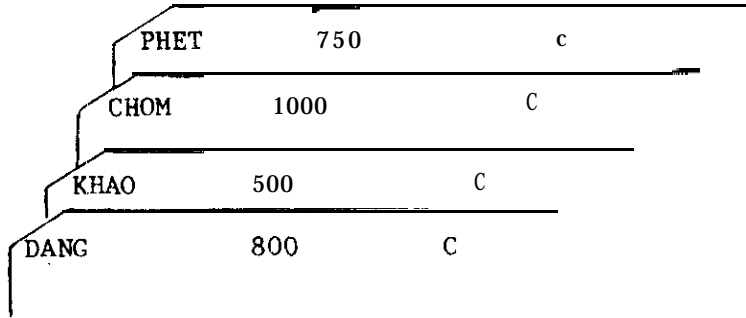
1. จงอธิบายความหมายในการใช้ Z-ADD, Z-SUB พร้อมทั้งยกตัวอย่าง
2. จุดประสงค์และประโยชน์ของการใช้ Test zone คืออะไร
3. คำสั่ง TAG ใช้เพื่อทำอะไร
4. ข้อควรระวังในการใช้คำสั่ง GOTO และ TAG มีอะไรบ้าง
5. MOVE และ MOVEL แตกต่างกันอย่างไรร
6. บริษัท REB ต้องการที่จะหายยอดรวมของ Account Receivable โดยที่ข้อมูลใน field ดังกล่าวนี้ถ้ามีค่าเป็นลบก็ให้เปลี่ยนเป็นบวกเสียก่อนแล้วจึงนำไปหายยอดรวม ให้เขียนคำสั่งใน C-Form โดยที่ให้พิมพ์ผลที่ได้ในกระดาษรายงานโดยวางรูปแบบตามความเหมาะสม โดยมีเงื่อนไขของ Input Record ดังนี้

MULTIPLE LAYOUT FOAM

CUST. NUMBER	NAME	ACCOUNT RECEIVABLE	SISMAN	
99999 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80		xxxxxx,xx 99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
10010	JOHN CASTORELLI	00109867	79211	
99999 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80		99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
10020	JAKE ZUCARELLA	1000010	25911	
99999 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80		99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
10050	STEVE ZUCKERMAN	00010	73591	
99999 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80		99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
10070	MORRIS SCHWARTZ	3000203	28761	
99999 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80		99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
10090	ZOE PASTERNAK	400000	63881	
99999 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80		99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	99999 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	

Multiple Record Input Processing

ในบทที่แล้วได้กล่าวถึงกรณีที่มี Input File มี Record ชนิดเดียวกัน เช่น



เราจะอธิบาย Input File ได้ดังนี้คือ

I-Form

IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction: Graphic Punch

Page: 1 2

Program Identification: 75 76 77 78 79 80

Line	Form Type	Filename	Sequence Number (LN)	Number (LN)	Display (O)	Record Identifying Indicator	Record Identification Codes			Field Location		Field Name	Control Level (L, B)	Matching Fields or Changing Fields	Field Record Relation	Field Indicators			Sterling Sign Position		
							Position	Not (N)	Character	Position	Not (N)					Character	From	To		Plus	Minus
0 1	1	FILE	AA	10	80	00															
0 2	1																				
0 3	1																				
0 4	1																				

ปัญหาที่เกิดขึ้นก็คือถ้ามี Record หลาย ๆ ชนิดใน Input File เดียวกัน เราจะมีวิธีการเขียนคำสั่งใน I-Form อย่างไร ถ้าเราจะคัดแยก Record แต่ละชนิดออกจากกันและแยกอธิบายเป็นชุด ๆ ไปของ Record แต่ละชนิดยอมทำได้ ดังตัวอย่างต่อไปนี้ ซึ่งจะเห็นว่าเป็นวิธีการที่ค่อนข้างจะไม่มีประสิทธิภาพ และยาวเกินความจำเป็นอีกด้วย

RPG INPUT SPECIFICATIONS

Line	Filename	Sequence	Record Identifying Indicator	Record Identification Codes			Field Location		Field Name	General Purpose	Field Indicators			Starting Position
				Position	Not IN Character	Position	Not IN Character	Position			Not IN Character	From	To	
0.1	RECEIV	MM	46	80	CM									
0.2								1	6	50PNUM				
0.3								25	6	NNAME				
0.4								45	6	NNSTR1				
0.5								58	6	NNCITY				
0.6								60	6	NNSTATE				
0.7								65	6	NNZIP				
0.8								75	6	NNAMT				
0.9								79	6	NNDATE				
1.0								1	6	50PNUM				
1.1								25	6	NNNAME				
1.2								45	6	NNSTR1				
1.3								58	6	NNCITY				
1.4								60	6	NNSTATE				
1.5								65	6	NNZIP				
1.6								75	6	NNAMT				
1.7								79	6	NNDATE				

