

บทที่ 10

การเรียงลำดับและการค้นหาข้อมูล (Sorting and Searching)

บทนำ ระบบการประมวลผลข้อมูลของคอมพิวเตอร์นั้นจะประกอบด้วยพื้นฐานของกิจกรรมหลักๆ ดังนี้
คือ

- ◆ การเรียงลำดับข้อมูล (Sorting)
- ◆ การค้นหาข้อมูล (Searching)
- ◆ การคำนวณ (Calculating)

◆ **การเรียงลำดับ (Sorting)** นับเป็นกิจกรรมที่มีความถี่ในการใช้งานมากในระบบการทำงาน เกือบทุกกิจการ ดังนั้นการศึกษาเรื่องของการเรียงลำดับจึงเป็นหัวข้อที่สำคัญยิ่ง กิจกรรมในการเรียงลำดับนั้นจะประกอบด้วยปัจจัยและกรรมวิธีแยกย่อยดังนี้คือ

- การเลือกคีย์ที่จะใช้เรียงลำดับ
- กรรมวิธีในการเรียงลำดับ
- การเลือกคีย์ที่จะใช้เรียงลำดับ เกณฑ์ในการเลือกรายการที่จะนำไปเป็นคีย์สืบค้นนั้นจะประกอบด้วยข้อกำหนดดังนี้
 1. การนำไปใช้งานประยุกต์ (Application)
 2. การมีค่าเป็นเอกลักษณ์ (Unique)
 3. เป็นรายการที่สั้น (Short)

1. **การนำไปใช้งานประยุกต์** นั้นนับเป็นสิ่งที่สำคัญที่สุด เพราะถ้าหากการเรียงลำดับนั้นเลือกรายการที่ไม่ต้องกับวัตถุประสงค์ของผู้ใช้งานก็จะทำให้การเรียงลำดับนั้นไม่ก่อให้เกิดผลประโยชน์กับการนำไปใช้งาน ตัวอย่างเช่น การเรียงลำดับรายการของเจ้าของโทรศัพท์โดยการเรียงลำดับตามหมายเลขของโทรศัพท์ ซึ่งจะส่งผลให้การสืบค้นไม่ตรงกับความต้องการของผู้ใช้งานเพราะโดยปกติผู้ใช้งานมักจะสืบค้นผู้ใช้โทรศัพท์โดยอาศัยจากชื่อของผู้เป็นเจ้าของโทรศัพท์มากกว่าที่จะสืบค้นจากหมายเลขโทรศัพท์ ดังนั้นจะเห็นได้ว่าสมุครายนามของผู้ครอบครองโทรศัพท์จะเรียงลำดับตามรายชื่อของเจ้าของโทรศัพท์ นอกจากนี้ ในกรณีของงานจัดการเรื่องการประกาศห้องสอบ หรือประกาศผลการสอบ หรือในกรณีของการสอบถามห้องพักของคนไข้ก็จะใช้ชื่อเป็นรายการสอบถาม ดังนี้ เป็นต้น

2. การมีค่าเป็นเอกลักษณ์ (Unique) ลักษณะของรายการที่จะนำมาเป็นคีย์นี้ในการสืบค้นนั้นจะถูกนำไปใช้ในการเรียงลำดับ ดังนั้นถ้าหากว่ามีข้อมูลในรายการนั้นเกิดการซ้ำเกิดขึ้น เราไม่สามารถจะระบุได้ว่าระเบียบข้อมูลใดคือข้อมูลที่เรากำลังต้องการ ในกรณีที่เกิดปัญหาเช่นนี้และเราไม่สามารถหลีกเลี่ยงไปใช้รายการข้อมูลอื่นมาแทน เราอาจจะหาทางออกเพื่อสร้างลักษณะที่เป็นเอกลักษณ์ได้ โดยการนำรายการอื่นมาทำหน้าที่เป็น Second Key โดยการให้รายการแรกที่เราเลือกมาเรียงลำดับแล้วทำหน้าที่เป็น Primary Key หรือถ้าเกิดมีปัญหาว่าใช้ถึง 2 รายการแล้วยังไม่เกิดลักษณะเอกลักษณ์ก็อาจจะต้องเพิ่ม Third Key ขึ้นมาอีกก็ได้ ตัวอย่างของกรณีนี้ก็คือการจัดการหนังสือในห้องสมุด ซึ่งมักจะปรากฏว่า ชื่อเรื่องหนังสือมักจะซ้ำๆกัน เช่นเดียวกับชื่อผู้แต่ง ดังนั้นอาจจะต้องมีการใช้คีย์นี้ไปในการเรียงลำดับถึง 3 รายการคือ ชื่อหนังสือ (Title) , ชื่อผู้แต่ง (Author) , ปีที่พิมพ์(Edition) ดังนั้นการจะสืบค้นหนังสือเล่มใดเล่มหนึ่งอาจจะต้องใช้คีย์นี้ถึง 3 รายการ โดยที่

- Title : Primary Key
- Author : Second key
- Edition : Third Key

Logical Key = Title + Author + Edition

3. เป็นรายการที่สั้น (Short) การใช้รายการข้อมูลที่สั้นจะส่งผลให้การเรียงลำดับหรือแม้กระทั่งการสืบค้นดำเนินการได้อย่างรวดเร็ว เหตุผลที่เป็นเช่นนี้ก็เพราะว่า โปรแกรมการเรียงลำดับนั้นจะประกอบด้วยพื้นฐานหลัก 2 ประการที่ทำงานมากที่สุด คือ

- การเปรียบเทียบ (Compare)
- การเคลื่อนย้าย (Move)

ตัวอย่างของการเปรียบเทียบ String A และ String B กำหนดให้แต่ละ String ยาวเท่ากับ 16 Bytes จะปรากฏการเปรียบเทียบดังนี้

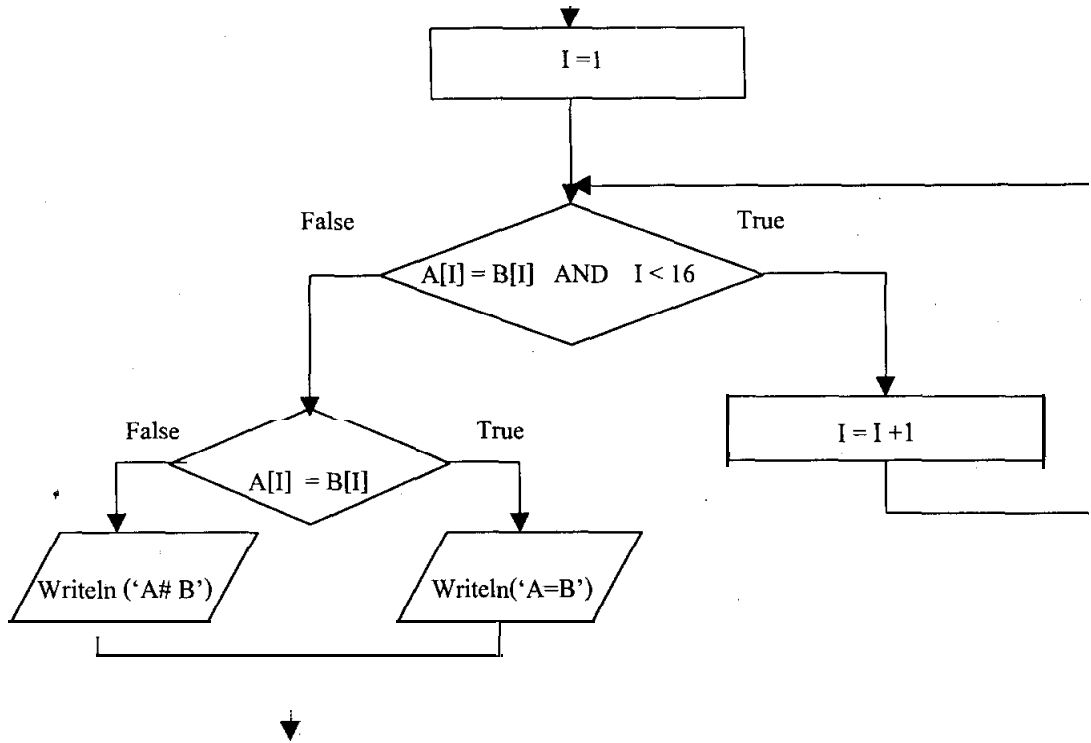
Array A

T	H	I	L	A	N	D	*	I	S	*	L	N							
											A[1]	A[2]							A[16]

Array B

T	H	I	L	A	N	D	&	I	&	A	L	AN							
											B[1]	B[2]							B[16]

จากภาพที่ปรากฏนี้ถ้าทำการเปรียบเทียบ String A ,B (Array A , B) นั้นจะต้องดำเนินการเปรียบเทียบทีละอักขระตั้งแต่ A[1] : B[1] ,.....,A[16] : B[16] เรื่อยไปยกเว้น ในกรณีที่มิมีคู่ใดคู่หนึ่งระหว่าง A[I] : B[I] ; I = 1,...,16 ไม่เท่ากันก็จะออกจากการเปรียบเทียบ แต่ถ้ายังเท่ากันก็จะเปรียบเทียบไปเรื่อยจนสิ้นสุด Array คือขนาด 16 Cell(ICel = 1 Byte) การเปรียบเทียบดังกล่าวนี้สามารถเขียนเป็นส่วนหนึ่งของผัง โปรแกรมได้ดังนี้



จากผัง โปรแกรมที่แสดงมานี้จะเห็นได้ว่ายิ่งข้อมูลมีความยาวมากเพียงใดการเปรียบเทียบก็จะยิ่งใช้เวลานานขึ้น นอกจากความยาวของอะเรย์จะส่งผลถึงการเปรียบเทียบแล้วยังส่งผลถึงการย้ายที่อยู่ระหว่าง 2 อะเรย์คืออะเรย์ยาวจะใช้เวลาในการสลับที่ย้ายข้อมูลมากกว่าอะเรย์ขนาดสั้น

กรรมวิธีในการเรียงลำดับ

กรรมวิธีในการเรียงลำดับ จะแบ่งเป็น 2 ประเภทตามลักษณะที่ปรากฏให้เห็นคือ

A. แบบ Physical Sort หมายถึงการเรียงลำดับจริงๆทางกายภาพ เช่นเรียงตามลำดับความสูงขณะเข้าแถวของนักเรียน

B. แบบ Logical Sort หมายถึงการเรียงลำดับแบบเสมือนจะเป็นการเรียงลำดับแบบไม่ปรากฏการเรียงตามที่ตั้งที่แท้จริง ตัวอย่างเช่นลูกค้าที่รับบัตรคิวเข้ามาใช้บริการของธนาคาร พอ

รับบัตรคิวเสร็จก็แยกย้ายกันไปนั่งรอที่นั่ง โดยจะนั่งตามที่นั่งว่างๆ ไม่มีการเรียง ลำดับตามบัตรคิว ในขณะที่ธนาคารจะเรียกลูกค้าตามบัตรที่ได้รับเรียงกันไป ซึ่งจะเป็นการเรียงลำดับ โดยที่ลูกค้าไม่ต้องไปยืนเข้าแถวรอเรียงเข้ารับการบริการครั้งนี้จะเป็นการเรียงลำดับแบบ Logical Sort

- การเรียงลำดับแบบที่เป็น Physical Sort นั้นเราจะแบ่งออกเป็น 2 วิธีใหญ่ในการดำเนินการคือ

การคือ

1. การเรียงลำดับแบบภายใน (Internal Sort)
2. การเรียงลำดับแบบภายนอก (External Sort)

การเรียงลำดับแบบภายใน (Internal Sort : Physical Sort)

หมายถึงการเรียงลำดับ โดยมีหลักการว่าจะต้องนำข้อมูลทั้งหมดเข้ามาจัดเก็บภายในสมองเครื่องก่อน ภายหลังจึงนำไปทำการเรียงลำดับตามดัชนีที่กำหนด กรรมวิธีการจัดเก็บข้อมูลภายในเครื่องก็มักจะใช้วิธีการเก็บแบบอะเรย์ อาจจะเป็นอะเรย์คู่ขนาน หรือไม่ก็เป็นอะเรย์ของระเบียนข้อมูล วิธีการเรียงลำดับแบบภายใน จะมีข้อดีในแง่ที่จะทำการเรียงลำดับได้อย่างรวดเร็วเพราะข้อมูลอยู่ภายในสมองเครื่องอยู่แล้ว แต่จะมีข้อจำกัดที่ว่าถ้ามีข้อมูลขนาดใหญ่จะไม่สามารถจัดเก็บภายในสมองเครื่องได้หมด การเรียงลำดับแบบนี้จะแยกออกเป็นวิธีย่อย โดยที่แต่ละวิธีย่อยก็จะพัฒนาไปเป็นวิธีต่างๆต่อไป เช่น การเรียงลำดับแบบ Bubble Sort , Quick Sort เป็นต้น ในที่นี้จะอธิบายเฉพาะหลักการใหญ่ๆเพื่อให้ผู้อ่านได้แนวคิดในการเรียงลำดับแต่ละวิธี

หลักการของการเรียงลำดับแบบภายในแบ่งเป็น 3 แบบคือ

- 1.1 การเรียงลำดับแบบเลือก (Selection Sort)
- 1.2 การเรียงลำดับแบบแทนที่ (Replacement Sort)
- 1.3 การเรียงลำดับแบบแทรก (Insertion Sort)

1.1 การเรียงลำดับแบบเลือก (Selection Sort) วิธีการเรียงลำดับโดยแนวคิดนี้จะมีลักษณะคล้ายๆกับพฤติกรรมกรรมการเลือกของมนุษย์ในการเรียงลำดับสิ่งของ พิจารณาจากขั้นตอนการปฏิบัติงานดังนี้

Array A [10] : Integer Unsorted Array

A[1] A[2]A[10]

34	21	15	7	9	65	8	2	1	48
----	----	----	---	---	----	---	---	---	----

Step0 : กำหนด I = 1

Step1 : ค้นหาว่า ข้อมูลที่เล็กที่สุดของอะเรย์ A ตั้งแต่ลำดับที่ I ,...,10 คือเท่าไรและอยู่ตำแหน่งที่เท่าไร จากตัวอย่างนี้ Small = 1 ; Point = 9 ทำการสลับที่ระหว่างตำแหน่ง A[I] กับ A[Point]
ผลลัพธ์ที่ได้คือ

A[1] A[2] A[10]

1	21	15	7	9	65	8	2	34	48
---	----	----	---	---	----	---	---	----	----

Step ที่ 1 จะได้ข้อมูลที่เล็กที่สุดย้ายมาอยู่ที่ตำแหน่งที่ 1 ขั้นตอนต่อไปก็จะดำเนินการเช่นเดียวกับ Step ที่ 1 เพียงแต่เราจะกระทำกับข้อมูลของอะเรย์ตั้งแต่ตำแหน่งที่ 2 ,...,10

Step 2 : ถ้า Sort ยังไม่ครบ 10 เชน ให้กลับไป Step1 ใหม่แต่ กำหนดให้

I = 2,3,... 10

Step 3 : การเรียงลำดับสิ้นสุดลง

ขั้นตอนที่กล่าวมานี้สามารถเขียนภาพที่เกิดขึ้นจากการกระทำได้ดังนี้

ระดับที่ 1 ก่อนจะเรียงลำดับ

A[1] A[2] A[10]

