

## ภาคผนวก C

# ระเบียบวิธีการใช้คอมพิวเตอร์ ในการประมวลผลข้อมูล

การใช้คอมพิวเตอร์ในการประมวลผลข้อมูลให้ตรงตามวัตถุประสงค์ที่ต้องการนั้น โดยทั่ว ๆ ไปจะประกอบด้วยระเบียบวิธี 4 ขั้นตอนใหญ่ ๆ คือ

1. วิเคราะห์ถึงโครงสร้างของระบบวิธีการประมวลผลข้อมูลเพื่อให้ได้ผลตามที่ต้องการ ขั้นตอนนี้เรียกว่าการวิเคราะห์ผังระบบ ซึ่งจะเป็นการแสดงขั้นตอนการปฏิบัติงานทั้งระบบ

2. วางรายละเอียดที่แสดงถึงแผนงานและขั้นตอนการทำงาน รวมทั้งการใช้ตรรกวิทยา (logic) เพื่อจะประมวลผลข้อมูล ขั้นนี้เรียกว่าการเขียนผังโปรแกรม ขั้นนี้จะแสดงรายละเอียดการทำงานของโปรแกรมหนึ่ง ๆ โดยเฉพาะ (ในขั้นนี้อาจใช้เครื่องมืออื่น ๆ แทนผังโปรแกรมได้ดังจะกล่าวถึงในหัวข้อเกี่ยวกับเครื่องมือที่ใช้ในการพัฒนาโปรแกรม)

3. อาศัยสิ่งที่เขียนในขั้นที่ 2 เราเขียนคำสั่งที่ใช้กับคอมพิวเตอร์ด้วยภาษาที่เราเลือกใช้ให้เหมาะสม ขั้นนี้เรียกว่าการเขียนโปรแกรม (Program coding) หรือการถอดรหัส (โปรแกรมคือชุดของคำสั่ง)


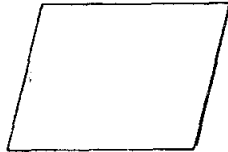
4. ตรวจสอบคำสั่งในขั้นที่ 3 ว่าให้ผลลัพธ์ (output) ตามความต้องการหรือไม่ ถ้ามีผิดพลาดจะได้ทำการแก้ไขหรือทำการตรวจแก้คำสั่ง (debuging) ความผิดพลาดที่ตรวจพบอาจเกิดจากการเขียนภาษาคอมพิวเตอร์ไม่ถูกต้องตามไวยากรณ์หรือกฎเกณฑ์ของภาษา ซึ่งจะถูกตรวจพบในตอนที่ตัวแปลภาษาทำการแปล เราเรียกความผิดชนิดนี้ว่า Syntax error หรือ Compile-time error นอกจากนั้นความผิดพลาดอาจเกิดจากขั้นตอนในการทำงาน วิธีการทำงาน การใช้ตรรกวิทยาผิด ข้อมูลเข้าผิดพลาดและอื่น ๆ ซึ่งจะตรวจพบในตอนวิ่งโปรแกรม (run program หรือ execute program) เราเรียกความผิดพลาดชนิดนี้ว่า (logic error หรือ run-time error)

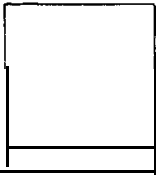
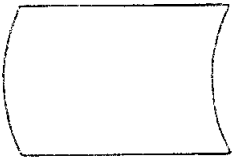


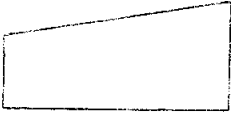
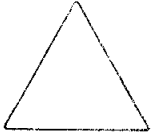
## 1. ผังระบบ (System flowchart)

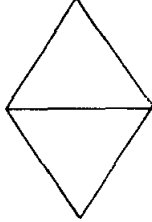
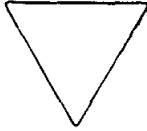



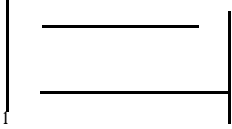
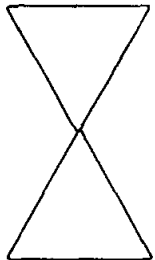
การที่จะสั่งคอมพิวเตอร์ให้ทำงานเพื่อให้ได้ผลลัพธ์หรือข้อมูลออกตามที่ต้องการนั้น สิ่งสำคัญที่ต้องปฏิบัติคือการกำหนดขั้นตอนการปฏิบัติงาน และวิธีที่ใช้ปฏิบัติของงานทั้งหมด นอกจากนั้นยังต้องทราบอีกว่าข้อมูลที่จะนำมาใช้นั้นเก็บอยู่ในตัวกลางชนิดใด และมีการส่งต่อข้อมูลไปในลักษณะใด รวมถึงท้ายที่สุดได้ข้อมูลออกมาในรูปของตัวกลางชนิดใด การกำหนดขั้นตอนตั้งแต่ป้อนข้อมูลเข้า การส่งต่อข้อมูลจนถึงได้ข้อมูลออก เรารวมเรียกว่า การวางแผนผังระบบซึ่งจะช่วยให้ผู้ที่มีหน้าที่หรือเกี่ยวข้องกับกระบวนการประมวลผลข้อมูลด้วยคอมพิวเตอร์ทราบว่าข้อมูลแต่ละขั้นตอนในการประมวลผลข้อมูลนั้นเก็บอยู่ในรูปใด และมีการเคลื่อนย้ายข้อมูลอย่างไรไปในรูปใด

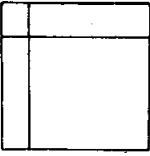

งานประมวลผลข้อมูลด้วยคอมพิวเตอร์นั้น โดยทั่วไปนั้นถ้าดำเนินงานง่าย ๆ และไม่ยุ่งยากซับซ้อน จะวางแผนโดยการเขียนอธิบายเป็นภาษาเขียนนั้นก็สามารถทำได้และเข้าใจกันได้ แต่ในกรณีที่เป็นงานขนาดใหญ่และมีขั้นตอนในการทำงานยุ่งยากมาก วิธีที่ง่ายและสะดวกก็คือ การเขียนผังระบบโดยใช้สัญลักษณ์ที่ตั้งขึ้นเพื่อให้เข้าใจเป็นอย่างเดียวกัน การตั้งสัญลักษณ์ขึ้นมาเป็นมาตรฐานเพื่อใช้ช่วยในการเขียนผังระบบการประมวลผลนั้นจะทำให้สามารถเขียนและอ่านได้ง่าย นอกจากนั้นยังช่วยลดข้อผิดพลาดในการหลงลืมขั้นตอนต่าง ๆ หรือการใช้ตัวกลางในการเก็บข้อมูลผิดพลาดประการอื่นอีกด้วย

### สัญลักษณ์ทั่ว ๆ ไปที่ใช้ในผังระบบ

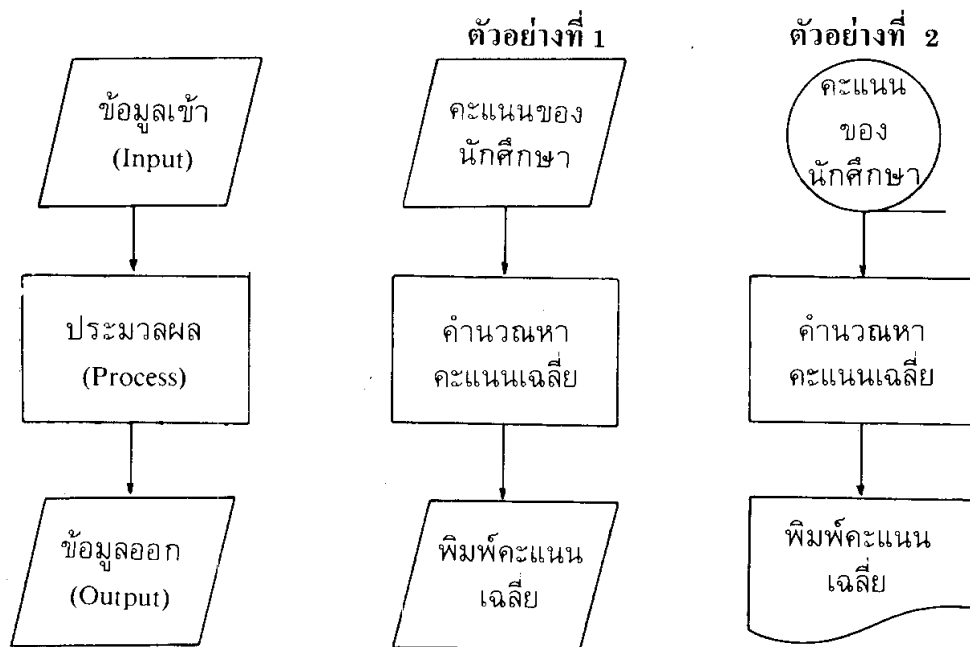
ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์
1. การประมวลผลข้อมูล (Process symbol) ใช้แทนงานประมวลผลที่ต้องปฏิบัติ	รูป □ ผืนผ้า 
2. การนำข้อมูลเข้า/ออก (Input/Output หรือ I/O symbol) ใช้แทนการนำข้อมูลเข้าหรือการนำข้อมูลออก ใช้เมื่อไม่เจาะจงตัวกลางที่จะใช้ (ถ้าต้องการระบุตัวกลางให้ใช้รูปที่แทนเฉพาะตัวกลางได้ ดูจากเรื่องผังโปรแกรม)	รูป □ ด้านขนาน 

ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์
3. การปฏิบัติงานช่วย (Auxiliary operation) เป็นงานที่ใช้เครื่องจักรช่วยในการประมวลผลข้อมูล แต่ไม่ได้อยู่ภายใต้การควบคุมของคอมพิวเตอร์ เช่น การใช้เครื่อง unit record ช่วย	รูป □ จัตุรัส 
4. หน่วยความจำเชื่อมต่อตรง (Online storage) เป็นตัวกลางในการเก็บข้อมูลที่อยู่ภายใต้การควบคุมของตัวซีพียูโดยตรง เช่น เทปแม่เหล็ก จานแม่เหล็ก	
5. หน่วยความจำไม่เชื่อมต่อตรง (Offline storage) เป็นตัวกลางในการเก็บข้อมูลที่อยู่นอกการควบคุมของตัวซีพียู	
6. การปฏิบัติงานประมวลผลเฉพาะกิจ (Manual operation) เช่น การตรวจสอบการคำนวณด้วยเครื่องคิดเลขโดยใช้คนตรวจสอบ (manual checking)	รูปสี่เหลี่ยมคางหมู 
7. การป้อนข้อมูลเข้าด้วยมือ (Manual input) เป็นการป้อนข้อมูลเข้าทางแป้นพิมพ์ที่ console typewriter ในขณะที่คอมพิวเตอร์กำลังปฏิบัติงาน	
8. การแยกข้อมูล (Extract symbol) เป็นการแยกข้อมูลบางส่วนที่มีลักษณะตามต้องการออกมาจากแฟ้มข้อมูล	

ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์
9. การเรียงลำดับ (Sort symbol) เป็นการเรียงลำดับข้อมูล โดยอาจเรียงตามข้อมูลเชิงปริมาณจากมากไปน้อย หรือจากน้อยไปมาก หรืออาจเรียงตามลำดับตัวอักษรก็ได้	
11. การรวมแฟ้มข้อมูล (Merge symbol) เป็นการรวมแฟ้มข้อมูล ตั้งแต่ 2 แฟ้มข้อมูลเข้าด้วยกัน	
11. จุดต่อเนื่อง (Connector) ที่อยู่ในหน้าเดียวกัน	
12. จุดต่อเนื่องที่อยู่คนละหน้า (Offpage connector)	
13. สายสื่อสาร (Communication link) แทนการต่อสายสื่อสาร หรือการส่งข้อมูลอย่างอัตโนมัติจากที่หนึ่งไปยังอีกที่หนึ่ง เช่น โทรพิมพ์ โทรศัพท์	
14. ลูกศร (arrow หรือ flowline) แสดงทิศทางการประมวลผล ชั้นต่อ ๆ ไป	
15. การตรวจทานเรียงลำดับ (Collate symbol )	

ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์
16. หน่วยความจำหลัก (Core storage)	
17. การเขียนคำอธิบายประกอบ (Comment annotation) ใช้สำหรับเขียนคำอธิบายรายละเอียดเพิ่มเติมในส่วนของผังระบบ	

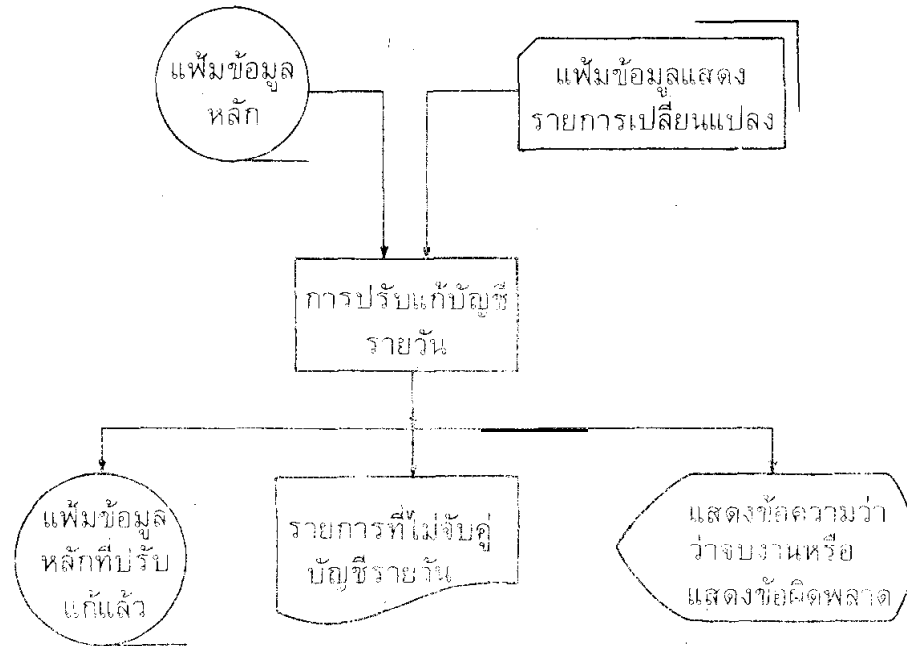
ระบบการประมวลผลข้อมูลมีผังระบบแสดงได้ง่าย ๆ เป็น 3 บล็อก คือ ข้อมูลเข้า การประมวลผล และข้อมูลออก ดังแสดงในผังระบบอย่างพื้นฐาน (Basic system flowchart)



ตัวอย่างที่ 1 แสดงระบบการทำคะแนนเฉลี่ยตั้งแต่อ่านคะแนนเข้าไป ทำการคำนวณ แล้วพิมพ์คะแนนเฉลี่ยที่คำนวณได้ การแสดงขั้นการทำงานนี้เป็นไปอย่างหยาบ ๆ ไม่ได้เขียนขึ้นเพื่อการที่จะนำไปเขียนคำสั่งสั่งให้คอมพิวเตอร์ปฏิบัติตาม บางครั้งเราเรียกผังระบบ

ว่า Outline หรือ macro flowchart จากตัวอย่างที่ 2 นั้น ถ้าเราเก็บคะแนนของนักศึกษาไว้ในเทปแม่เหล็กเราอาจใช้รูปแทนเทปแม่เหล็กแทนรูปสี่เหลี่ยมด้านขนาน และอาจใช้รูปแทนกระดาษจากเครื่องพิมพ์แสดงการพิมพ์ลงบนกระดาษ โดยใช้เครื่องพิมพ์ (Printer)

**ตัวอย่างที่ 3** ผังระบบแสดงการปรับแก้ (updating) บัญชีรายวัน



ในภาพเพิ่มข้อมูลหลัก (Master file) คือบัญชีรายวันซึ่งเก็บอยู่ในเทปแม่เหล็ก ส่วนเพิ่มข้อมูลแสดงรายการเปลี่ยนแปลง (Transaction file) อยู่ในบัตรเจาะรู สัญลักษณ์นั้นเป็น card file เพิ่มข้อมูลทั้ง 2 เป็นข้อมูลเข้า การประมวลผลในที่นี้คือการปรับแก้เพิ่มข้อมูลหลัก โดยใช้รายการข้อมูลในเพิ่มข้อมูลแสดงรายการเปลี่ยนแปลง ผลจากการปรับแก้ (updating) จะเก็บและแสดงข้อมูลออกมาดังนี้ คือ 1) เพิ่มข้อมูลหลักที่ปรับแก้แล้ว (updated master file) จะเก็บไว้ในเทปแม่เหล็ก 2) พิมพ์รายการข้อมูลที่ไม่จับคู่ไม่ได้ (unmatched item) ซึ่งอาจเกิดจากการบันทึกเลขที่ประจำเรคคอร์ดที่จะแก้ไขไม่ถูกต้อง จึงไม่สามารถค้นพบเรคคอร์ดนั้น ๆ ในเพิ่มข้อมูลหลักได้ นอกจากนั้นยังพิมพ์บัญชีรายวันออกมาด้วย และ 3) เมื่อจบบางหรือมีข้อความแสดงความผิดพลาดมันจะแสดงออกมาทางจอภาพ

## 2. เครื่องมือสำหรับการพัฒนาตรรกะของโปรแกรม (Tools for logic development)

ในที่นี้จะกล่าวถึงสิ่งต่อไปนี้คือ

1. ผังโปรแกรม (Program flowchart)
2. ผังโปรแกรมแบบโครงสร้าง (Structure program flowchart)
3. รหัสเทียม (Pseudocode)
4. ตารางการตัดสินใจ (Decision table)
5. Hierarchy (Structure) chart
6. ไฮโป (HIPO)
7. ผังภาพของแนสซี-ซีในเดอร์แมน

### 2.1 ผังโปรแกรม (Program flowchart)

ผังโปรแกรม (Program flowchart หรือ logic diagram) เป็นผังภาพที่ใช้เป็นเครื่องมือที่โปรแกรมเมอร์ใช้เพื่อพัฒนาตรรกะของโปรแกรม ผังโปรแกรมจะเน้นที่ลำดับของการแปลงรูปข้อมูลเข้า (input) ไปเป็นข้อมูลออก (output) หรือกล่าวอีกอย่างหนึ่งว่าผังโปรแกรมจะแสดงให้เห็นว่าข้อมูลในเพิ่มข้อมูลเข้ากลายเป็นเพิ่มข้อมูลออกและ/หรือรายงานที่เราต้องการได้อย่างไร

#### เหตุผลในการเขียนผังโปรแกรม

1. เนื่องจากว่าผังโปรแกรมเป็นไดอะแกรมหรือรูปแสดงตรรกะของโปรแกรม การวาดผังโปรแกรมจะทำให้โปรแกรมเมอร์มองเห็นอย่างชัดเจนว่าโปรแกรมของเขาจะทำอะไร
2. ผังโปรแกรมเป็นส่วนหนึ่งของเอกสารประกอบโปรแกรม (program documentation) เราอาจใช้มันเพื่อเป็นเครื่องสื่อสารระหว่างโปรแกรมเมอร์ด้วยกัน ระหว่างโปรแกรมเมอร์และนักวิเคราะห์ (Analysts) หรือระหว่างโปรแกรมเมอร์และผู้ใช้โปรแกรม (users) เนื่องจากผังโปรแกรมเป็นอิสระจากภาษาที่ใช้เขียนโปรแกรม ดังนั้น จึงไม่จำเป็นต้องรู้ภาษาคอมพิวเตอร์มาก่อนเพื่อที่จะทำความเข้าใจหรือทบทวนตรรกะที่แสดงไว้ในผังโปรแกรม
3. ผังโปรแกรมช่วยให้โปรแกรมเมอร์ทดสอบคำตอบต่างๆ ที่อาจเกิดขึ้นได้ สำหรับปัญหานั้น ๆ โดยที่ยังไม่ต้องเขียนโปรแกรม ผังโปรแกรมเป็นเครื่องมือสำหรับการตรวจสอบด้วยมือ (desk checking) อย่างง่าย ๆ นั่นคือโปรแกรมเมอร์สามารถใช้ข้อมูลตัวอย่างเป็นข้อมูลเข้า เพื่อติดตามดูการทำงานตามผังโปรแกรมว่าจะเกิดผลลัพธ์ตามที่เรต้องการหรือไม่ ถึงแม้ว่าการตรวจสอบนั้นอาจจะทำที่ขั้นใด ๆ ของขบวนการเขียนโปรแกรมก็ได้ แต่มันจะง่ายกว่าที่จะค้นหาและแก้ไขข้อผิดพลาดทางตรรกะ (logic error) ก่อนที่จะลงมือเขียนโปรแกรม

## อุปสรรคและข้อเสียของการเขียนผังโปรแกรม


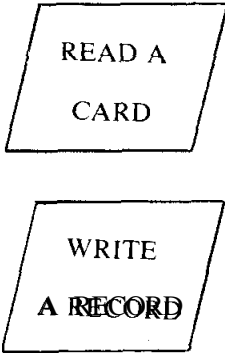
1. การเขียนผังโปรแกรมนั้นเสียเวลา แต่อย่างไรก็ดี เวลาที่ใช้ในการเขียนผังโปรแกรมที่ดีจะได้รับการชดเชยจากการประหยัดเวลาในการเขียนและทดสอบโปรแกรม (program testing)

2. ผังโปรแกรมแสดงวิธีที่ข้อมูลเข้ากลายเป็นข้อมูลออกว่าทำอะไร (how) แต่ไม่ได้แสดงว่าทำไม (why) จึงต้องทำขั้นนั้นขั้นนี้ ดังนั้น จึงจะเป็นประโยชน์มากถ้าท่านจะเขียนคำอธิบาย (comment) ในผังโปรแกรมของท่านซึ่งจะเห็นตัวอย่างต่อไป

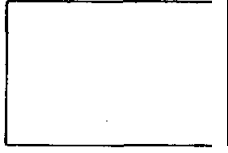
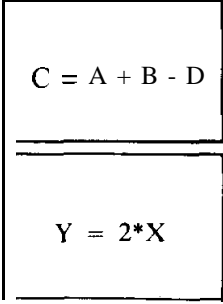
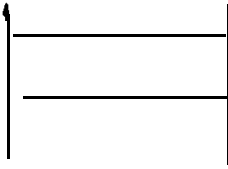
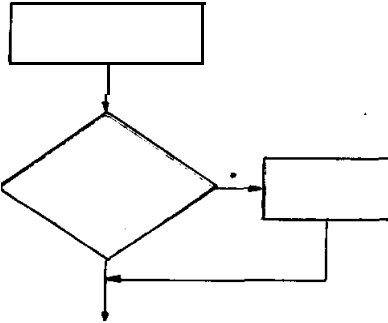

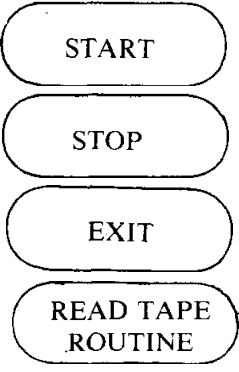
3. ผังโปรแกรมสำหรับปัญหาหนึ่งนั้น โปรแกรมเมอร์แต่ละคนไม่จำเป็นต้องเขียนเหมือนกัน ทั้งนี้ขึ้นอยู่กับประสบการณ์ของโปรแกรมเมอร์


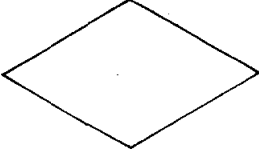
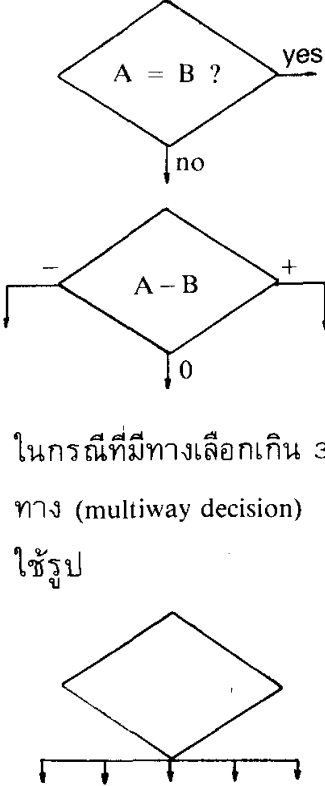

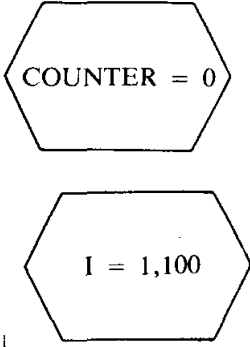
## สัญลักษณ์มาตรฐานที่ใช้ในการเขียนผังโปรแกรม

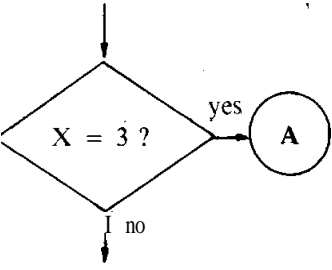

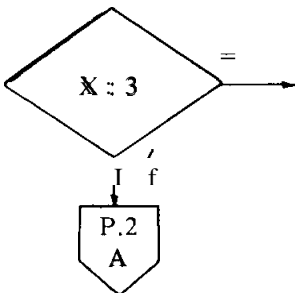
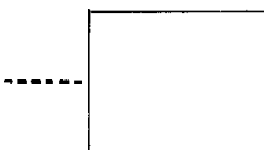
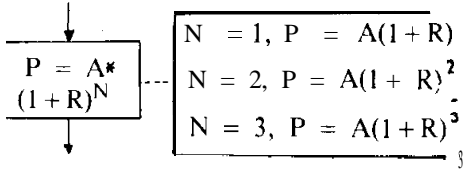
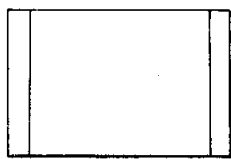
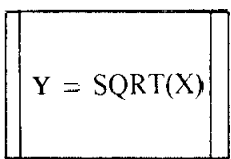
สัญลักษณ์ต่อไปนี้เป็นสัญลักษณ์ของสถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา (American National Standard Institute : ANSI)

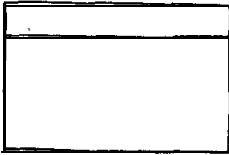
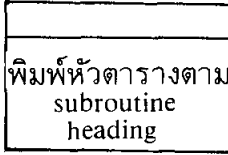
ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์	ตัวอย่าง
1. การนำข้อมูลเข้า/ออก (I/O symbol) ใช้สำหรับระบุตัวกลางในการนำข้อมูลเข้า/ออกโดยเขียนข้อความลงในรูปสี่เหลี่ยมด้านขนาน ใช้รูปนี้เป็นรูปทั่วไปในการแสดงขั้นของการนำข้อมูลเข้า/ออก	รูป <input type="checkbox"/> ด้านขนาน 	




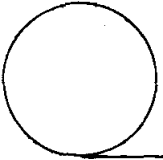
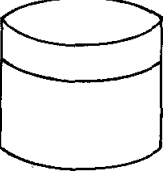

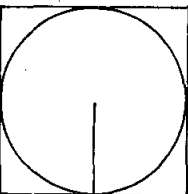
ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์	ตัวอย่าง
<p>2. การประมวลผลข้อมูล (Process symbol) ใช้แสดงการประมวลผลข้อมูล เช่น คำสั่งในการกำหนดค่าให้ตัวแปร การเคลื่อนย้ายข้อมูล เป็นต้น</p>	<p>รูป □ ผืนผ้า</p> 	
<p>3. การแสดงทิศทาง (flowline symbol) ใช้เพื่อเชื่อมต่อสัญลักษณ์ต่าง ๆ และแสดงลำดับการปฏิบัติงาน อาจใช้เส้นที่ไม่มีหัวลูกศรก็ได้ ขนาดยาวหรือสั้นตามต้องการ ทิศทางตามปกตินั้นคือจากบนลงล่าง และจากซ้ายไปขวาของกระดาษ</p>	<p>รูปลูกศร</p> 	
<p>4. จุดเริ่มต้นและจุดจบ (terminal symbol) ใช้กำหนดจุดเริ่มต้นและจุดจบของโปรแกรมและรูทีน (routine)</p>		

ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์	ตัวอย่าง
<p>5. การตัดสินใจ (decision symbol) ใช้แสดงสิ่งที่จะไปปฏิบัติโดยขึ้นกับเงื่อนไขในสัญลักษณ์นี้</p>	<p>รูป  ขั้วหลวมตัด</p> 	 <p>ในกรณีที่มีทางเลือกเกิน 3 ทาง (multiway decision) ใช้รูป</p>
<p>6. การเตรียม (Preparation symbol) ใช้ในการกำหนดชื่อของข้อมูลและรูปแบบที่จะถูกใช้ในข้อมูลเข้า ข้อมูลออก และที่ที่ใช้ทำงานและการกำหนดค่าเริ่มต้นให้แก่ที่ ๆ จะใช้ทำงาน (work area)</p>	<p>รูป 6 เหลี่ยม</p> 	

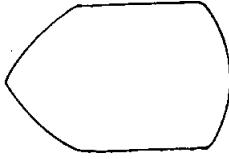
ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์	ตัวอย่าง
<p>7. การต่อเนื่อง (Connector symbol) ใช้เชื่อมต่อส่วนต่าง ๆ ของผังโปรแกรมเมื่อการใช้ลูกศรในการแสดงทิศทางจะก่อให้เกิดความสับสน นอกจากนั้นยังใช้เมื่อกระดาษยาวไม่พอ</p>	<p>รูปวงกลม</p> <p style="font-size: 2em; text-align: center;">0</p>	
<p>8. การต่อเนื่องคนละหน้า (Offpage connector symbol) ใช้เชื่อมต่อส่วนของผังโปรแกรมจากหน้าหนึ่งไปยังอีกหน้าหนึ่ง</p>	<p>รูป 5 เหลี่ยม</p> 	
<p>9. การเขียนคำอธิบายประกอบ (Comment annotation symbol) ใช้เขียนคำอธิบายเพิ่มเติมเพื่อให้ผังโปรแกรมชัดเจนขึ้น</p>		
<p>10. การอ้างถึงโปรแกรมย่อยภายนอก (External subroutine symbol) ใช้เพื่อเข้าถึง (perform หรือ call) โปรแกรมย่อยที่มีอยู่แล้วนอกโปรแกรม</p>		

ชื่อสัญลักษณ์ในการกระทำ/แทน	รูปสัญลักษณ์	ตัวอย่าง
11. การอ้างถึงโปรแกรมย่อยภายใน (Internal subroutine symbol) ใช้เพื่อเข้าถึงโปรแกรมย่อยที่มีอยู่แล้วในโปรแกรม		

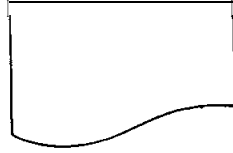
ในการนำข้อมูลเข้า/ออกจากระบบตัวกลางโดยใช้รูปสัญลักษณ์แทนตัวกลางต่าง ๆ ดังนี้

1. บัตรเจาะรู (Punched card) 
2. เทปแม่เหล็ก (Magnetic tape) 
3. จานแม่เหล็ก (Magnetic disk) 
4. ดรัมแม่เหล็ก (Magnetic drum) 
5. ดิสเกตต์ (Diskette) 

6. จอภาพ  
(Cathode ray tube : CRT)



7. กระดาษพิมพ์, เอกสาร  
(Document)



8. เทปกระดาษ  
(Punched tape)



### ข้อแนะนำในการเขียนผังโปรแกรม

เพื่อให้สามารถทำความเข้าใจผังโปรแกรมได้ง่าย และง่ายต่อการอ่าน ข้อแนะนำ  
ดังนี้ คือ

1. การเขียนผังโปรแกรมควรจะเขียนจากบนลงล่าง และจากซ้ายไปขวาของกระ-  
ดาษ ถ้าเป็นไปได้ให้ใช้ลูกศรทุกครั้งทิศทางไม่เป็นดังที่กล่าวข้างต้นหรืออาจใช้ลูกศร  
แสดงทั้งหมดก็ได้

