

บทที่ 10 : Tables

บทนี้จะเป็นเรื่องเกี่ยวกับการจัดเตรียม Input data ให้อยู่ในรูปที่สะดวกแก่การประมวลผล และสามารถประหยัดทั้งเวลาและค่าใช้จ่ายด้วย แต่ก่อนที่จะกล่าวถึงวิธีการดังกล่าวจะขอกล่าวถึงสิ่งที่เกี่ยวข้องกับวิธีการดังกล่าวเพื่อให้เข้าใจความหมาย และแนวทางของสิ่งเหล่านี้เสียก่อน เพื่อเป็นการปูพื้นฐานไปยังเรื่องที่จะใช้ต่อไป

ตาราง (Table) ก็คือชุดหนึ่งของข้อมูล (Informations) ซึ่งมีความเกี่ยวพันกัน ข้อมูลที่กล่าวถึงนี้อาจเป็น Numeric หรือ Alpha ก็ได้ โดยปกติแล้วตารางจะประกอบด้วยข้อมูล 2 ส่วน ส่วนที่ 1 เรียกว่า Argument (Index) ส่วนที่ 2 เรียกว่า Function (Content) โดยที่ Argument มีหน้าที่จะเป็นส่วนที่ชี้ว่าความหมายของมันคืออะไร ความหมายในที่นี้ก็คือ Function นั้นเอง หรืออาจจะกล่าวได้อีกนัยหนึ่งว่า Argument ก็คือส่วนที่แสดงหรือชี้ความหมายของเนื้อหาที่เราต้องการทราบนั่นเอง ยกตัวอย่างเช่น

01	หมายถึง	January
02	หมายถึง	February
.
..
12	หมายถึง	December

นั่นหมายความว่า ถ้าเราบ่งว่าเดือน 01 ก็หมายถึงเดือน January นั้นเอง ดังนั้นตารางที่เราสร้างก็จะเขียนได้ดังนี้คือ

Argument	Function
01	January
02	February

Argument	Function
03	March
04	April
05	May
06	June
07	July
08	August
09	September
10	October
11	November
12	December

รหัสของแต่ละเดือน (ตัวเลข 01-12) ก็ทำหน้าที่เป็น Argument โดยที่ความหมายของตัวเลขที่จะโยงไปรู้ว่าเป็นเดือนอะไร ก็คือ Indicator นั้นเอง จากจุดนี้ผู้อ่านคงจะพอนึกภาพออกถึงการนำประโยชน์ในการสร้างตารางไปใช้กับการเขียนคำสั่งเพื่อให้เครื่องทำรายงานได้โดยง่ายขึ้น ตัวอย่างเช่น ถ้าต้องการให้พิมพ์รายงานออกมาในรูปของเดือนว่าเป็นเดือน January หรืออื่น ๆ นั้น เราก็ไม่ต้องป้อนข้อมูลเข้าไปเป็นชื่อเดือนเลย เพียงแต่ใช้รหัสแทนว่าเป็นรหัสอะไรตั้งแต่ 01-12 เมื่อรู้รหัสแล้วก็ให้เครื่องนำรหัสไปเทียบกับตารางที่เราสั่งให้เก็บไว้ในหน่วยความจำ ก็จะได้ค่าของ function ออกมาโดยอาศัย Argument เป็นตัวชี้เสนอ วิธีการนี้เราจะประยุกต์ให้นำไปใช้กับข้อมูลอย่างอื่นที่พอจะจัดเป็นตารางได้ ยกตัวอย่างเช่น ถ้าเราจะสร้างตารางในการคิดอัตรา ชั่วโมงในการทำงานของคนงานในหน่วยงานใด ๆ โดยอาศัยระดับของคนงานเป็น Argument ก็จะช่วยให้เราประหยัดค่าใช้จ่ายในการเตรียมข้อมูลเพราะถ้ามีคนงานอยู่ในระดับเดียวกันมาก ๆ เราจะอาศัยตารางเดียวแต่ใช้ได้กับทุกคนเลย โดยไม่ต้องเจาะข้อมูลที่เกี่ยวกับอัตรา ชั่วโมงในการทำงานให้กับคนงานแต่ละคน