34	21	15	7	9	65	8	2	1	48
----	----	----	---	---	----	---	---	---	----

ระดับที่ 2

A[1] A[2] A[10]

1	21	15	7	9	65	8	2	34	48
---	----	----	---	---	----	---	---	----	----

ระดับที่ 3

A[1] A[2] A[10]

1	2	15	7	9	65	8	2	1	34	48
---	---	----	---	---	----	---	---	---	----	----

ระดับที่ 4

A[1] A[2] A[10]

12		7	15	9	65	8	2	1	34	48
----	--	---	----	---	----	---	---	---	----	----

ระดับที่ 5

A[1] A[2] A[10]

1	2	7	8	9	65	15	2	1	34	48
---	---	---	---	---	----	----	---	---	----	----

ระดับที่ 6

A[1] A[2] A[10]

1	2	7	8	9	65	15	21	34	48
---	---	---	---	---	----	----	----	----	----

ระดับที่ 7

A[1] A[2] A[10]

1	2	7	8	9	15	65	21	34	48
---	---	---	---	---	----	----	----	----	----

ระดับที่ 8

A[1] A[2] A[10]

1	2	7	8	9	15	21	65	34	48
---	---	---	---	---	----	----	----	----	----

ระดับที่ 9

A[1] A[2]A[10]

1	2	7	8	9	15	21	34	65	48
---	---	---	---	---	----	----	----	----	----

ระดับที่ 10 ระดับสุดท้ายของการเรียงลำดับเสร็จสิ้นลง

A[1] A[2] A[10]

1	2	7	8	9	.	15	21	34	48	65
---	---	---	---	---	---	----	----	----	----	----

จากขั้นตอนต่างๆที่เขียนมาหลายๆเราจะนำมาออกแบบเป็นผังโปรแกรมในการดำเนินการเรียงลำดับจากน้อยไปหามากได้ดังนี้

ข้อกำหนดของปัญหา ให้รับข้อมูลจากแป้นพิมพ์เป็นประวัติของนักเรียนจำนวนหนึ่งที่มีไม่เกิน 100 คน ให้นำไปเรียงลำดับตามรหัสประจำตัว

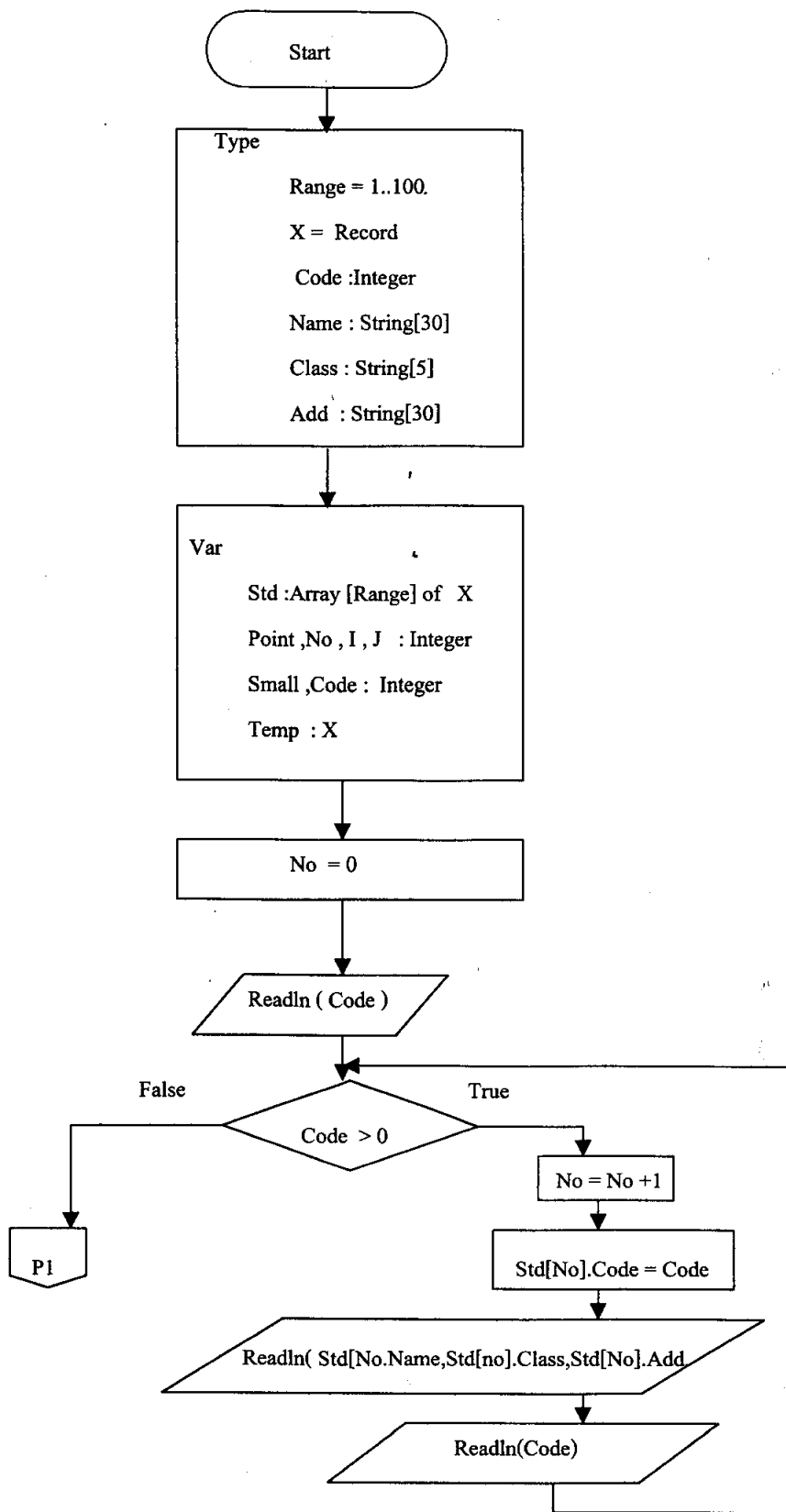
กำหนดโครงสร้างข้อมูลดังนี้

Code : Integer

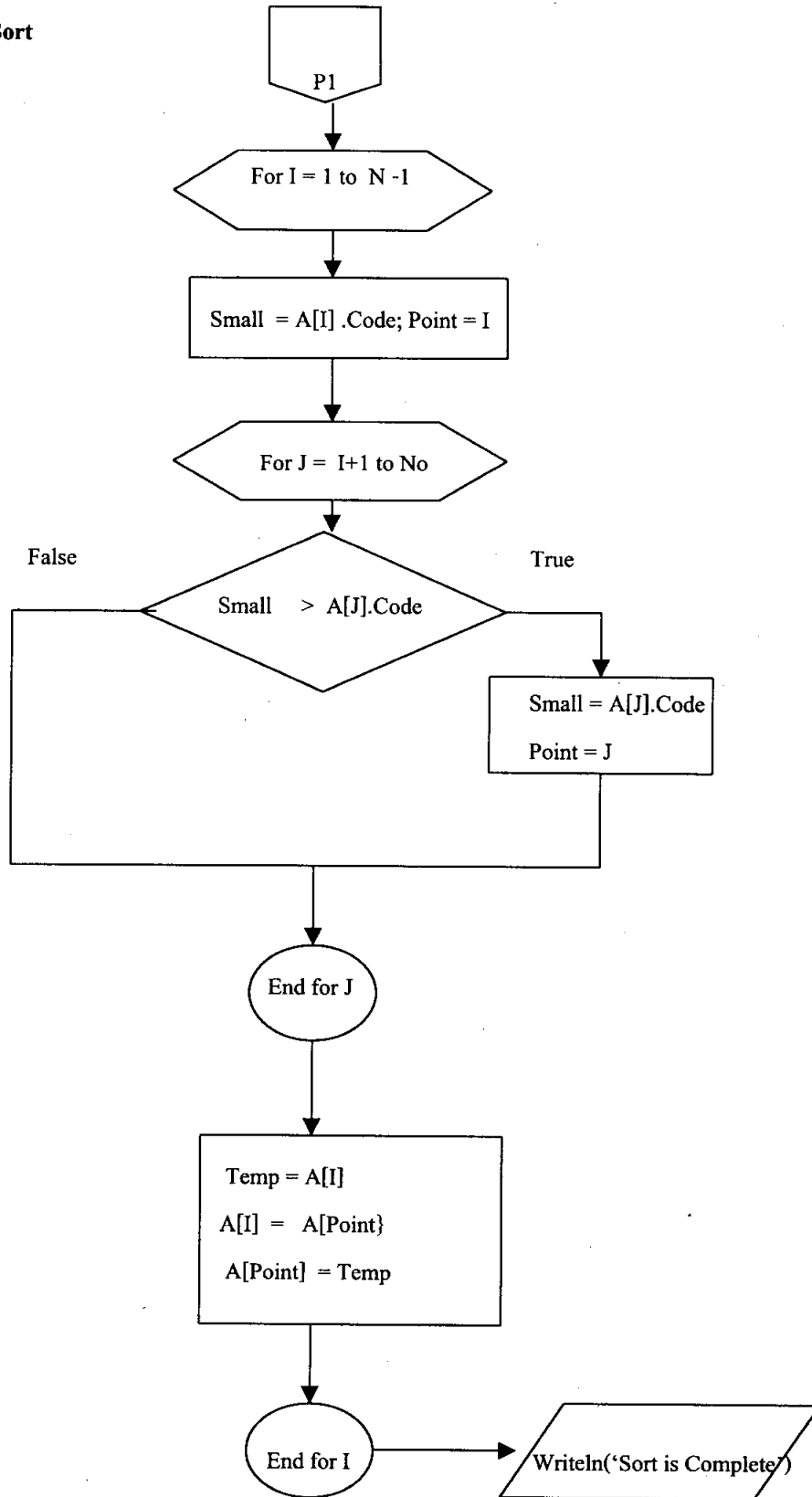
Name : String[30]

Class : String[5]

Add : String[30]



Selection Sort



1.2 การแทนที่ (Replacement Sort) วิธีการเรียงลำดับโดยแนวคิดนี้จะมีลักษณะคล้ายกับการแข่งขันกีฬาแบบจับคู่ (ผู้ชนะกับคู่ต่อสู้คนต่อไป)

Array A [10] : Integer Unsorted Array

A[1] A[2]A[10]

34	21	15	7	9	65	8	2	1	48
----	----	----	---	---	----	---	---	---	----

Step 0 : กำหนดให้ $I = 1, \dots, 9$

Step 1 : จับคู่เปรียบเทียบระหว่าง A[1] :A[2] , A[2] :A[3] ,..., A[9] :A[10] การเปรียบเทียบแต่ละคู่ นั้น ถ้าตัวใดใหญ่กว่าก็ให้สลับตัวที่ใหญ่กว่าให้ไปอยู่ในลำดับถัดไป ผลลัพธ์ที่ได้คือ

A[1] A[2]A[10]

21	15	7	9	34	8	2	1	48	65
----	----	---	---	----	---	---	---	----	----

Step ที่ 1 จะได้ข้อมูลที่ใหญ่ที่สุดย้ายมาอยู่ที่ตำแหน่งที่ 10 ขั้นตอนต่อไปก็จะดำเนินการเช่นเดียวกับ Step ที่ 1 เพียงแต่เราจะกระทำกับข้อมูลของอะเรย์ ตั้งแต่ตำแหน่งที่ 1 ,...,9

Step 2 : กลับไปที่ Step 1 ดำเนินการเหมือนเดิมเพียงแต่การจับคู่เปรียบเทียบลดลง โดยการตัด Array A ตัวสุดท้ายทิ้งไป ทำจนกระทั่งเหลือ คู่เดียวคือ A[1] : A[2]

Step 3 : การเรียงลำดับสิ้นสุดลง

ขั้นตอนที่กล่าวมานี้สามารถเขียนภาพที่เกิดขึ้นจากการกระทำได้ดังนี้

ระดับที่ 1

ก่อนจะเรียงลำดับ

A[1] A[2]A[10]

34	21	15	7	9	65	8	2	1	48
----	----	----	---	---	----	---	---	---	----

ระดับที่ 2

A[1] A[2]A[10]

21	15	7	9	34	8	2	1	48	65
----	----	---	---	----	---	---	---	----	----

ระดับที่ 3

A[1] A[2]A[10]

15	7	9	21	8	2	1	34	48	65
----	---	---	----	---	---	---	----	----	----

ระดับที่ 4

A[1] A[2]A[10]

7	9	15	8	2	1	21	34	48	65
---	---	----	---	---	---	----	----	----	----

ระดับที่ 5

A[1] A[2]A[10]

7	9	8	2	1	15	21	34	48	65
---	---	---	---	---	----	----	----	----	----

ระดับที่ 6

A[1] A[2]A[10]

7	8	2	1	9	15	21	34	48	65
---	---	---	---	---	----	----	----	----	----

ระดับที่ 7

A[1] A[2]A[10]

7	2	1	8	9	15	21	34	48	65
---	---	---	---	---	----	----	----	----	----

ระดับที่ 8

A[1] A[2]A[10]

2	1	7	8	9	15	21	34	48	65
---	---	---	---	---	----	----	----	----	----

ระดับที่ 9

A[1] A[2]A[10]