2. ควรใช้ดินสอแทนปากกา ทั้งนี้เพื่อความสะดวกในการลบและแก้ไข ใช้ไม้บรรทัด  
สำหรับการเขียนสัญลักษณ์ซึ่งเรียกว่าเทมเพลต (template) เพื่อให้ได้รูปที่ประณีต เทมเพลตเป็น  
แผ่นพลาสติกหรือโลหะที่ฉลุเป็นรูปสัญลักษณ์ต่าง ๆ ท่านเพียงแต่ลากเส้นไปตามรอยฉลุก็จะทำ  
ให้ท่านวาดรูปสัญลักษณ์ได้ง่ายและสะดวก เทมเพลตตัวอย่างนั้นมีสัญลักษณ์ที่ใช้ได้ทั้งสำหรับ  
การเขียนผังภาพและผังโปรแกรม

3. หลีกเลี่ยงการตัดกันของเส้นแสดงทิศทางไหลของงาน ใช้สัญลักษณ์แสดงการต่อ  
เนื่องเมื่อจำเป็น

### ก่อนเขียนผังโปรแกรมควรทราบสิ่งต่อไปนี้ คือ

1. ชื่อของรายการข้อมูลหรือตัวแปร (Data name หรือ Variable) ซึ่งได้แก่ชื่อที่เราตั้งขึ้น  
ตามกฎเกณฑ์ (มีหลักเกณฑ์ต่างกันในแต่ละภาษาที่ใช้เขียนคำสั่ง) หมายถึงส่วนของหน่วย-  
ความจำหลักที่คอมพิวเตอร์จะเตรียมไว้ให้เพื่อใช้เก็บข้อมูลตามที่โปรแกรมเมอร์ต้องการ

โดยที่ที่ส่วนนั้นของหน่วยความจำจะมีชื่อตามที่ตั้งขึ้น สิ่งที่เกิดขึ้นในที่ดังกล่าวอาจเปลี่ยนแปลงโดยคำสั่งในโปรแกรมได้

ดังเช่นในภาษาฟอร์แทรน ชื่อที่ใช้ได้จะประกอบด้วย 1-6 ตัวอักษรซึ่งอาจเป็นตัวเลขหรือตัวอักษร แต่อักษรตัวแรกของชื่อต้องเป็นตัวอักษร

ตัวอย่างเช่น N, ANS, N25, NSF, SEX, AGE, STATUS เป็นต้น

2. เครื่องหมายเท่ากับ "=" ในทางคอมพิวเตอร์หมายความว่า ให้นำสิ่งที่อยู่ทางขวามือของเครื่องหมายเท่ากับไปเก็บไว้ในที่ที่มีชื่อตามที่ปรากฏทางซ้ายมือของเครื่องหมายเท่ากับ

ตัวอย่าง คำสั่ง  $N = N + 5$

ในที่นี้ N เป็นตัวแปรหรือชื่อ และ 5 เป็นค่าคงที่ (constant) คำสั่งนี้หมายความว่า เดิมในหน่วยความจำหลักที่มีชื่อว่า N มีค่าเท่าใด ให้เพิ่มค่าอีก 5 แล้วเก็บไว้แทนที่เดิม ดังนั้น ถ้าเดิม N มีค่า 20 เก็บอยู่ หลังจากทำตามคำสั่ง  $N = N + 5$  แล้ว ที่ชื่อ N จะเก็บค่าใหม่เป็น 25 (ค่า 20 จะหายไป)

ก่อนทำตามคำสั่ง  $N = N + 5$

N

20
----

หลังทำตามคำสั่ง  $N = N + 5$

N

25
----

3. เครื่องหมายในการคำนวณและการเปรียบเทียบ

เครื่องหมาย	ความหมาย
+	บวก
-	ลบ
*	คูณ
/	หาร
**	ยกกำลัง (ในภาษาฟอร์แทรน)
=	เท่ากับ
≠	ไม่เท่ากับ
>	มากกว่า
≥	มากกว่าหรือเท่ากับ
≤	น้อยกว่า
<	น้อยกว่าหรือเท่ากับ
:	เปรียบเทียบกับ





## ขั้นตอนในการวิเคราะห์โจทย์เพื่อเขียนผังโปรแกรม

1. หาสิ่งที่โจทย์ต้องการ จดออกมาเป็นรายการให้ครบ

2. ข้อมูลเข้า (Input)

– ตั้งชื่อรายการข้อมูล สำหรับข้อมูลทุกรายการที่จะอ่านจากตัวกลางเข้าไปเก็บในหน่วยความจำ

– ดูที่มาของข้อมูลเข้าว่าจะต้องอ่านจากตัวกลางอะไร

– ดูว่าค่าของตัวแปรใดบ้างที่เราสามารถสร้างขึ้นเองโดยคำสั่งภายในโปรแกรม (Data initialization)

3. วิธีการประมวลผลข้อมูล ดูข้อ 1. และรูปแบบของข้อมูลออก (Output format)

ที่ต้องการประกอบกัน

– สูตรที่ใช้ (formula)

– ตัวแปรที่เกิดขึ้นระหว่างการประมวลผล รวมทั้งตัวแปรที่ใช้กับข้อมูลออก

– เงื่อนไข (condition) ในการจัดจำแนกหรือในการเปลี่ยนทิศทางการทำงาน

– การเรียงลำดับ (sorting)

– การนับ (counting)

และอื่น ๆ

4. ข้อมูลออก (Output)

ข้อมูลที่ต้องแสดงให้ดูจากรูปแบบของข้อมูลออกที่ได้กำหนดไว้

วิธีการกำหนดเงื่อนไข (condition)

เงื่อนไขที่เราใช้ในการจัดจำแนกหรือในการเปลี่ยนแปลงทิศทางการทำงานในผังโปรแกรมนั้นจะถูกทดสอบ ผลการทดสอบเงื่อนไขจะนำไปสู่การปฏิบัติงานที่ต่างกัน ทั้งนี้ขึ้นอยู่กับปัญหา ในหัวข้อนี้จะศึกษาวิธีการกำหนดเงื่อนไขแบบง่าย ๆ และวิธีการที่จะรวมเงื่อนไขง่าย ๆ เพื่อให้ได้รูปแบบของการควบคุมที่มีประสิทธิภาพ

**การเปรียบเทียบอย่างง่าย** เป็นการเปรียบเทียบระหว่างข้อมูล 2 ตัว (ซึ่งอาจจะเป็นตัวแปรทั้งคู่หรือเป็นค่าคงที่ทั้งคู่) เช่น อาจถามว่าค่าของตัวแปร **WEIGHT** เท่ากับ 50 กก.หรือไม่ ( $is\ WEIGHT = 50?$ ) หรือถามว่า **N** มีค่ามากกว่าหรือเท่ากับ 20 หรือไม่ ( $is\ N \geq 20?$ ) การทดสอบเหล่านี้เราถือว่าเป็นการเปรียบเทียบอย่างง่าย และผลการเปรียบเทียบคือจริง (true) หรือ

ไม่จริง (false) และเป็นสิ่งที่จะถูกนำมาประกอบกันเป็นการทดสอบที่ซับซ้อนขึ้น ในการเปรียบเทียบเรามีตัวดำเนินการในการเปรียบเทียบ (relational operator) 6 ตัว ซึ่งการเขียนเงื่อนไขในการเปรียบเทียบหรือที่เรียกนิพจน์เปรียบเทียบ (relational expression) ในภาษาต่าง ๆ จะใช้สัญลักษณ์แตกต่างกันไปในการเขียนคำสั่งในโปรแกรม ตัวอย่างเช่น

สัญลักษณ์ทางคณิตศาสตร์	ความหมาย	ในภาษาฟอร์แทรน	ในภาษาปาสกาล	ในภาษาพีแอลเอ็น *
=	เท่ากับ	.EQ.	=	=
≠	ไม่เท่ากับ	.NE.	<>	≠
>	มากกว่า	.GT.	>	>
≥	มากกว่าหรือเท่ากับ	.GE.	>=	>=
<	น้อยกว่า	.LT.	<	<
≤	น้อยกว่าหรือเท่ากับ	.LE.	<=	<=

\*ในภาษาพีแอลเอ็นยังมีอีก 2 ตัว คือ  $\neg >$  (ไม่มากกว่า) และ  $\neg <$  (ไม่น้อยกว่า)

รูปแบบการเปรียบเทียบอย่างง่าย (เงื่อนไขอย่างง่าย)

ข้อมูล 1 ตัวดำเนินการในการเปรียบเทียบ ข้อมูล 2

การรวมกันของเงื่อนไขอย่างง่ายโดยใช้ตัวดำเนินการตรรกะ (logical operator หรือ boolean operator)

ตัวดำเนินการตรรกะ คือ และ (AND) หรือ (OR) นิเสธ (NOT)

ในภาษาฟอร์แทรนใช้ .AND. .OR. และ .NOT. ตามลำดับ

เมื่อเราต้องการจะทำสิ่งหนึ่งโดยที่การกระทำนั้นจะขึ้นอยู่กับเงื่อนไข 2 เงื่อนไขที่เป็นอิสระต่อกันและต่างก็มีค่าเป็นจริง เราจะรวมเงื่อนไขทั้ง 2 ให้เป็นเงื่อนไขเดียว โดยใช้ตัวดำเนินการตรรกะ “และ (AND)” แต่ถ้าการกระทำนั้นขึ้นอยู่กับกรณีที่เงื่อนไขใดเงื่อนไขหนึ่งใน 2 เงื่อนไขนั้นเป็นจริงหรือทั้ง 2 เงื่อนไขเป็นจริงแล้ว เราจะรวมเงื่อนไขทั้ง 2 ให้เป็นเงื่อนไขเดียว

โดยใช้ตัวดำเนินการตรรกะ “หรือ (OR)” ส่วนตัวดำเนินการตรรกะ “นิเสธ (NOT)” นั้นใช้เมื่อต้องการผลตรงข้ามกับเงื่อนไขที่มีอยู่

ตารางแสดงผลการใช้ตัวดำเนินการตรรกะกับเงื่อนไข 2 เงื่อนไข คือ เงื่อนไข A และ เงื่อนไข B

A	B	A AND B	A OR B	NOT A	NOT B
true	true	true	true	false	false
true	false	false	true	false	true
false	true	false	true	true	false
false	false	false	false	true	true

**ตัวอย่างที่ 1** ในการตรวจสอบว่าคน ๆ หนึ่งเป็นผู้หญิงโสดอายุ 18-23 ปีหรือไม่

ถ้ากำหนดให้ตัวแปร SEX เก็บข้อมูลเรื่องเพศ (SEX = 1 คือชาย และ SEX = 2 คือหญิง)

ตัวแปร STATUS เก็บข้อมูลเรื่องสถานภาพสมรส (ในที่นี้ถ้า STATUS = 1 คือเป็นโสด)

และตัวแปร AGE เก็บข้อมูลเรื่องอายุ

เราต้องการทดสอบว่า

Is SEX = 2?                      เงื่อนไขคือ SEX = 2

Is STATUS = 1?                    เงื่อนไขคือ STATUS = 1

Is  $18 \leq AGE \leq 23$ ?            เงื่อนไขคือ  $18 \leq AGE \leq 23$

เงื่อนไข  $18 \leq AGE \leq 23$  นั้นเป็นเงื่อนไขที่เกิดจากการรวมเงื่อนไข 2 เงื่อนไขเข้าด้วยกัน

โดยใช้ตัวดำเนินการตรรกะ AND นั่นคือ

$18 \leq AGE \leq 23$  อาจเขียนเป็น  $AGE \geq 18$  AND  $AGE \leq 23$

เงื่อนไขทั้ง 3 เงื่อนไข สามารถรวมกันเพื่อทดสอบผู้หญิงโสดอายุ 18-23 ปี จะเป็น

ดังนี้

SEX = 2 AND STATUS = 1 AND ( $AGE \geq 18$  AND  $AGE \leq 23$ )

ถ้าเงื่อนไขย่อยเป็นจริง เงื่อนไขรวมนี้จะเป็นจริงด้วย

ตัวอย่างที่ 2 ในการตรวจสอบเพื่อคัดเลือกเอาคนผู้ชายหรือคนที่มีอายุเกิน 30 ปี (ซึ่งอาจเป็นชายหรือหญิงก็ได้) เราจะใช้เงื่อนไขย่อย 2 เงื่อนไข คือ SEX = 1 และ AGE > 30 แล้วรวมทั้ง 2 เงื่อนไขโดยใช้ตัวดำเนินการตรรกะ OR นั่นคือเงื่อนไขรวม คือ

$$\text{SEX} = 1 \text{ OR AGE} > 30$$

จะเห็นได้ว่าเงื่อนไขรวมจะเป็นจริงเมื่อเงื่อนไขย่อยเป็นจริงทั้งคู่ หรือเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริงก็ได้ ดังนั้นผลการคัดเลือกจะได้กลุ่มคนซึ่งอาจประกอบด้วย

1. ผู้ชายอายุเท่าใดก็ได้ (รวมพวกที่อายุเกิน 30 ปีด้วย)
2. ผู้ชายหรือผู้หญิงที่มีอายุเกิน 30 ปี

เงื่อนไขรวมในตัวอย่างทั้ง 2 นั้นเราอาจเรียกว่านิพจน์ตรรกะ (logical expression)

ซึ่งจะมีค่าเป็นจริง (true) หรือเป็นเท็จ (false) ได้เท่านั้น

รูปแบบของการเขียนนิพจน์ตรรกะ

$$\text{(NOT) เงื่อนไข 1} \left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\} \text{(NOT) เงื่อนไข 2} \left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\} \text{(NOT) เงื่อนไข 3}$$

โดยที่ { } หมายถึงเลือกตัวใดตัวหนึ่ง (choose one)  
 ( ) หมายถึงจะใช่หรือไม่ใช่ก็ได้ (option)

โปรดสังเกตว่า นิพจน์เปรียบเทียบนั้นเป็น subset ของนิพจน์ตรรกะ นั่นคือนิพจน์เปรียบเทียบทุกนิพจน์ถือเป็นนิพจน์ตรรกะ แต่นิพจน์ตรรกะไม่ใช่निพจน์เปรียบเทียบเสมอไป เช่น

ตัวแปรตรรกะ L1 และ L2 (L1 มีค่าเป็นจริงหรือเท็จ, L2 มีค่าเป็นจริงหรือเท็จ)

นิพจน์ตรรกะ เช่น L1 AND L2

L1 OR L2

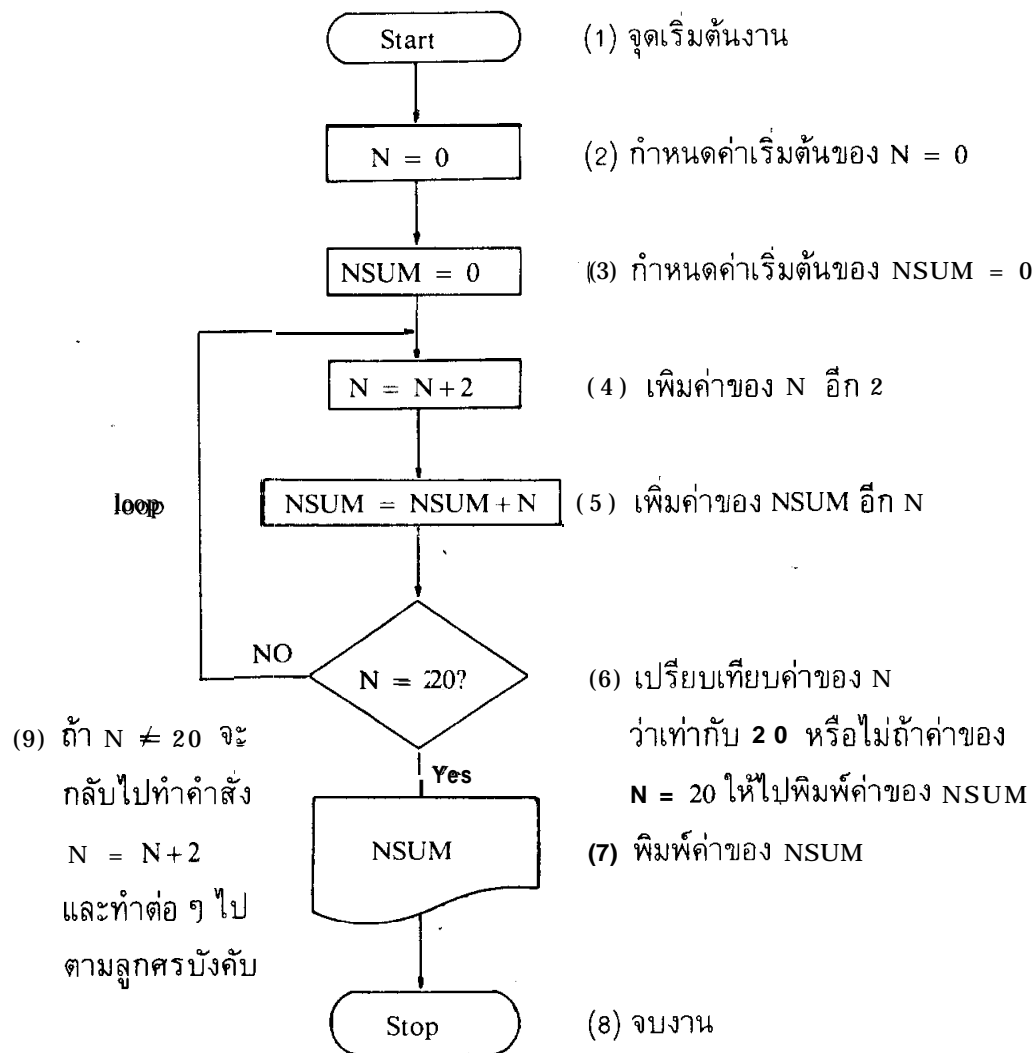
NOT L1

ซึ่งไม่มีนิพจน์ใดเป็นนิพจน์เปรียบเทียบเลย

รายละเอียดเพิ่มเติมจะได้เรียนในภาษาแต่ละภาษา

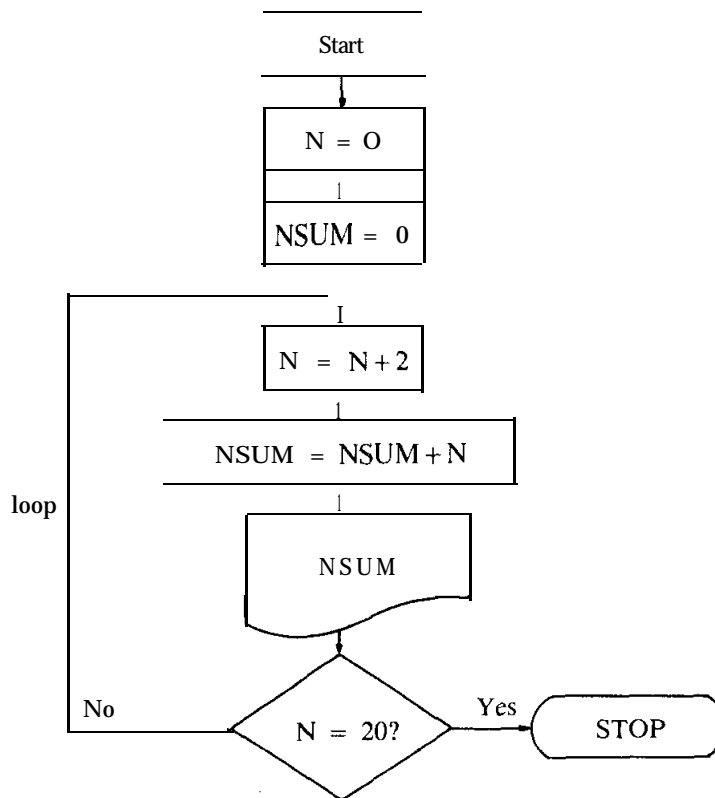
ตัวอย่าง Program Flowchart

ตัวอย่างที่ 1



Program flowchart นี้จะหาค่าของ  $NSUM = 2 + 4 + 6 + \dots + 20 = 110$   
 และจะพิมพ์ค่าของ  $NSUM$  1 ค่า คือ 110

**หมายเหตุ** คำสั่งที่อยู่ใน loop จะถูกทำซ้ำโดยไม่ต้องสั่งซ้ำหลาย ๆ ครั้งในการสร้าง loop ทุกครั้งต้องอาศัย Decision Block ในการควบคุมจำนวนครั้งของการทำซ้ำ และควบคุมการออกจาก loop ให้พิจารณาจาก Program flowchart ต่อไปนี้เพื่อเปรียบเทียบกับตัวอย่างที่ 1

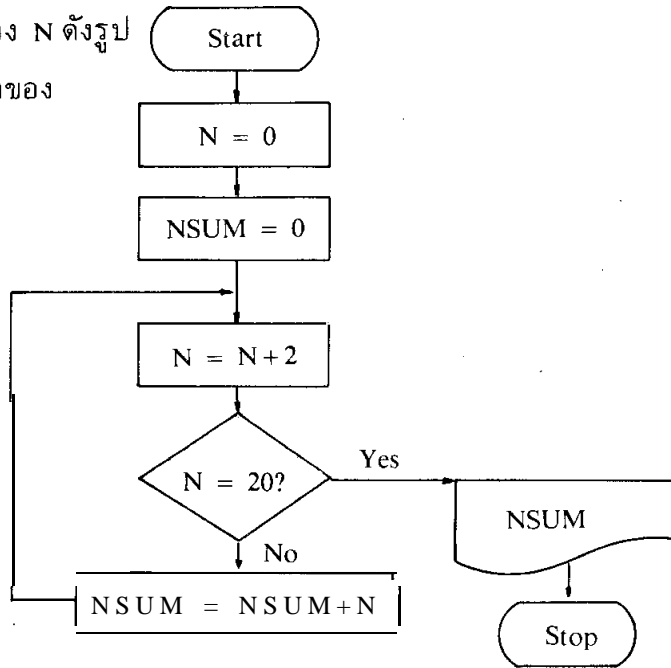


จะเห็นว่า block ที่ให้พิมพ์ค่าของ NSUM เข้าไปอยู่ใน loop (ในตัวอย่างที่ 1 อยู่นอก loop) ซึ่งทำให้ผลของคำสั่งเปลี่ยนแปลงไป

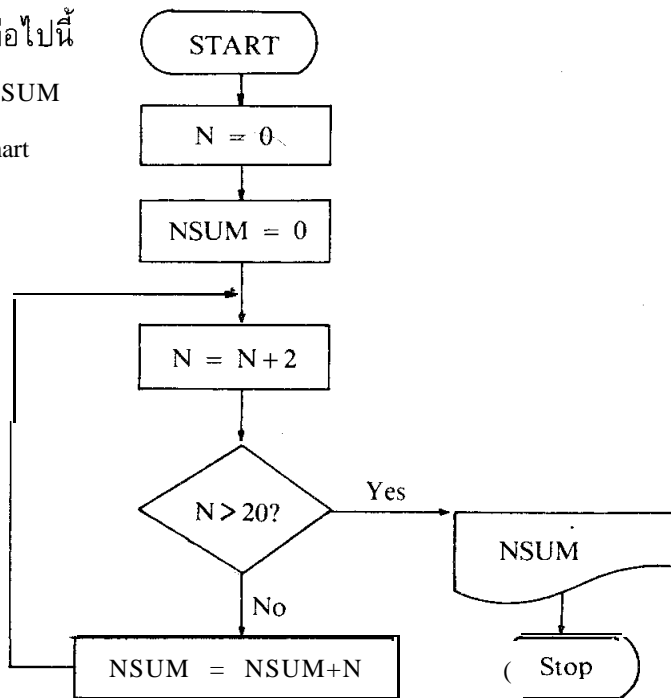
Program flowchart นี้จะหาค่าของ NSUM และพิมพ์ค่าของ NSUM 10 ค่า ดังนี้

เมื่อ	N	มีค่าเป็น 2	NSUM = 2	พิมพ์ค่า NSUM = 2
	N	มีค่าเป็น 4	NSUM = 6	พิมพ์ค่า NSUM = 6
	N	มีค่าเป็น 6	NSUM = 12	พิมพ์ค่า NSUM = 12
	N	มีค่าเป็น 8	NSUM = 20	พิมพ์ค่า NSUM = 20
	N	มีค่าเป็น 10	NSUM = 30	พิมพ์ค่า NSUM = 30
	N	มีค่าเป็น 12	NSUM = 42	พิมพ์ค่า NSUM = 42
	N	มีค่าเป็น 14	NSUM = 56	พิมพ์ค่า NSUM = 56
	N	มีค่าเป็น 16	NSUM = 72	พิมพ์ค่า NSUM = 72
	N	มีค่าเป็น 18	NSUM = 90	พิมพ์ค่า NSUM = 90
และ	N	มีค่าเป็น 20	NSUM = 110	พิมพ์ค่า NSUM = 110

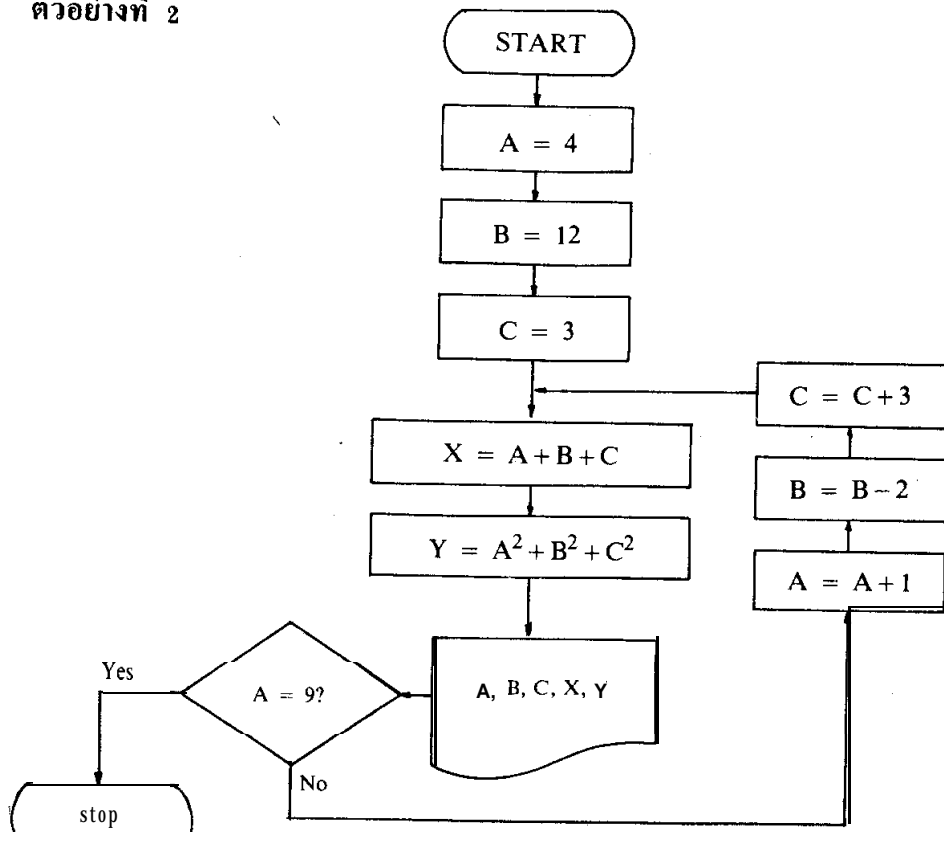
เมื่อย้ายที่ block ที่ทดสอบค่าของ N ดังรูป  
 Program flowchart นี้จะพิมพ์ค่าของ  
 NSUM = 90 เท่านั้น  
 (ไม่ได้บวก 20 ใน NSUM)



พิจารณา Program flowchart ต่อไปนี้  
 จะพบว่าผลของการพิมพ์ค่า NSUM  
 เท่ากับผลจาก Program flowchart  
 ในตัวอย่างที่ 1



ตัวอย่างที่ 2



Program flowchart นี้จะหาค่าของ X และ Y จากค่าของ A, B, และ C ทั้งหมด 6 ชุด แต่ละชุดมีค่าของ A, B และ C ดังนี้

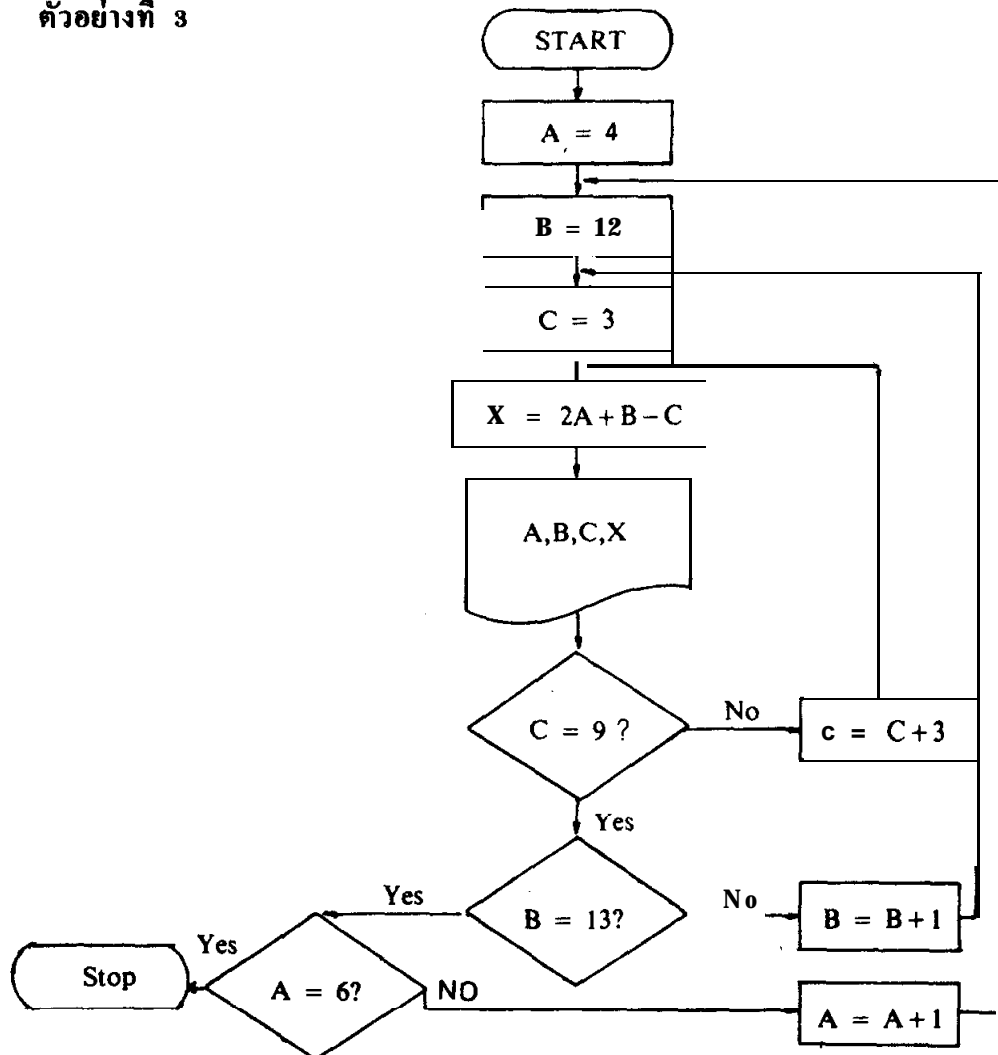
ชุด	A	B	C
1	4	12	3
2	5	10	6
3	6	8	9
4	7	6	12
5	8	4	15
6	9	2	18



ผลการพิมพ์จากผังโปรแกรมในตัวอย่างที่ 2

A	B	C	X	Y
4	12	3	19	169
5	10	6	21	161
6	8	9	23	181
7	6	12	25	229
8	4	15	27	305
9	2	18	<b>29</b>	409