การที่จะให้เครื่องคอมพิวเตอร์รับตารางที่จะใช้ไปเก็บก่อน ในสมองนั้น หมายความว่าก่อนที่จะมีการ process input data นั้น เราจำเป็นต้องอาศัยวิธีการจัดเตรียมคำสั่งเพื่อที่จะให้เครื่องรับตารางเข้าไปก่อนซึ่งการที่จะให้เครื่องอ่านตารางเข้าไปนั้น เราจะจัดกระทำกับตารางโดยให้ถือเสมือนว่า

ตารางก็คือ Input File อันหนึ่งนอกเหนือจาก Input File ที่เราจะ process ปัญหาที่ตามมาก็คือจะมีวิธีการอย่างไรที่จะให้เครื่องรับรู้ตารางที่ป้อนเข้าไปเป็น File ที่มีลักษณะแตกต่างจาก Input File ที่จะ process เพราะตารางจะเป็น File เฉพาะที่เก็บไว้เพื่อประโยชน์ในการ process ของ Input File อื่นโดยที่ตัวของมันเองอยู่เฉย ๆ การที่จะแยกความแตกต่างของ Table File กับ File อื่น ๆ ทำได้โดยใช้ Extension Specification เข้าช่วยโดย E-Form คือกลุ่มของคำสั่งที่จะตามมาเป็นอันดับต่อจาก F-Form วิธีการเขียนคำสั่งในกรณีที่มีการใช้ตารางก็คือ ให้ถือว่าตารางก็คือ Input File อันหนึ่ง ดังนั้นจึงต้องมีการป้อนชื่อของ Table ใน File นอกเหนือจาก Input File และ Output File อื่น ๆ

หลักการใช้ Table ให้ถือว่าเป็น Input File ตั้งชื่อโดยใช้เงื่อนใจของการตั้งชื่อที่เคยกล่าวมาแล้ว ปกติเป็น Input File ส่วนในสคมภ์ที่ 16 ให้ใช้ T เพราะว่าเป็น Table และสคมภ์ที่ 39 ให้ใส่ "E" เพราะจะมีการอธิบาย file นี้ใน E-Form อีกครั้งหนึ่ง ส่วนรายละเอียดอื่น ๆ ก็เหมือนกับ Input File ที่เคยกล่าวมาแล้วทุกประการ (กลับไปดูเรื่องการใช้ F-Form) ตัวอย่างการใช้คำสั่งแสดงถึงการใช้ Table และความสัมพันธ์ของการใช้ F-Form กับ E-Form

1. Extension Specification = E-Form

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date, Program, Punching Instruction, Graphic Punch, Page, Program Identification

Control Card Specifications

Table with columns for Line, Form Type, Core Size to Complete, Core Size to Execute, etc.

File Description Specifications

Table with columns for Line, File Name, File Type, File Description, Mode of Processing, Device, Symbolic Device, etc.

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Date, Program, Punching Instruction, Graphic Punch, Page, Program Identification

Extension Specifications

Table with columns for Line, Form Type, Record Sequence of the Chaining File, To Filename, Table or Array Name, etc.

Line Counter Specifications

Table with columns for Line, Form Type, Filename, Line Number, Channel Number, etc.

การเขียนคำสั่งใน E-Form ในสคมภ์ที่ 6 ของ E-Form จะต้องป่งถึงชนิดของมันโดยการใช้ E ในทุกคำสั่ง

สคมภ์ที่ 11-18 คือ From File Name ให้เขียนชื่อของ Table File ที่ป่งไว้ใน F-Form

สคมภ์ที่ 27-32 คือ Table Name คือชื่อของตารางโดยมีหลักเกณฑ์ในการตั้งชื่อว่า Table Name ต้องขึ้นต้นด้วย TAB... อีก 3 characters ถัดไปจะเป็นอะไรก็ได้หรืออาจจะไม่ใช่ครบทั้ง 6 characters ก็ได้ เช่นอาจจะตั้งชื่อว่า TABARG, TABA1, TAB1, TAB# เป็นต้น