RPG INPUT SPECIFICATIONS

Line	Filename	Sequence	Record Identifying Indicator	Record Identification Codes			Field Location		Field Name	General Purpose	Field Indicators			Starting Position
				Position	Not IN Character	Position	Not IN Character	Position			Not IN Character	From	To	
0.1	SA	48	80	CP										
0.2								1	6	50PNUM				
0.3								25	6	PNAME				
0.4								45	6	PSTR1				
0.5								58	6	PCITY				
0.6								60	6	PSTATE				
0.7								65	6	PZIP				
0.8								75	6	PAMT				
0.9								79	6	PDATE				
1.0								1	6	50PNUM				
1.1								25	6	CNAME				
1.2								45	6	CSTR1				
1.3								58	6	CCITY				
1.4								60	6	CSTATE				
1.5								65	6	CZIP				
1.6								75	6	CAMT				
1.7								79	6	CDATE				

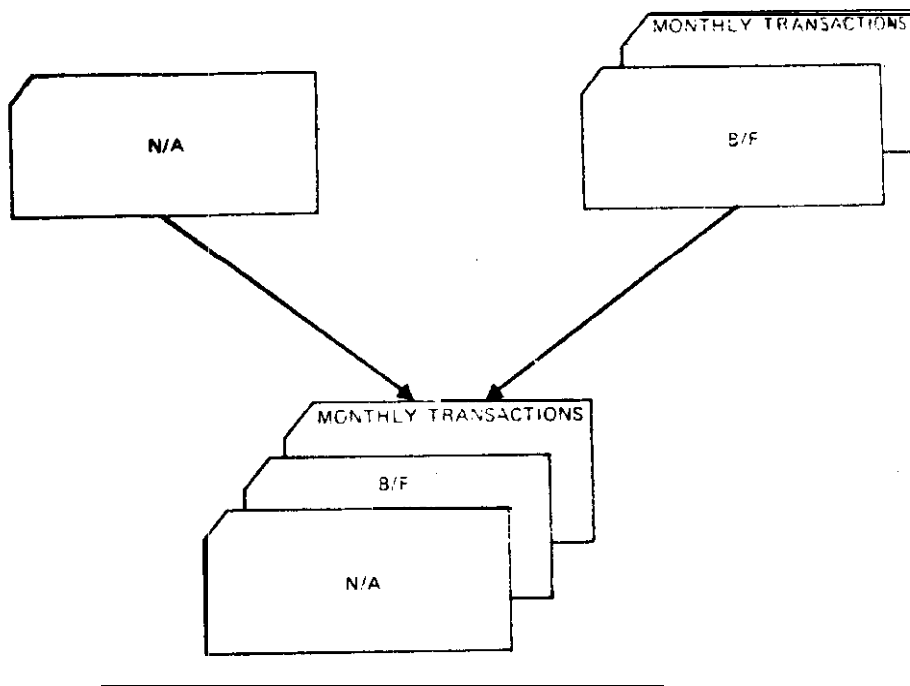
เนื่องจากวิธีการดังกล่าวที่ยกตัวอย่างมานี้ค่อนข้างจะยุ่งยากเราอาจใช้ OR เข้ามาช่วยได้เพื่อให้มีการเขียนคำสั่งง่ายเข้า แต่มีข้อแม้ว่า format ของ Input Record แต่ละชุดจะต้องเหมือนกันดังตัวอย่างเช่นนี้เป็นต้น ถ้า format ของ Input เกิดไม่เหมือนกันเราก็ไม่สามารถใช้ OR ได้

I-Form

10 IINFILE	AA	46	80	CN
20 I	OR	47	80	DF
30 I	OR	48	80	CP
40 I	OR	49	80	CC
50 I				1 50NUM
60 I				6 25 NAME
70 I				26 45 STRT
80 I				46 58 CITY
90 I				59 60 STATE
100 I				61 650ZIP
110 I				66 752AMT
120 I				76 790DATE