ตัวอย่างที่ 3



Program flowchart นี้แสดงการหาค่าของ A, B และ C ทั้งหมด 18 ชุด แต่ละชุดมี  
ค่า ของ A, B และ C ดังนี้

ชุด	A	B	C	ชุด	A	B	C	ชุด	A	B	C
1	4	12	3	7	5	12	3	13	6	12	3
2	4	12	6	8	5	12	6	14	6	12	6
3	4	12	9	9	5	12	9	15	6	12	9
4	4	13	3	10	5	13	3	16	6	13	3
5	4	13	6	11	5	13	6	17	6	13	6
6	4	13	9	12	5	13	9	18	6	13	9

#### ตัวอย่างที่ 4

ในงานประมวลผลข้อมูลงานหนึ่ง มีบัตร Input Card อยู่ 500 บัตร

กำหนด Input Card format

Field ที่	Column	รายการข้อมูล	ตั้ง Variable
1	1-3	เลขประจำบัตร (เลขที่แบบสอบถาม)	N 0
2	4	เพศ	SEX
3	5-6	อายุ	AGE
4	7	สถานภาพสมรส	STATUS

กำหนดรหัสให้ข้อมูลแสดงคุณภาพ (Qualitative Data)

เพศ : ชาย = 1

หญิง = 2

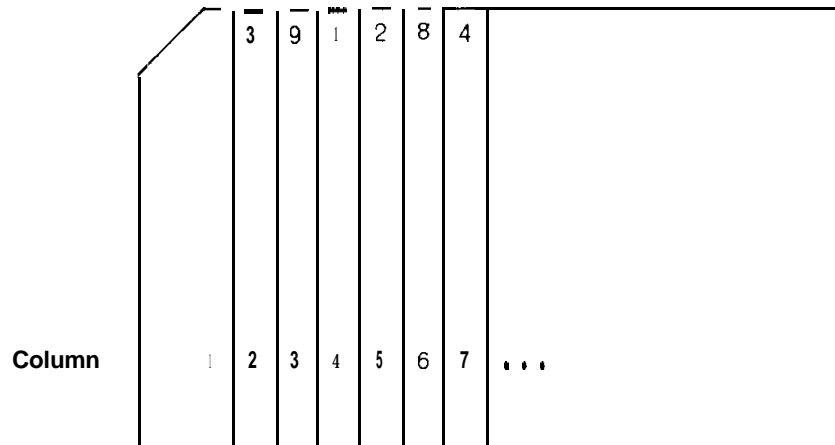
สถานภาพสมรส : โสด = 1

แต่งงานแล้ว = 2      หย่า = 3

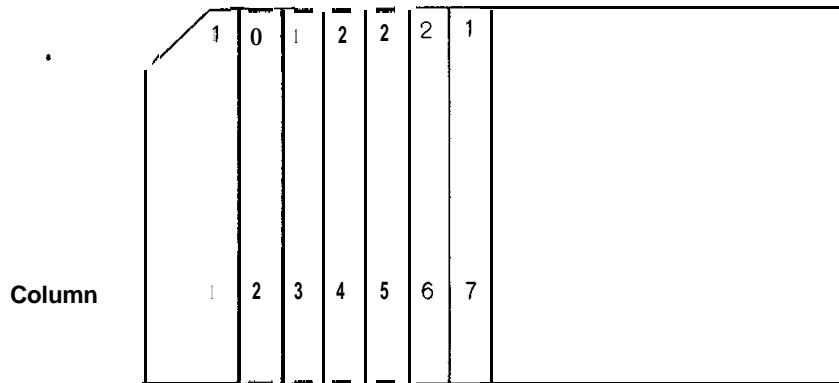
ม่าย (คู่สมรสตาย) = 4

## ตัวอย่าง Input card

1. ข้อมูลจากแบบสอบถามที่ 39 เพศชาย อายุ 28 เป็นม่าย



2. ข้อมูลจากแบบสอบถามที่ 101 เพศหญิง อายุ 22 เป็นโสด



### 4.1 จงเขียน Program Flowchart เพื่อ

1. นับจำนวนผู้หญิงทั้งหมด
2. นับจำนวนผู้หญิงโสดที่มีอายุระหว่าง 18 ถึง 23 ปี
3. พิมพ์รายการของผู้หญิงที่มีลักษณะในข้อ 2

กำหนด Output format

คือ head {

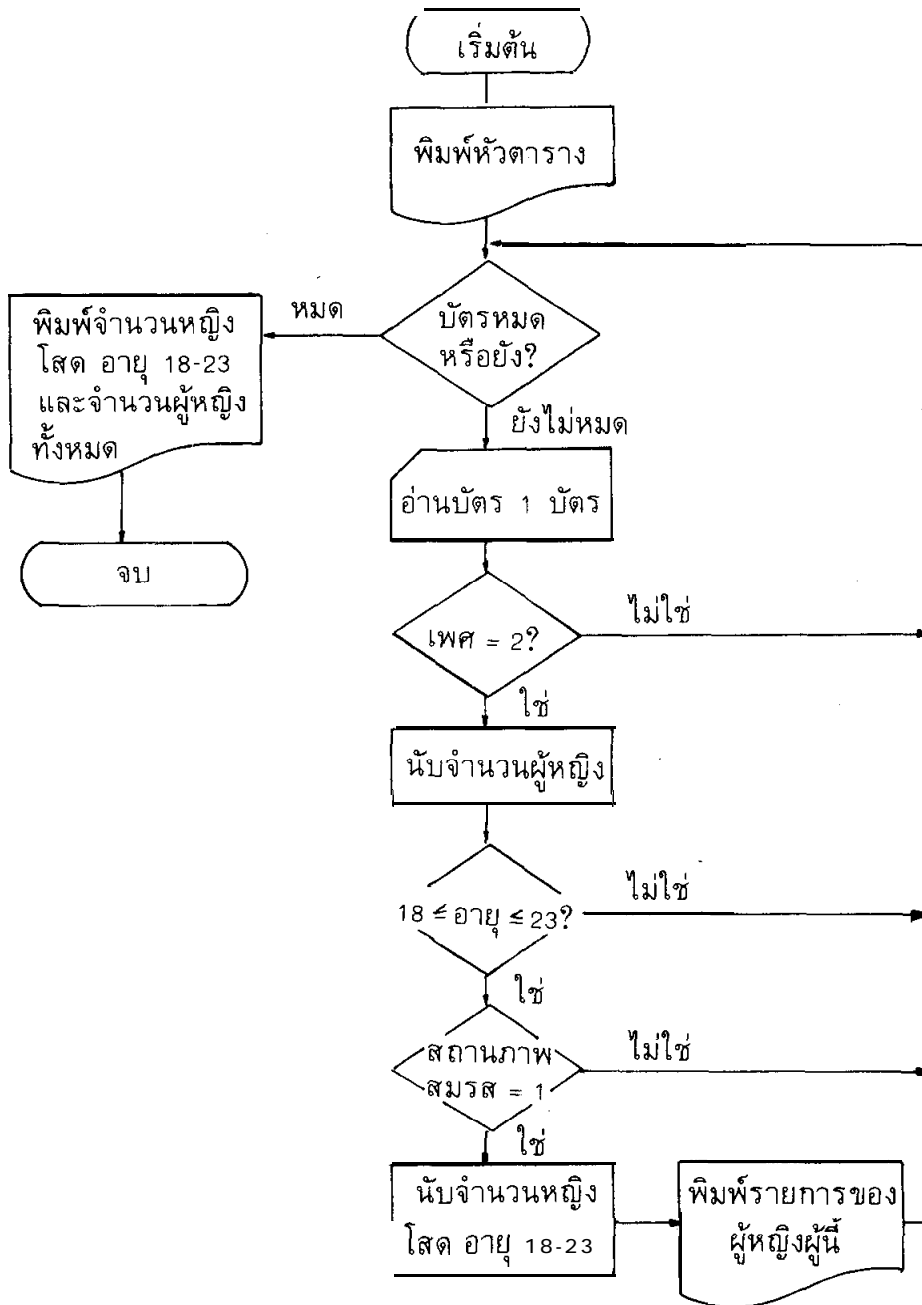
REPORT CS 200				
NUMBER	SEX	AGE	MARITAL	STATUS
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

คำอธิบายค่าที่พิมพ์ →

SINGLE FEMALE (AGE 18-23)	= XXX,	PERSONS
FEMALE	= XXX	PERSONS

จำนวนที่คอมพิวเตอร์จะนับให้  
ค่าของ Variable  
NSF และ NF ตามลำดับ

เราอาจเขียน Flowchart เพื่อแสดงลำดับของการทำงานดังนี้

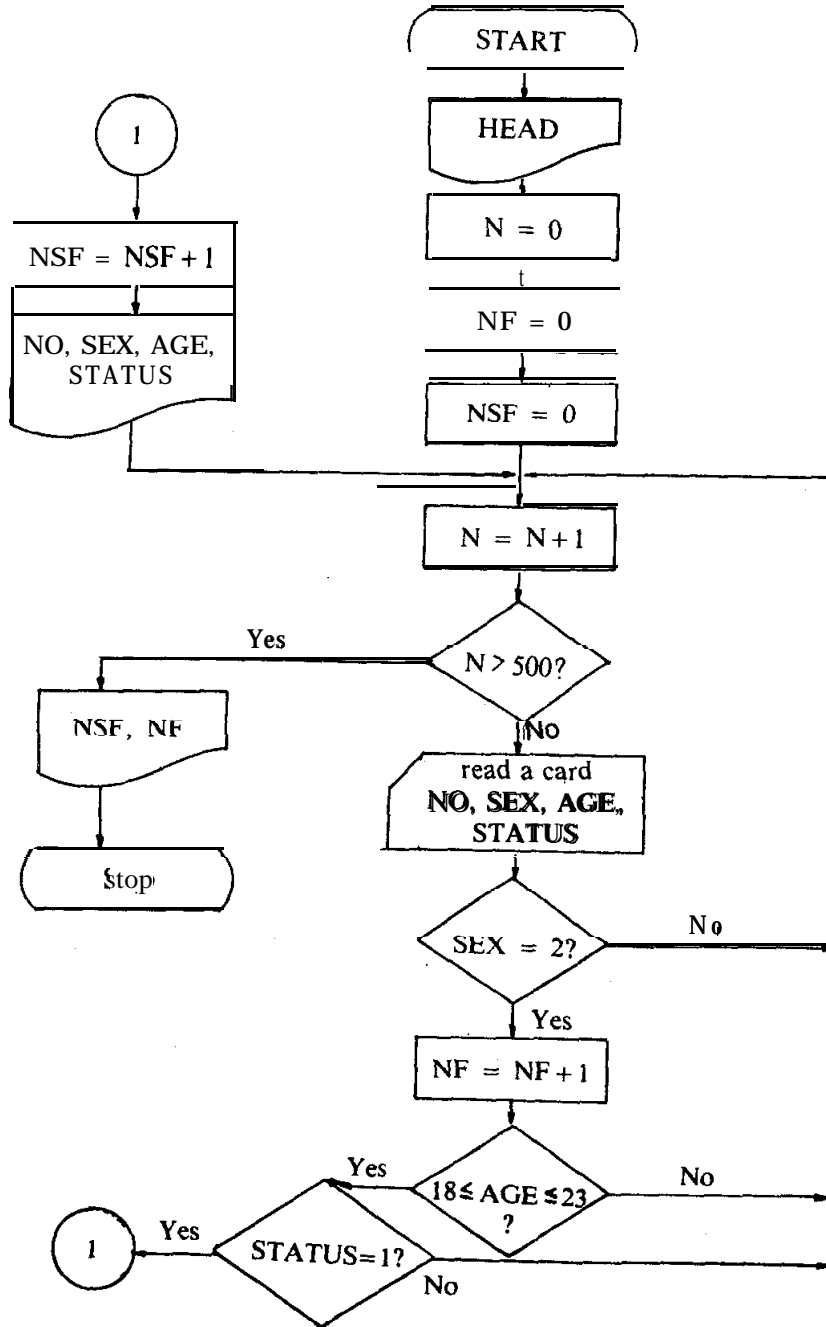


ทำข้อ 4.1 โดยใช้ Variable และแสดงวิธีสังคอมพิวเตอร် ให้นำสิ่งที่ต้องการ นอกจาก Variable 4 ตัวที่ตั้งไว้แล้วในโจทย์ เราจะตั้ง Variable เพิ่มอีกดังนี้ คือ

N ใช้นับบัตรที่อ่าน

NF ใช้นับจำนวนผู้หญิง

NSF ใช้นับจำนวนผู้หญิงโสดที่มีอายุระหว่าง 18-23 ปี



**ข้อสังเกต** เราให้ค่าเริ่มต้นของ Variable ที่ใช้นับเป็น 0 เช่น  $NF = 0$  เมื่อคอมพิวเตอร์ทำตามคำสั่ง  $NF = NF + 1$  แล้ว  $NF$  จะมีค่าเท่ากับ 1 และทุกครั้งที่คำสั่ง  $NF = NF + 1$  ถูกทำค่าของ  $NF$  จะเพิ่มขึ้นทีละ 1 (ซึ่งคือการนับนั่นเอง) คำสั่ง  $NF = 0$  จะถูกทำเพียง 1 ครั้งคือก่อนเริ่มนับเท่านั้น

#### 4.2 จงเขียน Program flowchart เพื่อ

1. นับจำนวนผู้หญิงทั้งหมด
2. นับจำนวนผู้หญิงโสดที่มีอายุระหว่าง 18-23 ปี
3. คำนวณหว่าจำนวนในข้อ 2 เป็นกี่เปอร์เซ็นต์ของจำนวนในข้อ 1

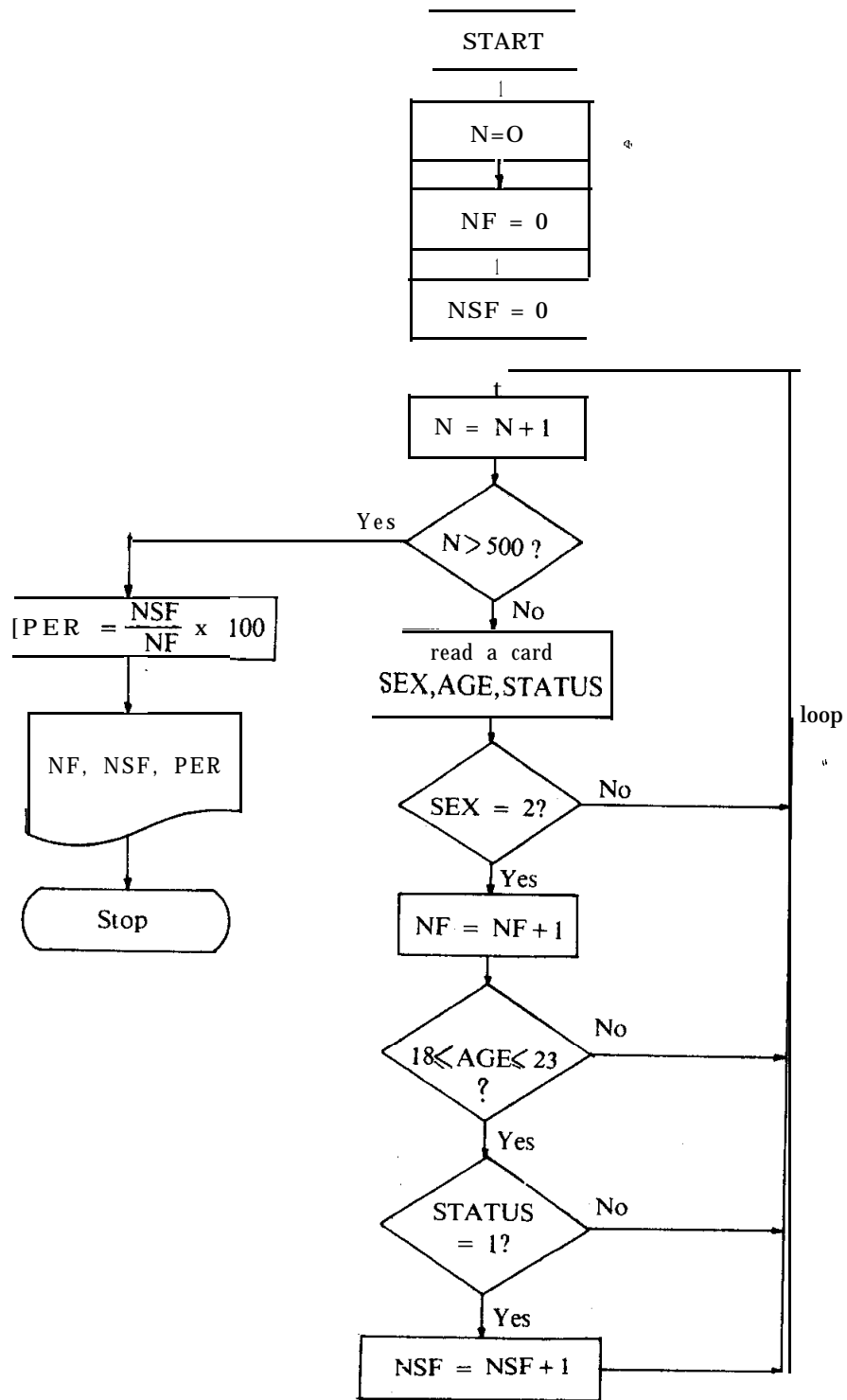
**กำหนด Output format**

FEMALE = XXX PERSONS

SINGLE FEMALE (AGE 18-23) = XXX PERSONS, THAT IS XX.XX %

**ข้อสังเกต** ในข้อ 4.2 นี้ เลขประจำบัตรซึ่งถือเป็น identification data เราไม่จำเป็นต้องอ่านเลขประจำบัตรเข้าไปเก็บในหน่วยความจำ เพราะในการประมวลผลข้อมูลตามความต้องการ เราไม่ได้ใช้ทำอะไร (เลขประจำบัตร จะลงใน Input Card เพื่อประโยชน์ในตอนตรวจทานข้อมูล (Verify) ในบัตร เพื่อให้ทราบว่าข้อมูลในบัตรมาจากแบบสอบถามใด) แต่ในข้อ 4.1 เราต้องอ่านเข้าไปเก็บในหน่วยความจำด้วย เพราะต้องเอามาใช้ตอนพิมพ์ผล

การหาเปอร์เซ็นต์และพิมพ์จำนวนที่นับได้จะทำเมื่อได้จำแนกบัตรครบทั้ง 500 บัตรแล้ว ดังนั้น block ของคำสั่งที่ให้หาเปอร์เซ็นต์ และ block ของการพิมพ์จำนวนที่นับได้ต้องอยู่นอก loop

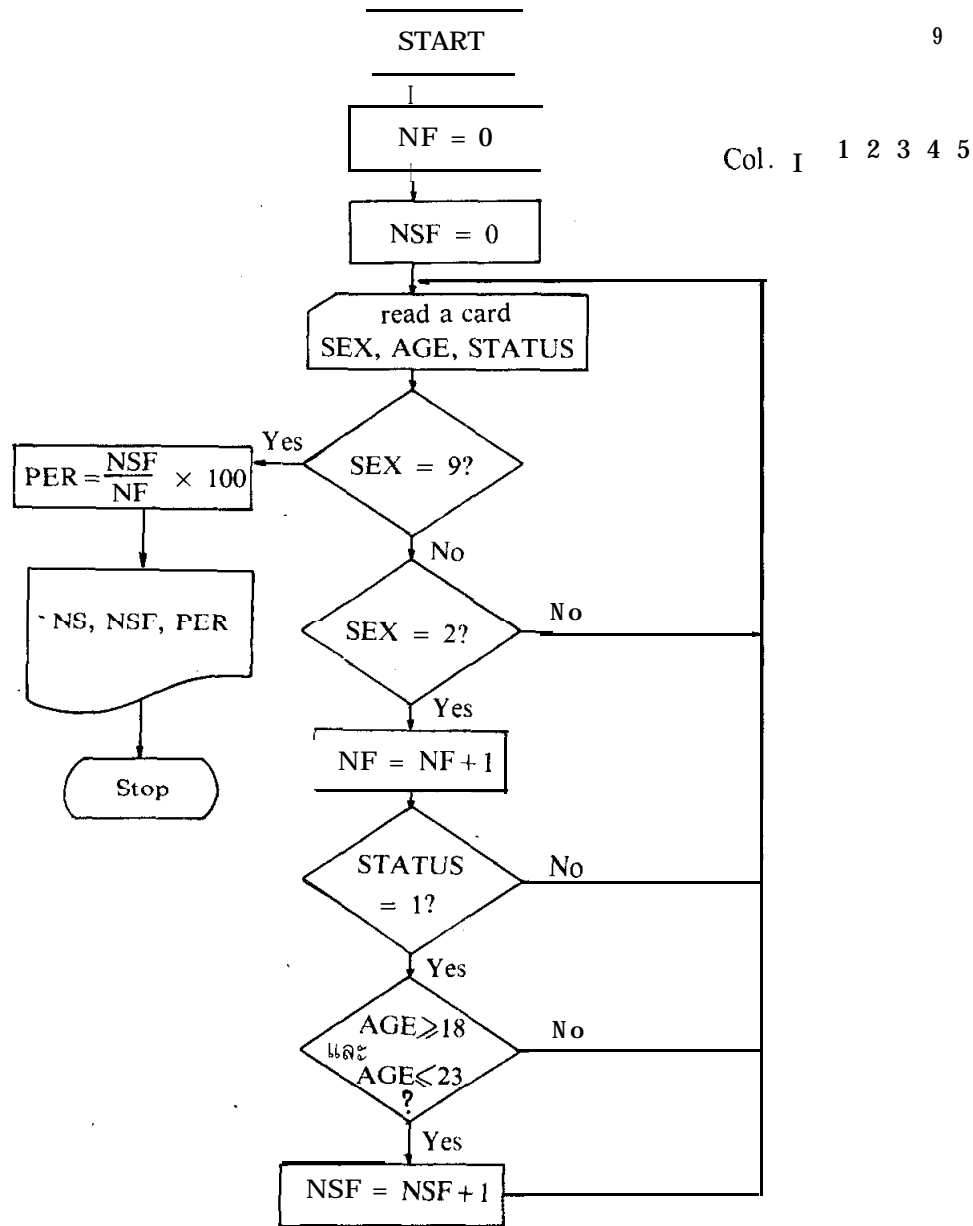




### 4.3 ใช้ Output format ข้อ 4.2

ควบคุมการอ่านบัตรข้อมูล โดยเจาะบัตรใบสุดท้าย (last card) เพิ่มขึ้นอีก 1 บัตร ใส่ไว้เป็นบัตรใบสุดท้ายของ data deck (ในข้อ 4.1 และ 4.2 ไม่ใช่บัตรใบสุดท้าย ควบคุมการอ่านบัตรโดยวิธีนับบัตร) ดังนั้น ข้อ 4.3 ไม่ใช่ Variable N

บัตรใบสุดท้าย เจาะเลข 9 ใน field ที่ 2



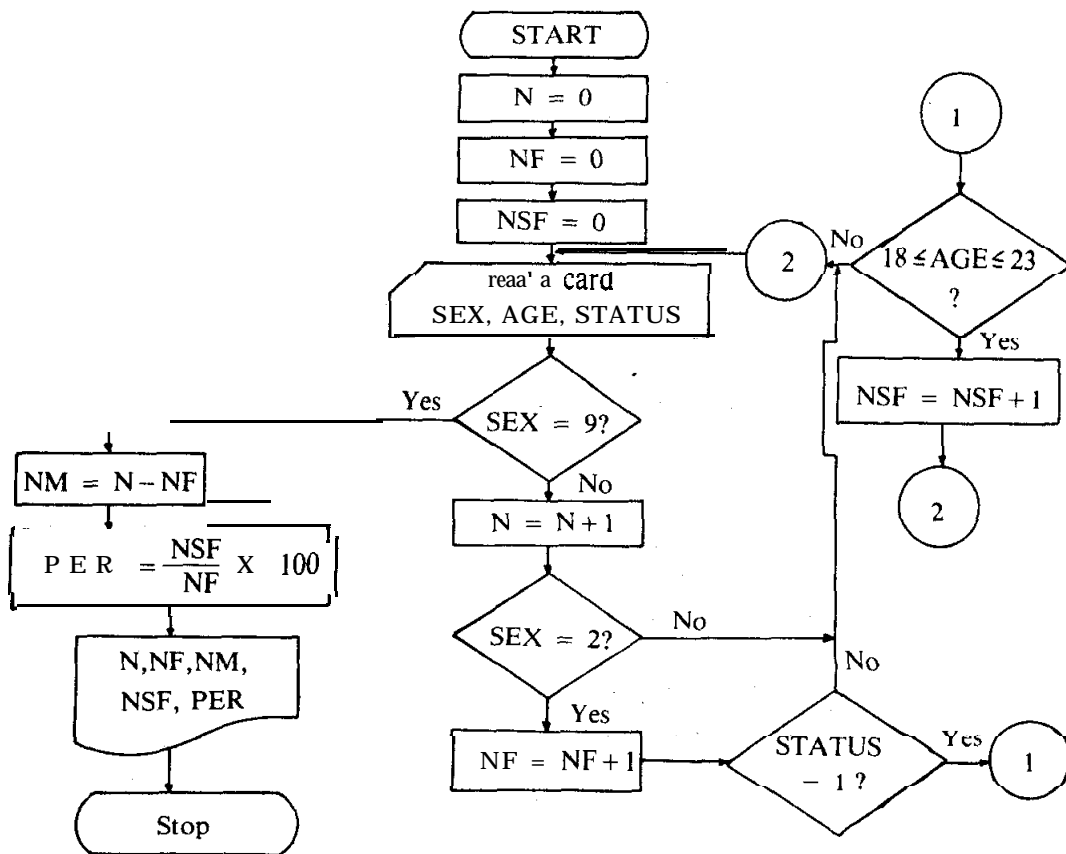
#### 4.4 ในกรณีที่เรามิทราบจำนวนบัตรข้อมูล (data card) ทั้งหมดว่ามีเท่าใด?

เราสามารถควบคุมการอ่านบัตรโดยใช้บัตรใบสุดท้าย (last card) ดังตัวอย่างในข้อ 4.3 เราอาจให้คอมพิวเตอร์ นับบัตรที่เป็นบัตรข้อมูลว่ามีทั้งหมดกี่บัตรได้ (ไม่นับบัตรที่เป็น last card)

ในกรณีที่ใช้ Variable N นับจำนวน data card วิธีการนี้อาจทำเพื่อตรวจสอบว่าบัตร Input ครบตามจำนวนที่ต้องมีอยู่หรือไม่

กำหนด **Output format** (เพิ่มจากข้อ 4.2 คือ นับจำนวนผู้ชายและนับจำนวน data card ทั้งหมด)

TOTAL	=	XXX	PERSONS
FEMALE	=	x x x	PERSONS
MALE	=	x x x	PERSONS
SINGLE FEMALE (AGE 18-23)	=	x x x	PERSONS
THAT IS	=	x . x x	%



### ตัวอย่างที่ 5 การติดดอกเบี้ยเงินฝาก (ดอกเบี้ยทบต้น)

ในการฝากเงินกับธนาคาร จะพบว่าถ้าเราฝากเงินต้น 100 บาท และธนาคารคิดอัตราดอกเบี้ย 8% พอดีครบปีที่ 1 เราจะได้เงินฝากทั้งสิ้น (รวมดอกเบี้ย) เท่ากับ 108 บาท

พอสิ้นสุดปีที่ 2 เงินฝากในธนาคารของเราก็จะกลายเป็น  $108 + 8.64 = 116.64$  บาท โดยที่เงิน 8.64 บาท ก็คือดอกเบี้ยที่คิดจากเงินฝาก 108 นั้นเอง

ดังนั้น ดอกเบี้ยที่คิดในแต่ละปีจะมีจำนวนเพิ่มขึ้นตามจำนวนเงินต้นที่เปลี่ยนแปลงในแต่ละปี

โดยทั่ว ๆ ไปถ้าเป็นดอกเบี้ยทบต้นเราจะคิดจำนวนเงินที่จะได้รับเมื่อสิ้นสุดปีที่  $N$  ได้โดยคิดจากสูตรได้ดังนี้

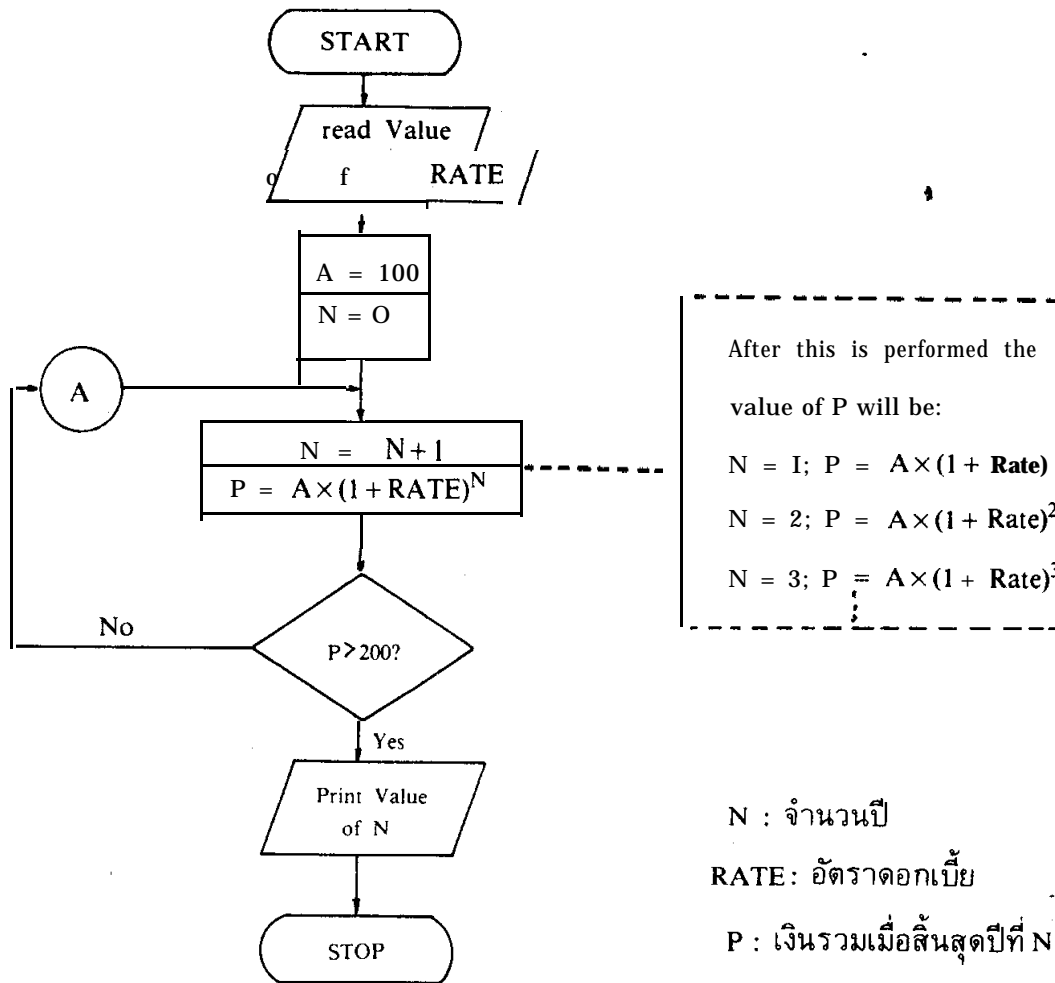
$$\text{จำนวนเงินที่รับเมื่อสิ้นสุดปีที่ } N = \text{เงินต้น} \times (1 + \text{อัตราดอกเบี้ย})^N$$