สคมภ์ที่ 36-39 เป็นส่วนที่จะป่งถึงจำนวน entries ในตาราง ตัวอย่างเช่น ตารางประกอบ ด้วย

01

02

12

ดังนั้นตารางนี้มีจำนวน 12 entries

หมายเหตุ ในตัวอย่างนี้จะกล่าวเฉพาะตารางที่มีเฉพาะ Argument ก่อน ส่วนตารางที่ประกอบด้วย ทั้ง Argument และ Function จะกล่าวถึงในภายหลัง

สคมภ์ที่ 40-42 เป็นส่วนที่อธิบายถึงความยาวของแต่ละ entry ในตารางนั้น ดังนั้นจาก ตัวอย่างที่กล่าวถึงแล้วจะมีความยาว 2 ตำแหน่ง (01-12) ใน 1 entry การที่ต้องป่งจำนวน entries และความยาวของแต่ละ entries นั้น ก็เพื่อที่จะป่งถึงจำนวนหน่วยความจำที่จะต้องเก็บไว้ใน ตาราง

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

Extension Specifications

Line	Form Type	Record Sequence of the Chaining File		To Filename	Table or Array Name	Number of Entries Per Rowset	Number of Entries Per Table or Array	Length of Entry	P = Packed (B = Binary) Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed (B = Binary) Chaining Positions Sequence (A/D)	Comments
		Number of the Chaining Field	From Filename										
0 1	E				TABLE	1	1	1	X	CA			
0 2	E												
0 3	E												
0 4	E												
0 5	E												
0 6	E												
0 7	E												
0 8	E												
0 9	E												
1 0	E												

Line Counter Specifications

Line	Form Type	Filename	1		2		3		4		5		6		7		8		9		10		11		12	
			Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number
1 1	L																									
1 2	L																									
1 3	L																									

สคมภ์ที่ 44 ใช้เป็นตำแหน่งเพื่อบ่งทศนิยมในแต่ละ entry (แต่ละ entry ต้องมีจำนวนทศนิยมเท่ากัน) ในกรณีที่มีข้อมูล Alpha ให้เว้นว่างในสคมภ์ที่ 44 แต่ถ้าเป็น Numeric ต้องบ่งทศนิยมด้วย (0-9) ถ้าไม่มีทศนิยมก็ต้องใส่ศูนย์ ข้อจำกัดก็คือ field size ของ entry จะต้องไม่เกิน 15 ตัว (ในกรณีของข้อมูลชนิด Numeric)

สคมภ์ที่ 45 ใช้เพื่อบ่งว่าตารางที่ใช้เป็นตารางที่ใช้เป็นตารางในลักษณะใด เช่น ถ้าเป็นตารางที่เรียง entry จากน้อยไปหามาก (Ascending Order) ก็ให้ใส่ A ในส่วนนี้ แต่ถ้าเป็นตารางที่เรียง entry จากมากไปหาน้อย (Decending Order) ก็ให้ใส่ D ในส่วนนี้ แต่ถ้าไม่มีการเรียงลำดับในลักษณะใดเลย ก็ให้เว้นว่างส่วนนี้ไว้

ตัวอย่างของข้อมูลที่มีการจัดเรียงในลักษณะต่าง ๆ

ข้อมูลในลักษณะเช่นนี้ก็จะมีความหมายว่า 1 entry บรรจุใน 1 Record ดังนั้นคำสั่งที่สมบูรณ์ใน E-Form สำหรับข้อมูลชุดนี้ก็คือ

IBM

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Extension Specifications

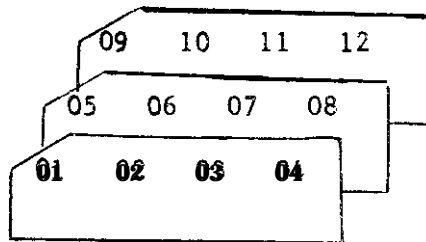
Line	Form Type	Record Sequence of the Chaining File		To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	Table or Array Name (Alternating Format)	Length of Entry	Comments
		Number of the Chaining Field	From Filename								
01	E		FILETABLE		TABLE	1	1	1			
02	E										
03	E										
04	E										
05	E										
06	E										
07	E										
08	E										
09	E										
10	E										