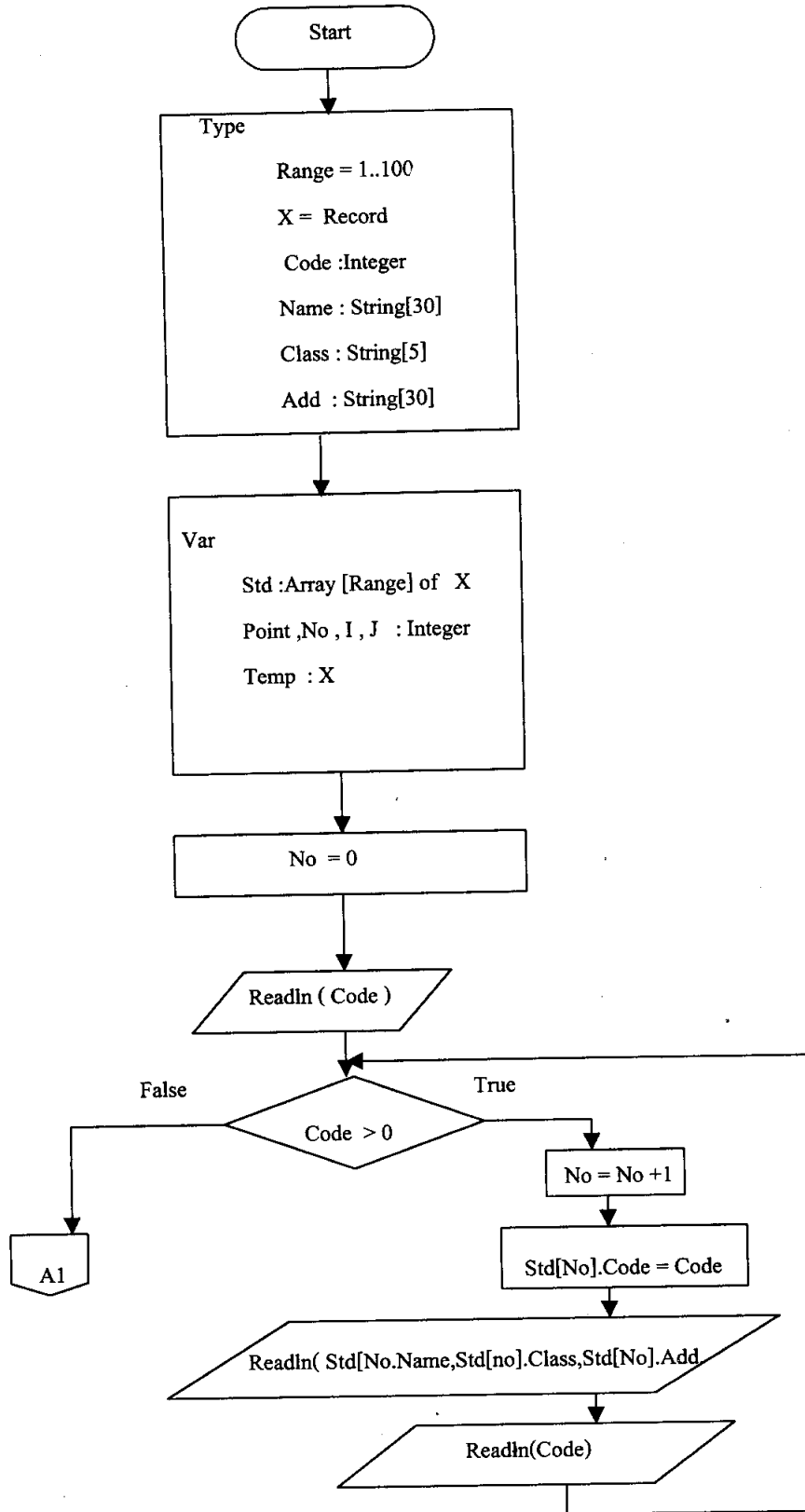
2	1	7	8	9	15	21	34	48	65
---	---	---	---	---	----	----	----	----	----

ระดับที่ 10

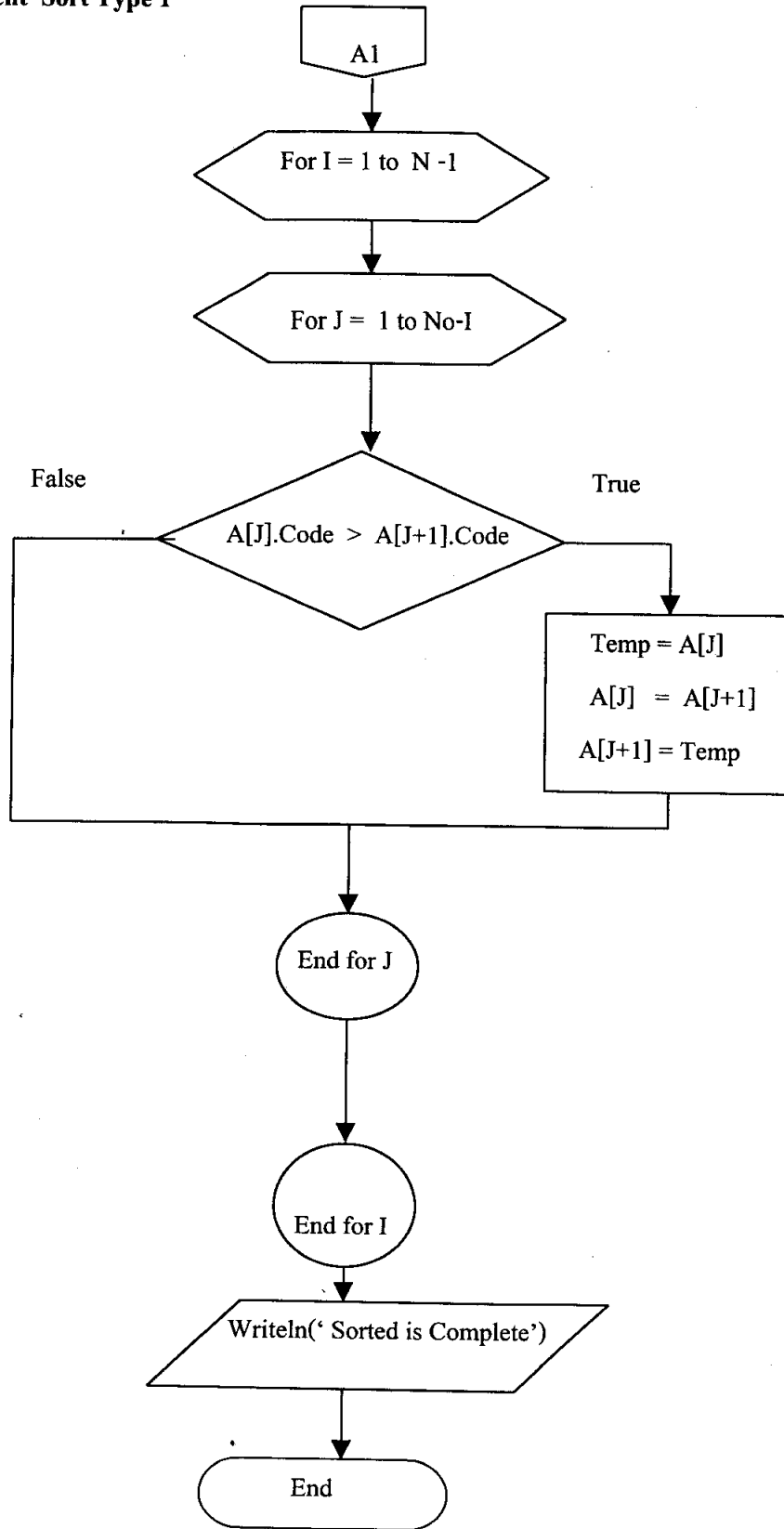
A[1] A[2]A[10]

1	2	7	8	9	15	21	34	48	65
---	---	---	---	---	----	----	----	----	----

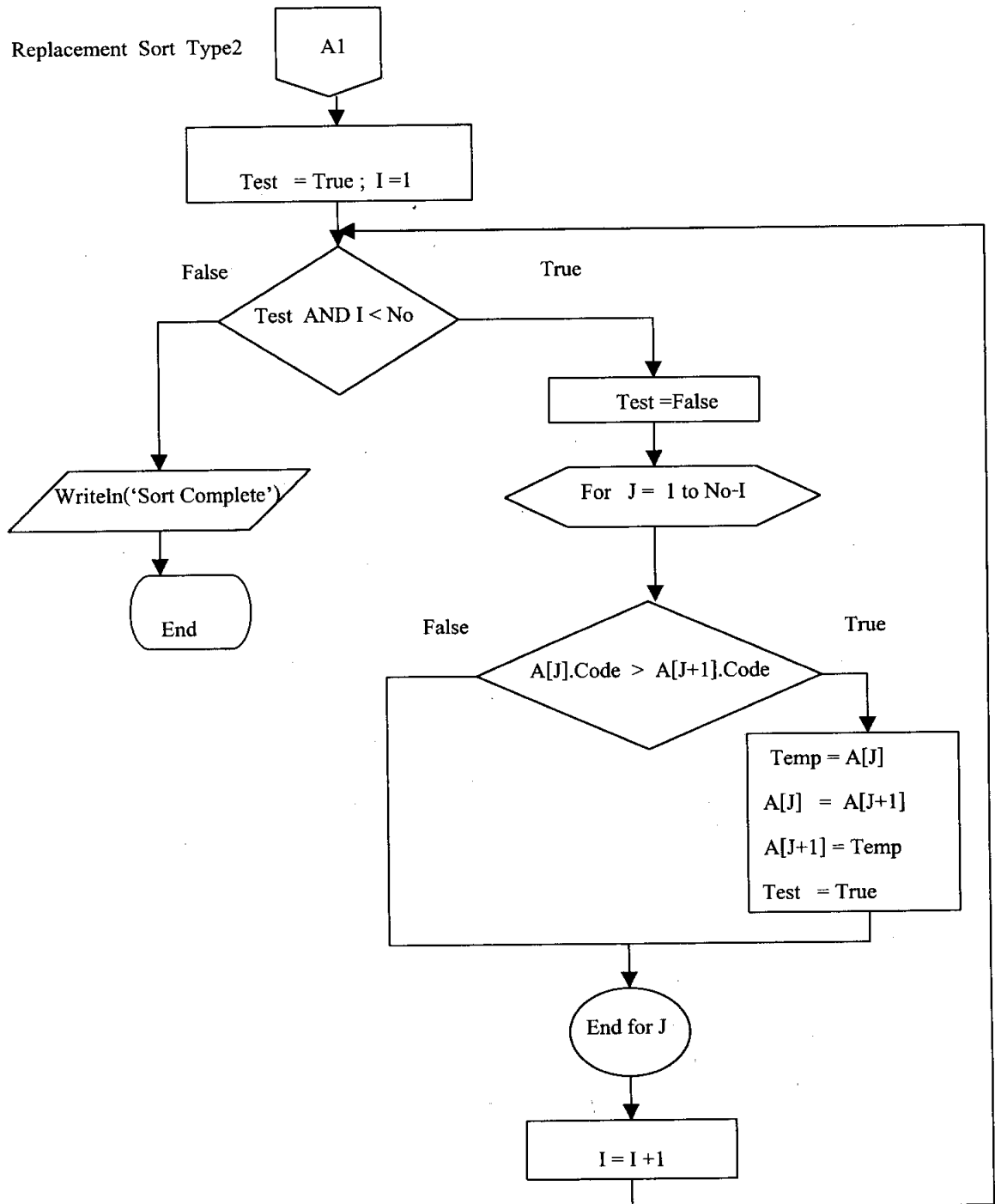
จากขั้นตอนดังกล่าวเรานำมาเขียนเป็นผังโปรแกรมได้ดังนี้



Replacement Sort Type 1



จากผังของ Replacement Sort ดังกล่าว เราสามารถที่จะปรับปรุงให้มีประสิทธิภาพดียิ่งขึ้น เพื่อให้ใช้ในกรณีที่ ถ้าข้อมูลของอะเรย์นั้น Sort เสร็จเรียบร้อยแล้ว ก็จะสามารถหลุดจากการตรวจสอบได้ โดยไม่ต้องวน Loop I หลายๆ ครั้ง ดังนั้นในส่วนของ การ Sort โดยวิธีนี้จะพัฒนาใหม่ได้ดังนี้



1.3 การแทรก (Insertion Sort) - วิธีการเรียงลำดับโดยแนวคิดนี้จะมีลักษณะคล้ายกับการเรียงลำดับแบบหน้าไฟ

A[1]	34
A[2]	21
A[3]	15
A[4]	7
A[5]	9
A[6]	65
A[7]	8
A[8]	2
A[9]	1
A[10]	48

ขั้นตอนการทำงานจะปรากฏเป็นลำดับดังนี้

ระดับที่ 1

A[0]	21
A[1]	34
A[2]	21
A[3]	15
A[4]	27
A[5]	9
A[6]	65
A[7]	8
A[8]	2
A[9]	1
A[10]	48

$A[0] = 21$ ไฟใบที่ 2 เทียบกับไฟใบที่ 1 ถ้าเล็กกว่าให้ร่นไฟใบที่ 1 ไปอยู่ที่ตำแหน่งที่ 2 แทน แต่ถ้าไฟใบที่ 2 ใหญ่กว่าไฟใบที่ 1 คำนึงให้ไฟใบที่ 2 อยู่ที่ตำแหน่งเดิม คือ ตำแหน่งที่ 2

ระดับที่ 2

A[0]	15
A[1]	21
A[2]	34
A[3]	15
A[4]	27
A[5]	9
A[6]	65
A[7]	8
A[8]	2
A[9]	1
A[10]	48

A[0] = 15 ไพบ่ที่ 3 เทียบกับไพบ่ที่ 1,2 เพื่อหาที่แทรก
 ตามตัวอย่างนี้ ที่แทรกคือตำแหน่งที่ 1 ดังนั้นจะมี
 การร่นตำแหน่งที่ 2 ไปที่ตำแหน่งที่ 3 และตำแหน่งที่
 1 ไปอยู่ที่ตำแหน่งที่ 2 และตำแหน่งที่ 1 แทรกด้วย
 ตำแหน่งที่ 0

ระดับที่ 3

A[0]	27
A[1]	15
A[2]	21
A[3]	34
A[4]	27
A[5]	9
A[6]	65
A[7]	8
A[8]	2
A[9]	1
A[10]	48

A[0] = 27 ไพบ่ที่ 4 เทียบกับไพบ่ที่ 1,2,3 เพื่อหาที่แทรก
 ตามตัวอย่างนี้ ที่แทรกคือตำแหน่งที่ 3 ดังนั้นจะมี
 การร่นตำแหน่งที่ 3 ไปที่ตำแหน่งที่ 4 และตำแหน่งที่
 3 นำข้อมูลจากตำแหน่งที่ 0 มาแทรกแทน

ระดับที่ 4

A[0]	9
A[1]	15
A[2]	21
A[3]	27
A[4]	34
A[5]	9
A[6]	65
A[7]	8
A[8]	2
A[9]	1
A[10]	48

A[0] = 9 ไพบ่ที่ 5 เทียบกับไพบ่ที่ 1,2,3,4 เพื่อหาที่แทรก
 ตามตัวอย่างนี้ ที่แทรกคือตำแหน่งที่ 1 ดังนั้นจะมี
 การร่นตำแหน่งที่ 4 ไปที่ตำแหน่งที่ 5 และตำแหน่งที่
 4 รับ ตำแหน่งที่ 3 ตำแหน่งที่ 3 รับตำแหน่งที่ 2
 ตำแหน่งที่ 2 รับตำแหน่งที่ 1 ส่วนตำแหน่งที่ 1
 แทรกด้วยตำแหน่งที่ 0

ระดับที่ 5

A[0]	65
A[1]	9
A[2]	15
A[3]	21
A[4]	27
A[5]	34
A[6]	65
A[7]	8
A[8]	2
A[9]	1
A[10]	48

A[0] = 65 ไพบ่ที่ 6 เทียบกับไพบ่ที่ 1,2,3,4,5 เพื่อหาที่แทรก
 ตามตัวอย่างนี้ เนื่องจากตำแหน่งที่ 6 เป็นข้อมูลที่ใหญ่
 กว่าทุกตำแหน่งตั้งแต่ 1,2,...,5 ดังนั้น A[6] จึงอยู่
 ตำแหน่งเดิม

ระดับที่ 6

A[0]	8
A[1]	9
A[2]	15
A[3]	21
A[4]	27
A[5]	34
A[6]	65
A[7]	8
A[8]	2
A[9]	1
A[10]	48

A[0] = 8 ไฟโบที่ 7 เทียบกับไฟโบที่ 1,2,3,4,5,6 เพื่อหาที่แทรก
ปรากฏว่าแทรกที่ตำแหน่งที่ 1 ดังนั้นจึงต้องทำการร่น
ตั้งแต่ 6 ไป 7 ตำแหน่งที่ 5 ไป 6 ดังนี้เรื่อยไป