ดังนั้นจำนวนเงินเมื่อสิ้นสุดปีที่ 3 ในเมื่อเงินต้นเป็น 100 บาท อัตราดอกเบี้ย 8% จะคิดได้ดังนี้

$$\begin{aligned} \text{จำนวนเงินที่ได้รับเมื่อสิ้นสุดปีที่ 3} &= 100 \times (1 + 8/100)^3 \\ &= 100 \times (108/100)^3 \\ &= 125.97 \text{ บาท} \end{aligned}$$

กรณีที่ต้องการตรวจสอบรายละเอียดต่อไปว่า ถ้าเราต้องการให้เงินฝาก 100 บาท ดอกเบี้ย 8% ต่อปี กลายเป็นเงิน 200 บาท (จำนวนเงินสองเท่าของเงินฝากครั้งแรก) เราจะต้องใช้เวลาฝากเงินนานกี่ปี

จากปัญหานี้ถ้าเราจะใช้เครื่องคอมพิวเตอร์ช่วยในการวิเคราะห์ เราก็อาจสร้าง Flowchart เพื่อแก้ปัญหานี้ คือ



ใน Flowchart นี้ จะแสดงการทำงานของเครื่องดังนี้ เริ่มทำงาน แล้วอ่านค่าของอัตราดอกเบี้ย

RATE = 8% โดยที่เงินต้นคือ P = 100 จำนวนปี N = 0

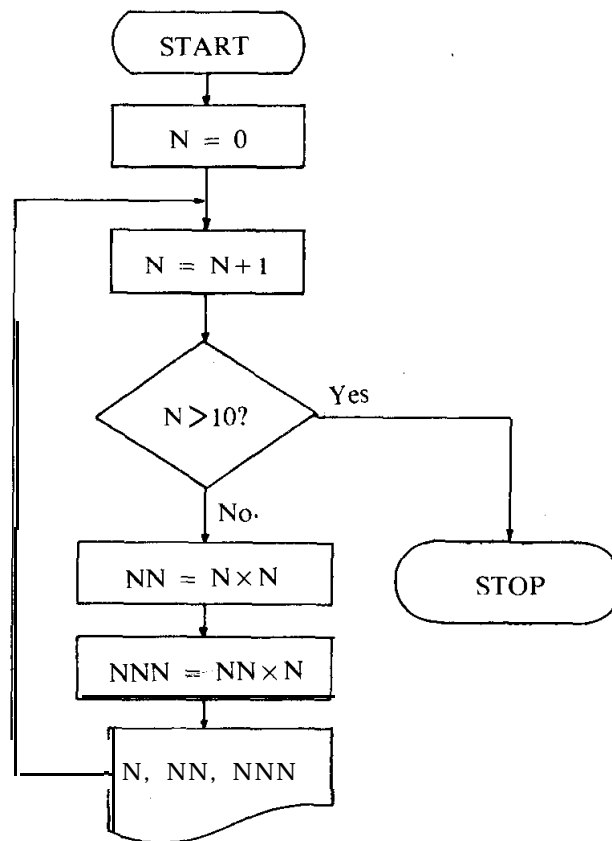
ในปีที่ 1  $N = N + 1$  เงินต้น P จะกลายเป็น  $100 \times (1 + 8/100)^1 = 108$

ต่อจากนั้นมาถึงการตัดสินใจว่า เปรียบเทียบว่า P มากกว่า 200 บาทหรือยัง ถ้ายังให้ไปที่จุดต่อหนึ่ง A ต่อไปคือ เปลี่ยนค่า  $N = N + 1$  ( $N = 1 + 1 = 2$ ) และ  $P = 100 \times (1 + 8/100)^2$  แล้วไปขึ้นการตัดสินใจเปรียบเทียบจำนวนเงิน P กับ 200 บาทอีก การทำงานของเครื่องจะเวียนเช่นนี้จนกว่า  $P > 200$  แล้วจึงไปทำงานขั้นต่อไปคือพิมพ์ค่าของ N ออกว่ามีค่าเท่าไร เราก็จะทราบจำนวนปีที่เราฝากเงินแล้วจะได้รับเงินฝากประมาณสองเท่าของเงินฝากครั้งแรก

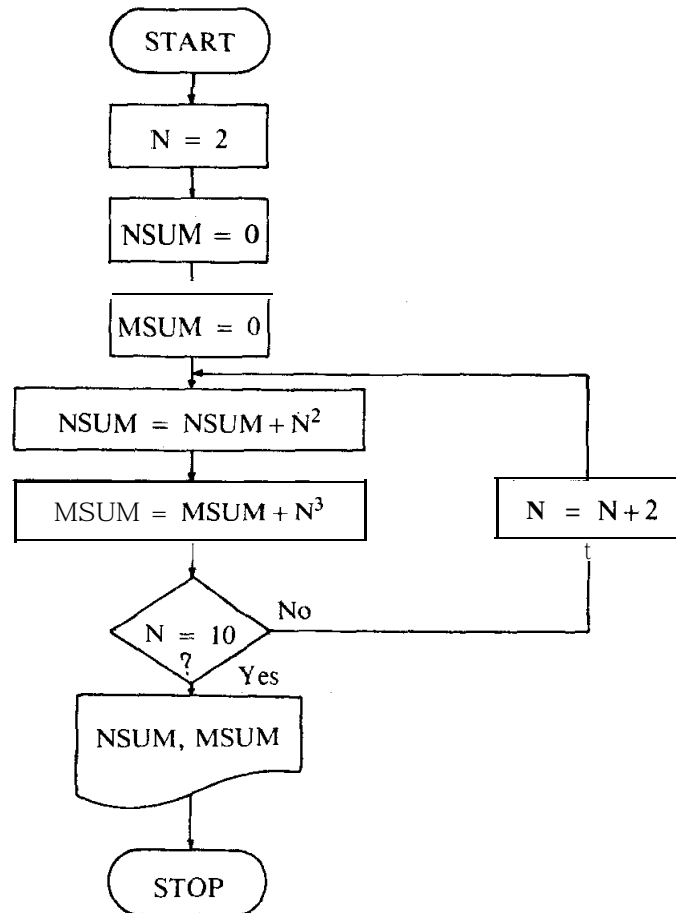
# แบบฝึกหัด

ข้อ 1. จากผังโปรแกรมที่กำหนดให้ จะพิมพ์ค่าของ 3 Variables N, NN และ NNN ก็ครั้ง  
มีค่าเท่าใดบ้าง ให้แสดงตามแบบที่กำหนดให้

ครั้งที่	ค่าของ		
	N	NN	NNN
1		<b>L</b>	
2			
⋮			

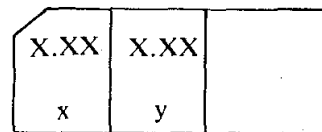


ข้อ 2. จากผังโปรแกรมที่กำหนดให้ จะพิมพ์ค่าของ NSUM และ MSUM เท่ากับเท่าใด



ข้อ 3. จงเขียนผังโปรแกรม เพื่ออ่านค่า X และ Y จากบัตร 10 บัตร

กำหนด INPUT CARD FORMAT



ถ้า  $x < y$

ให้หาค่า

$$Z = y^2 + x$$

ถ้า  $x > y$

ให้หาค่า

$$Z = \frac{x}{2} - y$$

ถ้า  $x = y$

ให้หาค่า

$$z = 2x$$

กำหนด OUTPUT FORMAT

X	Y	Z	คือ HEAD
---	---	---	
---	---	---	

ข้อ 4. ห้างสรรพสินค้าไทย ต้องการทำบัญชีเงินเดือนของพนักงานของห้าง 100 คน  
พนักงาน 1 คน มีข้อมูลบันทึกอยู่ในบัตร 1 บัตร

**กำหนด Input Card Format**

Field ที่	Column	รายการข้อมูล
1	1-6	รหัสประจำตัวพนักงาน
2	8-12	เงินเดือน (Salary)
3	13-14	จำนวนชั่วโมงที่ทำล่วงเวลา
4	15-17	อัตราค่าล่วงเวลา (บาท/ชม.)

สมมติว่าพนักงานทุกคนต้องเสียภาษีรายเดือนตามอัตราต่อไปนี้

เงินรายได้ (INCOME) น้อยกว่าหรือเท่ากับ 1,000 บาท ยกเว้นภาษี

1001-5000 บาท เสีย 3% ของรายได้

มากกว่าหรือเท่ากับ 5001 บาท เสีย 5% ของรายได้

(เงินรายได้ = เงินเดือน + เงินล่วงเวลา)

จงเขียนผังโปรแกรม เพื่อคำนวณเงินที่ห้างจะต้องจ่ายกับพนักงานแต่ละคนภายหลังจากหักภาษีไว้แล้ว

**กำหนด Output format**

THAI DEPARTMENT STORE

THE PAYROLL

EMPLOYEE NUMBER    INCOME (BAHT)    TAX (BAHT)    NETPAY (BAHT)

ข้อ 5. จงเขียนผังโปรแกรมเพื่อ

1. นับจำนวนนักเรียนที่สอบผ่านทั้ง 3 วิชา (คะแนนผ่าน : คะแนน  $\geq 60$ ) และได้คะแนนเฉลี่ยมากกว่า 80 คะแนน

2. พิมพ์รายการของนักเรียนที่มีลักษณะตามข้อ 1 พร้อมทั้งคะแนนรวม 3 วิชา และคะแนนเฉลี่ยของแต่ละคนด้วย

กำหนดให้มีนักเรียนทั้งหมด 1,000 คน (นักเรียน 1 คน มีข้อมูลบันทึกอยู่ในบัตร 1 บัตร)

กำหนด Card Format

Field ที่	Column	รายการข้อมูล
1	1-5	รหัสประจำตัว
2	7-11	คะแนนคณิตศาสตร์
3	13-17	คะแนนสังคม
4	18-23	คะแนนภาษาอังกฤษ

กำหนด Output Format

#### SCORE REPORT

STUDENT NUMBER	MATH	SOCIAL	ENGLISH	TOTAL	AVERAGE
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....

TOTAL = XXX PERSONS



ข้อ 6. โรงพยาบาลแห่งหนึ่งเก็บข้อมูลของคนที่ยินดีจะบริจาคโลหิตให้เมื่อโรงพยาบาลต้องการ คนทั้งหมดมี 1,500 คน แต่ละคนมีข้อมูลอยู่ในบัตร 1 บัตร โดยการกำหนดรหัสของกลุ่มเลือด ดังนี้

group A = I    group B = 2

group AB = 3    group O = 4

**กำหนด Card Format**

Field ที่	Column	รายการข้อมูล
1	1-6	รหัสประจำตัว (Code)
2	8	group เลือด
3	I0-I 1	อายุ
4	13-17	น้ำหนัก
5	19-24	ความสูง
6	26-32	เบอร์โทรศัพท์ที่ติดต่อได้ตลอดเวลา (สมมุติว่าทุกคนมี)

วันหนึ่งโรงพยาบาลต้องการเลือด Group O (โอ)

จึงเขียนผังโปรแกรม เพื่อเลือกและพิมพ์ record ของคนที่มีเลือด Group O และนับจำนวนคนที่มีเลือด Group O ทั้งหมดด้วย

**กำหนด Output Format**

HOSPITAL XY  
BLOOD TYPE O

NUMBER	CODE	AGE	WEIGHT	HEIGHT	TELEPHONE-NUMBER
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
<b>TOTAL = XXX PERSONS</b>					

ข้อ 7. เขียนผังงานเพื่อที่จะคำนวณ พ.ท.ของสามเหลี่ยมและพิมพ์ผลลัพธ์ออกมา โดยที่ให้ เครื่องอ่านข้อมูลเข้า คือด้านทั้งสามด้านของสามเหลี่ยมเข้าไปเก็บไว้ที่ Storage A,B,C ก่อนหาพื้นที่ของสามเหลี่ยม ต้องตรวจสอบก่อนว่าด้านทั้ง 3 ที่กำหนดให้มันเป็นด้านทั้ง 3 ของสามเหลี่ยมจริง (นั่นคือ 2 ด้านใด ๆ รวมกันย่อมยาวกว่าด้านที่เหลือ)

$$\text{กำหนดสูตร : พ.ท.ของสามเหลี่ยม} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{โดยที่} \quad s = \frac{a+b+c}{2}$$

สมมุติบัตรใบสุดท้ายมีค่า A = 0 และถ้าด้านทั้ง 3 ไม่เป็นด้านของสามเหลี่ยมให้พิมพ์ "NOT A TRIANGLE" ที่คอลัมน์ REMARK

กำหนด Output Format

AREA OF TRIANGLES					
NO	A	B	C	AREA	REMARK
⋮	⋮	⋮	⋮	⋮	⋮

ข้อ 8. เขียนผังงานในการที่จะอ่านอุณหภูมิเป็นองศาฟาเรนไฮท์และแปลงผลลัพธ์ได้ออกมาเป็นองศาเซลเซียส และพิมพ์ผลลัพธ์ออกมา

ข้อ 9. เขียนผังงานในการอ่านข้อมูลเข้าสี่ค่าเข้าไปเก็บไว้ที่ Storage A, B, C, D แล้วให้หาค่าสูงสุดของข้อมูลชุดนี้ออกมาแล้วพิมพ์ผลลัพธ์ที่ได้ออกมา

ข้อ 10. บริษัทแห่งหนึ่งจะจ่ายเงินโบนัสให้แก่คนงานโดยคิดจากเงินเดือนและยอดขายในแต่ละปี การจ่ายจะจ่ายเป็นไปตามหลักเกณฑ์ดังนี้

- เงินโบนัสจะเป็น 3% ของยอดขาย ถ้ายอดขายน้อยกว่าสี่เท่าของเงินเดือน
- เงินโบนัสจะเป็น 4% ของยอดขาย ถ้ายอดขายเกินสี่เท่าของเงินเดือน แต่ไม่หกเท่า
- เงินโบนัสจะเป็น 5% ของยอดขาย ถ้ายอดขายมีค่าเป็นหกเท่าหรือมากกว่าหกเท่าของเงินเดือน

จงเขียนผังงานตั้งแต่เริ่มอ่านข้อมูล (ข้อมูลคือเงินเดือน และยอดขายทั้งปีของพนักงานขายแต่ละคน) จนกระทั่งคำนวณหาโบนัสสิ้นปีและพิมพ์ผลออกมา

กำหนดบัตรใบสุดท้าย : ให้ ECODE = 0

กำหนด Output format

BONUS REPORT				
EMPLOYEE CODE	SALARY	AMOUNT SOLD	BONUS	

} head

ข้อ 11. จาก Quadratic form  $ax^2 + bx + c = 0$

จงเขียนผังงานให้เครื่องอ่าน ส.ป.ส. a,b,c แล้วคำนวณหาค่าของ X และพิมพ์ออกมา

โดยการใช้สูตร

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

หมายเหตุ ถ้าค่าของ  $(b^2 - 4ac)$  เป็นบวก, x จะมีค่ารากสมการสองค่า

ถ้าค่าของ  $(b^2 - 4ac)$  เป็นลบ ให้พิมพ์ผลออกมาว่า "The roots are complex" ที่

REMARK

และ ถ้าค่าของ  $(b^2 - 4ac)$  เป็นศูนย์, x จะมีค่า  $-b/2a$

กำหนดบัตรใบสุดท้าย : A<0

กำหนด Output format

QUADRATIC EQUATIONS						
NO	A	B	C	ROOT 1	ROOT 2	REMARK

} head

## 2.2 ผังโปรแกรมแบบโครงสร้าง (Structured program flowchart)

เท่าที่ผ่านมาผังโปรแกรมที่กล่าวมาแล้วเป็นผังโปรแกรมแบบไม่มีโครงสร้าง (unstructured program flowchart หรือ straight flowchart) นั่นคือตรรกะของผังโปรแกรมเริ่มจากจุดเริ่มต้น (Start) ถึงจุดจบ (Stop) ซึ่งมีซับซ้อนเพียงจำนวนน้อยเท่านั้น ผังโปรแกรมหากกล่าวใช้ได้ดีกับโปรแกรมสั้น ๆ แต่ถ้าท่านมีผังโปรแกรมที่ยาวและมีการเปลี่ยนทิศทางการทำงานมากมาย ตรรกะของงานจะยากที่จะติดตาม ในหลาย ๆ กรณีผังโปรแกรมแบบไม่มีโครงสร้างที่มีขนาดใหญ่ขึ้นยากมากที่จะติดตามดูผล มันซับซ้อนเกินกว่าที่จะช่วยทำให้ตรรกะของงานดูง่ายขึ้นตามจุดประสงค์ของการเขียนผังโปรแกรม

ในกลางปี 1960 ได้มีการศึกษาเกี่ยวกับความซับซ้อนของตรรกะและความสามารถในการทำงานของโปรแกรมเมอร์ การศึกษาดังกล่าวแสดงว่าโปรแกรมเมอร์ที่เขียนภาษาโคบอลคนหนึ่ง ๆ นั้นตรวจแก้ไขคำสั่งที่ผิดได้โดยเฉลี่ย 10 บรรทัด/วัน นั้นหมายความว่าถ้าโปรแกรมประกอบด้วยคำสั่ง 500 คำสั่ง มันอาจจะต้องใช้เวลาถึง 50 วัน ที่จะตรวจสอบโปรแกรมนั้นเสร็จ จากผลการศึกษาทำให้โปรแกรมเมอร์และคนอื่น ๆ จึงได้เริ่มต้นค้นหาวิธีการเขียนโปรแกรมที่มีประสิทธิภาพ และผลที่ได้คือการเขียนโปรแกรมแบบโครงสร้างนั่นเอง

### เหตุผลในการใช้เทคนิคการเขียนโปรแกรมแบบโครงสร้าง

ประโยชน์บางประการของเทคนิคการเขียนโปรแกรมแบบโครงสร้าง คือ

1. ถ้าปราศจากวิธีมาตรฐานในการแก้ปัญหา โปรแกรมเมอร์จะใช้เวลามากกว่าที่จำเป็นในการหาวิธีที่เหมาะสมและในการพัฒนาโปรแกรม การเขียนโปรแกรมแบบโครงสร้าง (Structured program) คือวิธีที่เราจะใช้

2. ผังโปรแกรมแบบไม่มีโครงสร้างส่วนมากจะมีการเปลี่ยนทิศทางมากมายซึ่งเป็นผลให้มีการเปลี่ยนลำดับของการประมวลผล สิ่งนี้ทำให้ผู้อื่นนอกจากโปรแกรมเมอร์ผู้เขียนโปรแกรมนั้นประสบความสำเร็จยากในการดัดแปลงโปรแกรม (บางครั้งยังยากสำหรับโปรแกรมเมอร์ผู้เขียนมันขึ้นเอง) ดังนั้น ราคาและความยุ่งยากในการเก็บรักษาโปรแกรมจึงเพิ่มมากขึ้น เนื่องจากการเขียนโปรแกรมแบบโครงสร้างไม่ใช่คำสั่ง GO TO จึงทำให้โปรแกรมเมอร์หันไปเขียน modular program (คือโปรแกรมที่แบ่งออกเป็นส่วนตัวต่าง ๆ ตามหน้าที่ของมัน ซึ่งเรียกว่า ซับรูทีน) ซึ่งง่ายแก่การดัดแปลงมากกว่าโปรแกรมแบบไม่มีโครงสร้าง บางครั้งเราเรียกโปรแกรมแบบโครงสร้างว่า โปรแกรมที่ปราศจาก GO TO (GO-TO-less)

3. เนื่องจากโปรแกรมแบบโครงสร้างมีส่วนย่อย ๆ มากกว่าโปรแกรมปกติ มันจึงง่ายในการทดสอบและแก้ไขมากกว่า ส่วนย่อย ๆ หรือซักรูทีที่สามารถถูกแยกทดสอบและพบข้อผิดพลาดได้ง่ายกว่า

4. เป็นการง่ายที่จะแยกโปรแกรมแบบโครงสร้างออกเป็นส่วน ๆ และกำหนดให้โปรแกรมเมอร์หลายคนช่วยกันเพื่อเพิ่มความเร็วในการเขียนโปรแกรม

5. ในระบบที่เป็นระบบ virtual storage นั้น ส่วนของโปรแกรมจะถูกเก็บในหน่วยความจำหลัก ณ เวลาหนึ่ง ๆ ซึ่งระบบนี้จะทำให้ลดความต้องการขนาดของหน่วยความจำหลักสำหรับการปฏิบัติตามโปรแกรมหนึ่ง ๆ ลงได้ ส่วนของโปรแกรมซึ่งไม่ได้เก็บในหน่วยความจำหลักถูกเก็บอยู่ในหน่วยความจำสำรอง (secondary storage) จนกว่าเราต้องการใช้มัน ส่วนต่าง ๆ ของโปรแกรมจะถูกย้ายไปมาระหว่างหน่วยความจำหลักและหน่วยความจำสำรองตลอดการปฏิบัติงานในโปรแกรม ในโปรแกรมซึ่งมีการเปลี่ยนทิศทางของการทำงานมากมาย การย้ายส่วนต่าง ๆ ของโปรแกรมไปมาอย่างต่อเนื่องตลอดเวลาจะสร้างความไม่มีประสิทธิภาพอย่างมาก ซึ่งเราเรียกว่า Thrashing โปรแกรมแบบโครงสร้างจะช่วยทำให้เกิด thrashing น้อยมาก

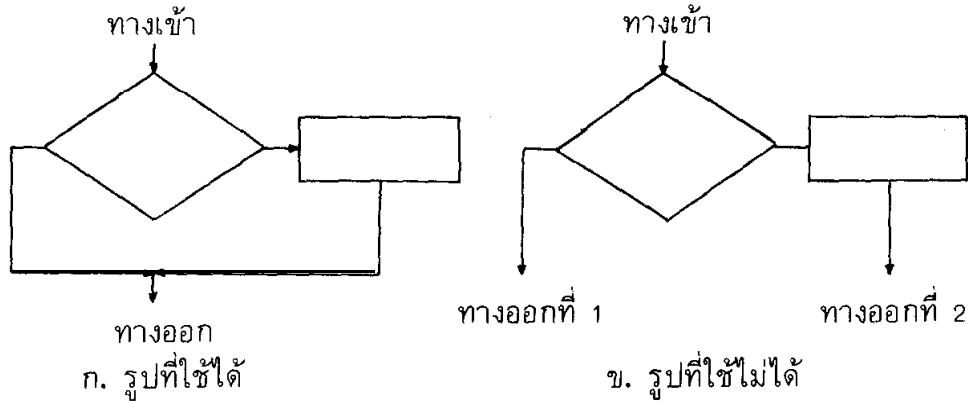
### กฎในการเขียนผังโปรแกรมแบบโครงสร้าง

ในการเขียนผังโปรแกรมแบบโครงสร้างนั้นควรจะได้คำนึงถึงกฎต่อไปนี้ คือ

1. มีโครงสร้างพื้นฐานเพียง 3 รูปเท่านั้นคือ Sequence, Decision (selection) และ Looping (iteration)

2. สัญลักษณ์แทนจุดต่อเนื่องอาจไม่ถูกใช้ในการสร้างลูป (loop) สัญลักษณ์ดังกล่าวควรจะถูกใช้เพื่อเชื่อมต่อการประมวลผลข้อมูลจากคอลัมน์หนึ่งหรือหน้าหนึ่งไปยังอีกคอลัมน์หรืออีกหน้าหนึ่งตามลำดับเมื่อเรามีที่ไม้เพียงพอ

3. ในโครงสร้างแต่ละรูปจะมีทางเข้า 1 ทาง และทางออก 1 ทางเท่านั้น



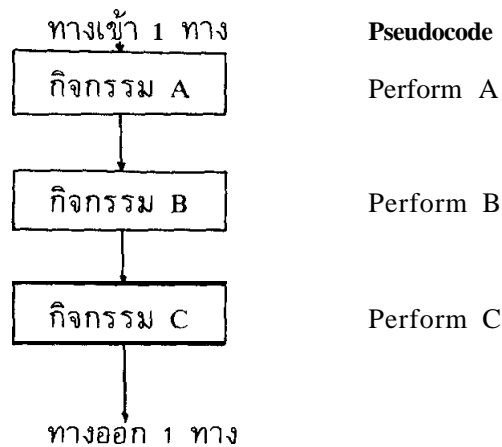
4. รูปโครงสร้างอาจซ้อน (nest) กันได้ นั่นคือรูปหนึ่งอาจซ้อนอยู่ในอีกรูปหนึ่งได้

5. ไม่ใช่คำสั่ง GO TO ดังได้กล่าวมาแล้วข้างต้นว่าเทคนิคการเขียนโปรแกรมอีกชนิดหนึ่งคือ modular programming ซึ่งหลายองค์การใช้อยู่ ใน modular program นั้นคำสั่ง GO TO อาจถูกใช้ในซับรูทีน แต่จะไม่ใช้ในการโยกย้ายจากซับรูทีนหนึ่งไปยังอีกซับรูทีนหนึ่ง นั่นหมายความว่าเราจะเข้าไปในซับรูทีนได้ที่จุดเริ่มต้นของมันและออกจากซับรูทีนที่จุดจบของมันเท่านั้น

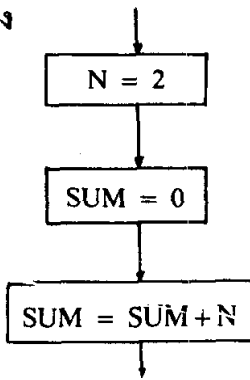
6. มีคำสั่ง Stop เพียง 1 คำสั่งในโปรแกรมหนึ่ง ๆ เท่านั้น และคำสั่งนี้จะต้องอยู่ในโปรแกรมหลัก (Main program)

### รูปโครงสร้างพื้นฐานของผังโปรแกรมแบบโครงสร้าง

1. Sequence structure รูปนี้จะมีสัญลักษณ์ใด ๆ ก็ได้ยกเว้น Decision symbol



ตัวอย่าง



คำสั่งในภาษาฟอร์แทรน

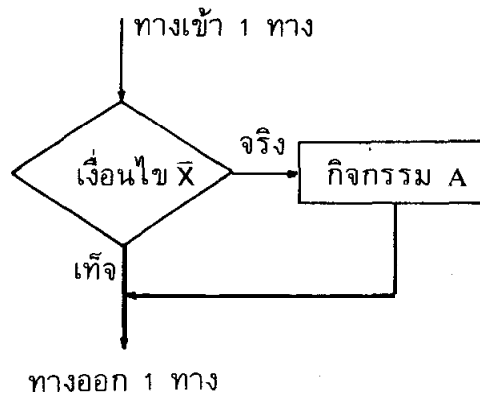
```

N = 2
SUM = 0
SUM = SUM+N
  
```

2. **Decision structure** ใช้เพื่อแสดงการเลือกซึ่งเป็นผลจากค่าของเงื่อนไข  
**IF-THEN-ELSE** ใช้เพื่อแสดงการเลือกซึ่งเป็นผลจากค่า 2 ค่า คือค่าถูกหรือผิดของ

เงื่อนไข

รูปที่ 1

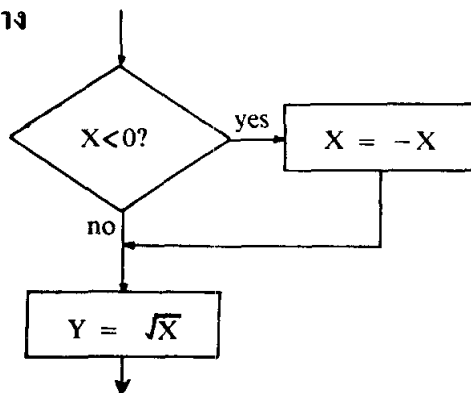


Pseudocode

```

IF (X) THEN
  Perform A
ENDIF
  
```

ตัวอย่าง

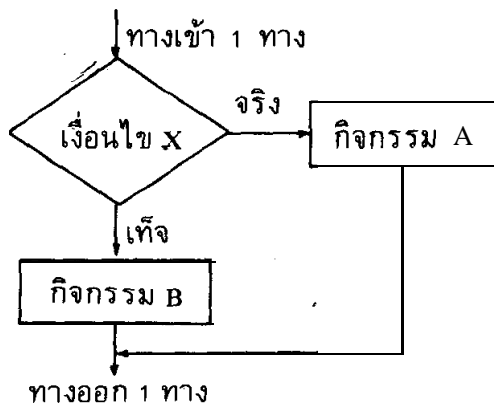


คำสั่งในภาษาฟอร์แทรน 77

```

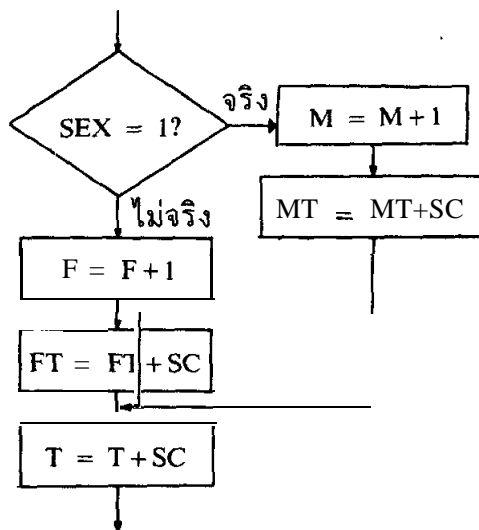
IF (X .LT.0) THEN
  X = -X
ENDIF
Y = SQRT (X)
  
```

รูปที่ 2



Pseudocode  
 IF (X) THEN  
   Perform A  
 ELSE  
   Perform B  
 ENDIF

ตัวอย่าง

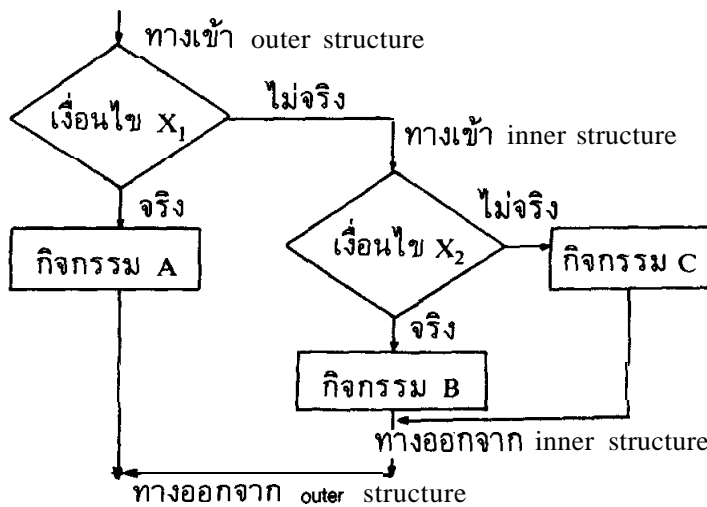


คำสั่งในภาษาฟอร์แทรน 77

```

  IF (SEX.EQ.1) THEN
    M = M + 1
    MT = MT + SC } บล็อก IF
  ELSE
    F = F + 1
    FT = FT + SC } บล็อก ELSE
  ENDIF
  T = T + SC
  
```

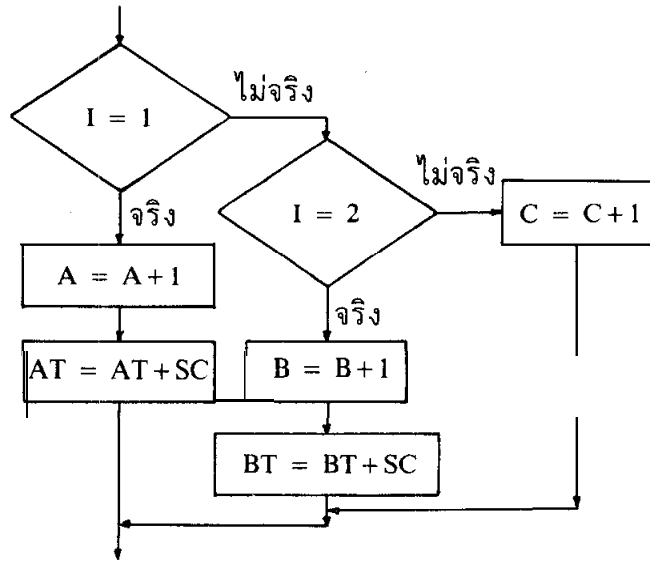
IF - THEN - ELSE - IF



Pseudocode  
 IF (X<sub>1</sub>) THEN  
   Perform A  
 ELSE IF (X<sub>2</sub>) THEN  
   Perform B  
 ELSE  
   Perform C  
 ENDIF  
 ENDIF