Line Counter Specifications

Line	Form Type	Filename	1		2		3		4		5		6		7		8		9		10		11		12	
			Line Number	F.L. or Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number
11	L																									
12	L																									
13	L																									

การสร้างข้อมูลในลักษณะที่ 1 Record บรรจุ 1 entry จะเป็นวิธีการที่ค่อนข้างจะฟุ่มเฟือย ดังนั้นเราอาจจะลดจำนวน data card แทนที่จะใช้ถึง 12 ใบ เราอาจใช้ data card 1 ใบบรรจุหลาย ๆ entries ก็ได้

ตัวอย่างเช่น



ดังนั้นจะได้ 1 Record ประกอบด้วย 4 entries

หลักเกณฑ์การใช้เมื่อ 1 Record บรรจุได้มากกว่า 1 entry

1. Argument แรกของแต่ละ Record ต้องเริ่มที่สคมภ์ที่ 1 เสมอ
2. ทุก ๆ entry ต้องมีความยาวเท่า ๆ กัน
3. ทุก ๆ entry จะต้องต่อเนื่องกัน ไม่อนุญาตให้ 1 entry อยู่ต่าง Record กัน
4. Record แต่ละ Record จะต้องมีจำนวน entry เท่า ๆ กัน ยกเว้น Record สุดท้ายอาจจะมีจำนวน entry น้อยกว่าได้
5. ข้อมูลของ entry ที่จะ load เข้าไป จะต้องมีจำนวน entry ในตารางเท่ากับที่กำหนดไว้ในสคมภ์ที่ 36-39 ของ E-Form
6. entry เดียวกันจะต้องอยู่ใน Record เดียวกัน จะแยกออกมาอยู่คนละ Record ไม่ได้
7. ถ้ากำหนด L หรือ R ในสคมภ์ที่ 43 ของ E-Form จะมีความหมายว่าความยาวของ entry ที่บ่งไว้ใน E-Form นั้นจะรวมเครื่องหมายเข้าไปด้วย
8. Format ของตารางจะต้องเป็น Fixed Format เสมอ ดังนั้น สคมภ์ที่ 10 ของ F-Form จะต้องเป็น F สำหรับการใส่ตาราง (Table)

หมายเหตุ ถ้าสคมภ์ที่ 43 ใส่ P หรือ B ความหมายก็คือข้อมูลเป็น Packed หรือ Binary Format ซึ่งจะ loaded ในช่วงของ Object Time

จุดประสงค์ที่สำคัญของการใส่ตารางก็คือการเปลี่ยนแปลงข้อมูล โดยใช้ Argument โยงไปสู่ Function ที่เราต้องการ แต่จากตัวอย่างที่ยกมาจะเห็นว่าเราเพียงแค่สร้างตารางของ Argument (รหัสของเดือนเท่านั้น ตามตัวอย่างที่เคยยกมาให้ดู) แต่ค่าของ Argument คือ Function เรายังไม่ได้สร้าง วิธีการสร้าง Function นี้สามารถกระทำได้ 2 วิธีคือ

วิธีที่ 1 เรียกว่าการสร้าง Alternating Table สร้าง Function เป็นตารางอันใหม่หรือถือเป็นตารางที่ 2 ต่อจากการสร้างตารางของ Argument จากตารางเดิมเราจะสร้างได้ 2 ตารางใหม่ดังนี้คือ

FILE1 TABLE	FILE2 TABLE
01	JANUARY
02	FEBRUARY
03	MARCH
04	APRIL
05	MAY
06	JUNE
07	JULY
08	AUGUST
09	SEPTEMBER
10	OCTOBER
11	NOVEMBER
12	DECEMBER