ระดับที่ 7

A[0]	I 2 I
A[1]	8
A[2]	9
A[3]	15
A[4]	21
A[5]	27
A[6]	34
A[7]	65
A[8]	2
A[9]	1
A[10]	48

A[0] = 2 ไฟโบที่ 8 เทียบกับไฟโบที่ 1,2,3,4,5,6,7 เพื่อหาที่แทรก
ปรากฏว่าแทรกที่ตำแหน่งที่ 1 ดังนั้นจึงต้องทำการร่น
ตั้งแต่ 7 ไป 8 ตำแหน่งที่ 6 ไป 7 ดังนี้เรื่อยไป

ระดับที่ 8

A[0]	1
A[1]	2
A[2]	8
A[3]	9
A[4]	15
A[5]	21
A[6]	27
A[7]	34
A[8]	65
A[9]	1
A[10]	48

A[0] = 1

ไฟโบที่ 9 เทียบกับไฟโบที่ 1,2,3,4,5,6,7,8 เพื่อหาที่
แทรกปรากฏว่าแทรกที่ตำแหน่งที่ 1 ดังนั้นจึงต้องทำการ
ร่นตั้งแต่ตำแหน่งที่ 8 ไป 9 ตำแหน่งที่ 7 ไป 8 ดังนี้เรื่อย

ระดับที่ 9

A[0]	48
A[1]	1
A[2]	2
A[3]	8
A[4]	9
A[5]	15
A[6]	21
A[7]	27
A[8]	34
A[9]	65
A[10]	48

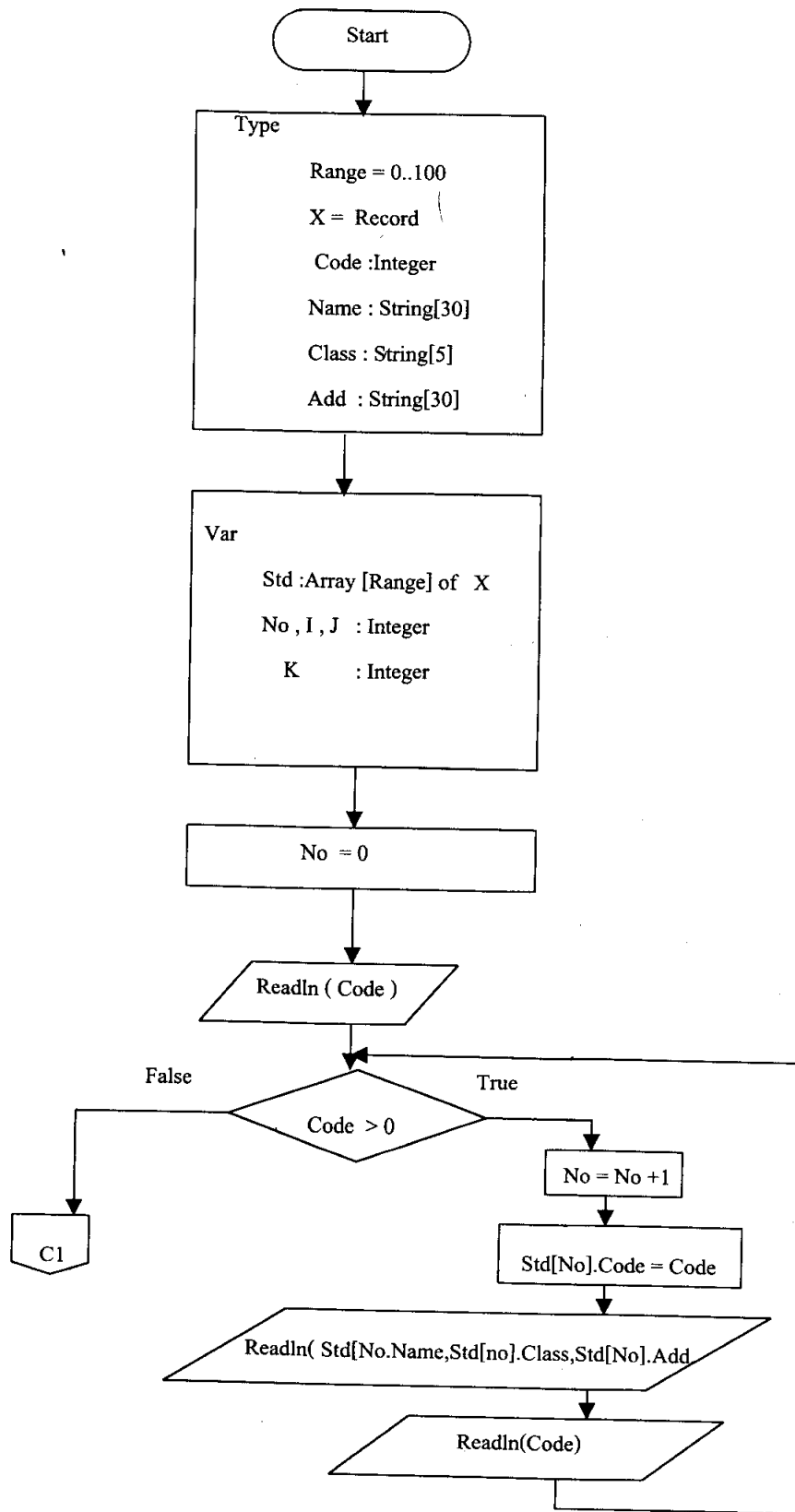
A[0] = 48

ไฟโบที่ 10 เทียบกับไฟโบที่ 1,2,3,4,5,6,7,8,9 เพื่อหา
แทรกปรากฏว่าแทรกที่ตำแหน่งที่ 9 ดังนั้นจึงต้องทำการ
ร่นตั้งแต่ตำแหน่งที่ 9 ไป 10 ตำแหน่งที่ 0 แทรกไปที่
ตำแหน่งที่ 9

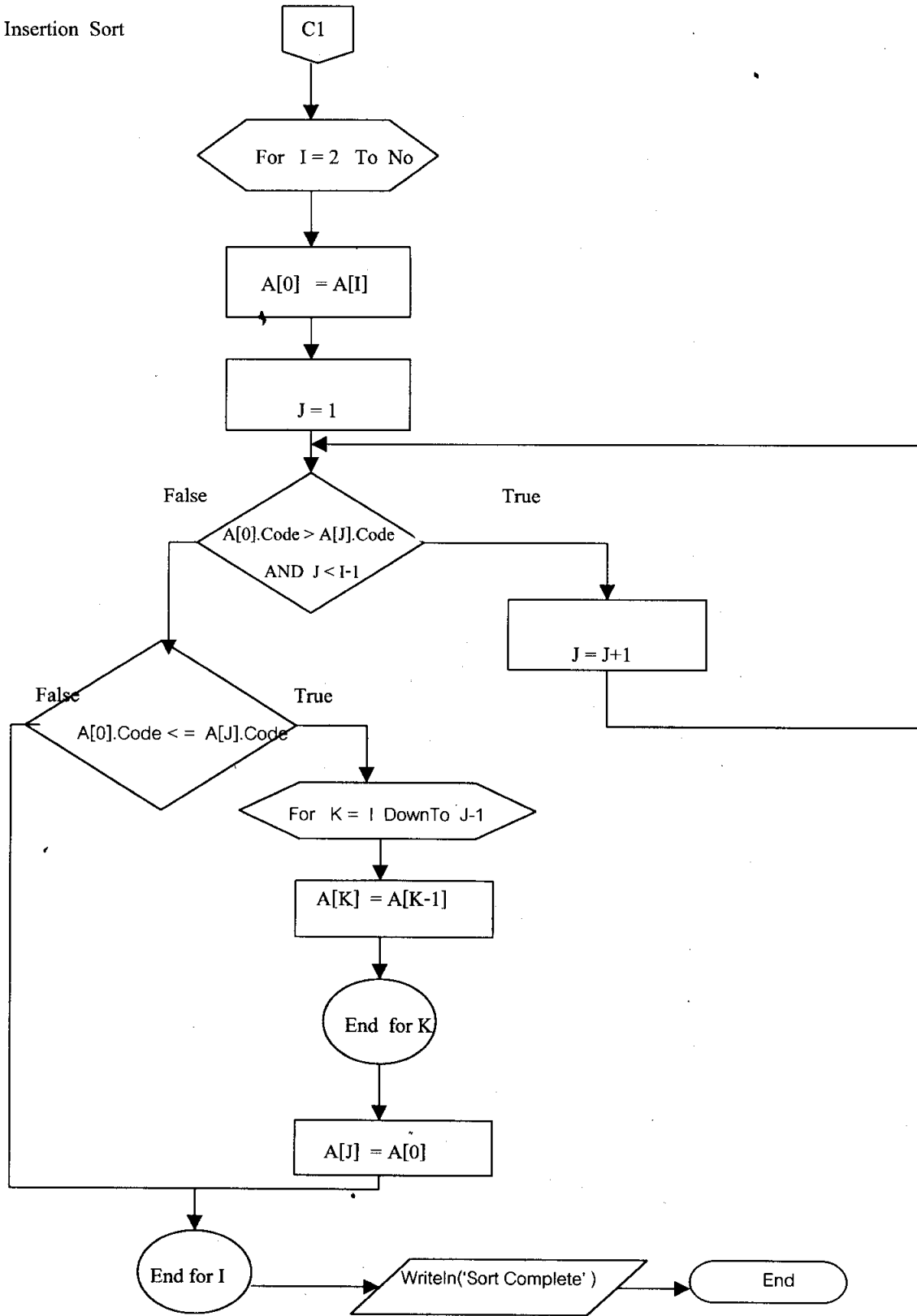
การเรียงลำดับสิ้นสุดลง

A[0]	48
A[1]	1
A[2]	2
A[3]	8
A[4]	9
A[5]	15
A[6]	21
A[7]	27
A[8]	34
A[9]	48
A[10]	65

จากขั้นตอนดังกล่าวเรานำมาเขียนเป็นผังโปรแกรมได้ดังนี้



Insertion Sort



การเรียงลำดับแบบภายนอก (External Sort :Physical Sort)

การเรียงลำดับแบบนี้มักจะใช้กับแฟ้มข้อมูลขนาดใหญ่ที่เกินกำลังความสามารถในการที่จะอ่านข้อมูลจากแฟ้มบนดิสก์ (Disk or Diskette) มาจัดเก็บบนพื้นที่ของสมองเครื่องได้ การเรียงลำดับโดยวิธีนี้มักจะใช้ร่วมกับวิธีการของการเรียงลำดับแบบภายใน การเรียงลำดับโดยวิธีนี้จะแบ่งออกเป็นหลายประเภทด้วยกันเช่น

- ❖ Merge Sort
- ❖ Radix Sort
- ❖ Heap Sort (การเรียงลำดับแบบ Heap Sort นั้นเป็นวิธีการเรียงลำดับแบบหนึ่งของการเรียงลำดับแบบ Internal Sort คือจัดโครงสร้างให้เป็น Heap ภายใน Memory โดยใช้หลักของ Priority Queue แล้วนำไปประยุกต์ร่วมกับวิธีการของ Overlapping ระหว่าง Memory กับ External File ถ้าสนใจให้ไปอ่านในเรื่องการดำเนินการแบบ Heap Sort ในเรื่องของ File Processing)

ในหนังสือเล่มนี้จะขออธิบายเฉพาะวิธีการของ Merge Sort ซึ่งเป็นวิธีที่นิยมใช้กันมากในการเรียงลำดับกับแฟ้มข้อมูลแบบ Sequential File

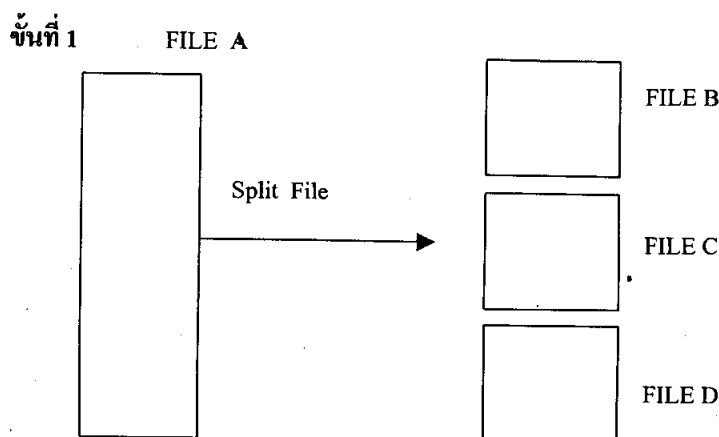
แนวทางในการทำ Merge Sort จะดำเนินการดังนี้คือ

ขั้นตอนที่ 1 นำแฟ้มข้อมูลนั้นมาแบ่งเป็นแฟ้มย่อยๆ โดยที่แต่ละแฟ้มนั้นจะมีขนาดที่สามารถอ่านเข้าไปเก็บในสองเครื่องได้หมดในคราวเดียว

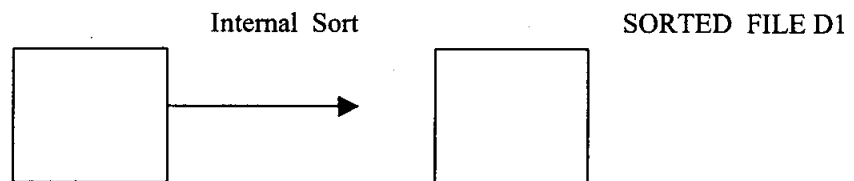
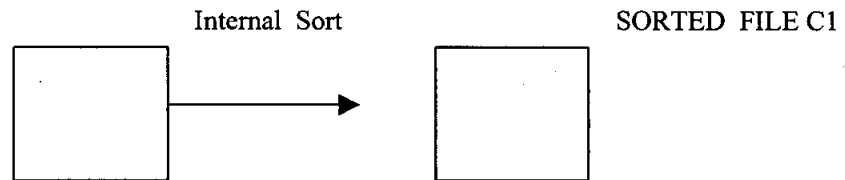
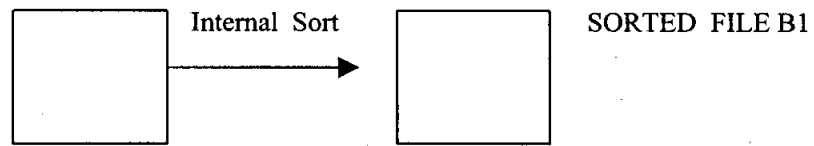
ขั้นตอนที่ 2 ดำเนินการเรียงลำดับแต่ละแฟ้มย่อยๆ ให้เรียบร้อย

ขั้นตอนที่ 3 นำแฟ้มย่อยๆ นั้นมาผสาน (Merge) เข้าด้วยกันเป็นแฟ้มใหม่ที่เรียงลำดับเรียบร้อยแล้ว