ตัวอย่าง

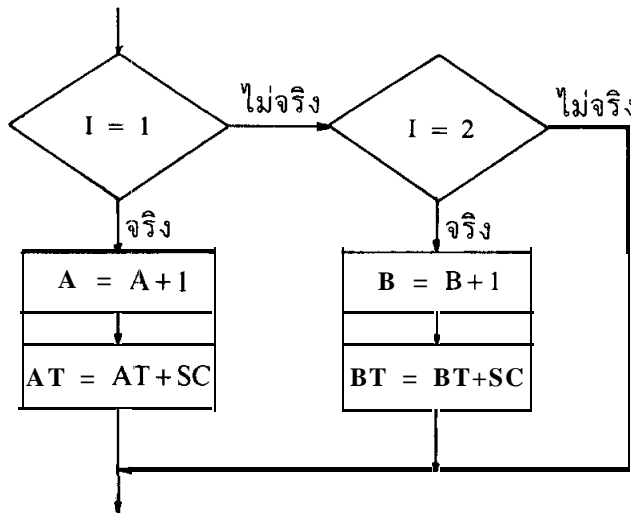


คำสั่งในภาษาฟอร์แทรน 77

```

IF (LEQ. 1). THEN
    A = A + 1
    AT = AT + SC
ELSE IF (LEQ.2) THEN
    B = B + 1
    BT = BT + SC
ELSE
    C = C + 1
ENDIF
  
```

ตัวอย่าง

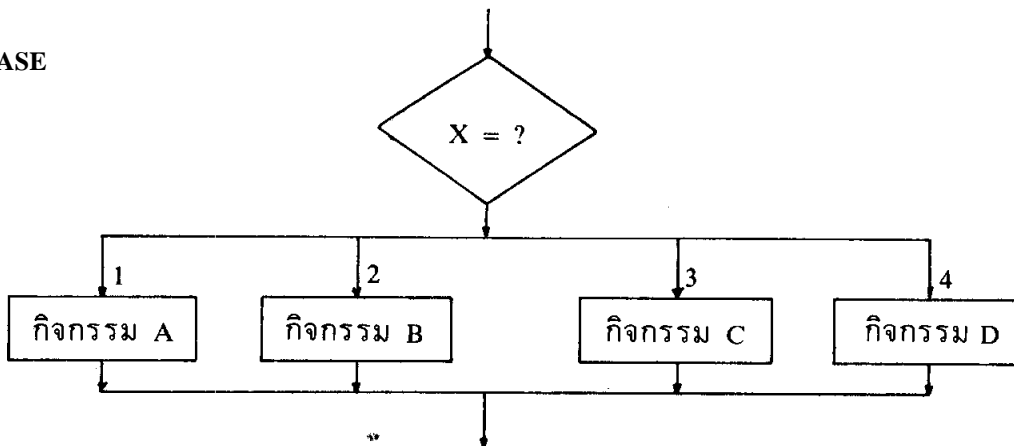


คำสั่งในภาษาฟอร์แทรน 77

```

IF (LEQ. 1) THEN
    A = A + 1
    AT = AT + SC
ELSE IF (LEQ.2) THEN
    B = B + 1
    BT = BT + SC
ENDIF
  
```

CASE



**Pseudocode**

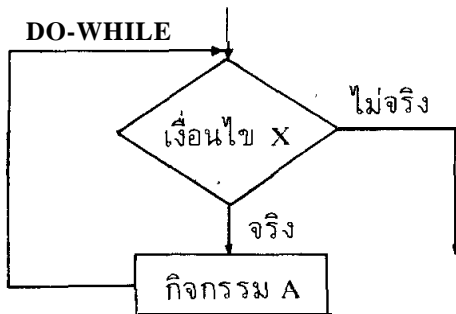
CASE X  
 X = 1, Perform A  
 X = 2, Perform B  
 X = 3, Perform C  
 X = 4, Perform D  
 END CASE

**ตัวอย่างคำสั่งในภาษาปาสคาล**

CASE REPLY OF  
 'N' : Y := 1;  
 'E' : X := 1;  
 'S' : y := 1;  
 'W' : x := -1;  
 END

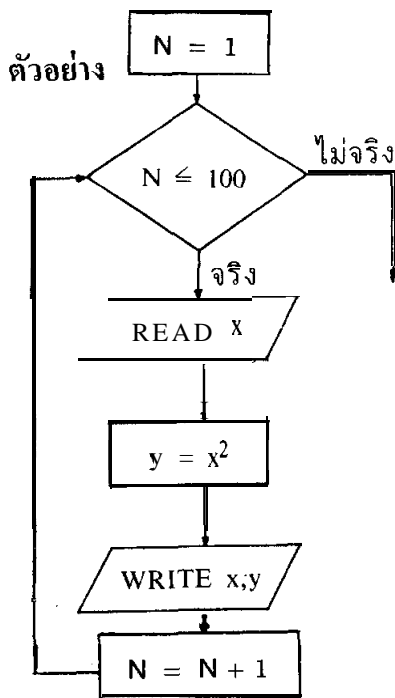
หมายเหตุ บางภาษาไม่มีคำสั่ง CASE แต่สามารถใช้คำสั่ง IF-THEN-ELSE-IF แทนได้

**3. Looping Structure** ใช้ควบคุมการกระทำซ้ำๆ ของคำสั่งหลายๆ ครั้ง



**Pseudocode**

DO WHILE X  
 Perform A  
 ENDDO

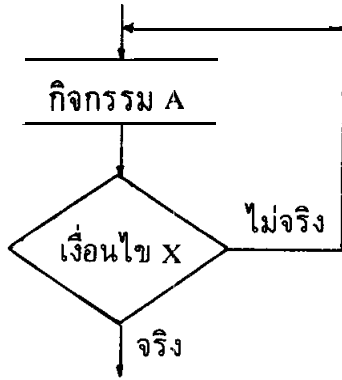


**คำสั่งในภาษาฟอร์แทรน -77**

```

N = 1
DO WHILE (N.LE. 100)
    READ (5,2) X
    Y = X**2
    WRITE (6,3) X, Y
    N = N + 1
END DO
2 | FORMAT (F5.2)
3 || FORMAT (2F9.4)
    
```

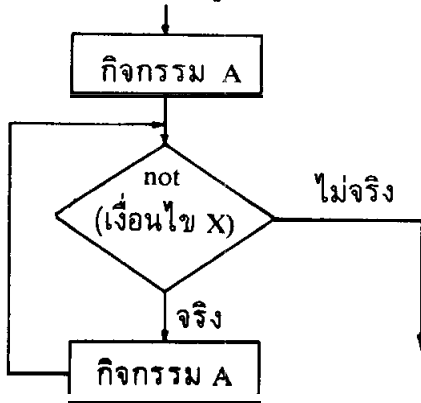
### DO-UNTIL



### Pseudocode

```
DO UNTIL X  
  Perform A  
ENDDO
```

ถ้าพิจารณารูปต่อไปนี้จะเห็นว่า DO-UNTIL เป็นส่วนกลับของ DO-WHILE



### Pseudocode

```
Perform A  
DO WHILE (not X)  
  Perform A  
END DO
```

ในรูป DO-WHILE นั้น กิจกรรม A อาจไม่ถูกกระทำเลย ถ้าเงื่อนไข X ไม่เป็นจริงตั้งแต่ครั้งแรก แต่ในรูป DO-UNTIL นั้น กิจกรรม A จะถูกกระทำอย่างน้อย 1 ครั้ง เพราะกิจกรรม A ถูกกระทำก่อนทดสอบเงื่อนไข X ว่าจริงหรือไม่

### 2.3 รหัสเทียม (Pseudocode)

เราอาจเขียน Pseudocode แทนผังโปรแกรมได้ ผังโปรแกรมแสดงตรรกะของงานเป็นรูปอย่างชัดเจน ส่วน Pseudocode นั้น เป็นการแสดงตรรกะของงานในลักษณะของภาษาเขียน โปรแกรมเมอร์จำนวนมากชอบใช้ Pseudocode โปรแกรมเมอร์บางกลุ่มนิยมใช้ผังโปรแกรมเพราะเขาต้องการที่จะมองเห็นรูปภาพของตรรกะให้ชัดเจน องค์กรบางแห่งต้องการให้มีผังโปรแกรมเป็นส่วนของเอกสารประกอบโปรแกรม ในขณะที่บางองค์กรต้องการใช้ Pseudocode บางองค์กรเปิดโอกาสให้โปรแกรมเมอร์ตัดสินใจเองว่าจะใช้วิธีใด ดังนั้นท่านควรจะทำ ความคุ้นเคยกับทั้ง 2 วิธี

#### กฎสำหรับการเขียน Pseudocode

ท่านเองเดียวกับการเขียนผังโปรแกรม การเขียน Pseudocode ต้องมีกฎเกณฑ์ในการเขียนเช่นกัน อย่างไรก็ตามวิธีการเขียน Pseudocode ก็มีแบบต่าง ๆ กัน ซึ่งยังไม่มีแบบใดที่เป็นมาตรฐาน กฎเกณฑ์ต่อไปนี้เป็นกฎที่องค์กรส่วนมากใช้เพื่อเขียน Pseudocode

1. Pseudocode ต้องไม่ขึ้นอยู่กับภาษาคอมพิวเตอร์ภาษาใด ๆ ให้พยายามหลีกเลี่ยงคำที่มีใช้เฉพาะภาษาใดภาษาหนึ่งเท่านั้น

2. ย่อหน้าบรรทัดเพื่อทำให้ Pseudocode ง่ายแก่การอ่านและเข้าใจ

3. ใช้อักษรตัวพิมพ์สำหรับคีย์เวิร์ด

4. เครื่องหมายวรรคตอนจะใช้หรือไม่ก็ได้

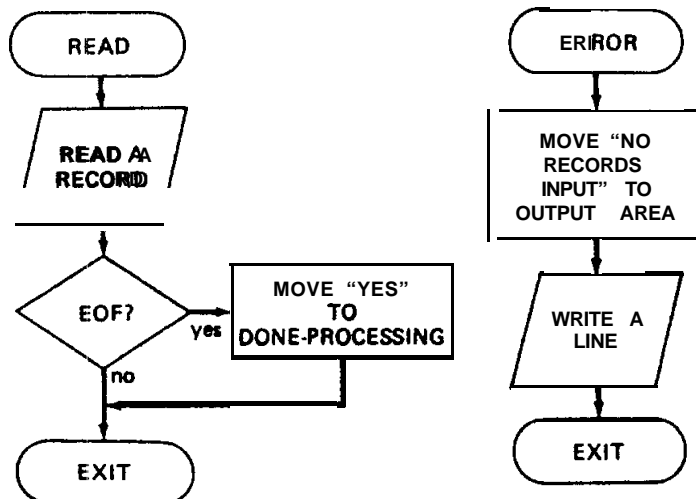
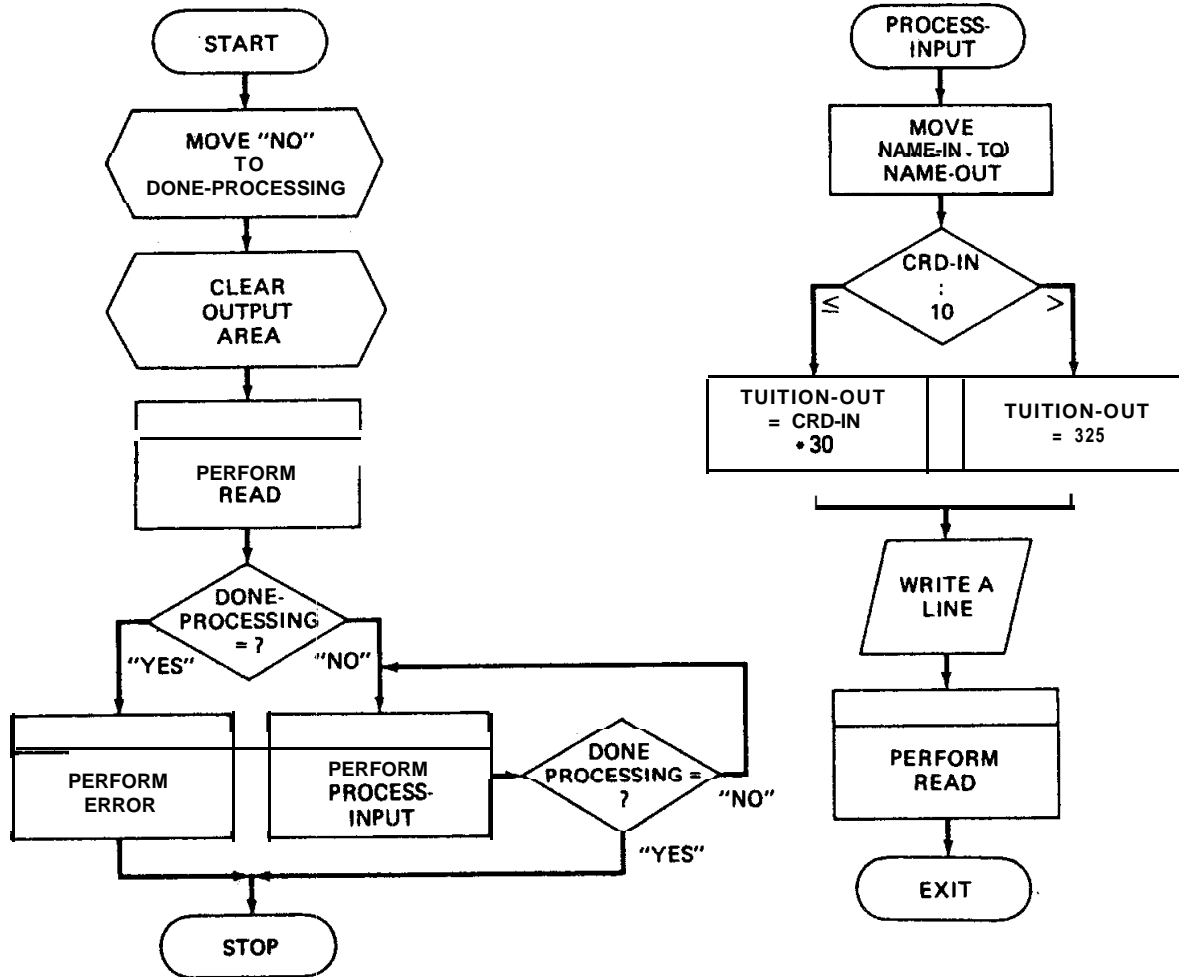
5. ทุกคำสั่ง IF ต้องจบด้วย ENDIF

6. ทุกคำสั่ง DO WHILE และ DO UNTIL ต้องจบด้วย ENDDO

7. โปรแกรมหลักจะต้องถูกแสดงไว้ก่อน ส่วนย่อย ๆ อื่น ๆ จึงใส่ตามมา

ตัวอย่างการเขียน Pseudocode สำหรับโครงสร้างพื้นฐานของผังโปรแกรมแบบโครงสร้างนั้นดูในหัวข้อ 5.2.2

ตัวอย่างของ Pseudocode เทียบกับผังโปรแกรม



ก. ผังโปรแกรม

```

MAINLINE
START
MOVE "No" to Done-Processing
CLEAR output area
DO Read routine
IF Done-Processing = "No" THEN
    DO Process-Input routine UNTIL Done-Processing = "Yes"
    ENDDO
ELSE DO Error routine
ENDIF
STOP

PROCESS-INPUT
MOVE Name-In to Name-Out
IF Crd-In is less than or equal to 10 THEN
    Tuition-Out = Crd-In . 30
ELSE Tuition-Out = 325
ENDIF
WRITE a line
DO Read routine
ENDPROCESS-INPUT

READ
READ a record
ATEND of file MOVE "Yes" to Done-Processing
ENDREAD

ERROR
MOVE "No Records Input" to output area
WRITE a line
ENDERROR

```

ข. Pseudocode สำหรับผังโปรแกรมใน ก.

## 2.4 ตารางการตัดสินใจ (Decision table)

เดิมนักวิเคราะห์ระบบ (System analyst) ใช้ตารางการตัดสินใจในการสื่อสารกับผู้ใช้ ถึงแม้ว่าโปรแกรมเมอร์หลาย ๆ คนจะพบว่าตารางการตัดสินใจ มีส่วนช่วยอย่างมากในการวิเคราะห์ตรรกะในการตัดสินใจที่ยุ่งยาก แต่ผังโปรแกรมทั้งผังไม่อาจถูกแทนด้วยตารางการตัดสินใจได้ ดังนั้น โปรแกรมเมอร์จึงอาจใช้ตารางการตัดสินใจประกอบกับผังโปรแกรม

ตัวอย่างที่ 1

Discount rate		Rule number	
		1	2
Application			
Decision stub	Condition statement		
	Purchases < \$100.00	T	F
Action stub	Action taken		
	Rate = 10%	X	
	Rate = 20%		X

ตารางแสดงการตัดสินใจถูกแบ่งออกเป็น 6 ส่วน ซึ่งถูกแบ่งด้วยเส้นคู่ ส่วนทั้ง 4 คือ Decision stub, Decision entry, Action stub, Action entry ส่วนชื่อของตารางและกฎต่าง ๆ แสดงไว้ส่วนบนของตาราง

เงื่อนไขที่จะถูกทดสอบเพื่อช่วยการตัดสินใจ แสดงไว้ในส่วนบนของตารางพร้อมกับค่าที่จะเป็นไปได้คือ T = true (หรือ Y = yes) และ F = false (หรือ N = no) ถ้าผลของการตัดสินใจไม่มีผลกระทบกับการกระทำเราอาจใส่ (hyphen) แทนที่ T หรือ F ใน Decision entry ก็ได้ การกระทำที่จะเกิดขึ้นซึ่งขึ้นอยู่กับ การตัดสินใจได้แสดงในส่วนล่างของตารางโดยการใช้ X ใน action entry ที่เหมาะสม

ตัวอย่างที่ 2 การให้สวัสดิการแก่คนงานในบริษัทแห่งหนึ่งเป็นไปตามกฎเกณฑ์ดังนี้

1. ผู้บริหารระดับสูง (Upper management) จะได้รับของบริษัที่ใช้ บริษัทจะประกันสุขภาพให้ หยุดพักผ่อนประจำปีได้ 4 อาทิตย์ ลาป่วยได้ 20 วัน ได้รับส่วนแบ่งกำไร และสามารถซื้อหุ้นของบริษัทได้

2. ผู้บริหารระดับกลาง (Middle management) จะได้รับการประกันสุขภาพ หยุดพักผ่อนได้ 4 อาทิตย์ ลาป่วยได้ 15 วัน และได้รับส่วนแบ่งกำไร

3. ผู้บริหารระดับต่ำ (Lower management) จะได้รับการประกันสุขภาพ หยุดพักผ่อนได้ 3 อาทิตย์ ลาป่วยได้ 10 วัน และได้รับส่วนแบ่งกำไร

4. พนักงานทั่วไป (Employee) จะได้รับการประกันสุขภาพ ลาป่วยได้ 10 วัน หยุดพักผ่อนได้ 3 อาทิตย์ ถ้าทำงานมากกว่าหรือเท่ากับ 5 ปี หรือหยุดพักผ่อนได้ 2 อาทิตย์ ถ้าทำงานต่ำกว่า 5 ปี

Benefits application	Rule Number				
	1	2	3	4	5
Condition statements					
Upper management	T	F	F	F	F
Middle management	F	T	F	F	F
Lower management	F	F	T	F	F
Regular employee	F	F	F	T	T
Employed less than 5 years	-	-	-	T	F
Action taken					
Company car	X				
Medical insurance	X	X	X	X	X
Profit sharing	X	X	X		
Stock options	X				
4 weeks vacation	X	X			
3 weeks vacation			X		X
2 weeks vacation				X	
20 sick days	X				
15 sick days		X			
10 sick days			x	x	x

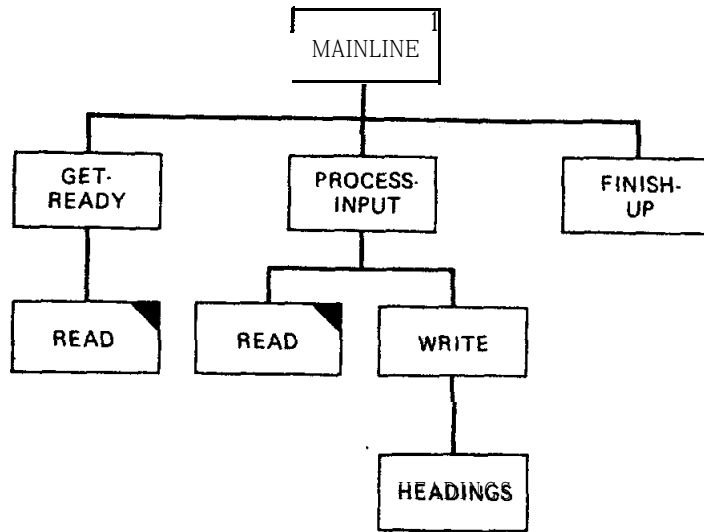


ตัวอย่างที่ 3

Bill collection application	Rule Number								
	1	2	3	4	5	6	7	8	9
Condition statements (How old is the bill?)									
30 days or less	Y	N	N	Y	N	N	Y	N	N
31-60 days	N	Y	N	N	Y	N	N	Y	N
over 60 days	N	N	Y	N	N	Y	N	N	Y
(What is the amount)									
\$1,000 or less	Y	Y	Y	N	N	N	N	N	N
\$5,000 or less	N	N	N	Y	Y	Y	N	N	N
over \$5,000	N	N	N	N	N	N	Y	Y	Y
Actions taken									
Send mild letter	X	X		X					
Send standard letter			x		X		X		
Send harsh letter						X		X	X
Collection agency			X			X			X
Stop credit over \$1,000						X			
Stop all credit									X
Go to rule 1, if more data	x	x	X	X	X	X	X	X	X

## 2.5 Hierarchy (Structure) chart

หลาย ๆ ครั้งที่เราไม่ทราบจะเริ่มต้นการเขียนผังภาพหรือ pseudocode อย่างไร hierarchy chart จะเป็นเครื่องมือที่มีประโยชน์ที่จะช่วยให้เราเริ่มต้นได้ hierarchy chart หรือ structure chart แสดงรูทिनที่เราจะใช้ทั้งหมดและความสัมพันธ์ของมัน



ตัวอย่าง hierarchy chart

ระดับแรกของแผนภูมิคือรูทินชื่อ MAINLINE ระดับถัดไปแสดงรูทินที่จะถูกเรียกใช้โดย MAINLINE คือรูทิน GET-READY, PROCESS-INPUT และ FINISH-UP ระดับถัดไปแสดงรูทินที่ถูกเรียกโดย GET-READY, PROCESS-INPUT และ FINISH-UP และเรื่อย ๆ ไป แผนภูมิจะแสดงรูทินทั้งหมดในโปรแกรม

โปรดสังเกตว่ารูทิน READ นั้นมีหลายอันแลเงาไว้ ทั้งนี้เพื่อแสดงว่ารูทิน READ ถูกเรียกใช้จากรูทินอื่นมากกว่า 1 รูทิน เราเรียกมันว่า Common module

เมื่อได้เขียน hierarchy chart เสร็จแล้ว โปรแกรมเมอร์ก็สามารถเขียนผังภาพและ pseudocode สำหรับรูทินที่แสดงไว้ในแผนภูมิ บางครั้งท่านอาจพบว่าได้ลืมนิรูทินหนึ่งหรือมากกว่าหนึ่งที่เราต้องใช้ในขณะที่เขียนผังภาพ ถ้าสิ่งนั้นเกิดขึ้นท่านก็เพียงแต่ดัดแปลง hierarchy chart แล้วก็เขียนผังภาพต่อไป

## 2.6 ไฮโป (HIPO)

HIPO ย่อมาจาก Hierarchy plus input-process-output บริษัท IBM เป็นผู้พัฒนา HIPO ขึ้น HIPO ประกอบด้วย hierarchy chart ของทั้งโปรแกรมและ IPO diagram (Input-process-output diagram) สำหรับแต่ละ routine (routine) รายละเอียดนั้นให้นักศึกษาสนใจ อาจหาอ่านเพิ่มเติมเอง

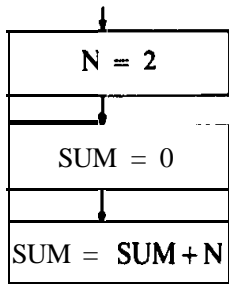
## 2.7 ผังภาพของแนสซี-ชไนเดอร์แมน

(Nassi-Schneiderman Chart : N-S chart)

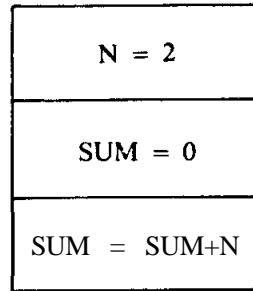
ผู้พัฒนาคือ I. Nassi และ Ben Schneiderman แต่ละ routine ใน N-S chart จะถูกแสดงโดยการใช้รูปสี่เหลี่ยมผืนผ้าขนาดใหญ่ เราใช้ N-S chart แสดง Sequence, decision (IF-THEN-ELSE และ CASE) และ looping (DO-WHILE) ได้ ใน N-S chart เราไม่สามารถแสดงคำสั่ง GO TO หรือการเปลี่ยนแปลงทิศทางแบบไม่มีเงื่อนไขได้

ต่อไปเป็นตัวอย่างของ N-S chart เทียบกับผังโปรแกรม

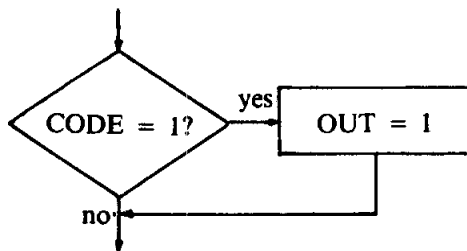
### 1. Sequence structure



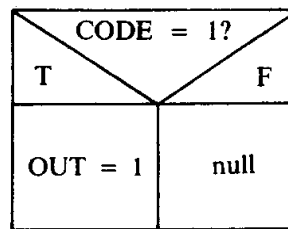
N-S chart

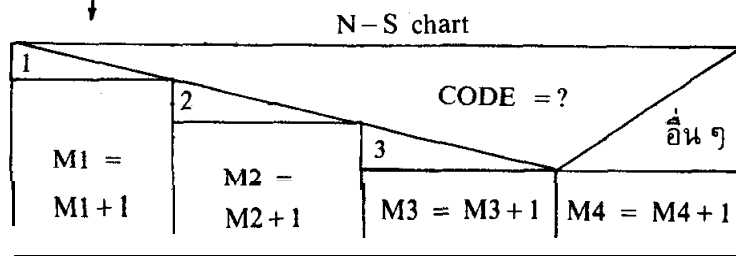
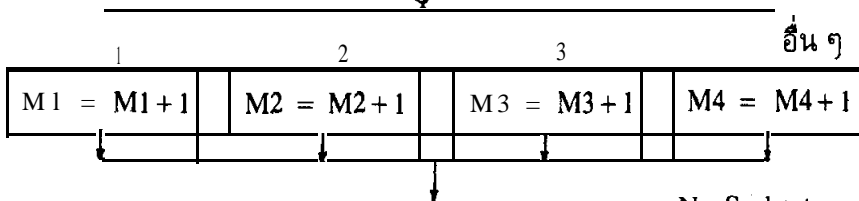
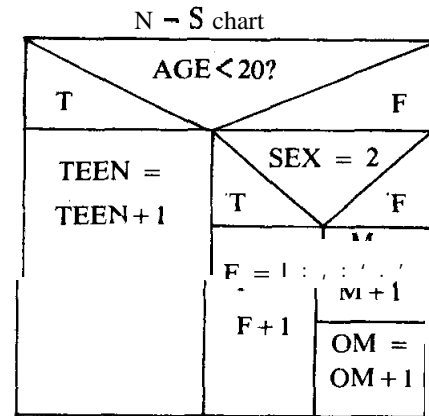
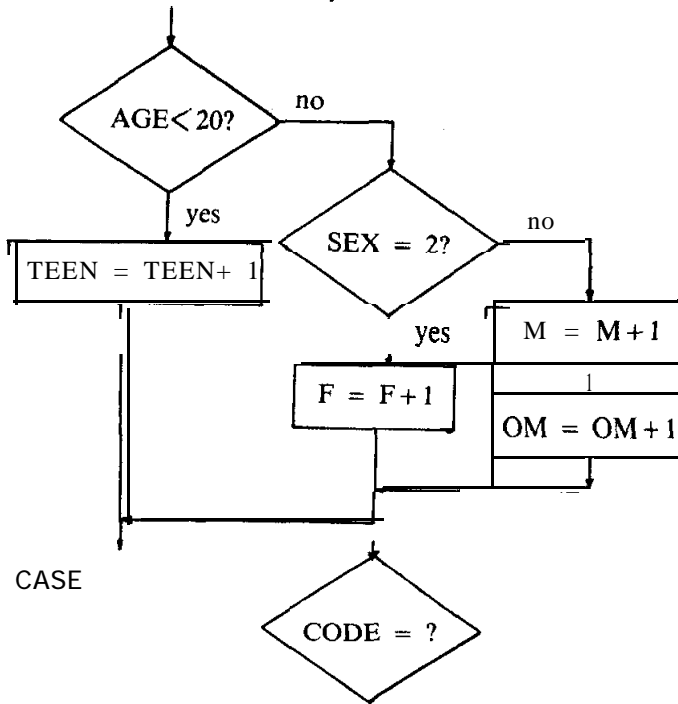
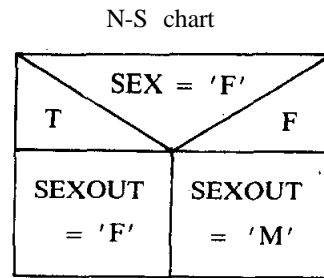
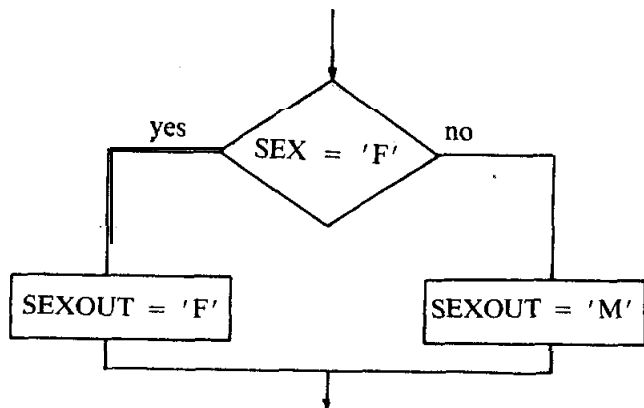


### 2. Decision Structure IF-THEN-ELSE

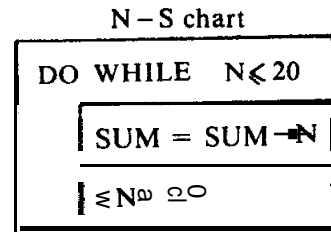
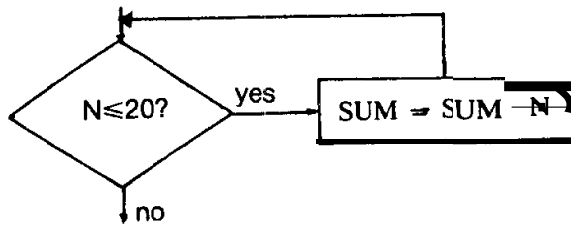