สำหรับการเขียนคำสั่งนี้ก็ดำเนินการตามวิธีที่เคยกล่าวมาแล้ว โดยที่ตารางทั้งสองอันจะเป็นอิสระต่อกัน การป้อนข้อมูลที่เป็นตารางเข้าเครื่องนั้นเราต้องระมัดระวังว่าจำนวนข้อมูลที่ป้อนไว้ใน E-Form จะต้องเท่ากับข้อมูลที่จะป้อนเข้าไป เพราะถ้าหากมีข้อมูลมากเกินไปที่ป้อนไว้เครื่องจะรับแต่ที่ป้อนไว้ และในทำนองเดียวกัน ถ้ามีข้อมูลน้อยกว่าที่ป้อนไว้มันจะรับเอา blank เข้ามาเป็น data เพิ่มเติมจนครบเท่าที่ป้อนไว้ หรือไม่ก็อาจไปรับเอา Input Data ของชุดอื่นมาในกรณีที่ไม่มีทางที่จะหาข้อมูลให้ครบเท่าที่ป้อนไว้ก็จะเกิด error ขึ้น

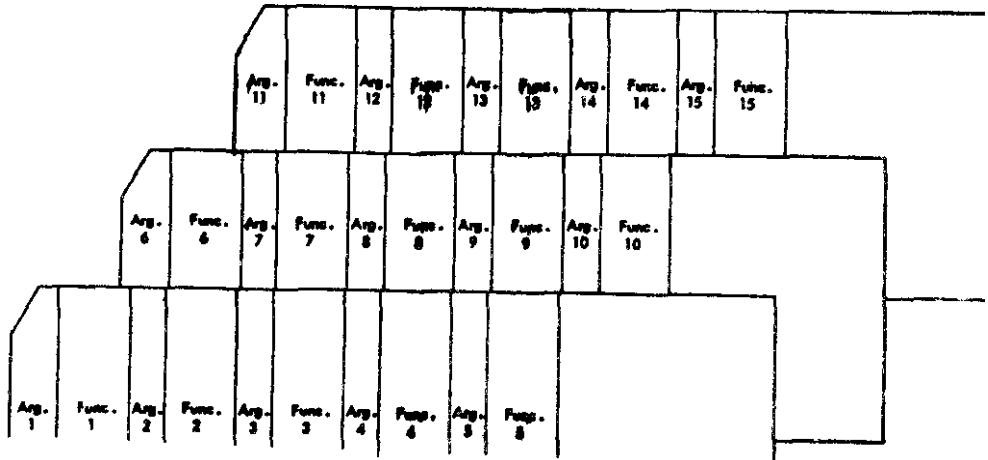
ตารางทั้ง 2 ที่สร้างขึ้นเพื่อเก็บในหน่วยความจำในรูปแบบนี้จะอยู่เป็นอิสระกัน เราสามารถสร้างความสัมพันธ์ของ 2 ตารางเพื่อประโยชน์ในการหาค่าของ Function ที่ตรงกับ Argument ที่ต้องการได้ ซึ่งเรื่องนี้จะขอกกล่าวในภายหลัง

วิธีที่ 2 เรียกว่าวิธี Alternating Argument and Function Table วิธีการนี้เป็นการสร้างให้ทั้ง Argument และ Function อยู่ใน Data File เดียวกัน

หลักการสร้างตารางลักษณะที่ 2 มีดังนี้คือ

1. เราต้องสร้าง Argument และ Function รวมอยู่ใน entry เดียวกัน
2. ในแต่ละ entry นั้นจะขึ้นต้นด้วย Argument หรือ Function ก่อนก็ได้
3. ใน entry เดียวกันจะแยก Argument และ Function ให้อยู่คนละ Record ไม่ได้
4. จำนวน entry ในแต่ละ Record ต้องเท่า ๆ กัน และเงื่อนไขอื่น ๆ ก็เขียนตามหลักของการสร้างตารางที่กล่าวมาแล้ว

ลักษณะของข้อมูลที่ใช้วิธีการ Alternating Argument and Function Table จะเป็นดังนี้
 ดังนั้น จากตัวอย่างในเรื่องของการสร้างชื่อของเดือน เราจะสร้างตารางของข้อมูลได้ดังนี้คือ