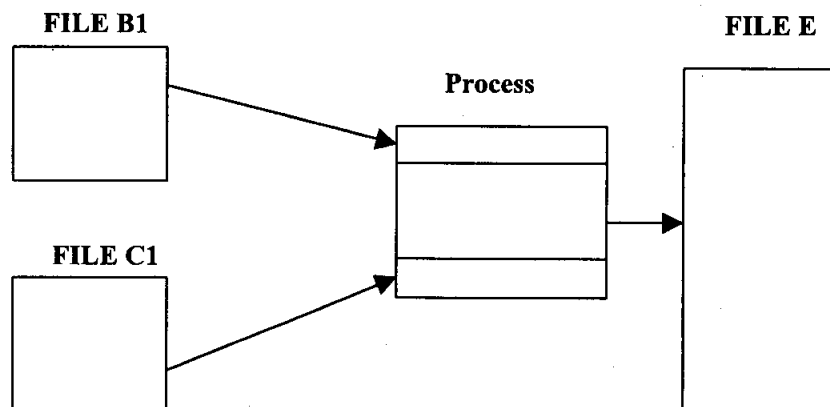
ขั้นตอนที่กล่าวมานี้จะเขียนเป็นรูปแบบดำเนินการได้ดังนี้



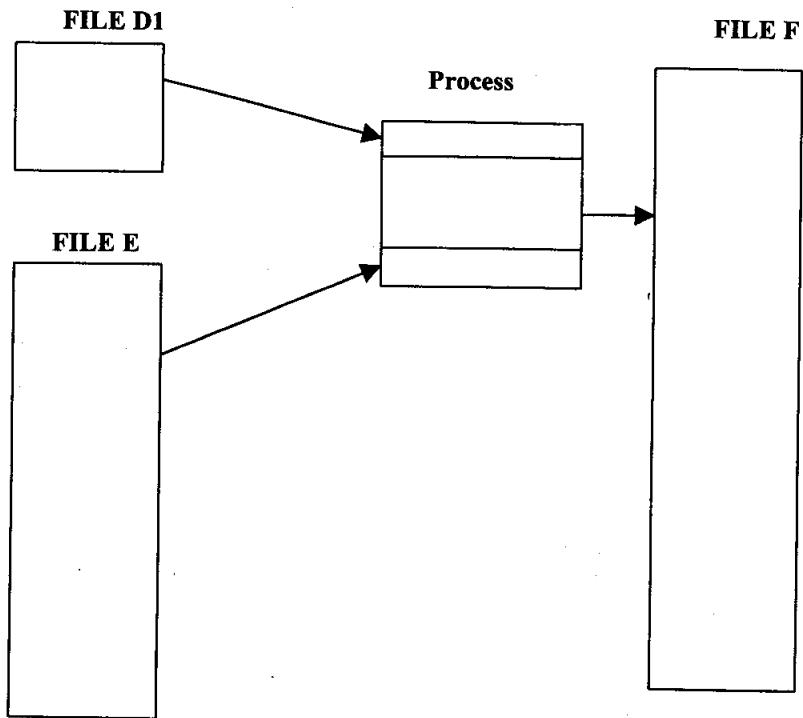
ขั้นที่ 2



ขั้นที่ 3 (Merge)



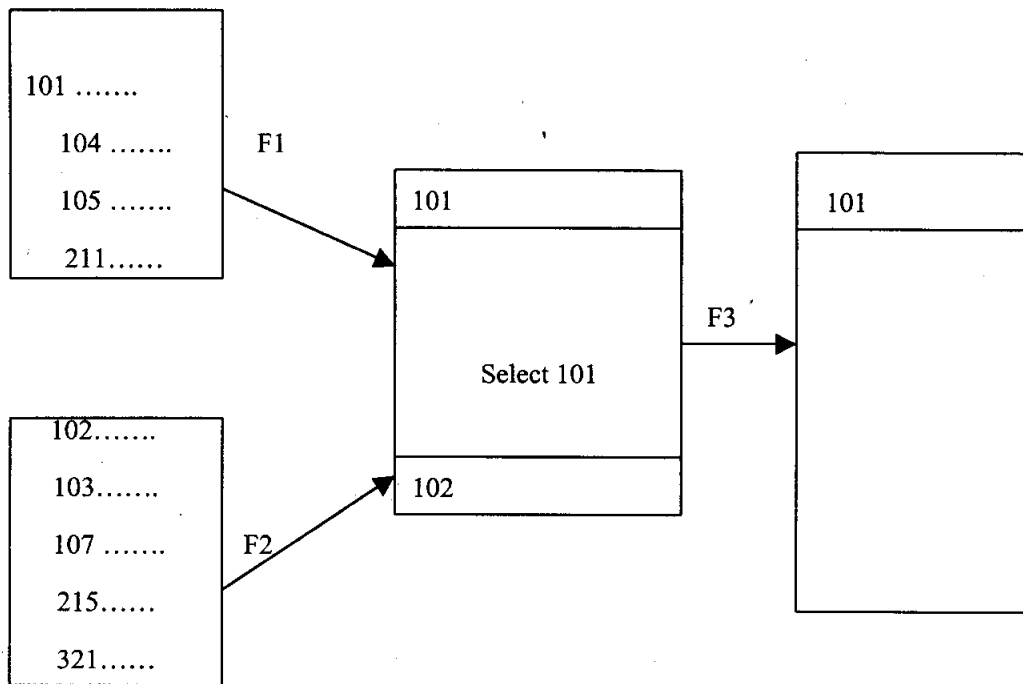
ขั้นที่ 4 (Merge) ขั้นสุดท้าย



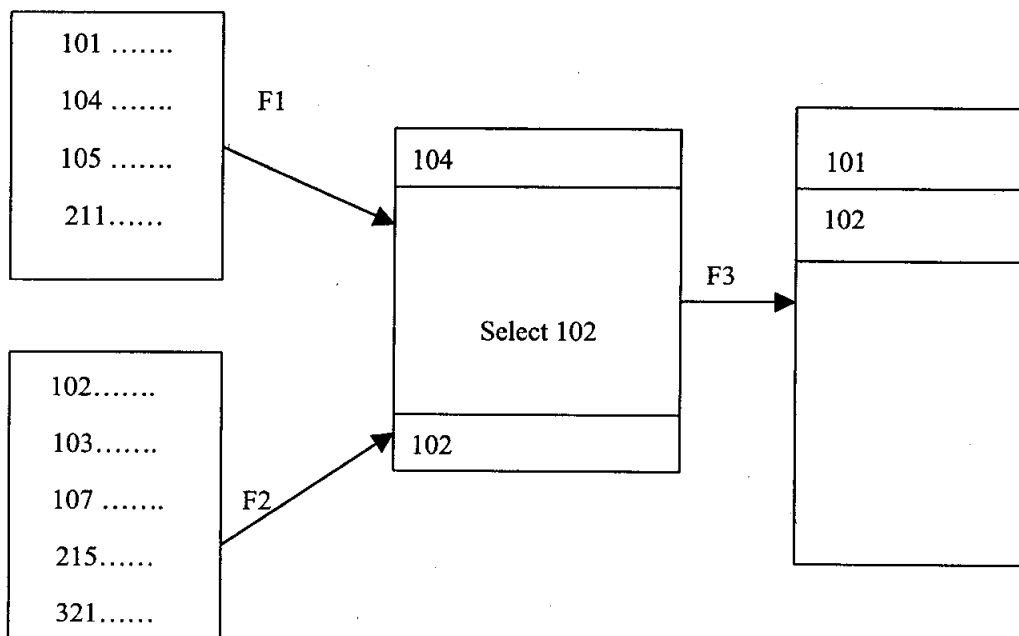
เราอาจจะรวมขั้นที่ 3 และ 4 เข้าด้วยกันในรูปของการผสานแบบ Multiway ก็ได้ โดยการควบคุมหลายๆ แฟ้มเข้าด้วยการอ่านเข้าไปผสานกัน

แต่เพื่อจะลดความยุ่งยากเนื่องจากผู้อ่านอาจจะพึงเริ่มศึกษาการออกแบบโปรแกรม ในที่นี้จะขออธิบายเฉพาะการทำ Merge Sort แบบ Two way คือใช้แฟ้มข้อมูลที่เรียงลำดับแล้วทีละ 2 แฟ้ม ดังนี้