ในงานบางชุดที่ต้องอาศัย Record หลายชนิดใน Input File เดียวกัน โดยที่ Record เหล่านั้นมีลักษณะแตกต่างกันออกไป ไม่สามารถอธิบาย Type ของ Record ได้ภายใน format เดียวกัน หรือในกรณีที่ต้องการให้มีการตรวจสอบลำดับของ Record แต่ละชนิดใน Input File นั้นหมายความว่าเรากำลังศึกษา Input File ที่มี Record หลาย ๆ ชนิด (Multiple Record) ซึ่งก็จะเปรียบเทียบได้กับการนำ Record แต่ละชุดจากแต่ละ Input File มารวมกันเป็น Input File เดียวกันโดยที่ Input File นี้มีหลาย ๆ Record

ตัวอย่างการรวม file 2 ชุด ให้เป็น data file ชุดเดียวกันนั่นเอง



จาก Multiple Record ใน Input File ที่ได้นี้ ถ้าหากไม่มีการตรวจสอบลำดับ (Sequence) ของแต่ละ-Record เราจะเขียนคำสั่งได้ในแบบเดียวกับที่เคยใช้ในตัวอย่างที่แล้วมาโดยการพิจารณาเพียงแต่ว่าแต่ละ Record มี format แตกต่างกันหรือไม่เท่านั้น แต่ถ้าหากมีเงื่อนไขนำไปสู่การ process เพื่อให้ Output ออกมาดังตัวอย่างต่อไปนี้

Record แรกจะทำให้เกิดผลที่ Output รูปของการเก็บ Address Record ที่ 2 จะต้องเป็น Record ของ Balance Foreword ส่วน Record ต่อไปจะเป็นรายการที่เปลี่ยนแปลงในแต่ละวัน (มาจาก MONTHLY TRANSACTIONS Records)

การเขียนคำสั่งอธิบายแต่ละ Record ตามเงื่อนไขในลำดับก่อนหลังจะเป็นดังนี้คือ

I-Form

```

10 IINFILE      0146 80 CN
20 I                                DATA
30 I                                DESCRIPTION
40 1            02 47 80 CF
50 I                                DATA
60 I                                DESCRIPTION
70 1            03 48 80 CP
80 I            OR 49 80 CC
90 I                                DATA
100 I           DESCRIPTION
  
```

จากคำสั่งมี Input File ชื่อว่า INFILE ซึ่งมี Record อยู่ 3 ชนิด ที่จะต้องอยู่เรียงลำดับคือ- Record ลำดับแรกมี N ที่สคมภ์ที่ 80 มี Record Indicator 46 ลำดับที่ 2 คือ Record ที่มี F ที่สคมภ์ที่ 80 ใช้ Indicator 47 ส่วน Record ลำดับที่ 3 มีอยู่ 2 ชนิดคือ ชนิดแรกมี P ที่สคมภ์ที่ 80 และใช้ Record Indicator 48 ส่วน Record ชนิดที่ 2 มี C ที่สคมภ์ที่ 80 ใช้ Record Indicator 49 ลำดับที่ใช้ในการตรวจสอบ Record คือ Sequence Code ซึ่งจะเริ่มที่ 01,02,03 เป็นตัวเลขที่ใช้ตรวจสอบ ซึ่งพิจารณาเทียบกับ Record ก็คือ Record ลำดับแรกที่เข้ามาก็คือ N/A (Name and Address) Record ลำดับที่สองก็คือ B/F (Balance Foreword) ส่วน Record ลำดับที่สาม (ลำดับสุดท้าย) ก็คือ- Monthly Transaction ซึ่งเครื่องจะตรวจสอบว่าลำดับของ Record เหล่านี้เข้ามาตามลำดับที่กำหนดไว้ตามคำสั่งในโปรแกรมหรือไม่ ถ้าหากลำดับผิดที่ไปจากที่ปงไว้จะเกิด error ขึ้น

จากตัวอย่างของ Input File นี้ถ้าเรากำหนดเงื่อนไขเพิ่มเติมว่าจำนวน Record ในแต่ละลำดับที่เข้ามา มีจำนวนเท่าไร เราจะต้องใช้สคมภ์ที่ 17 เป็นเงื่อนไขในการสั่งโดยที่ถ้ากำหนดให้ใช้ Code 1 หรือ N หรือ blank ในส่วนที่เป็น field number โดยที่แต่ละ Code มีความหมายดังนี้

1 = แสดงว่า Record ชนิดนี้ปรากฏใน Input File เพียง 1 Record เท่านั้น

N = แสดงว่า Record ชนิดนี้อาจมีปรากฏใน Input File ตามลำดับ (Sequence) ที่กำหนดมากกว่า 1Record ก็ได้

blank ใช้ในกรณีที่ Sequence ในสคมภ์ที่ 15-16 เป็น Alphabetic คือไม่มีการตรวจสอบลำดับของ Record ที่เข้ามา ถ้าเป็นการประมวลผล ดังนั้นจำนวน Record จะมีเท่าไรก็ได้