N-S chart



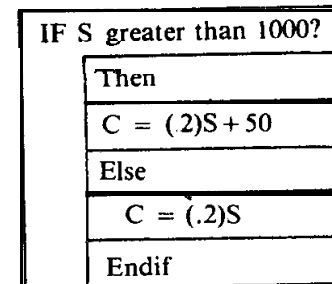
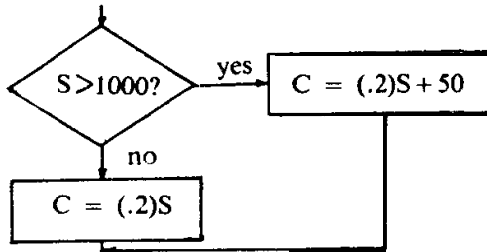


3. DO-WHILE structure

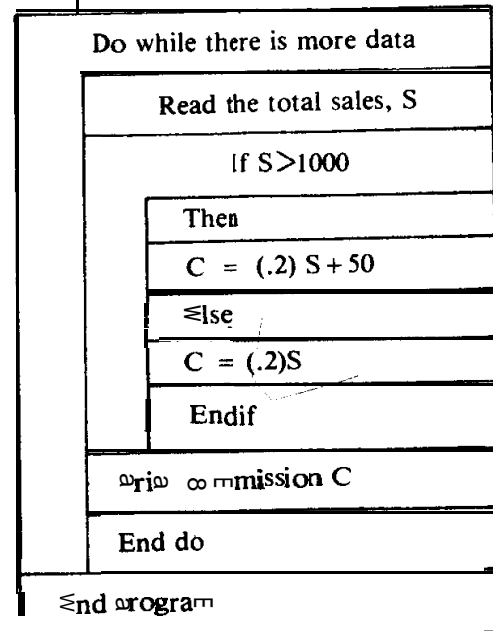
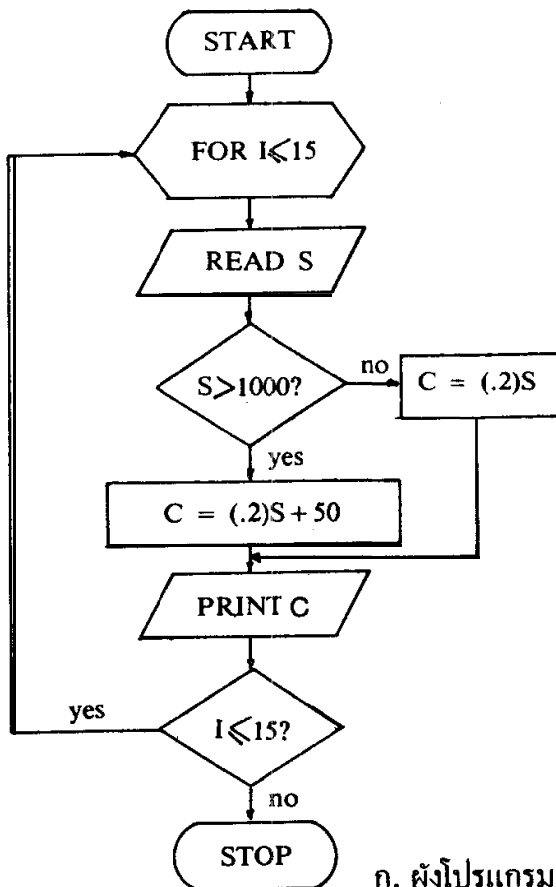


รูปแบบอื่นที่อาจพบในการเขียนผังโปรแกรมแบบโครงสร้าง ดูคล้ายแบบผสม

ระหว่าง N-S chart และ Pseudocode สำหรับรูป Decision structure



ตัวอย่างผังโปรแกรม

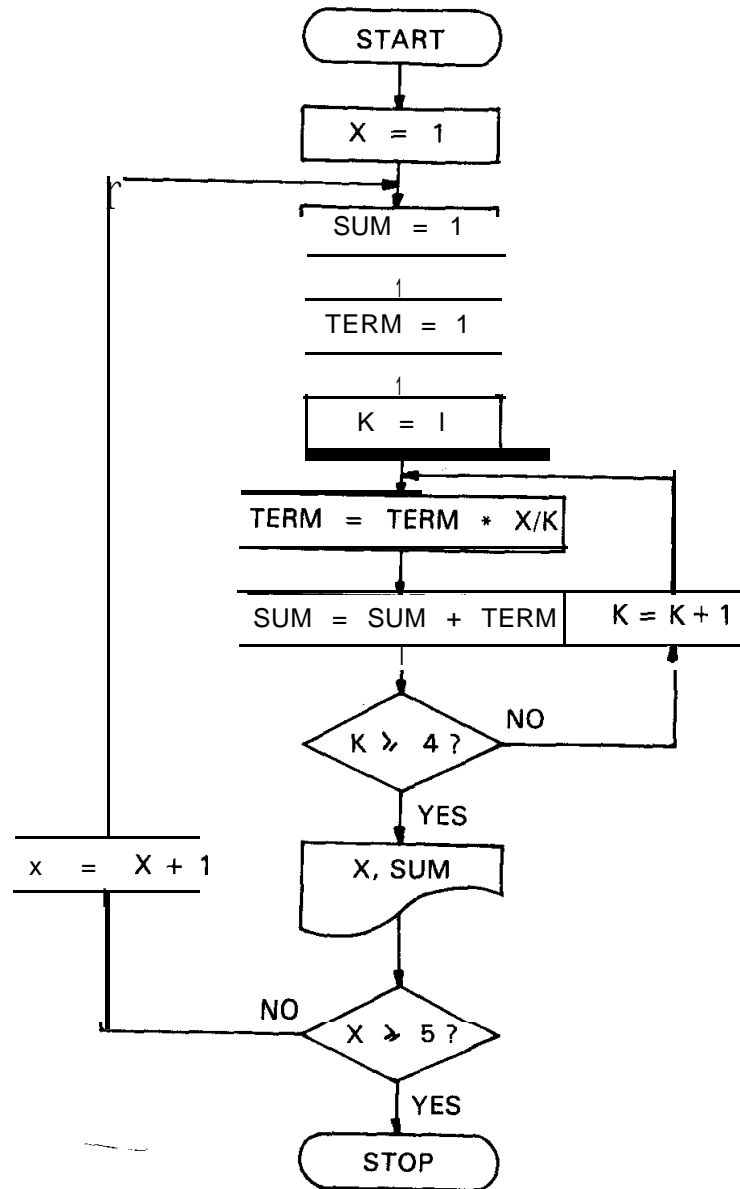


ข. ผังโปรแกรมแบบโครงสร้าง

ค. ผังโปรแกรมแบบเก่า

### ตัวอย่างผังโปรแกรมเพิ่มเติม

ตัวอย่าง 1 จากผังโปรแกรมที่กำหนดให้ จงบอกค่าของ X และ SUM ที่จะถูกพิมพ์



คำตอบ จะพิมพ์ค่าของ X และ  $SUM = \sum_{k=0}^4 \frac{X^k}{k!}$  เมื่อ  $X = 1.2, \dots, 5$

โดยที่  $k! = k(k-1) \dots 1, X^0 = 1, 0! = 1$

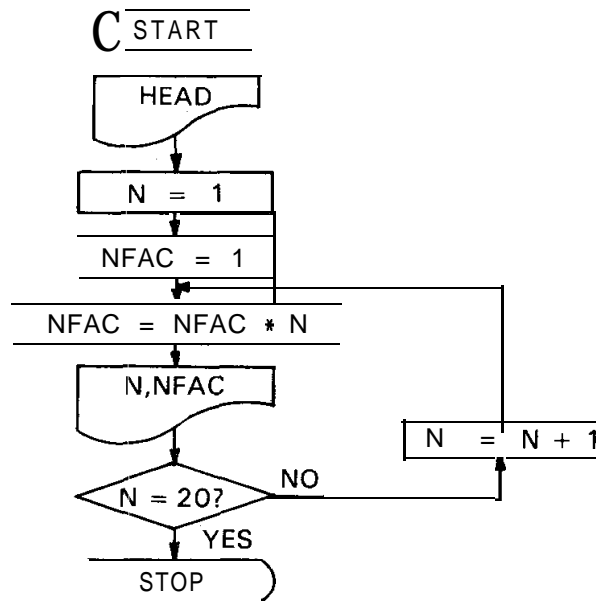
X = 1, SUM =	1 + 1 + 1/2 + 1/6 + 1/24	= 65124
X = 2, SUM =	1 + 2 + 4/2 + 8/6 + 16/24	= 168124
X = 3, SUM =	1 + 3 + 9/2 + 27/6 + 81/24	= 393/24
X = 4, SUM =	1 + 4 + 16/2 + 64/6 + 256/24	= 824/24
X = 5, SUM =	1 + 5 + 25/2 + 125/6 + 625/24	= 1569124

ตัวอย่างที่ 2 จงเขียนผังโปรแกรม เพื่อสร้างตาราง Factorial ของ 1-20

โดยที่  $N! = N(N-1)(N-2) \dots 2 \cdot 1$

กำหนด Output format

FACTORIAL TABLE		} head
NUMBER	FACTORIAL	
1	1	
20	20!	



ตัวอย่างที่ 8 บริษัทพรมแห่งหนึ่ง ประกาศลดราคาพรม 20% ให้กับลูกค้าที่ซื้อพรมมากกว่า 85 หลา

กำหนด format ของเรคคอร์ด (record)

field ที่	column	รายการข้อมูล	Variable
1	1 - 5	หมายเลขประจำตัวลูกค้า	CCODE
2	6 - 9	ราคาพรม/หลา	PER
3	10 - 13	จำนวนหลาที่สั่งซื้อ	YARD

จงคำนวณเงินที่ลูกค้าต้องจ่ายเป็นค่าพรมที่ซื้อ ในกรณีที่ลูกค้าต้องจ่ายน้อยกว่า 10,000.00 บาท ลูกค้าจะต้องจ่ายค่าขนส่งเพิ่มต่างหากอีก 100 บาท

กำหนดเรคคอร์ดสุดท้าย : CCODE = 99999

กำหนด Output format ให้พิมพ์หมายเลขประจำตัวลูกค้า, ความยาวพรมที่ซื้อ, ราคาพรมจริง และราคาที่ลูกค้าจะต้องจ่ายเมื่อหักส่วนลดหรือเพิ่มค่าขนส่งแล้วตามฟอร์มนี้

SIAM CARPET CO.LTD.

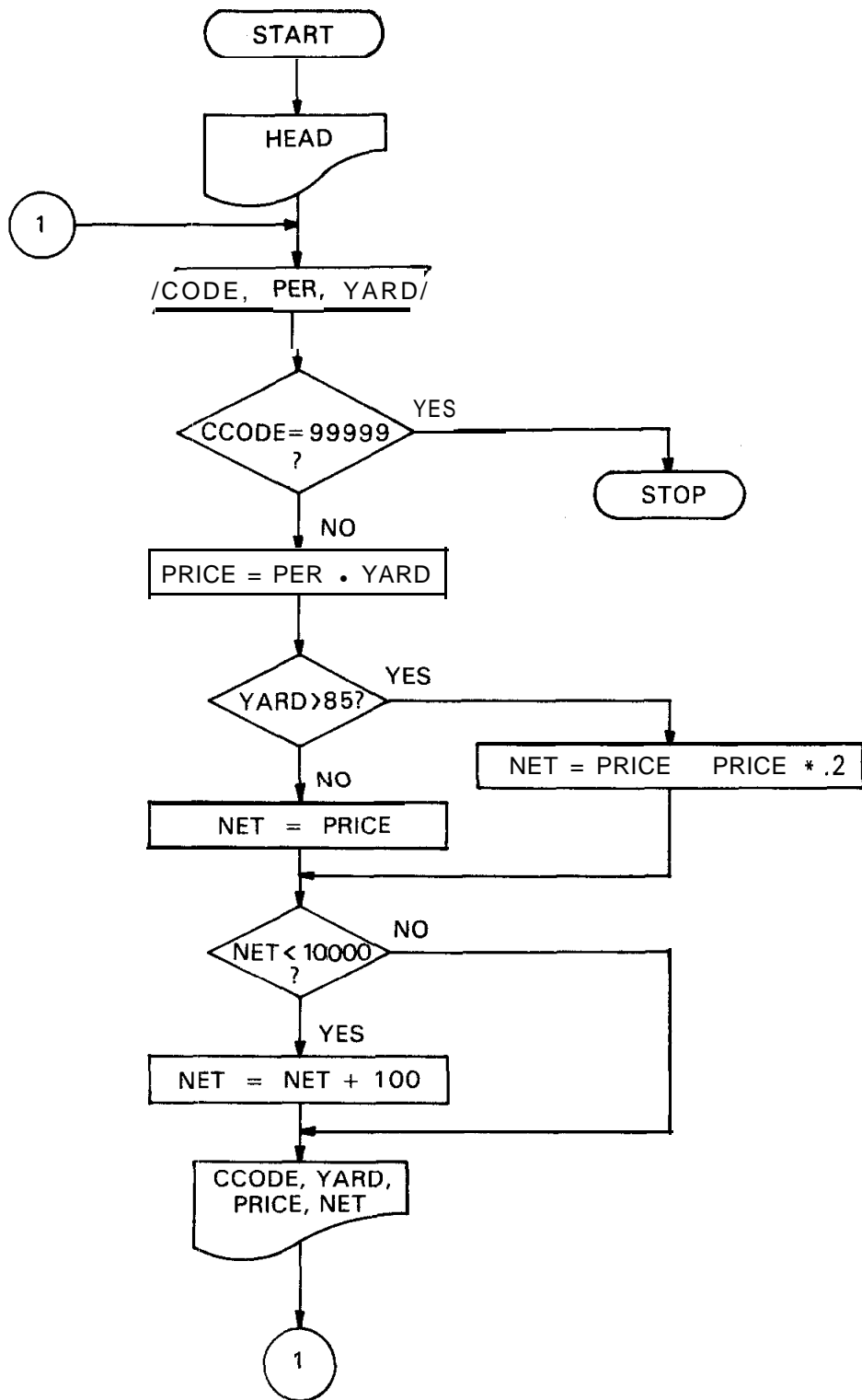
CUSTOMER CODE	NO.OF YARDS	PRICE (BAHT)	NET (BAHT)	}	head
---------------	-------------	--------------	------------	---	------

กำหนดให้

PRICE = ราคาพรม = PER \* YARD

NET = ราคาที่ต้องจ่ายเมื่อหักส่วนลดหรือเพิ่มค่าส่งแล้ว





ตัวอย่างที่ 4 จากข้อมูลของนักศึกษาที่สอบผ่านวิชา CS 215 ภาคฤดูร้อน/2526 จำนวน 22 คน แต่ละคนมีข้อมูลบันทึกอยู่ในหนึ่งเรคคอร์ด ซึ่งมี format ดังนี้

field ที่	column	รายการข้อมูล	Variable
1	1 - 8	รหัสประจำตัว	SCODE
2	9 - 40	ชื่อ-นามสกุล	NAME
3	41	เพศ : ชาย = 1, หญิง = 2	SEX
4	42 - 43	คะแนน	SCORE

จงเขียนผังโปรแกรมเพื่อ

1. พิมพ์รายชื่อของนักศึกษาที่สอบผ่านทั้งหมด รวมทั้งเกรดที่ได้ โดยใช้เกณฑ์  
ดังนี้

P :  $60 \leq \text{คะแนน} \leq 84$  และ

G :  $\text{คะแนน} \geq 85$

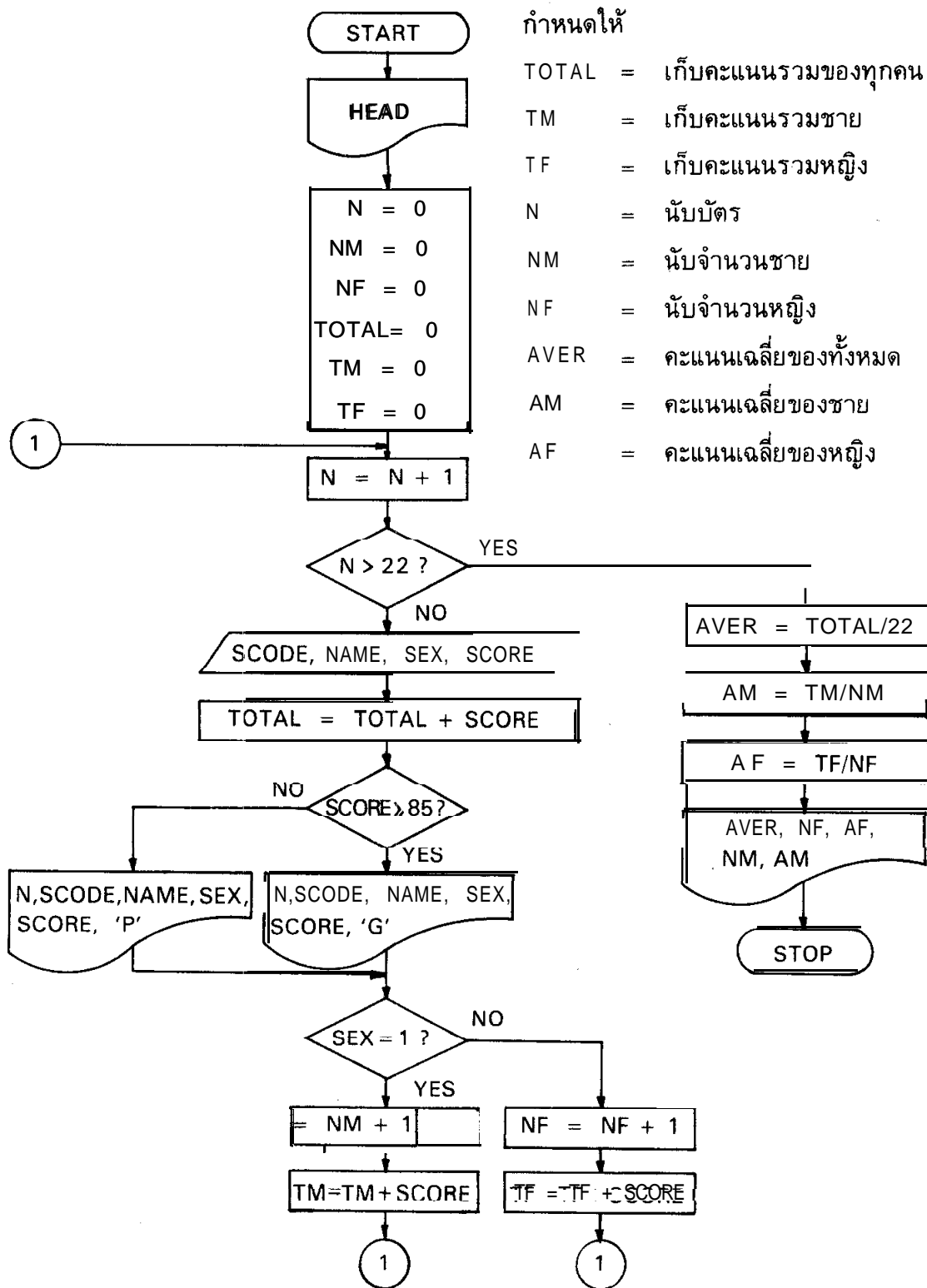
2. หาคะแนนเฉลี่ยของนักศึกษาที่สอบผ่านวิชา CS 215

3. นับจำนวนนักศึกษาชายและนักศึกษาหญิง และหาคะแนนเฉลี่ยของแต่ละกลุ่ม

ด้วย

กำหนด Output format

	CS 215 (S/2526)						
	GRADE	REPORT					head
NO.	STUDENT ID.	NAME	SEX	SCORE	GRADE		
1							
⋮							
22							
AVERAGE SCORE	=			<u>P O I N T S</u>			
NUMBER OF FEMALE	=			__ PERSONS			
FEMALE AVERAGE SCORE	=			__ POINTS			
NUMBER OF MALE	=			__ PERSONS			
MALE AVERAGE SCORE	=			_____ POINTS			



ตัวอย่างที่ 5 กำหนด Input เรคคอร์ด ให้ 20 เรคคอร์ด แต่ละเรคคอร์ดมีค่าของ X และ Y จงเขียนผังโปรแกรมเพื่อคำนวณหาค่า Z ตามเงื่อนไขดังนี้

ก) ถ้า  $X < 1000$  และ  $Y \leq 1500$  หาค่า  $Z = 2X - Y$

ข) ถ้า  $X \geq 1000$  และ  $Y > 1500$  หาค่า  $Z = X + Y$

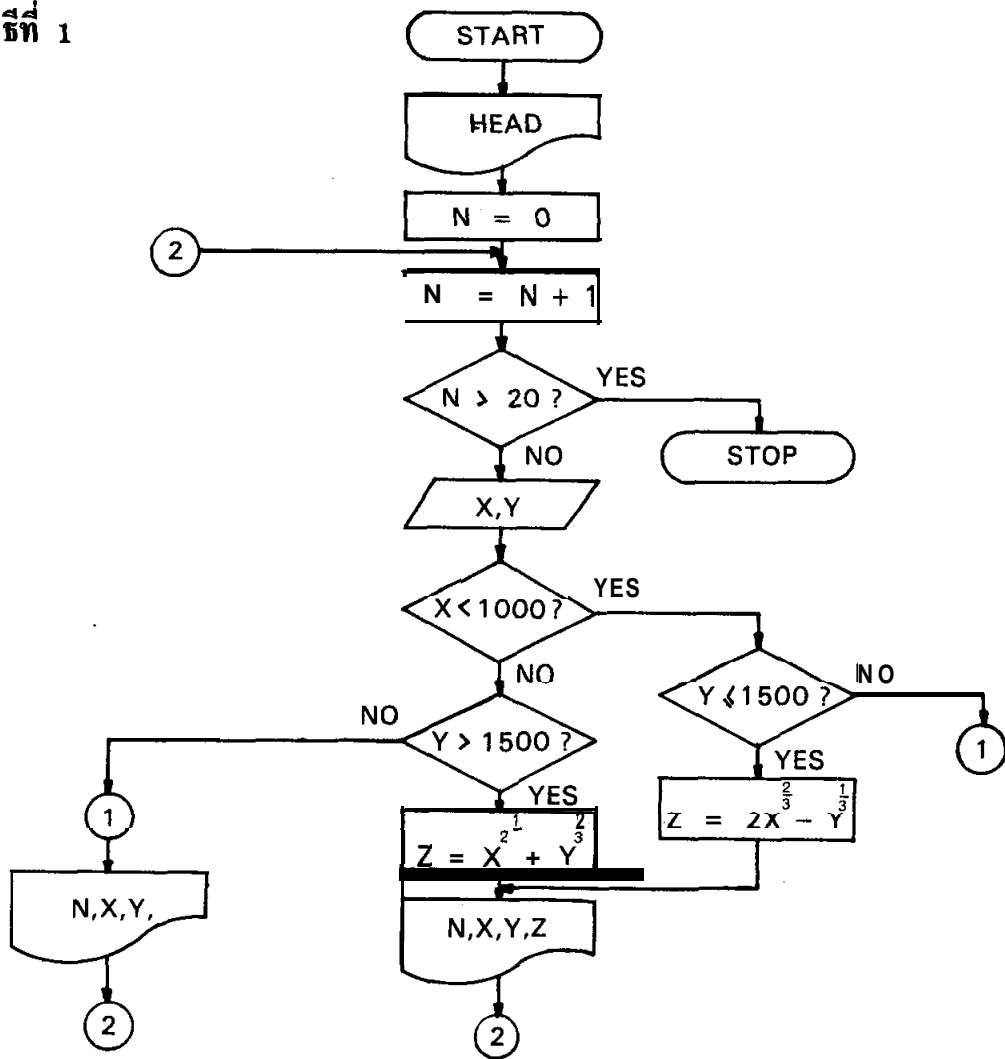
นอกจากนี้ไม่ต้องคำนวณค่า Z แต่ให้พิมพ์ค่า X และ Y ด้วย

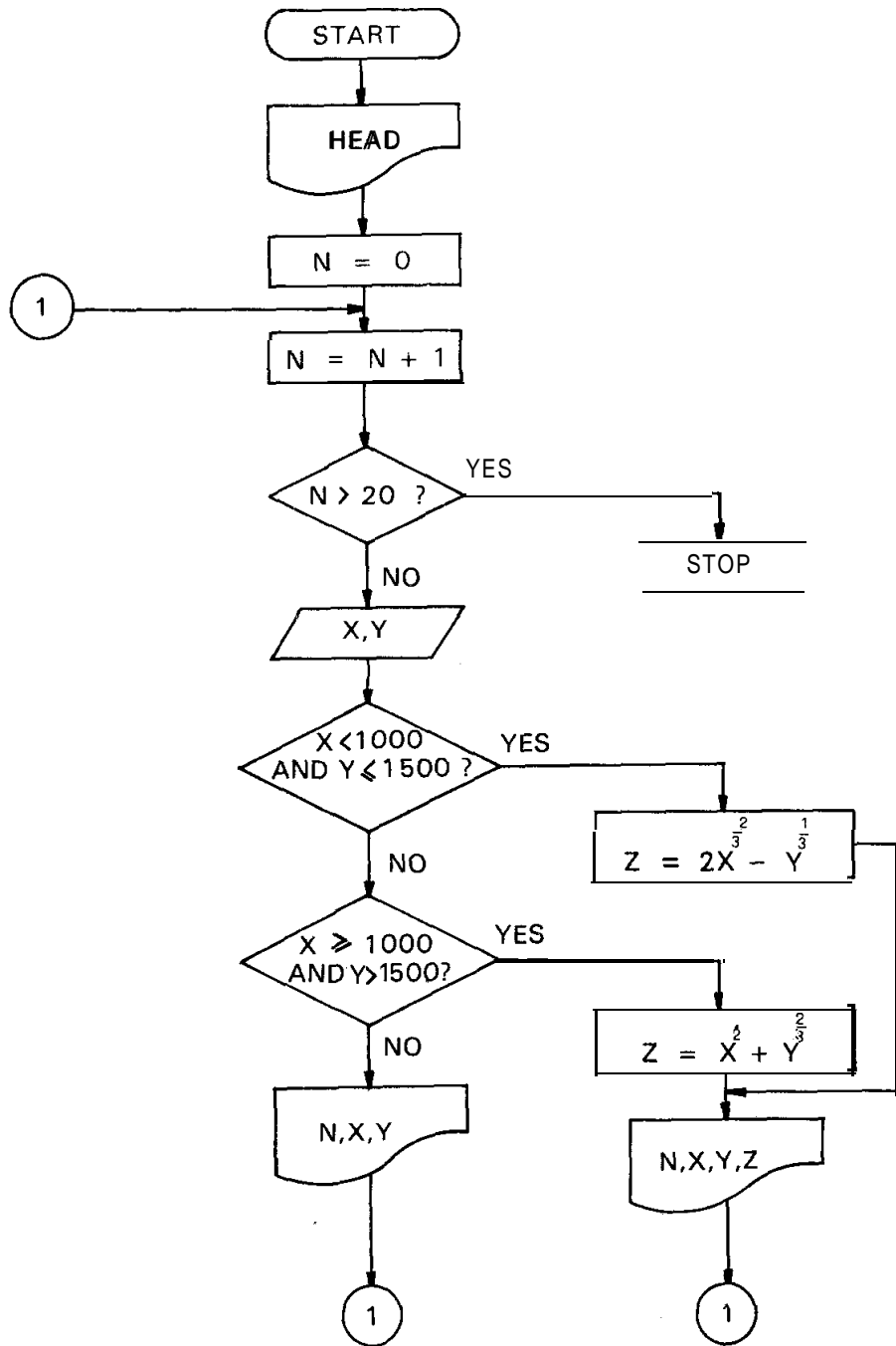
กำหนด Output format

COMPUTED Z - VALUES			
NO	X	Y	Z
⋮	⋮	⋮	⋮

} head

วิธีที่ 1



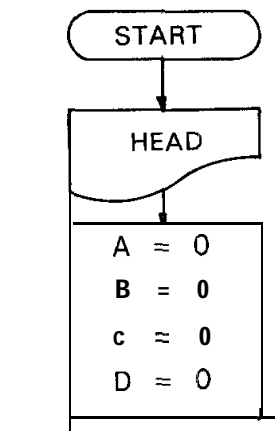


ตัวอย่างที่ 6 ในการเลือกตั้งกรรมการตำแหน่งหนึ่งของสมาคมแห่งหนึ่ง มีผู้สมัครรับเลือกตั้ง 3 คน โดยที่หมายเลข 1 คือ นาย A, หมายเลข 2 คือนาย B และหมายเลข 3 คือนาย C มีผู้มีสิทธิลงคะแนนเสียง 1500 คน การเลือกแต่ละครั้งจะถูกบันทึกใน 1 เรคคอร์ด ใน column 1 ตามหมายเลขของผู้ที่ได้รับการเลือก นั่นคือถ้าเลือกนาย B เลข 2 จะถูกเจาะลงใน column 1 ในกรณีที่ไม่มีใครเลยให้เจาะเลข 4 ใน column 1

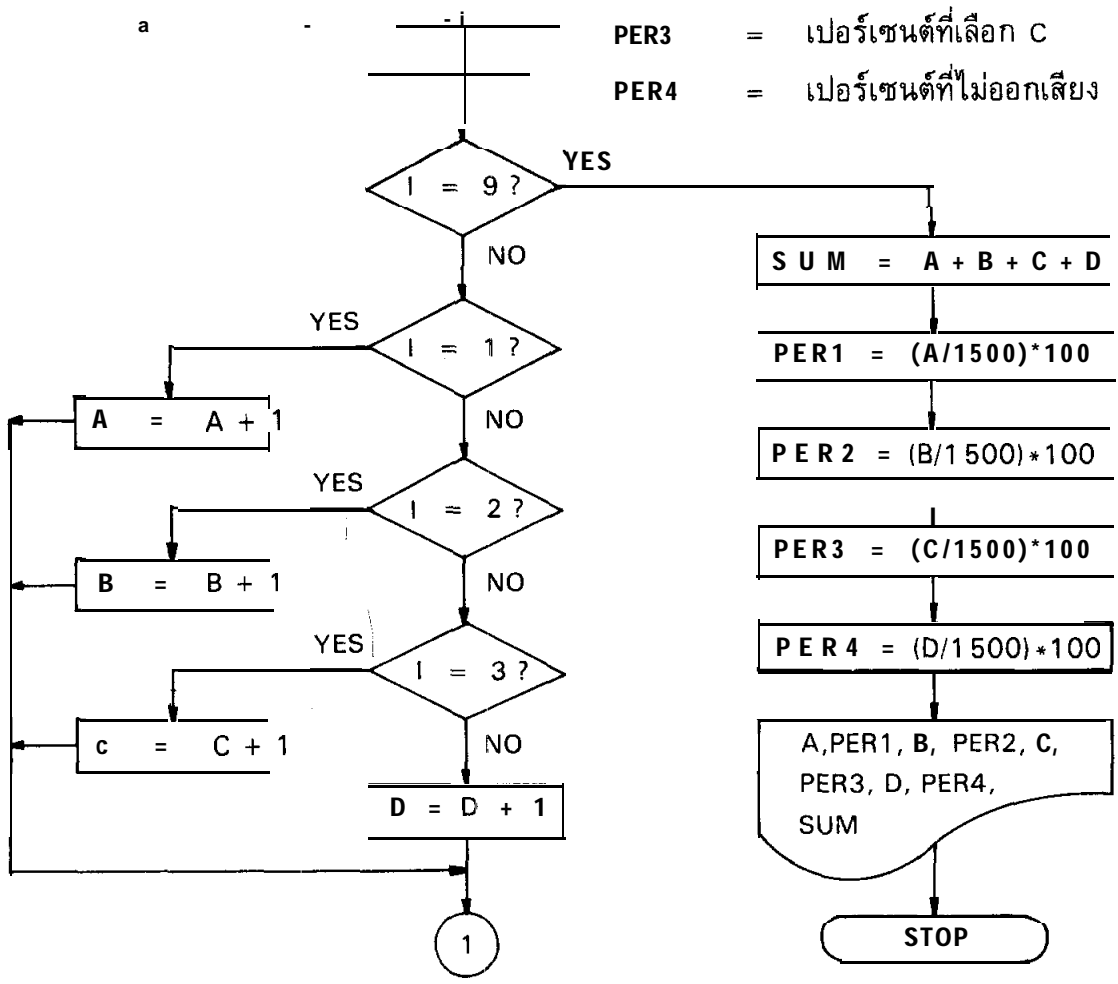
กำหนดเรคคอร์ดสุดท้าย บันทึกเลข 9 ที่ column 1

จงเขียนผังโปรแกรมเพื่อนับคะแนนเลือกตั้ง คำนวณเปอร์เซ็นต์ที่ผู้สมัครแต่ละคนได้รับ และพิมพ์ผลตาม Output format ที่กำหนดข้างล่างนี้

	CANDIDATE	VOTES
MR.A	= _____,	THAT IS <u>P E R C E N T</u>
MR.B	= _____,	THAT IS <u>P E R C E N T</u>
MR.C	= _____,	THAT IS ___ PERCENT
BAD VOTES	= _____,	THAT IS <u>P E R C E N T</u>
TOTAL VOTES	= _____,	THAT IS 100 PERCENT .