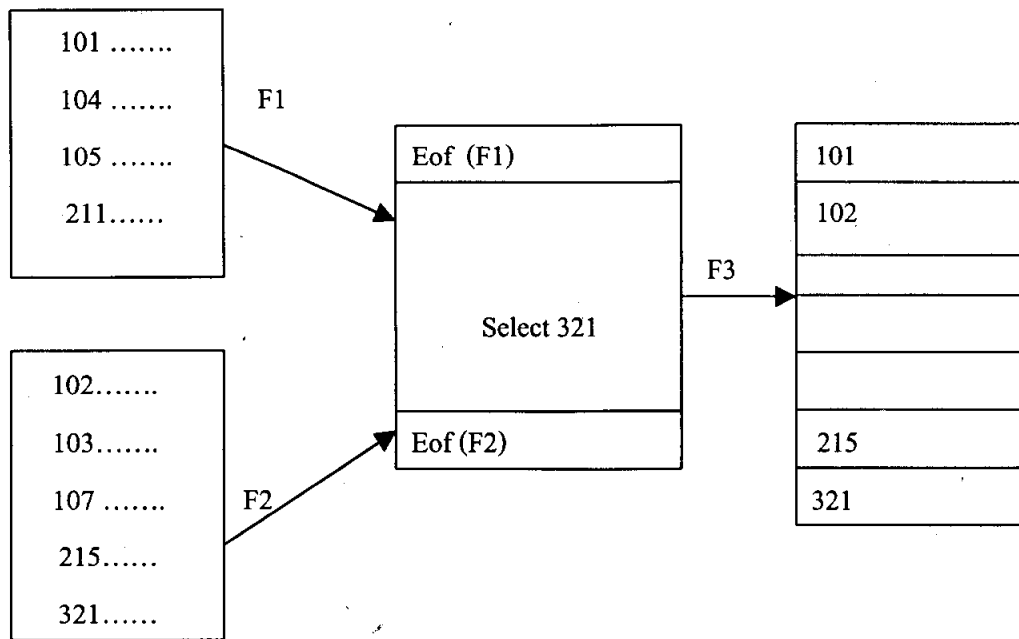
ขั้นที่ 1



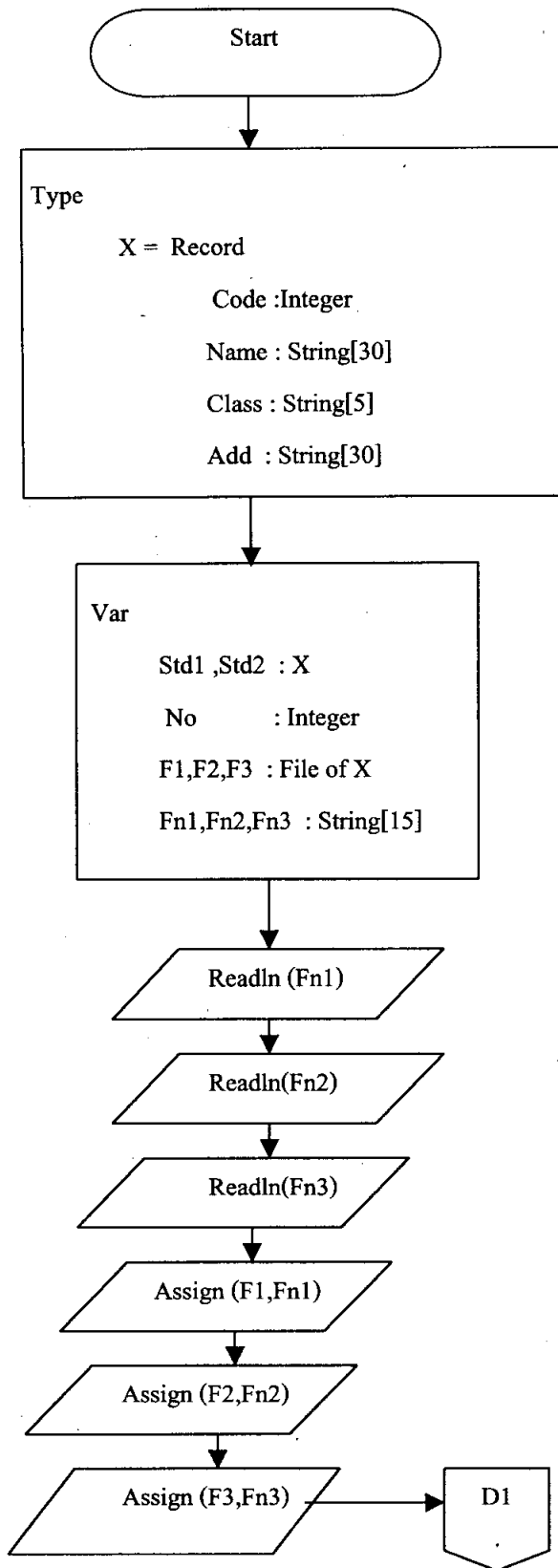
ขั้นที่ 2

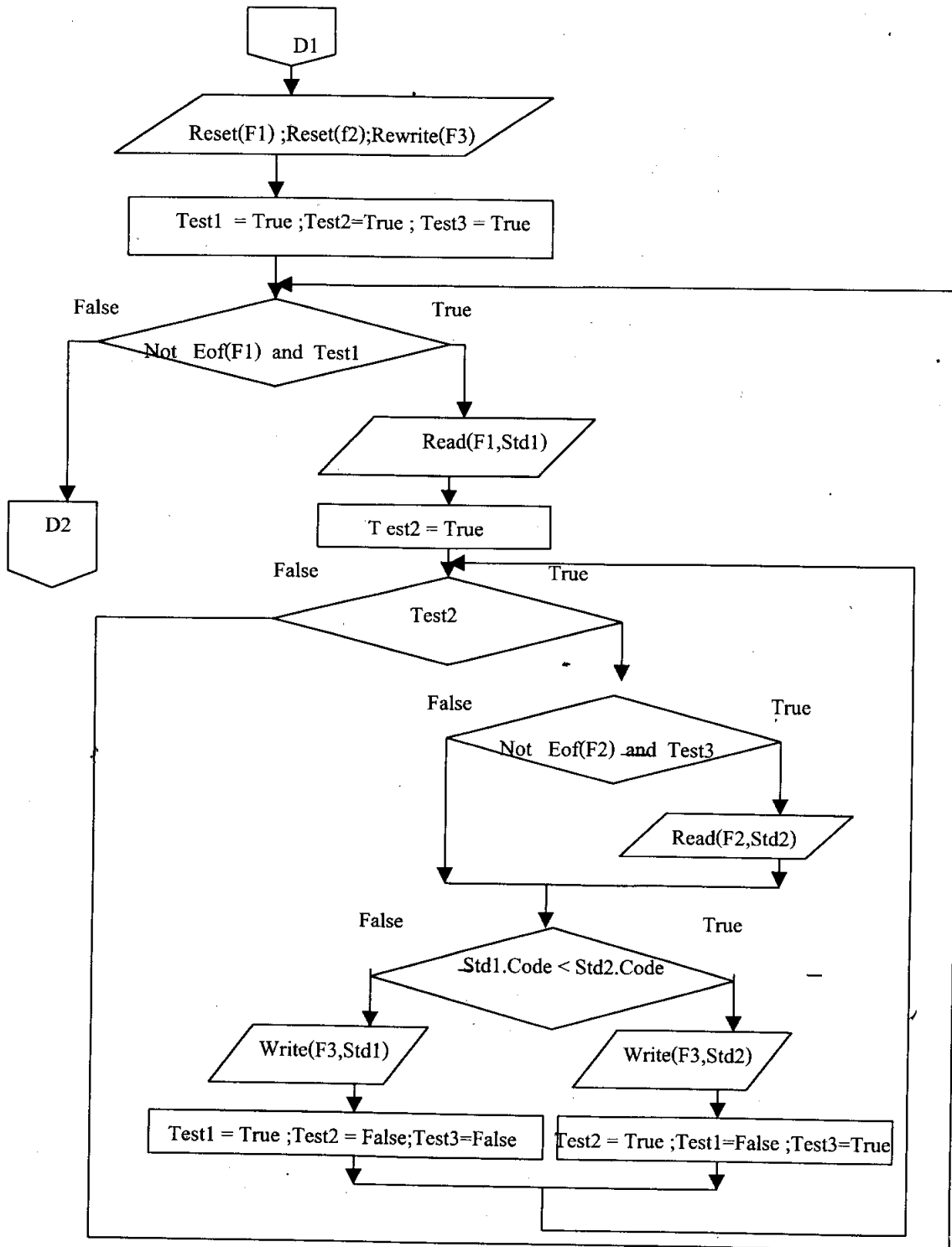


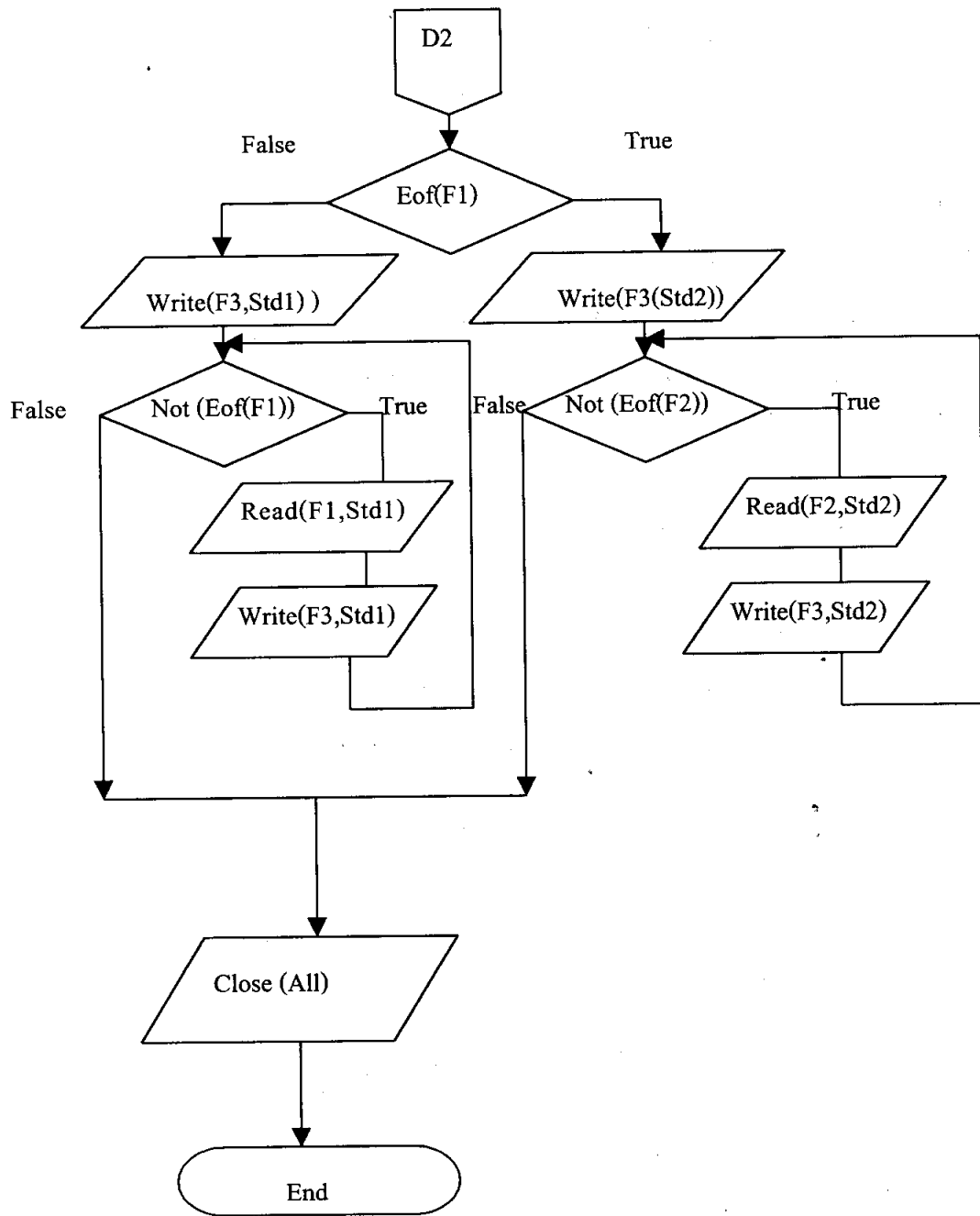
ขั้นสุดท้าย



ผังขั้นตอนดังกล่าวจะเขียนเป็นผังโปรแกรมได้ดังนี้







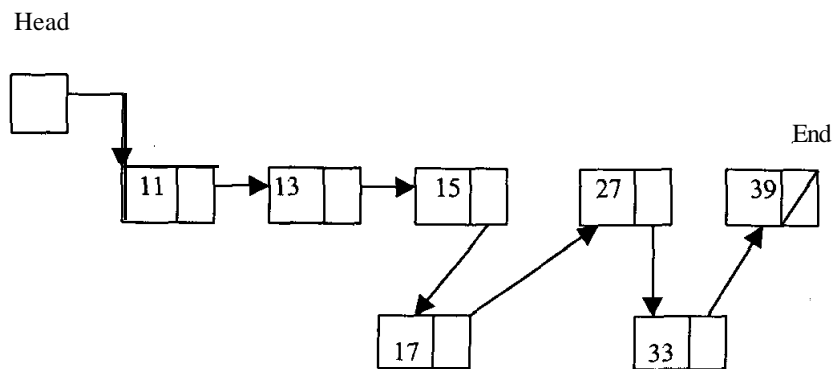
◆ **การค้นหา (Searching)** การค้นหานับว่าเป็นกิจกรรมที่ดำเนินการบ่อยมากทั้งในการประมวลผลแบบ กลุ่ม (Batch Processing) หรือแบบทันทีทันใด (Online Processing) ตัวอย่างการประมวลผลแบบกลุ่มที่เกี่ยวข้องกับการค้นหา เช่นการทำการวิเคราะห์การขาย (sale Analysis) เพื่อสร้างสารสนเทศสำหรับ ส่วนของ MIS (Management Information System) ส่วนกิจกรรมของการค้นหาในงานประเภททันทีทันใดนั้นจะพบมากในกิจกรรมของระบบสอบถาม กิจกรรมการค้นหานั้นจะพึงพิงอยู่กับกิจกรรมของการเรียงลำดับ นั่นหมายความว่า การเรียงลำดับและการจัดโครงสร้างข้อมูลจะส่งผลให้การสืบค้นเป็นไปได้อย่างรวดเร็วตามความต้องการของผู้ใช้

การสืบค้นนั้นจะแบ่งออกเป็น 2 วิธีคือ

1. การสืบค้นแบบเชิงเส้น (Linear Search)
2. การสืบค้นแบบทีละครึ่ง (Binary Search)

การสืบค้นทั้ง 2 วิธีนี้ได้กล่าวมาแล้วในบทที่ 7 ถึงวิธีการดำเนินงาน ดังนั้นเนื้อหาในส่วนนี้จะขออธิบายในส่วนของโครงสร้างข้อมูลนอกเหนือจากโครงสร้างของอะเรย์ปกติที่เราสร้างขึ้นเพื่อใช้ในการสืบค้นดังนี้คือ

● โครงสร้างแบบ Linked List โดยนำมาประยุกต์ในการเก็บข้อมูลแบบต่อเนื่องการ เราอาจจะเรียกว่า One Way List โดยที่ List จะมีความหมายถึงชุดของ Element โดยที่แต่ละ Element จะประกอบด้วย สารสนเทศ 2 ส่วนคือ ข้อมูล(Data) และรายการเชื่อมโยง (Link) ซึ่งจะทำหน้าที่เป็น Pointer รูปต่อไปนี้จะแสดงภาพการจัดเก็บข้อมูลที่เรียงลำดับแล้วที่จะส่งผลถึงการค้นหา



โครงสร้างข้อมูลแบบนี้จัดว่าเป็นการเรียงลำดับแบบเชิงตรรกะ โครงสร้างแบบนี้ถ้าเทียบกับ อะเรย์แล้วจะมีข้อดีและข้อเสีย ในส่วนที่เป็นข้อดีก็คือการลบ - แทรกข้อมูลจะกระทำได้ง่ายกว่าเพราะเพียงแต่เปลี่ยนตัวชี้(Pointer) เท่านั้น ในขณะที่ถ้าเป็นอะเรย์จะต้องทำการร่นหรือย้าย และในกรณีที่พื้นที่ที่จองไว้ไม่เพียงพอก็จะก่อให้เกิดปัญหาเพราะพื้นที่ของอะเรย์เป็นชนิด Static Variable

- โครงสร้างแบบ Inverted Array โดยใช้หลักการจัดเก็บแบบที่มีการเรียงลำดับได้หลายทาง ให้ออกจากตัวอย่างต่อไปนี้

Sorted By NAME HEAD

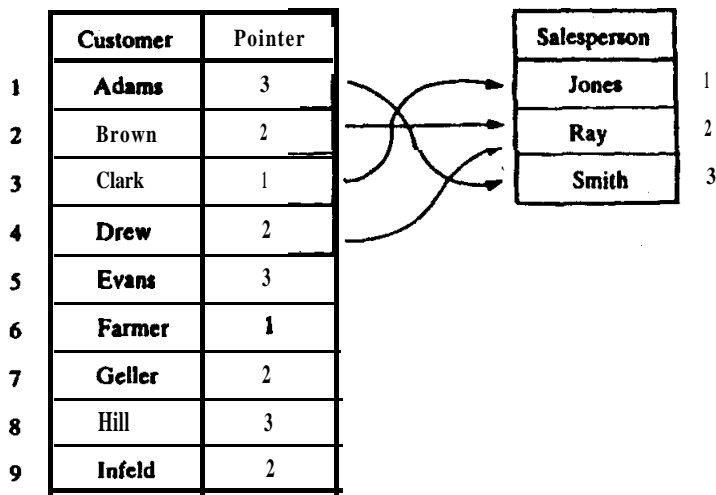
5

LOCATION	CODE	NAME	POINT
1	102	A5	6
2	109	A3	3
3	205	A4	1
4	208	A2	2
5	312	A1	4
6	345	A6	End

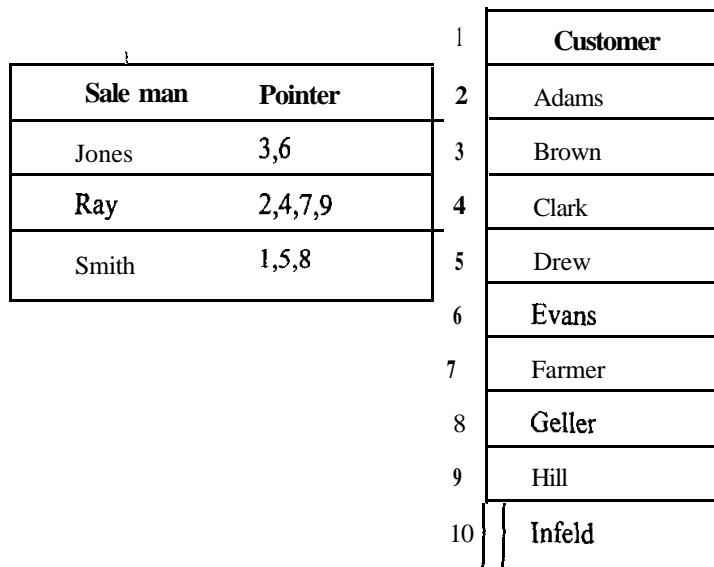
ตัวอย่างโครงสร้างข้อมูลตามที่ปรากฏนี้จะทำให้ผู้ใช้สามารถสืบค้นข้อมูลได้จาก รหัสประจำตัว เพราะว่าโครงสร้างข้อมูลนี้จะทำการ เรียงลำดับแบบกายภาพ (Physical Sort) และในขณะเดียวกันก็ทำการเรียงลำดับแบบตรรกะ (Logical Sort) ตามชื่อของคณงาน โดยใช้ Pointer ในการเชื่อมโยงบอกตำแหน่งลำดับถัดไป

- โครงสร้างแบบ Relation โดยใช้หลักการจัดเก็บแบบคล้ายกับการจัดการฐานข้อมูลแบบ Relational Database โดยการสร้างความสัมพันธ์ระหว่างอะเรย์ ซึ่งโดยกรรมวิธีที่นำมาประยุกต์นี้จะช่วยให้การดำเนินงานสืบค้นสิ่งที่ต้องการกระทำได้อย่างมีประสิทธิภาพ

ตัวอย่างการจัดโครงสร้างข้อมูลระหว่างข้อมูลของลูกค้ากับพนักงานขายโดยการใช้ Pointer



ตามความต้องการในการสืบค้นนี้เราอาจจะพัฒนาโครงสร้างใหม่ได้โดยการสร้างอะเรย์แบบระเบียบ โดยที่รายการข้อมูลแรกกำหนดให้เป็นชื่อของพนักงานขาย และรายการข้อมูลที่สอง กำหนดให้เป็นตำแหน่ง (Pointer) ที่จะชี้ไปที่ลูกค้าแต่ละคนจะมีชื่อดำเนินตรงที่ว่าพนักงานขายแต่ละคนจะดูแลลูกค้าจำนวนไม่เท่ากันดังนั้น การจัดโครงสร้างข้อมูลวิธีนี้จำเป็นจะต้องกำหนดเพื่อไว้ ดังตัวอย่างนี้