ข้อสังเกต ใน Sequence ถ้าเป็น Numeric แล้ว field ของ Number จะต้องเป็นเลข 1 หรือ N เท่านั้น นอกจาก Field Number จะเกี่ยวข้องกับจำนวน Record ที่เข้ามาในแต่ละลำดับแล้ว ยังมี field อื่นที่เข้ามาเกี่ยวข้องก็คือ field option (สคมภ์ที่ 18) field นี้ใช้สำหรับบ่งว่า Record ชนิดนั้นอาจจะมีหรือไม่มีก็ได้ วิธีการใส่ Code ก็คือ ถ้าใส่ 0 ใน field นี้จะหมายถึงว่ามี Record ชนิดนั้นหรือไม่มีก็ได้ แต่ถ้าใส่ blank ก็หมายความว่าต้องมี Record ชนิดนั้นปรากฏอยู่ตามจำนวนที่บ่งไว้ใน field number (นั่นหมายความว่าใส่ 0 ใน field นี้ได้ก็ต่อเมื่อมีการตรวจ Sequence ของ Record หรือสรุปสั้น ๆ ก็คือ Sequence เป็นตัวเลขนั่นเอง)

จากตัวอย่างที่ยกมาให้ดู ถ้าเรากำหนดว่า Record N/A เข้ามาเป็นลำดับแรกโดยที่ Record ชนิดนี้จะต้องมีปรากฏและมีเพียง 1Record เท่านั้น Record ที่เข้ามาเป็นอันดับที่ 2 ก็คือ Record-B/F โดยที่ Record ชนิดนี้จะมีหรือไม่มีก็ได้ ถ้ามีก็มีเพียง Record เดียวเท่านั้น ส่วน Record ชนิดนี้จะมีอยู่ 2 แบบคือแบบที่ 1 มี P ที่สคมภ์ที่ 80 แบบที่ 2 มี C ที่สคมภ์ที่ 80 โดยที่ MONTHLY TRANSACTION จะมีอยู่ที่ Record ก็ได้ หรือไม่มีก็ได้ ดังนั้น เราจะเขียนคำสั่งของ Input File ซึ่งประกอบด้วย Record ต่าง ๆ ที่กำหนดได้ดังนี้คือ

I-Form

10 I	INFILE	01	1	46	SO CN
20 I					
30 I					
40 I		02	1	47	80 CF
50 I					
60 I					
70 I					
80 I		03N0		48	80 CP
90 I		OR		49	80CC
100 I					
110 I					
120 I					
130 I					

กฎเกณฑ์เพิ่มเติมเกี่ยวกับการเขียนคำสั่งใน I-Form เมื่อ Input File มี Record หลายชนิดก็คือ ถ้า Record ต้องอาศัยการตรวจสอบลำดับทุก ๆ Record ก็ดำเนินงานตามตัวอย่างที่ให้ แต่ถ้าเกิดมี Record ชนิดที่ไม่ต้องการตรวจสอบลำดับเพิ่มเข้ามาเช่น ถ้าเราเพิ่ม Record ที่ป่ง วันที่ เข้าไปใน Input File เดิม (ในตัวอย่างที่แล้ว) อีกโดยที่ Record ที่เพิ่มเข้ามาใหม่ ไม่จำเป็นต้องมีการตรวจสอบลำดับ (Sequence Code คือ Alphabetic)

ดังนั้นการเขียนคำสั่งใน I-Form จะต้องมีการเปลี่ยนแปลงไปดังนี้คือ Record ชนิดที่เป็น Alphabetic Code จะต้องมาก่อน record ที่เป็น Numeric Code
การเขียนคำสั่งจะเป็นดังนี้คือ

RPG INPUT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Number (1-5) Control ID	Record Identifying Indicator	Record Identificat on Codes									Field Location		Field Name	Control Level (1-4) Matching Factor or Changing Factor	Field Record Position	Field Indicators			Starting Skip Position		
						1			2			3			From	To				Plus	Minus	Zero or Blank			
						Position	Not (N) Character	C/Z/D	Position	Not (N) Character	C/Z/D	Position	Not (N) Character	C/Z/D											
01	I	RECGRP	AA	40	PO	C*																			
02	I																								
03	I																								
04	I																								
05	I		01	48	PO	EM																			
06	I																								
07	I																								
08	I																								
09	I				01	47	PO	EM																	
10	I																								
11	I																								
12	I																								
13	I				01	48	PO	EM																	
14	I		OR	49	PO	EM																			
15	I																								