- A = ใช้นับคะแนนของนาย A
- B = ใช้นับคะแนนของนาย B
- C = ใช้นับคะแนนของนาย C
- D = ใช้นับผู้ไม่ออกเสียง
- I = หมายเลขที่เลือก
- SUM = จำนวนผู้มาใช้สิทธิ์
- PER 1 = เปอร์เซ็นต์ที่เลือก A
- PER2 = เปอร์เซ็นต์ที่เลือก B
- PER3 = เปอร์เซ็นต์ที่เลือก C
- PER4 = เปอร์เซ็นต์ที่ไม่ออกเสียง



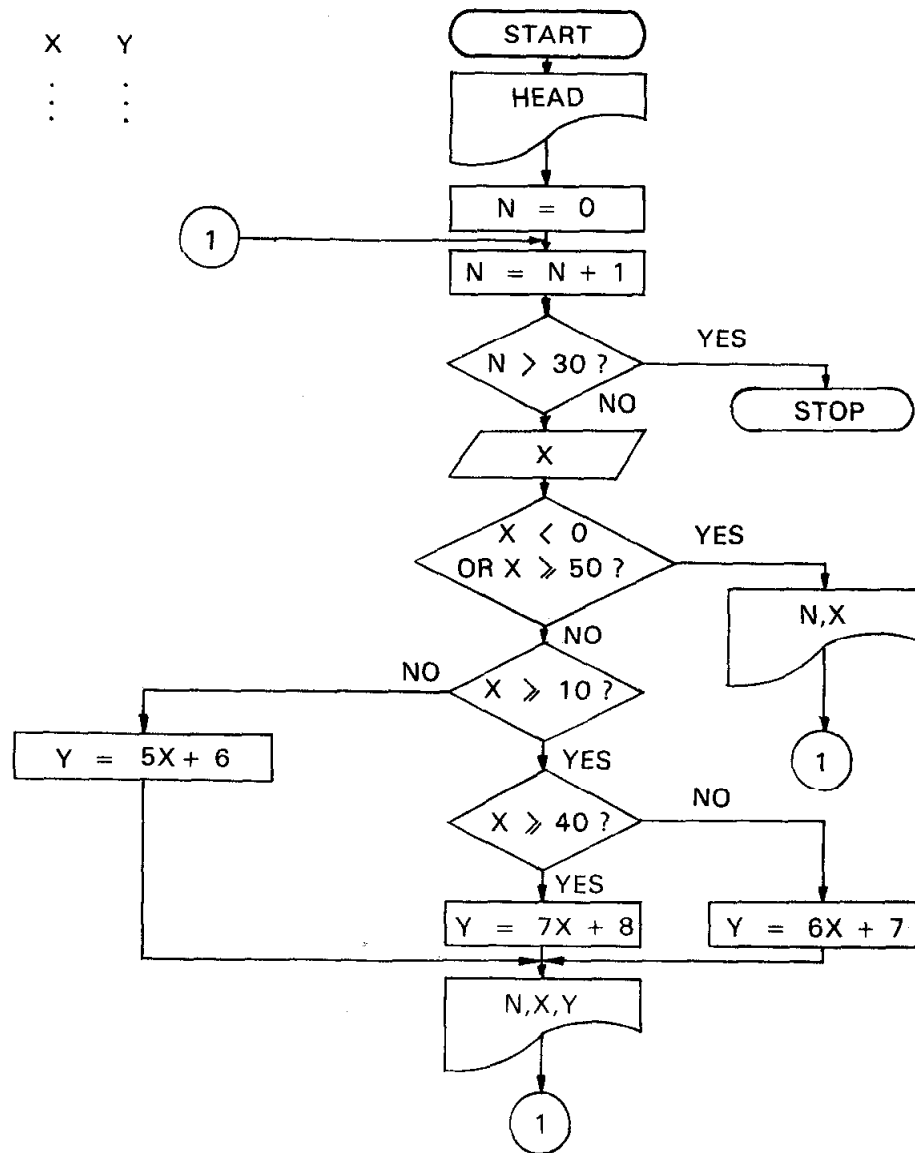
ตัวอย่างที่ 7 จงเขียนผังโปรแกรมเพื่ออ่านค่า X จาก 30 เรคคอร์ด แล้วคำนวณค่า Y ตามเงื่อนไขต่อไปนี้

$$Y = \begin{cases} 5X + 6 & \text{ถ้า } 0 \leq X < 10 \\ 6X + 7 & \text{" } 10 \leq X < 40 \\ 7X + 8 & \text{" } 40 \leq X < 50 \end{cases}$$

ถ้า X อยู่นอกขอบเขตเหล่านี้จะไม่คำนวณค่า Y แต่ให้พิมพ์ค่าของ X ด้วย

กำหนด Output format

NO	X	Y
⋮	⋮	⋮





### ตัวอย่างที่ 8

ในภาค 1/2528 นักศึกษาที่ลงทะเบียนเรียนวิชา CS 105 จะต้องกรอกแบบสอบถามซึ่งมีแบบฟอร์มดังนี้

โครงการศึกษาลักษณะบางประการของนักศึกษาที่เรียนวิชา CS 105  
ภาค 1/2528

จงเติมคำตอบลงในช่องว่างและขีด ✓ ลงใน  ของคำตอบที่ต้องการ

1. รหัสประจำตัว \_\_\_\_\_
2. เพศ  ชาย  
 หญิง
3. อายุ \_\_\_\_\_ ปี
4. คณะที่ศึกษา  นิติศาสตร์  
 บริหารธุรกิจ  
 มนุษยศาสตร์  
 ศึกษาศาสตร์  
 วิทยาศาสตร์  
 รัฐศาสตร์  
 เศรษฐศาสตร์

ก. จงกำหนดรหัส (Design code) ให้ข้อมูลแสดงคุณภาพ (Qualitative data) เพื่อความสะดวกในการถ่ายข้อมูลจากแบบสอบถามลงบนตัวกลาง

ข. จงกำหนดรูปแบบของเรคคอร์ด (Design record format) เพื่อบรรจุข้อมูลทั้ง 4 รายการ (item) ลงใน 1 เรคคอร์ด ใช้ข้อ ก. ประกอบการกำหนดรูปแบบของเรคคอร์ดด้วย

ค. จงกำหนดเรคคอร์ดสุดท้าย (Last record) เพื่อใช้ในการควบคุมการอ่านข้อมูลในกรณีที่เราไม่ทราบจำนวนเรคคอร์ดทั้งหมดที่มีอยู่

ง. จากเรคคอร์ดของนักศึกษาที่มีอยู่ (Student file) และใช้คำตอบข้อ ก.-ค. ประกอบคำตอบในข้อนี้ จงเขียนผังโปรแกรม (Program flowchart) เพื่อนับจำนวนนักศึกษาทั้งหมด

นักศึกษาชายคณะวิทยาศาสตร์ พิมพ์ข้อมูลทั้ง 4 รายการของนักศึกษาชายคณะวิทยาศาสตร์  
แต่ละคน พิมพ์จำนวนนักศึกษาชายคณะวิทยาศาสตร์ และจำนวนนักศึกษาทั้งหมดที่นับได้ด้วย

กำหนด Variable names ให้ข้อมูลทั้ง 4 รายการตามลำดับดังนี้

SCODE = รหัสประจำตัว

SEX = เพศ

AGE = อายุ

FAC = คณะ

กำหนด Output format

```

                                REPORT CS 105
                                1/2528
                                } head
                                MALES FROM FACULTY OF SCIENCE
                                STUDENT CODE   SEX   AGE   FACULTY
                                .
                                .

                                MALE ~ SCIENCE STUDENTS   = xx xx PERSONS
                                TOTAL STUDENTS             = xxx x PERSONS
    
```

### เฉลย

ก. ข้อมูลแสดงคุณภาพ (Qualitative data) ในที่นี้คือคำตอบเรื่องเพศ และคณะที่  
ศึกษา

กำหนดรหัสให้เพศ = 1 ถ้าเป็นชาย

= 2 ถ้าเป็นหญิง

- กำหนดรหัสให้คณะ = 1 ถ้าอยู่คณะนิติศาสตร์  
 = 2 ถ้าอยู่คณะบริหารธุรกิจ  
 = 3 ถ้าอยู่คณะมนุษยศาสตร์  
 = 4 ถ้าอยู่คณะศึกษาศาสตร์  
 = 5 ถ้าอยู่คณะวิทยาศาสตร์  
 = 6 ถ้าอยู่คณะรัฐศาสตร์  
 = 7 ถ้าอยู่คณะเศรษฐศาสตร์

ข. รูปแบบของเรคคอร์ด

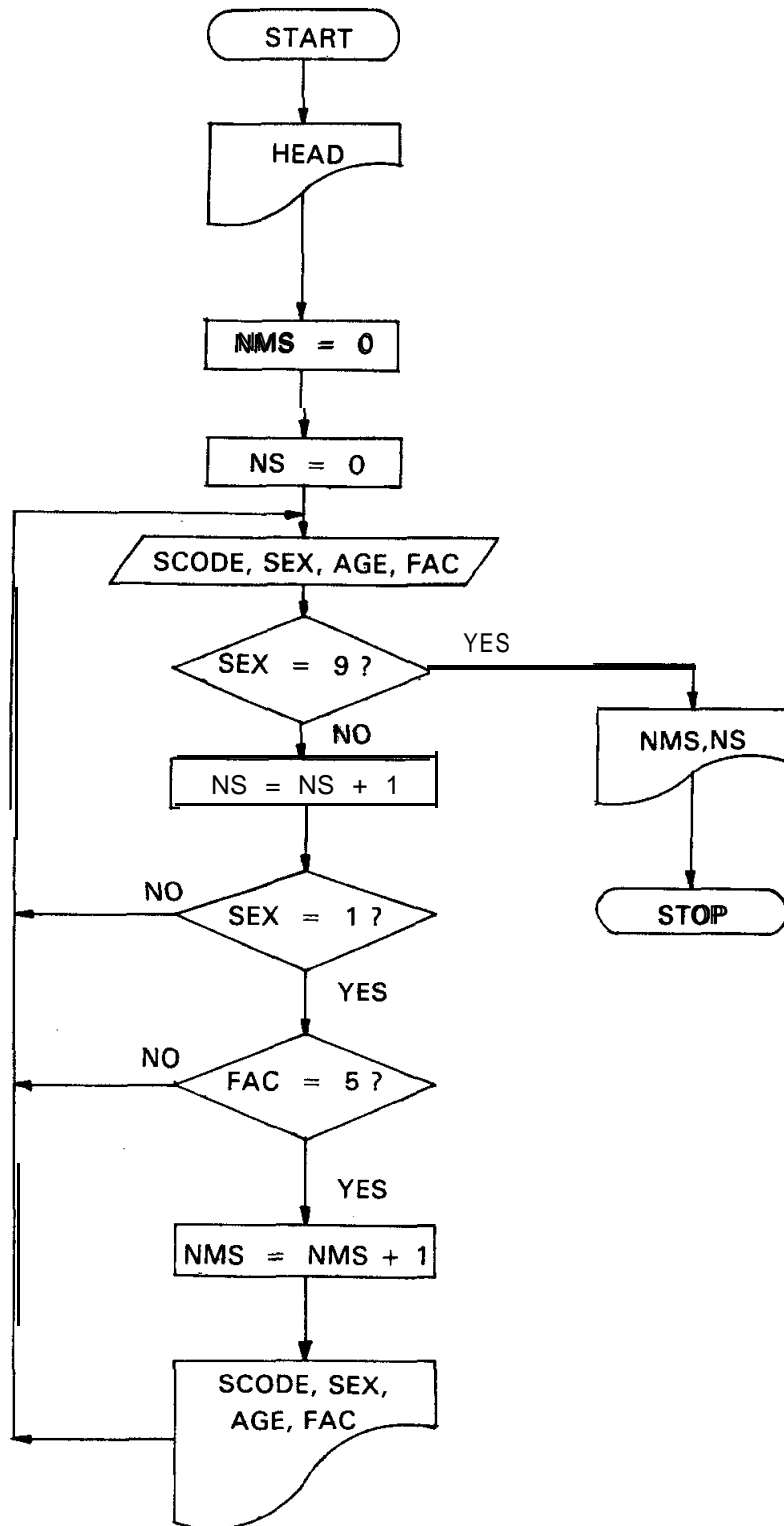
field ที่	Column	รายการข้อมูล
1	1 - 9	รหัสประจำตัว
2	10	เพศ
3	11 - 12	อายุ
4	13	คณะที่ศึกษา

หมายเหตุ field ที่ 2 และ 4 ใช้เพียง field ละ 1 Column เพื่อบันทึกข้อมูลที่เป็นรหัสที่กำหนดในข้อ ก.

ค. ให้เพศ = 9 ในเรคคอร์ดสุดท้าย นั่นคือ Column ที่ 10 ของเรคคอร์ดสุดท้ายจะบันทึกเลข 9 ไว้

ง. NS = นับจำนวนนักศึกษาทั้งหมด

NMS = นับจำนวนนักศึกษาชายคณะวิทยาศาสตร์



จ. (โจทย์เพิ่มเติม)

ถ้ามีเรคคอร์ดของนักศึกษา ซึ่งมีรูปแบบในข้อ ข. อยู่ 1170 เรคคอร์ด จงเขียนผังโปรแกรม (Program flowchart) เพื่อนับจำนวนนักศึกษารัฐกิจทั้งหมด นับจำนวนนักศึกษาคณะบริหารธุรกิจ พิมพ์ข้อมูลทั้ง 4 รายการของนักศึกษาหญิงคณะบริหารธุรกิจแต่ละคน พิมพ์จำนวนนักศึกษาหญิงคณะบริหารธุรกิจและจำนวนนักศึกษารัฐกิจทั้งหมดที่นับได้ด้วย

กำหนด Variable names

SCODE = รหัสประจำตัว

SEX = เพศ

AGE = อายุ

FAC = คณะ

กำหนด Output format

```
REPORT CS 105
          1/2528
FEMALES FROM FACULTY OF BUSINESS
STUDENT CODE  SEX  AGE  FACULTY
      .      .   .   .
      .      .   .   .
      .      .   .   .

FEMALE - BUSINESS STUDENTS = x x x x PERSONS
TOTAL BUSINESS STUDENTS   = xx xx PERSONS
```

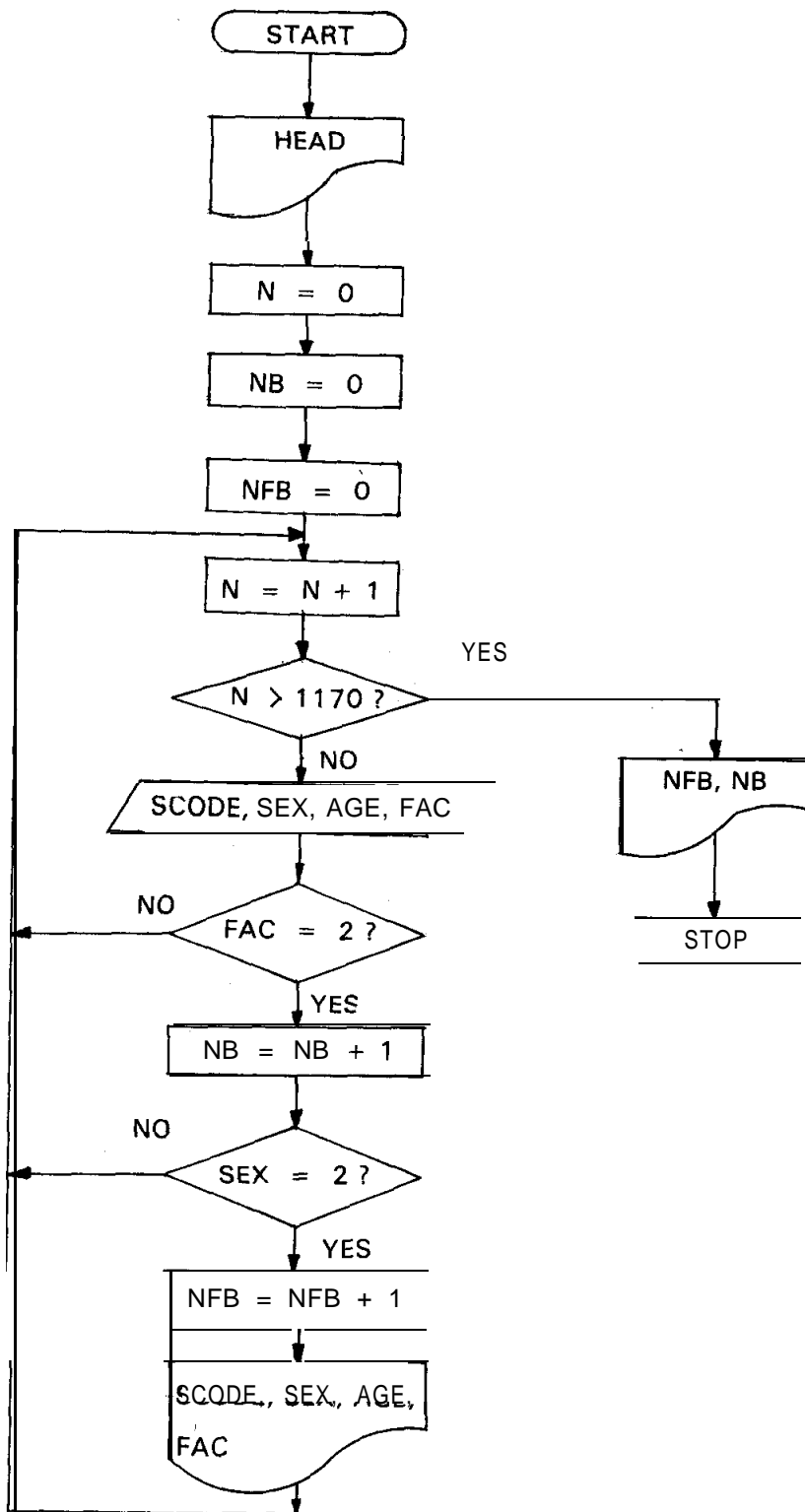
} head

เฉลย

N = นับจำนวนนักศึกษาทั้งหมด

NB = นับจำนวนนักศึกษาคณะบริหารธุรกิจ

NFB = นับจำนวนนักศึกษาหญิงคณะบริหารธุรกิจ



ตัวอย่างที่ 9 (โจทย์นี้คล้ายคลึงกับแบบฝึกหัดข้อ 4 ในแบบฝึกหัดของหัวข้อ 5.2.1 แต่ต่างกัน  
ในรายละเอียดบางอย่างใน Output format ที่ต้องการ)

9.1 ห้างสรรพสินค้าไทย ต้องการทำบัญชีเงินเดือนของพนักงานของห้าง จำนวน  
2000 คน พนักงาน 1 คน มีข้อมูลบันทึกอยู่ใน 1 เรคคอร์ด

กำหนด Input เรคคอร์ด format

field ที่	Column	รายการข้อมูล	Variable name
1	1 – 6	รหัสประจำตัวพนักงาน (Employee number)	ECODE
2	8 – 12	เงินเดือน (Salary)	SAL
3	13 – 14	จำนวนชั่วโมงที่ทำล่วงเวลา (Overtime hours)	HOUR
4	15 – 17	อัตราค่าล่วงเวลา (บาท/ชม.) (Overtime rate)	RATE

สมมติว่าพนักงานต้องเสียภาษีรายเดือน (หัก ณ ที่จ่าย) ตามอัตราดังต่อไปนี้  
ถ้ากำหนด เงินรายได้ (Income) = เงินเดือน (Salary) + เงินล่วงเวลา (Overtime)

แล้ว

- 1) เงินรายได้  $\leq$  1000 บาท ยกเว้นภาษี (TAX = 0)
- 2) เงินรายได้ 1001 – 5000 บาท เสียภาษี 3% ของรายได้
- 3) เงินรายได้ > 5000 บาท เสียภาษี 5% ของรายได้

จากสิ่งที่กำหนดให้ข้างต้น จงเขียนผังโปรแกรม (Program flowchart) เพื่อคำนวณ  
เงินที่ห้างจะต้องจ่ายให้กับพนักงานแต่ละคนภายหลังจากหักภาษีไว้แล้ว (NETPAY) นับจำนวน  
พนักงานที่มีเงินรายได้เกิน 5000 บาท และพิมพ์รายงานตาม Output format ที่กำหนดให้ต่อไปนี้

THAI DEPARTMENT STORE  
THE PAYROLL

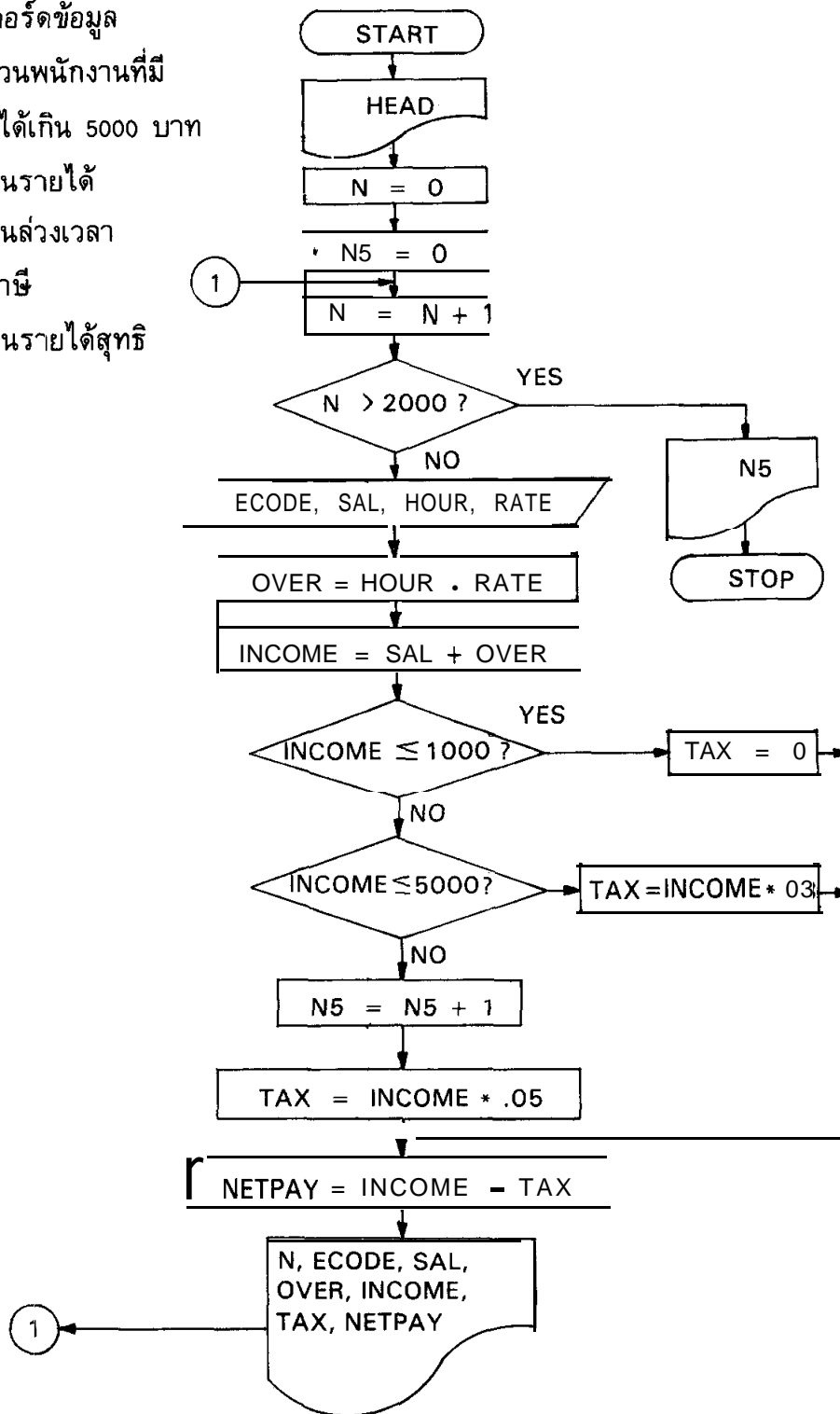
NO.	EMPLOYEE NUMBER	SALARY (BAHT)	OVERTIME (BAHT)	INCOME (BAHT)	TAX (BAHT)	NETPAY (BAHT)
. 1	-		-	-		
2	-		-	-	-	-
20'00	-	-		-	-	-

} head

NUMBER OF EMPLOYEES (INCOME>5000 BAHT) = x x x PERSONS



N = นับเรคคอร์ดข้อมูล  
 N5 = นับจำนวนพนักงานที่มี  
       เงินรายได้เกิน 5000 บาท  
 INCOME = เงินรายได้  
 OVER = เงินล่วงเวลา  
 TAX = ภาษี  
 NETPAY = เงินรายได้สุทธิ



9.2 ถ้าห้างสรรพสินค้าไทย ต้องการทำบัญชีเงินเดือนของพนักงานจำนวนหนึ่ง (สมมติว่าไม่ทราบจำนวนที่แน่นอน) พนักงานแต่ละคนมีข้อมูลบันทึกอยู่ใน 1 เรคคอร์ด ซึ่งมีรูปแบบเช่นเดียวกับใน 9.1

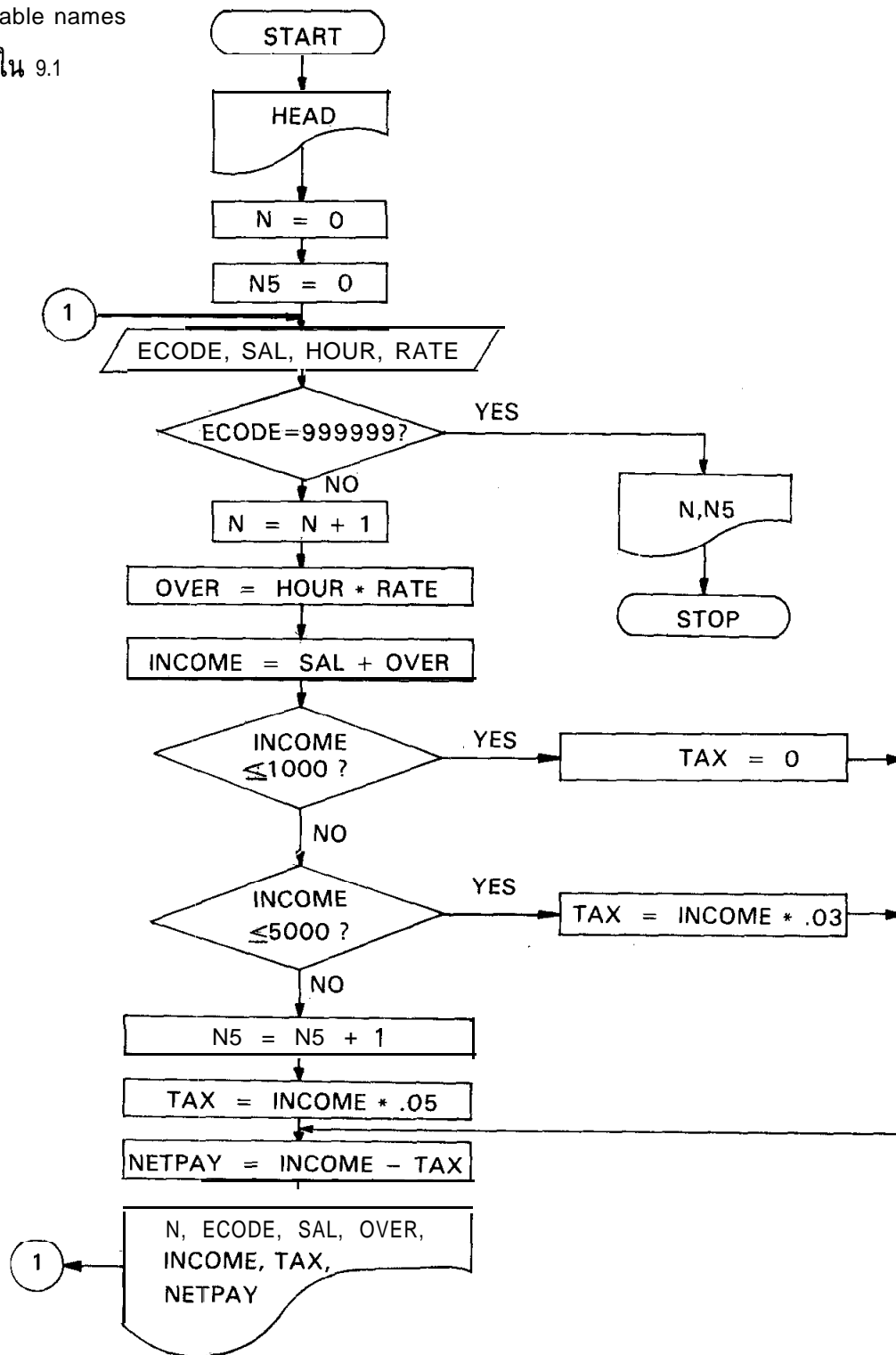
กำหนดเรคคอร์ดสุดท้าย (Last record) โดยให้ field ที่ 1 มีค่า 99999 ส่วน Column ที่เหลือเป็น blank column หมด

จากสิ่งที่กำหนดให้ข้างต้น จงเขียนผังโปรแกรม (Program flowchart) เพื่อคำนวณเงินที่ห้างจะต้องจ่ายให้กับพนักงานแต่ละคนภายหลังจากหักภาษีไว้แล้ว (NETPAY) นับจำนวนพนักงานทั้งหมด นับจำนวนพนักงานที่มีเงินรายได้เกิน 5000 บาท และพิมพ์รายงานตาม Output-format ที่กำหนดให้ต่อไปนี้

THAI DEPARTMENT STORE						
THE PAYROLL						
NO.	EMPLOYEE NUMBER	SALARY (BAHT)	OVERTIME (BAHT)	INCOME (BAHT)	TAX (BAHT)	NETPAY (BAHT)
1	.				.	.
2					.	.
					.	.
					.	.
					.	.
					.	.
					.	.
TOTAL NUMBER OF EMPLOYEES					= xxx	x PERSONS
* NUMBER OF EMPLOYEES (INCOME > 5000 BAHT)					= xxx	x PERSONS

} head

ใช้ variable names  
เหมือนใน 9.1



ตัวอย่างที่ 10 โรงพยาบาลแห่งหนึ่งเก็บข้อมูลของคนที่ยินดีจะบริจาคโลหิตให้เมื่อโรงพยาบาลต้องการ มีคนทั้งหมดที่มีรายชื่ออยู่ 2000 คน แต่ละคนมีข้อมูลอยู่ใน 1 เรคคอร์ด ซึ่งมีรูปแบบของเรคคอร์ด ดังนี้

field ที่	column	รายการข้อมูล	Data name
1	1 - 6	รหัสประจำตัว	CODE
2	7 - 38	ชื่อ นามสกุล	NAME
3	39 - 45	เบอร์โทรศัพท์	TELNO
4	46	กลุ่มเลือด	GROUP
5	47 - 48	อายุ	AGE
6	49 - 54	น้ำหนัก (mn.)	WEIGHT
7	55 - 57	ความสูง (ซม.)	HEIGHT

กำหนดรหัสให้กลุ่มเลือด 1 = เลือดกลุ่ม A  
 2 = " B  
 3 = " AB  
 4 = " O (กลุ่ม "โอ")

วันหนึ่งโรงพยาบาลต้องการเลือดกลุ่ม O จงเขียนผังโปรแกรม (ให้ใช้ Data name ที่กำหนดไว้ใน Card format) เพื่อนับและพิมพ์ record ของคนที่มีเลือดกลุ่ม O ทั้งหมด และหาน้ำหนักเฉลี่ยของคนกลุ่มนี้ด้วย

กำหนด Output format

```

                                HOSPITAL XYZ
                                BLOOD TYPE O
                                } head
NUMBER CODE NAME TEL.NO. AGE WEIGHT (KG) HEIGHT (CM)
1      :      :      :
2      :      :      :

```

TOTAL = x xx x PERSONS

AVERAGE WEIGHT = x x x.x x K.G.