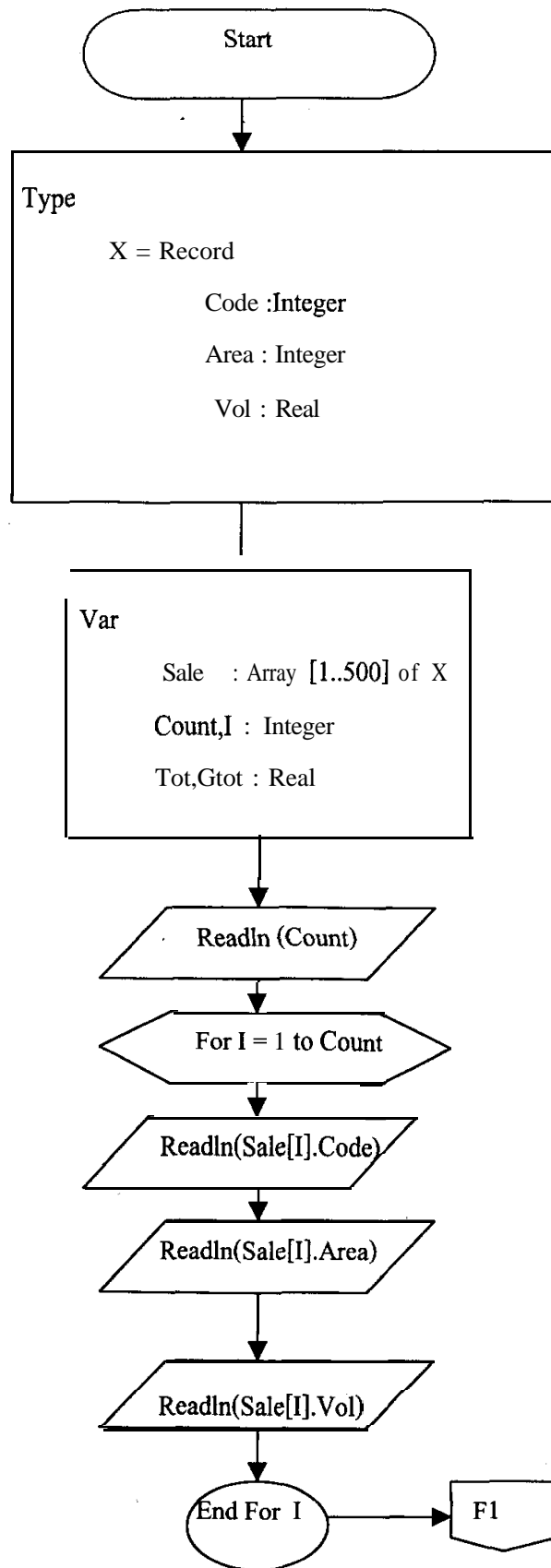
◆ **การคำนวณ (Calculating)** เป็นกิจกรรมที่นับเป็นหัวใจของระบบคอมพิวเตอร์ ด้วยเหตุที่ระบบคอมพิวเตอร์นั้นจะมีคุณลักษณะเด่นคือความเร็ว และระดับความถูกต้องในการคำนวณสูงกว่าการทำงานโดยมนุษย์

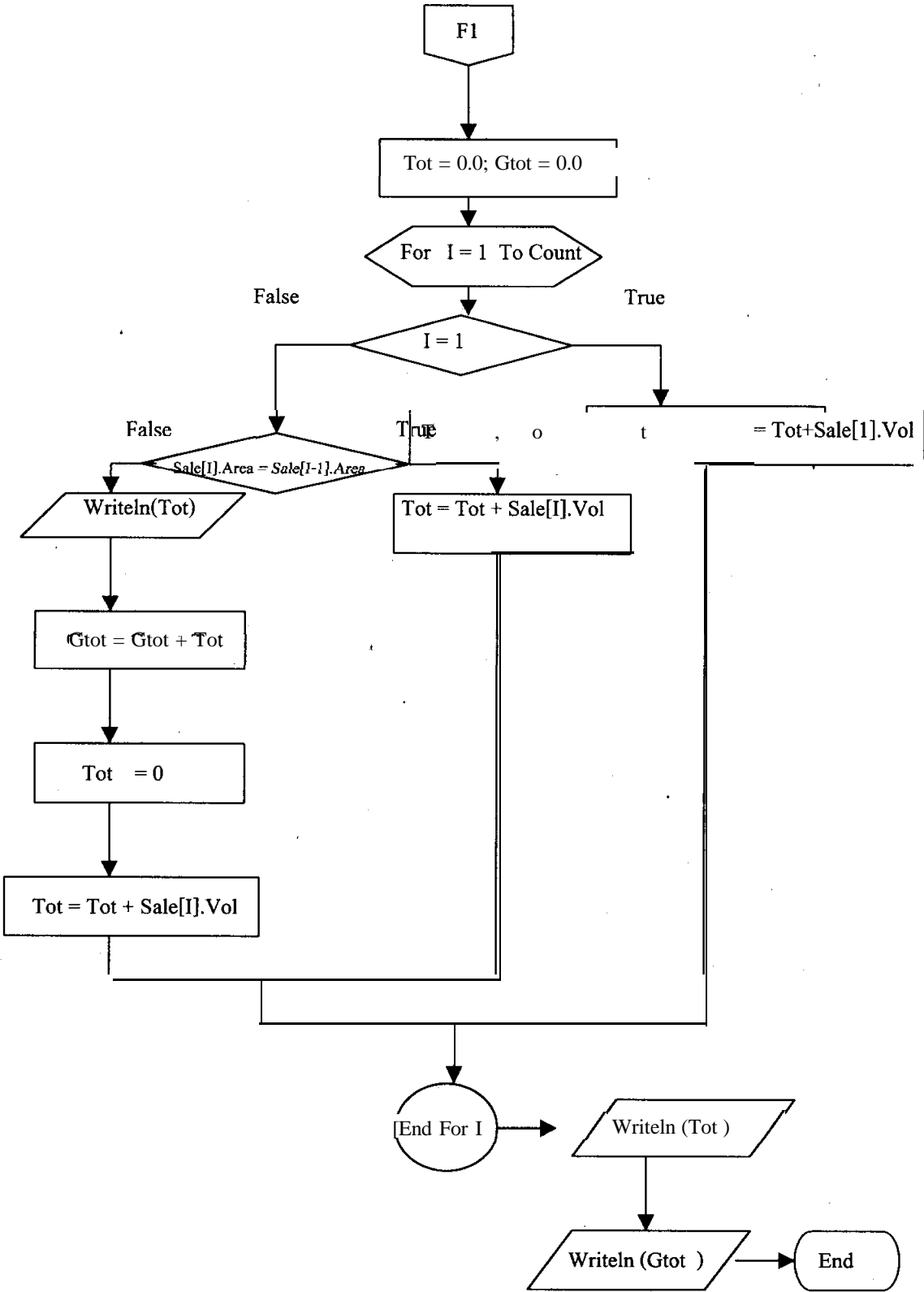
เนื้อหาในส่วนนี้จะจะนำประโยชน์ที่ได้จากการเรียงลำดับมาประยุกต์ในการคำนวณ เพื่อเป็นการประสานประโยชน์ระหว่างการเรียงลำดับกับการคำนวณ

ตัวอย่างกรณีศึกษา กำหนดให้เพิ่มข้อมูลหนึ่งซึ่งเป็นโครงสร้างของอะเรย์แบบระเบียบข้อมูลของพนักงานขายของบริษัทแห่งหนึ่ง โดยที่เพิ่มข้อมูลนี้จะเรียงลำดับตามพื้นที่การขาย การรับข้อมูลเพิ่มนี้ให้รับจากเป็นพิมพ์เข้าไปจัดเก็บในสมองเครื่อง ภายหลังให้คำนวณหายอดขายเป็นรายพื้นที่ของประเทศ และยอดขายรวมทั้งประเทศ กำหนดให้โครงสร้างข้อมูลปรากฏดังนี้

CODE	AREA	VOL
101	1	3457.50
102	1	5678.00
118	1	6000.50
202	2	3890.75
675	6	5690.00
725	7	8765.50
793	7	9800.00

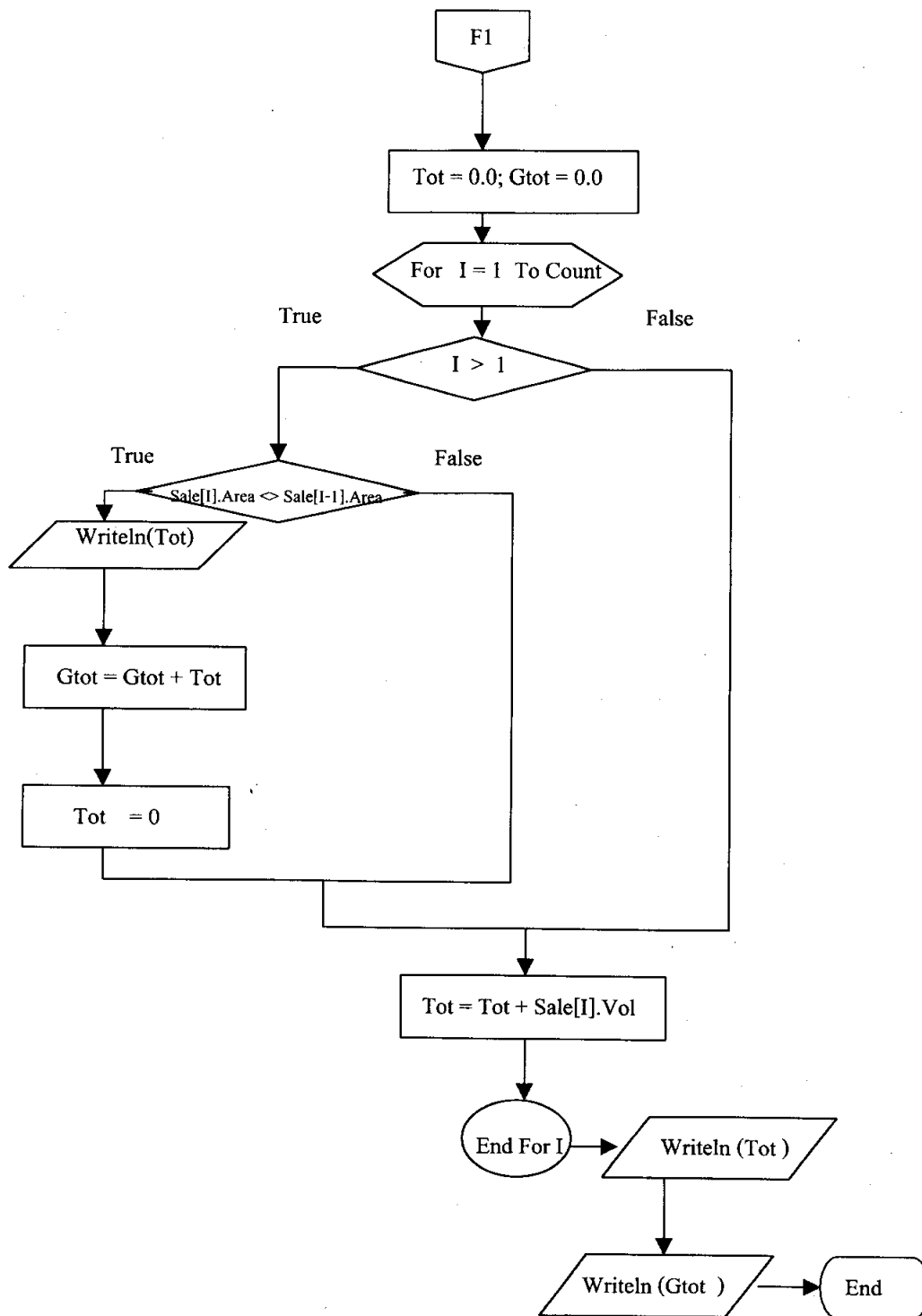
จงออกแบบโปรแกรม เพื่อทำการคำนวณสรุปยอดขายตามที่ต้องการ โดยการเขียนแสดงด้วยผังโปรแกรม



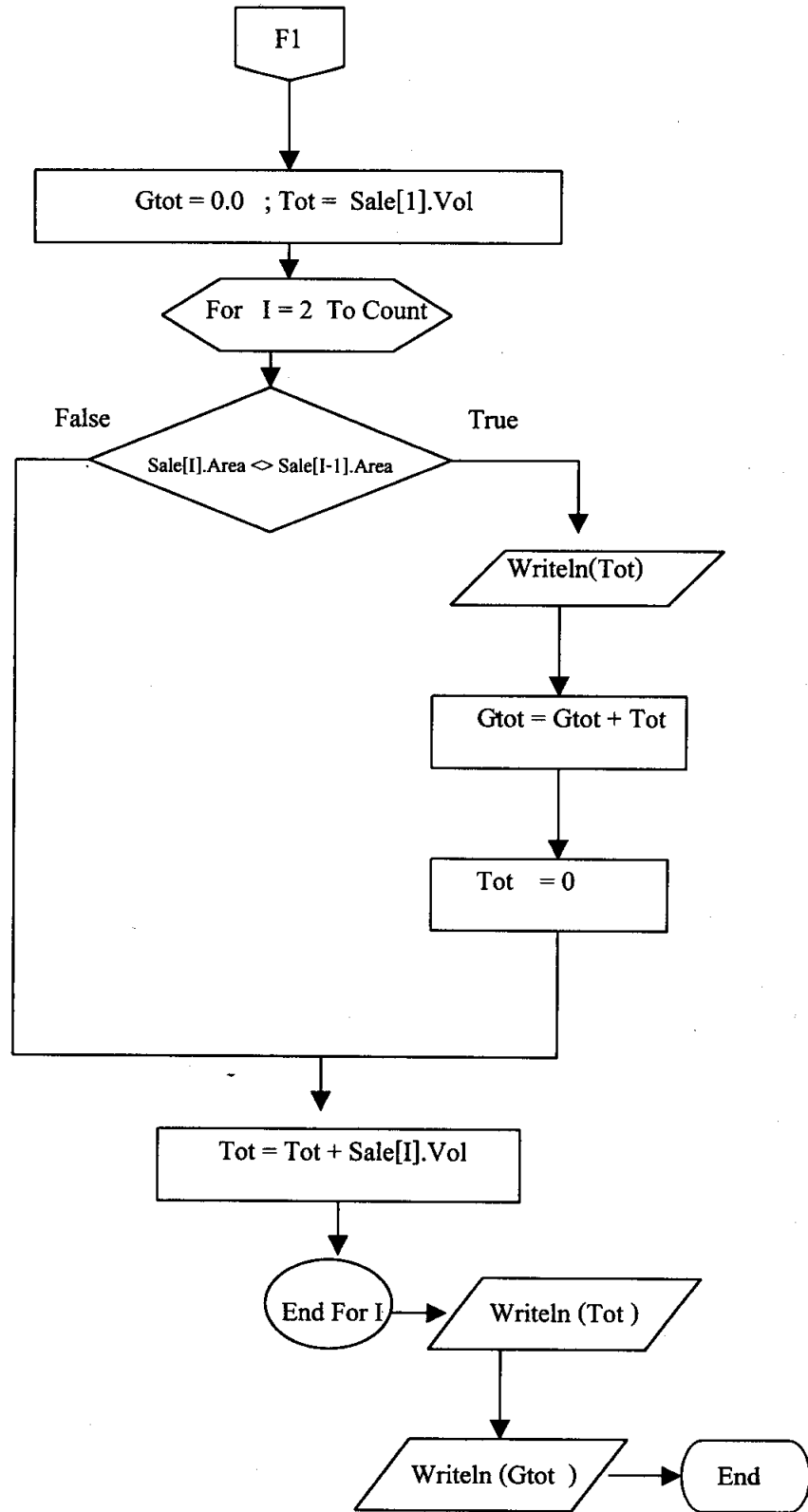


จากผังโปรแกรมในส่วนที่ดำเนินการคำนวณหาผลรวมยอดขายตามรายพื้นที่ เราอาจจะปรับเปลี่ยนให้เป็นทางเลือกที่ 2 และที่ 3 (Alternative 2 และ Alternative 3) ได้ดังนี้