การเขียนคำสั่งของ alternating Argument and Function

1. จะมีการป้อนว่ามี Table เดียวใน F-Form (แทนที่จะเป็น File เหมือนกับวิธีแรก)
2. ใน E-Form ในส่วนของสครมภ์ที่ 11-18 ก็ยังคงใช้ในลักษณะเดิม คือใช้ชื่อของ Table ที่ป้อนใน F-Form
3. สครมภ์ที่ 27-32 (Table Name) ให้ใช้เป็นชื่อของตารางสำหรับ Argument หรือ Function ก็ได้ ดูว่าค่าใดมาก่อน ถ้า Argument นำหน้าก็ให้เป็นชื่อของตารางสำหรับ Argument แต่ถ้าเราให้ Function นำหน้า ก็ให้ใช้ชื่อของตารางสำหรับ Function วิธีการตั้งชื่อให้ยึดหลักของ-Table Name ที่กล่าวมาแล้ว

4. สดมภ์ที่ 36-45 จะเป็นส่วนที่อธิบายถึงลักษณะของ Table Name ในสดมภ์ที่ 27-32 ในแต่ละ Record เป็นความยาวของ Argument ลักษณะการเรียงลำดับ และจำนวนทศนิยม

5. สดมภ์ที่ 46-51 เป็นส่วนที่ใช้สำหรับ Table Name ของข้อมูล ส่วนที่เหลือ ตัวอย่าง เช่น ถ้า Argument เป็นส่วนแรกของ entry และ Function เป็นส่วนที่ต่อมาจาก Argument ใน entry เมื่ออธิบาย Argument ไปแล้วในสดมภ์ที่ 27-45 เราก็ต้องอธิบาย Function ในสดมภ์ที่ 46-57 หลักการตั้งชื่อ Table ในส่วนนี้ก็ดำเนินการเหมือนกับการตั้ง Table Name ทุกประการ โดยที่บ่งว่าความยาวของ element ในส่วนนี้ยาวเท่าใดในสดมภ์ที่ 52-54 ข้อมูลอยู่ในรูป Packed หรือไม่ให้บ่งในสดมภ์ที่ 55 ส่วนสดมภ์ที่ 56 ใช้สำหรับบ่งว่ามีจำนวนทศนิยมกี่ตำแหน่ง ส่วนในสดมภ์ที่ 57 ใช้บ่งว่ามีการเรียงลำดับของข้อมูลที่เป็น entry นี้หรือไม่อย่างไร

จากข้อมูลในเรื่องเดือน ซึ่งเราสามารถจัดรูปของ Input Data ให้มี 2 Record โดยที่ใน 1 Record บรรจุ 7 entries (1 Data File = 12 entries) แต่ละ entry นั้นนำหน้าด้วย Argument ตามด้วย-Function ความยาวของ Argument มี 2 ตำแหน่ง ไม่มีทศนิยมเรียงลำดับจากน้อยไปหามาก ส่วน-Function มีความยาว 9 ตำแหน่งเป็นข้อมูล Alpha ไม่มีการเรียงลำดับ ถ้าเราตั้งชื่อของตารางนี้ว่า-FILETAB โดยที่ File นี้ประกอบด้วย TABMON (ชื่อของตารางสำหรับ Argument และ TABMTH. (ชื่อของตารางสำหรับ Function) เราจะเขียนคำสั่งได้ดังนี้

แบบฝึกหัด

1. จงสร้างตารางเพื่อเปลี่ยนจากรหัสตัวเลขไปสู่วันต่าง ๆ ในสัปดาห์ โดยใช้รหัสดังนี้

Code	Equivalent
01	Monday
02	Tuesday
03	Wednesday
04	Thursday
05	Friday
06	Sunday
07	Saturday

ให้เขียนคำสั่งเพื่ออ่านตารางนี้เข้าไปเก็บไว้ในหน่วยความจำเพื่อใช้ในการประมวลผลข้อมูล