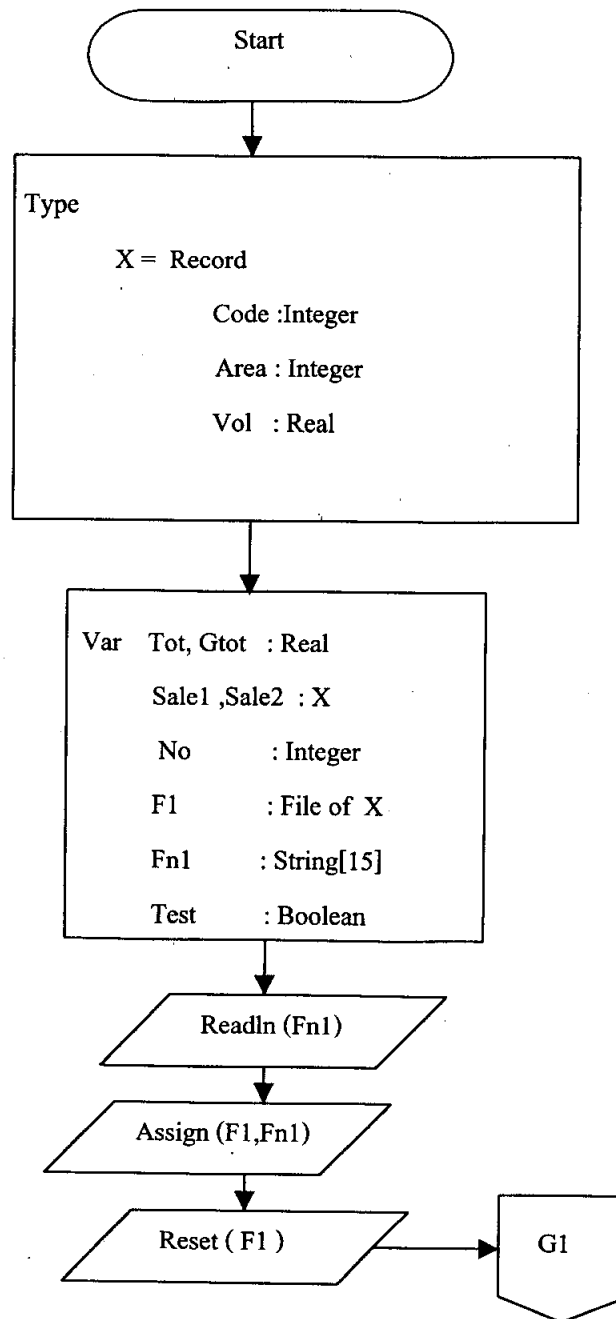
Alternative 2

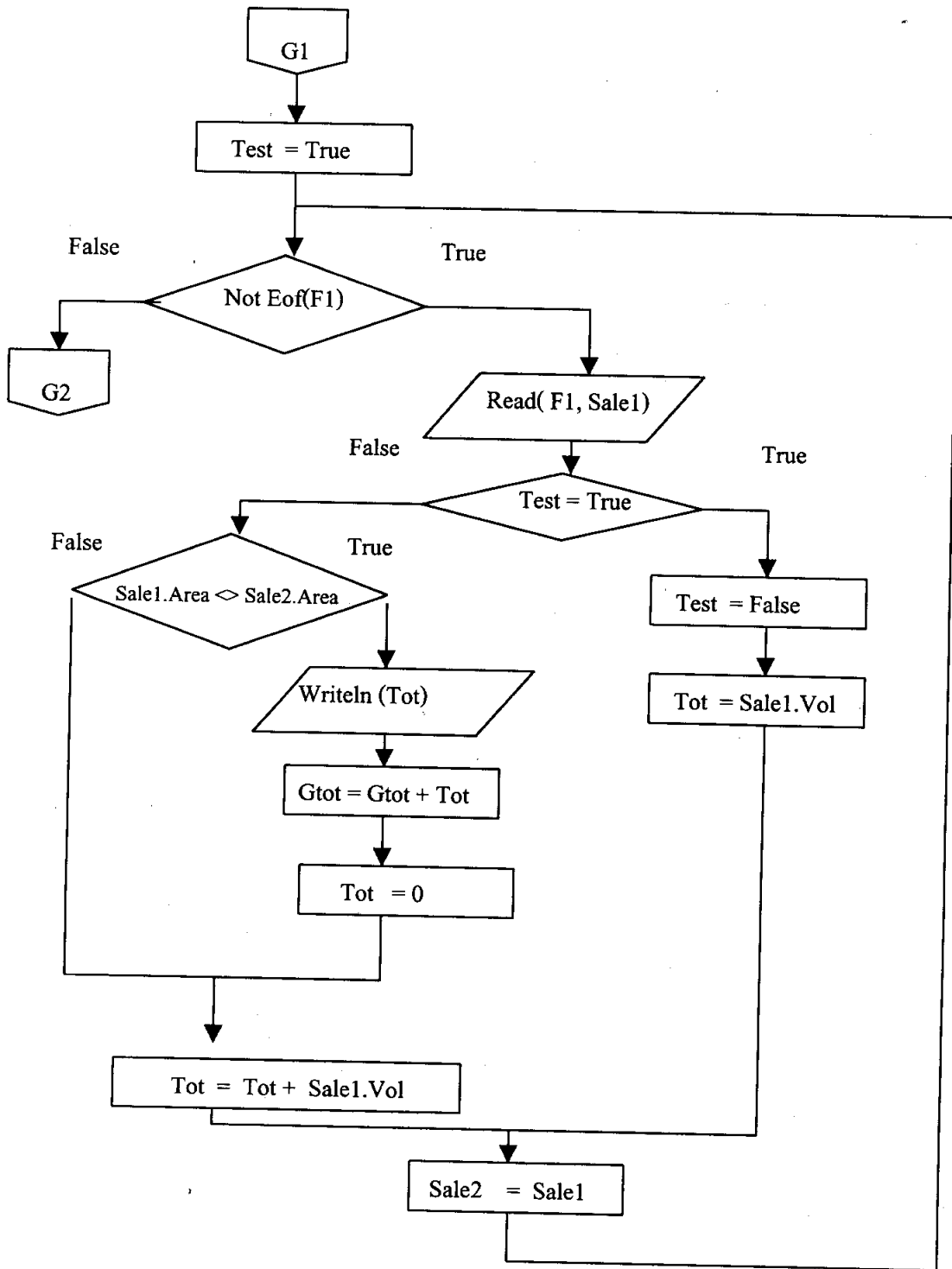


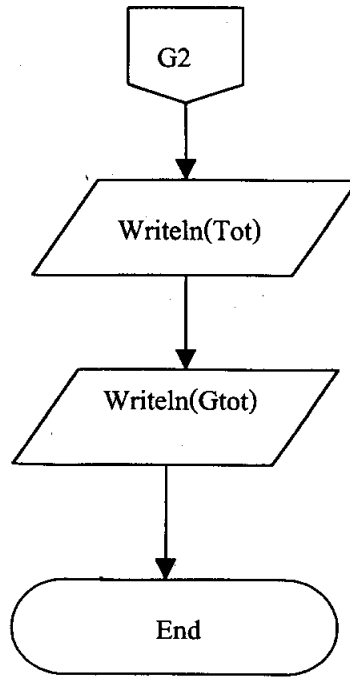
Alternative 3



การดำเนินงานตามตัวอย่างในการคำนวณผลสรุปของยอดขายโดยแยกไปตามรายพื้นที่นั้น ในกรณีที่ข้อมูลที่เรานำมาคำนวณได้มาจากเพิ่มถาวรบนสื่อ ดคยที่เพิ่มข้อมูลนั้นเก็บแบบเรียงลำดับแบบเดียวกับที่ปรากฏนั้นเราสามารถปรับผัง โปรแกรมให้เข้ากับระบบงานได้ ดังนี้





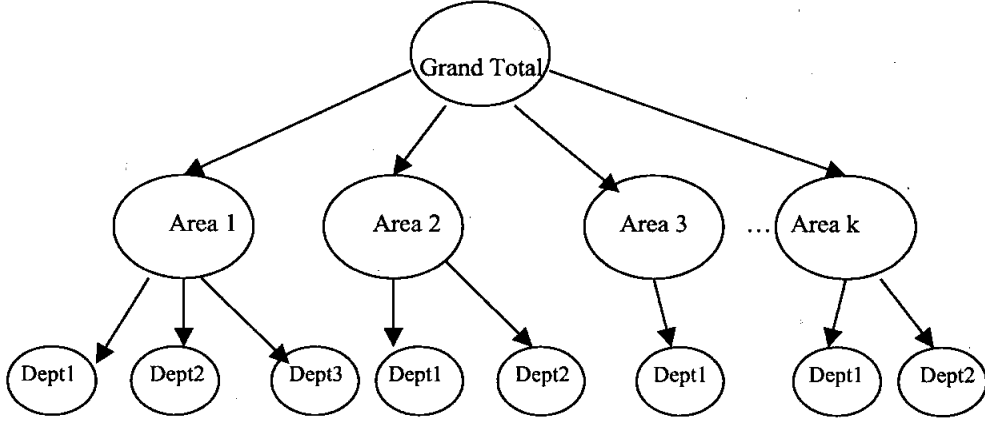


คำถามท้ายบท

1. จากโครงสร้างข้อมูลข้างล่างนี้ ให้นักศึกษาดำเนินการออกแบบเรียงลำดับข้อมูล โดยให้เรียงทั้ง Area และ Dept การออกแบบให้ดำเนินการโดยการเขียนแสดงด้วยผังโปรแกรม

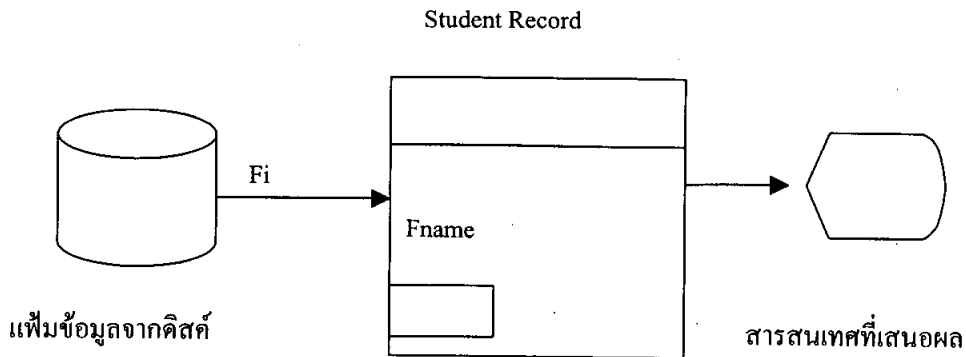
	Area	Dept	Code	Age	Address	Salary
Employee[1]						
Employee[50]						

2. จากงานดำเนินงานที่ได้ในข้อ 1 ให้ดำเนินการคำนวณหาเงินเดือนรวม โดยการแยกตาม Area และ Dept การออกแบบให้ดำเนินการโดยการเขียนแสดงด้วยผังโปรแกรม (ดูโครงสร้างของรายงานจากตัวแบบต่อไปนี้)



3 . จากระบบงานต่อไปนี้ ให้นำเพิ่มข้อมูลซึ่งนักศึกษาที่ลงทะเบียนในแต่ละวิชา โดยที่เพิ่มดังกล่าวประกอบด้วยระเบียนข้อมูลของนักเรียนแต่ละคน ซึ่งมีรายการข้อมูลคือรหัสประจำตัวและวิชาที่ลงทะเบียน 1 วิชา ไปดำเนินการสรุปเป็นรายวิชาและเรียงตามลำดับของรหัสของนักเรียนในแต่ละกระบวนวิชา การดำเนินงานให้เขียนผังโปรแกรม

ผังระบบในการทำงาน



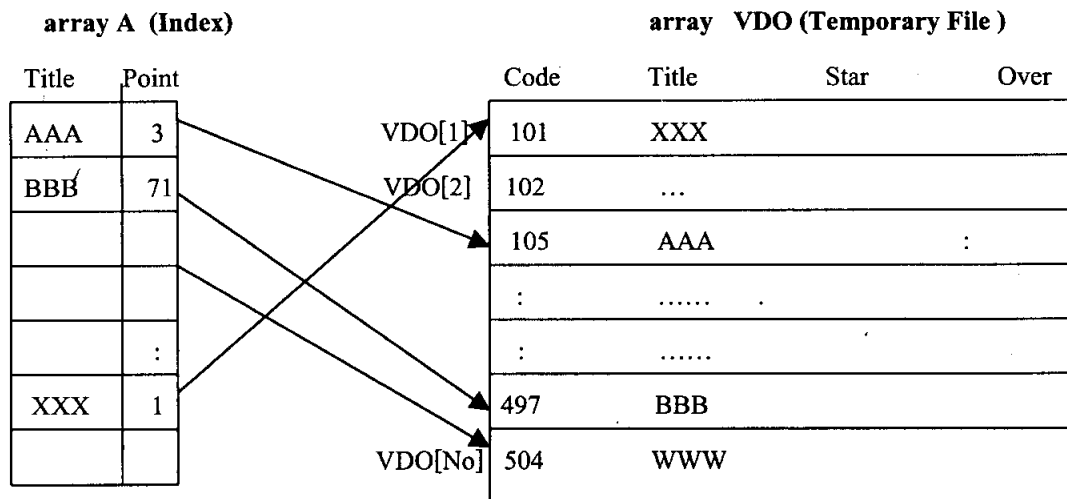
4. จากข้อมูลของชื่อวิดีโอสำหรับลูกค้า โดยที่วิดีโอแต่ละม้วนจะมีรายการข้อมูลต่อไปนี้คือ

- ❖ รหัสวิดีโอ (Code : Longint)
- ❖ ชื่อหนังวิดีโอ (Title : String[30])
- ❖ ดาราแสดงนำ (Star : String[40])
- ❖ เนื้อเรื่องย่อ (Over : String[100])

ลักษณะการดำเนินการจัดเก็บข้อมูล(กำหนดให้เรียงตาม Title)

	Code	Title	Star	Over
VDO[1]	301	ANT		
	205	:		
	106	:		
	:	.		
VDO[100]	72			

ให้ออกแบบโครงสร้างของ Logical Sort โดยการสร้าง คณิต Index ขึ้นมาใช้งาน รูปแบบในการสร้างข้อมูลเพื่อดำเนินการดังกล่าวจะปรากฏดังนี้



จงดำเนินงานดังกล่าวโดยการเขียนด้วยผังโปรแกรม

5. ในกรณีของเพิ่มข้อมูลรายชื่อผู้ใช้โทรศัพท์ซึ่งมีขนาดใหญ่ นั้นเราควรจะดำเนินการในงานเรียงตามรายชื่อของเจ้าของโทรศัพท์อย่างไรให้อธิบายและเขียนขั้นตอน (Algorithm) ประกอบการทำงาน