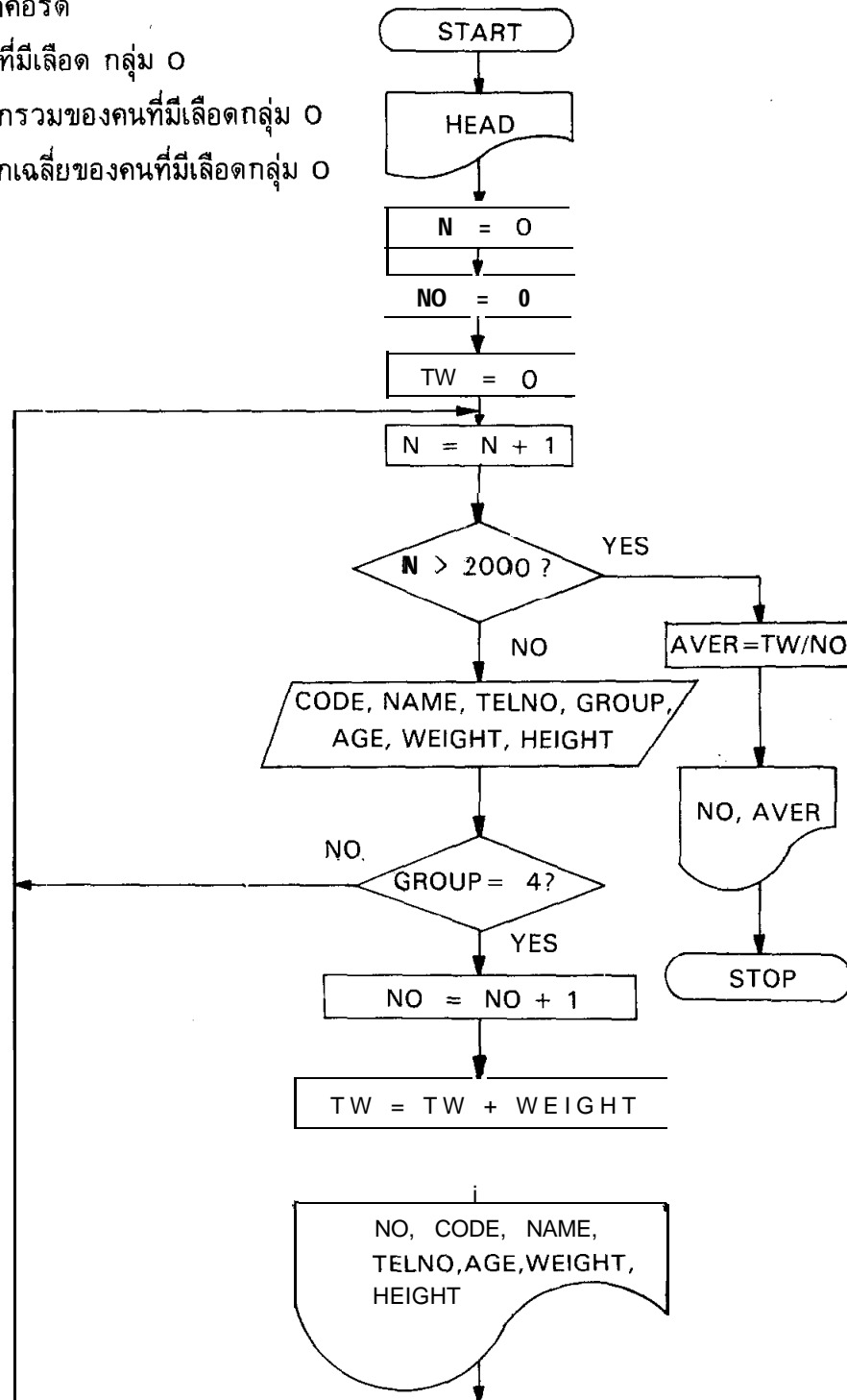
(โจทย์ข้อนี้ คล้ายคลึงกับแบบฝึกหัดข้อ 6 ในแบบฝึกหัดของหัวข้อ 5.2.1 บางส่วนเท่านั้น)

N = นับเรคคอร์ด

NO = นับคนที่มีเลือด กลุ่ม O

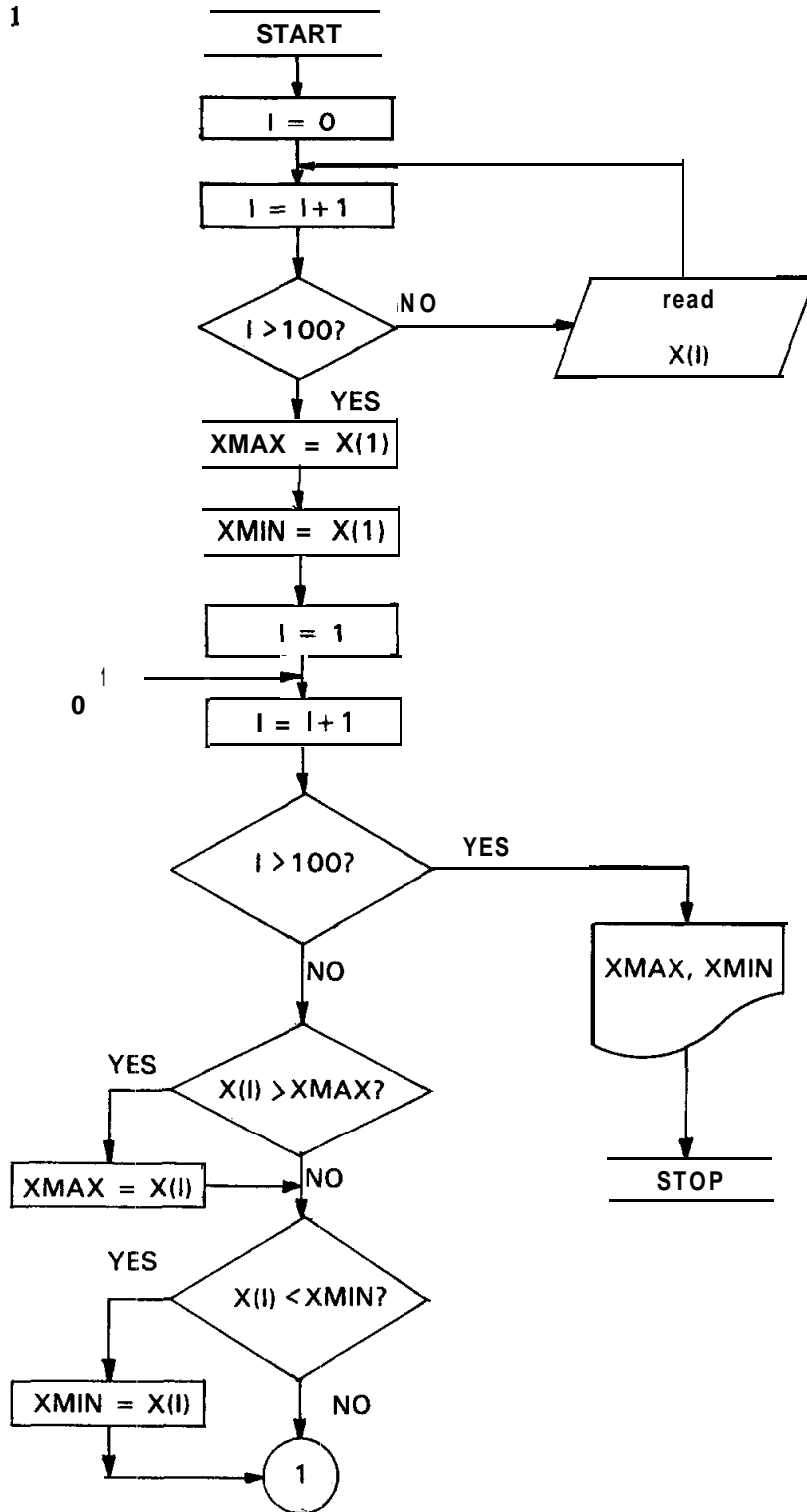
TW = น้ำหนักรวมของคนที่มีเลือดกลุ่ม O

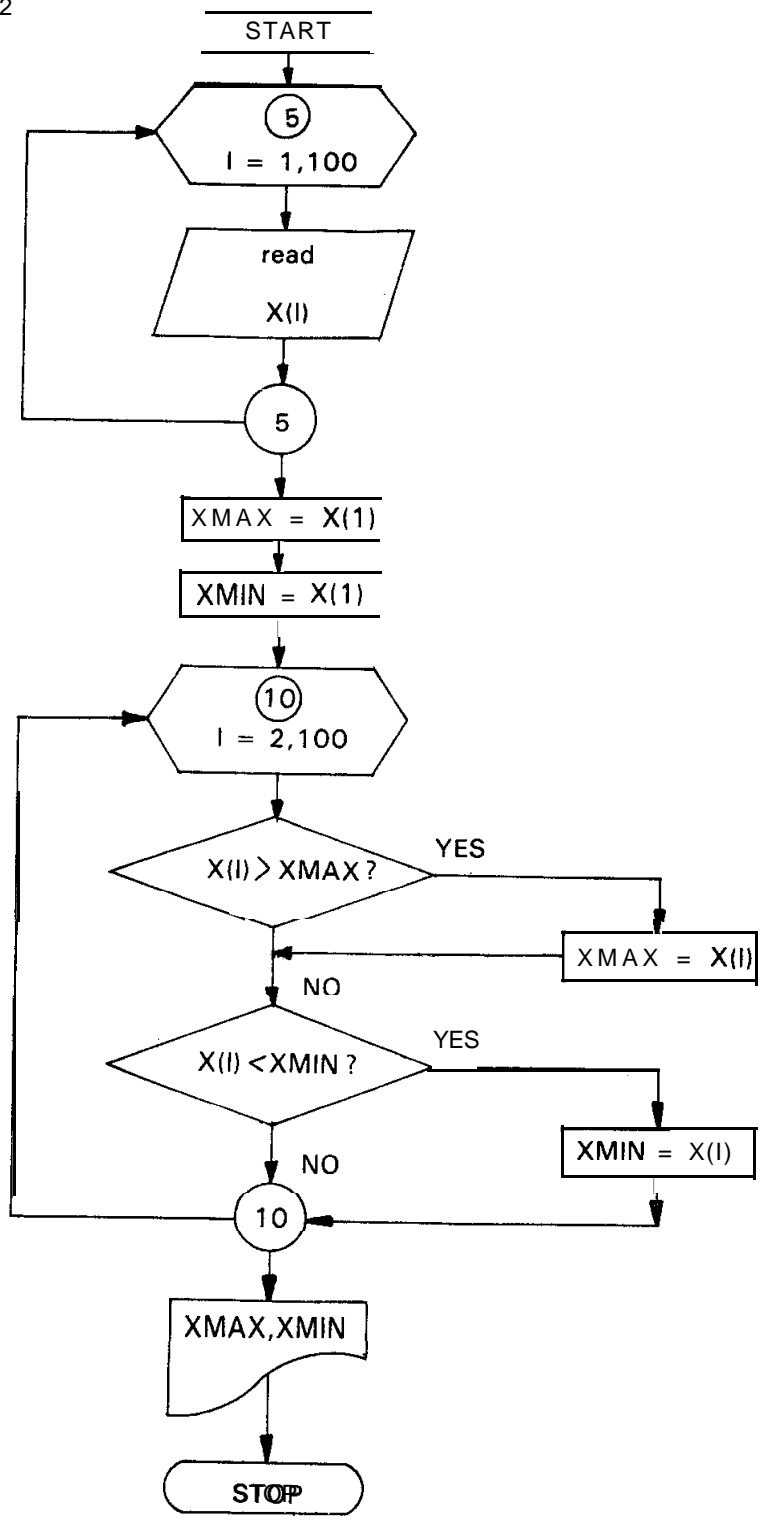
AVER = น้ำหนักเฉลี่ยของคนที่มีเลือดกลุ่ม O



ตัวอย่างที่ 11 จงเขียนผังโปรแกรมเพื่อหาและพิมพ์ค่าสูงสุดและค่าต่ำสุดของเลข 100 จำนวน  
วิเคราะห์โจทย์

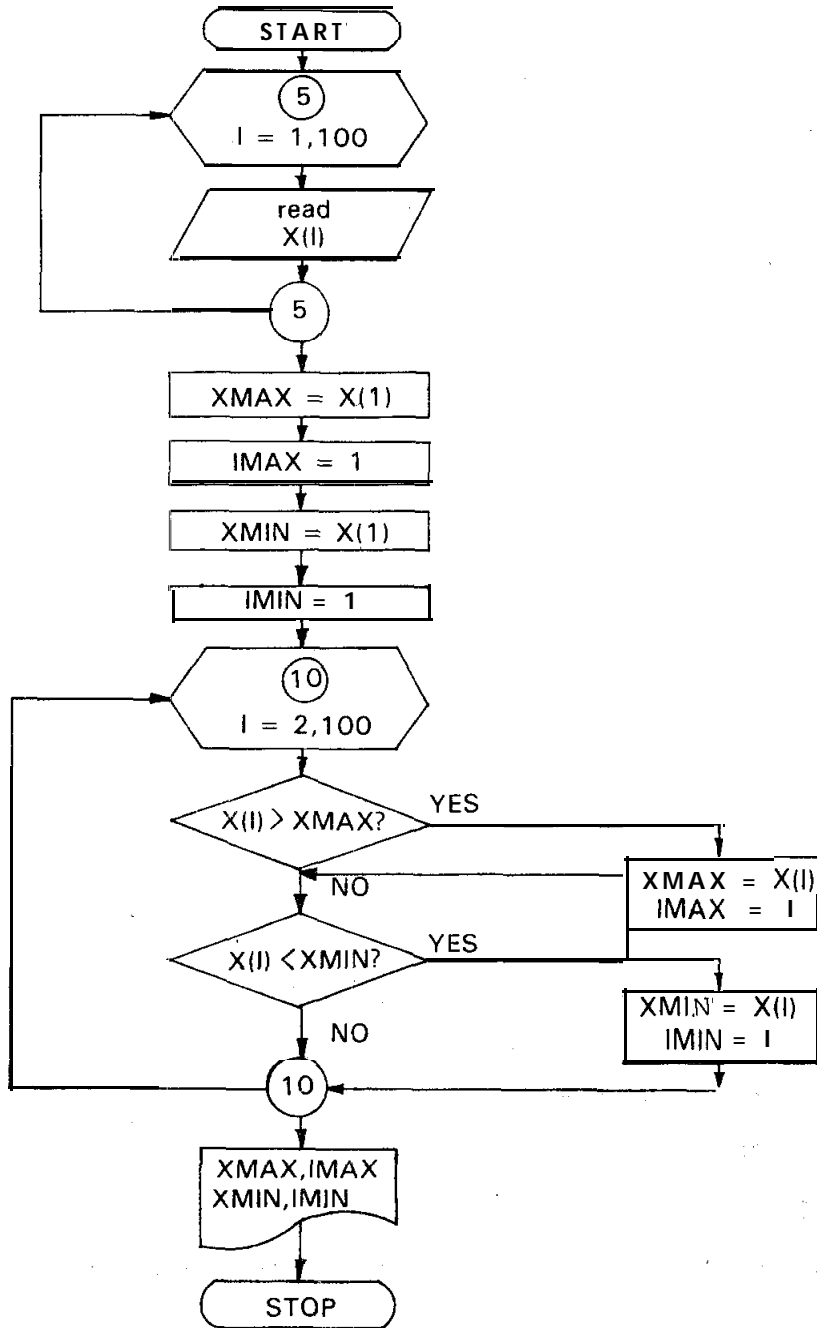
1. สิ่งที่โจทย์ต้องการ : ค่าสูงสุดและค่าต่ำสุดของเลข 100 จำนวน
2. ข้อมูลเข้า : อ่านจาก 100 เรคคอร์ดเข้าไปเก็บในแถวลำดับ 1 มิติ (one-dimensional array) ชื่อ X ซึ่งมีสมาชิก 100 ตัว คือ  $X(1), X(2), \dots, X(100)$
3. การประมวลผลข้อมูล :
  - ให้  $XMAX =$  ค่าสูงสุดของข้อมูลชุดนี้ และ  $XMIN =$  ค่าต่ำสุดของข้อมูลชุดนี้
  - กำหนด  $XMAX = X(1)$  ก่อนแล้วเปรียบเทียบ  $XMAX$  กับ  $X$  ที่เหลืออยู่ ถ้า  $X$  ตัวใดมากกว่า  $XMAX$  ให้แทนค่าของ  $XMAX$  ด้วยค่าของ  $X$  ตัวนั้น ๆ  $XMAX$  จะคงค่าสูงภายหลังการเปรียบเทียบเสมอ
  - กำหนด  $XMIN = X(1)$  ก่อน แล้วเปรียบเทียบ  $XMIN$  กับ  $X$  ที่เหลืออยู่ ถ้า  $X$  ตัวใดน้อยกว่า  $XMIN$  ให้แทนค่าของ  $XMIN$  ด้วยค่าของ  $X$  ตัวนั้น ๆ  $XMIN$  จะคงค่าต่ำภายหลังการเปรียบเทียบเสมอ
4. ข้อมูลที่ต้องแสดง ค่าของ  $XMAX$  และ  $XMIN$



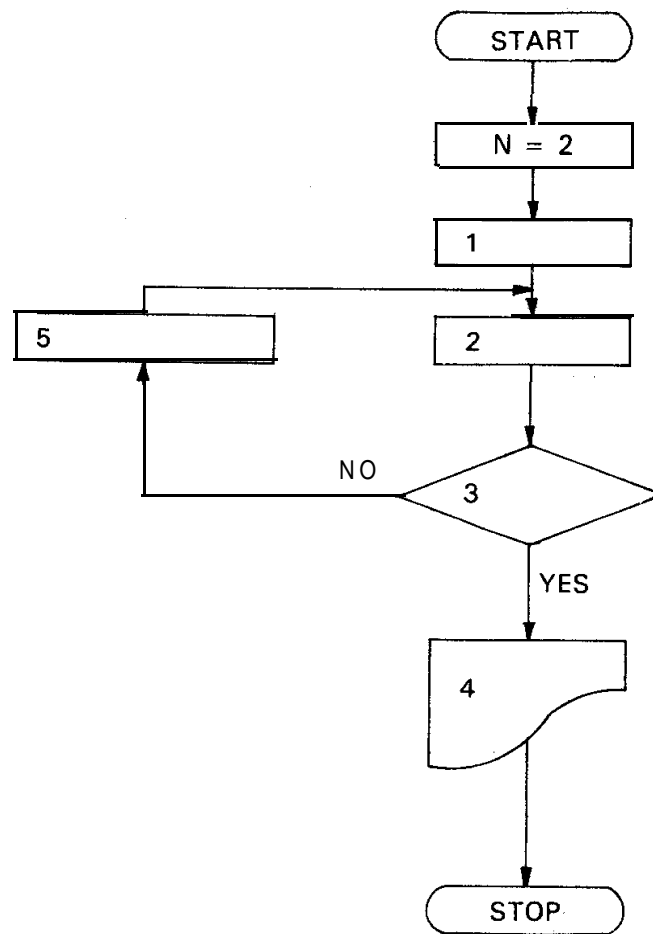




11.3 ถ้าต้องการทราบว่า  $X$  ตัวใดเป็นตัวที่มีค่าสูงสุด และ  $X$  ตัวใดเป็นตัวที่มีค่าต่ำสุด ทุกครั้งที่มีการเปรียบเทียบ  $X(I)$  กับ  $XMAX$  และ  $XMIN$  ถ้ามีการแทนค่าของ  $XMAX$  ด้วย  $X(I)$  ใด ๆ เราจะเก็บค่าของ  $I$  นั้น ๆ ไว้ด้วย โดยจะเก็บไว้ใน  $IMAX$  และถ้ามีการแทนค่าของ  $XMIN$  ด้วย  $X(I)$  ใด ๆ เราจะเก็บตัวที่ของ  $X$  คือ  $I$  ใน  $IMIN$



ตัวอย่างที่ 12 จงเติมคำสั่งลงใน Block หมายเลข 1 ถึง 5 ที่กำหนดไว้ในผังโปรแกรม (Program flowchart) เพื่อหาค่าของ  $NSUM = 2 + 4 + 6 + \dots + 20$



**เฉลยคำตอบ**

คำสั่ง  
am 1

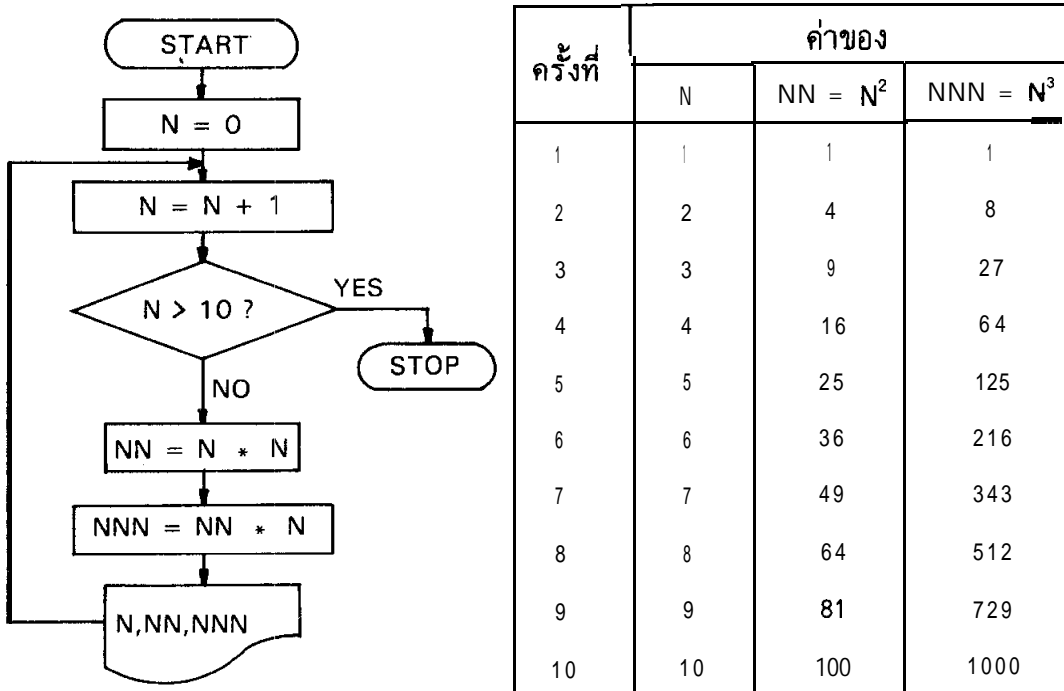
1.  $NSUM = 0$
2.  $NSUM = NSUM + N$
3.  $N = 20?$
4.  $NSUM$
5.  $N = N + 2$

คำสั่ง  
am 2

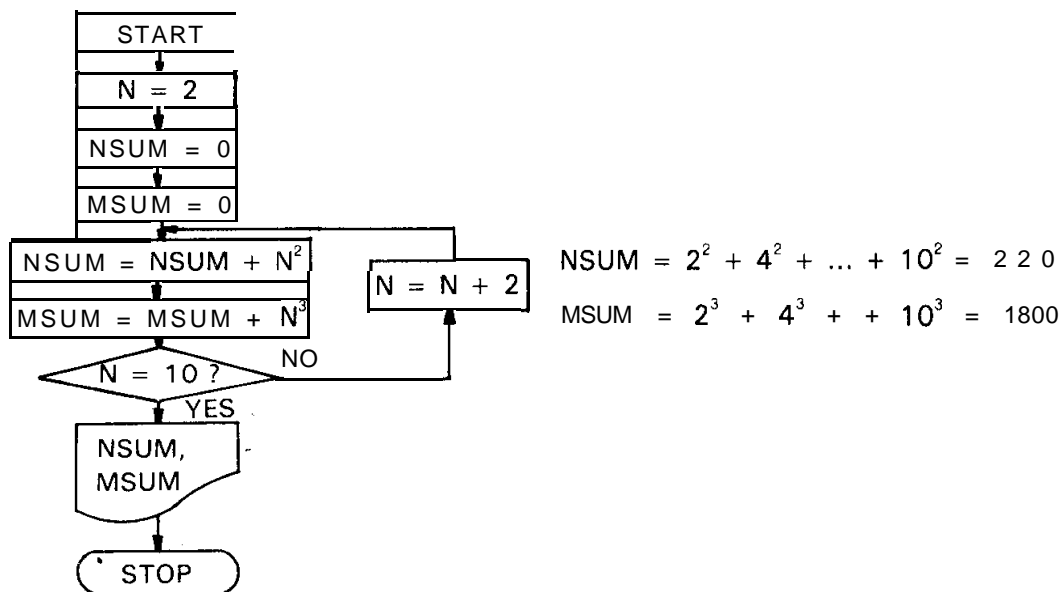
1.  $NSUM = 2$
2.  $N = N + 2$
3.  $N > 20?$
4.  $NSUM$
5.  $NSUM = NSUM + N$

## เฉลยแบบฝึกหัดของหัวข้อ 2.1

1. จากผังโปรแกรมที่กำหนดให้ จะพิมพ์ค่าของ 3 Variables N, NN และ NNN ที่ครั้งมีค่าเท่าใดบ้าง ให้แสดงตามแบบที่กำหนดให้

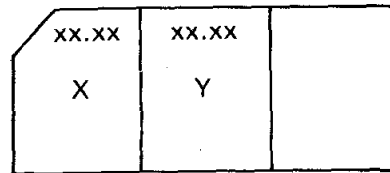


2. จากผังโปรแกรมที่กำหนดให้ จะพิมพ์ค่าของ NSUM และ MSUM เท่ากับเท่าใด



3. จงเขียนผังโปรแกรมเพื่ออ่านค่า x และ Y จากบัตร 10 บัตร

กำหนด Input card format



ถ้า  $X < Y$     ให้หาค่า     $Z = Y^2 + x$   
 ถ้า  $X > Y$     "                 $z = \frac{X}{2} - Y$   
 ถ้า  $X = Y$     "                 $Z = 2x$

กำหนด Output format

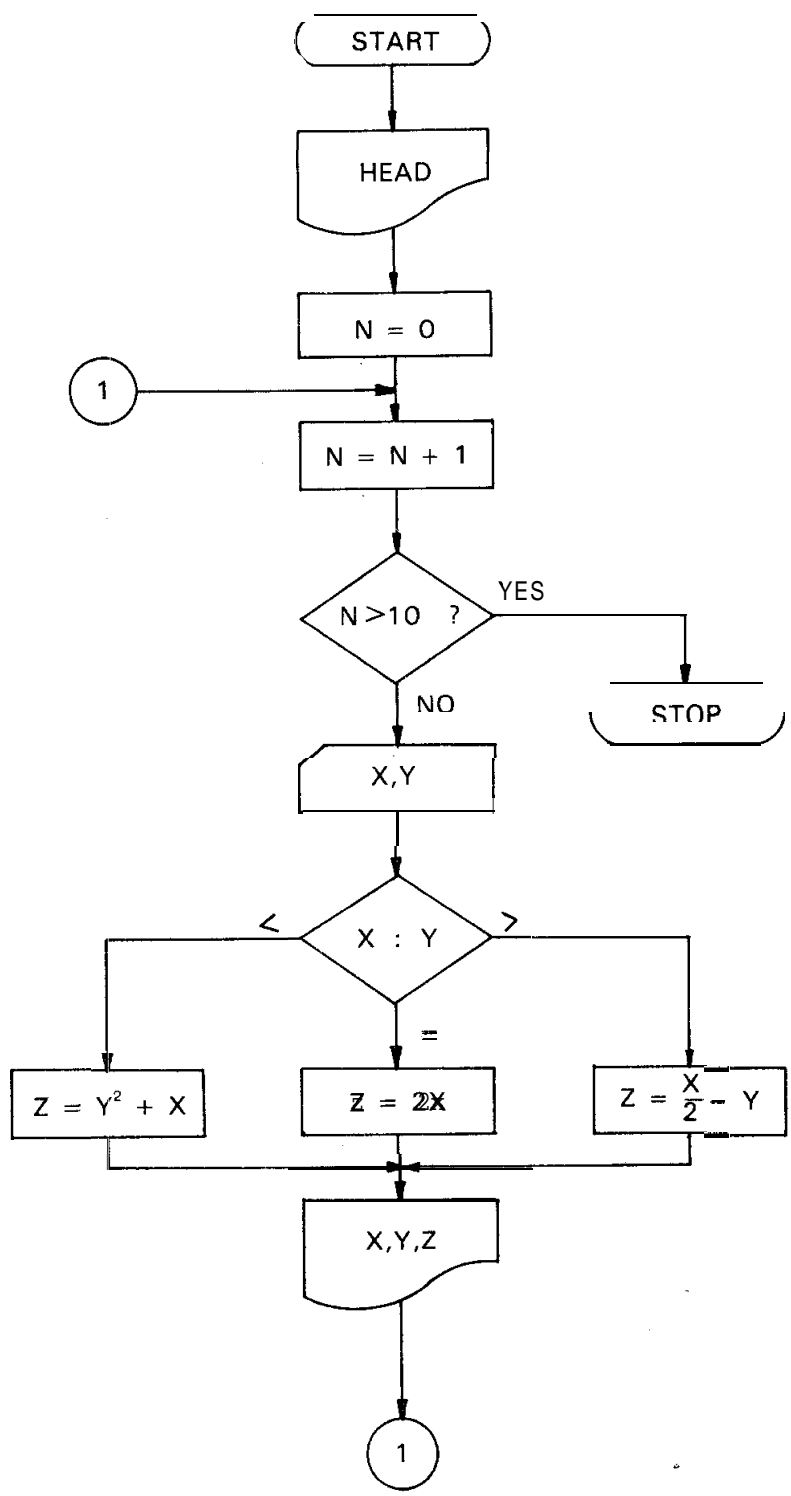
COMPUTED Z VALUES			} head
X	Y	Z	
-	-	-	
-	-	-	
-	-	-	

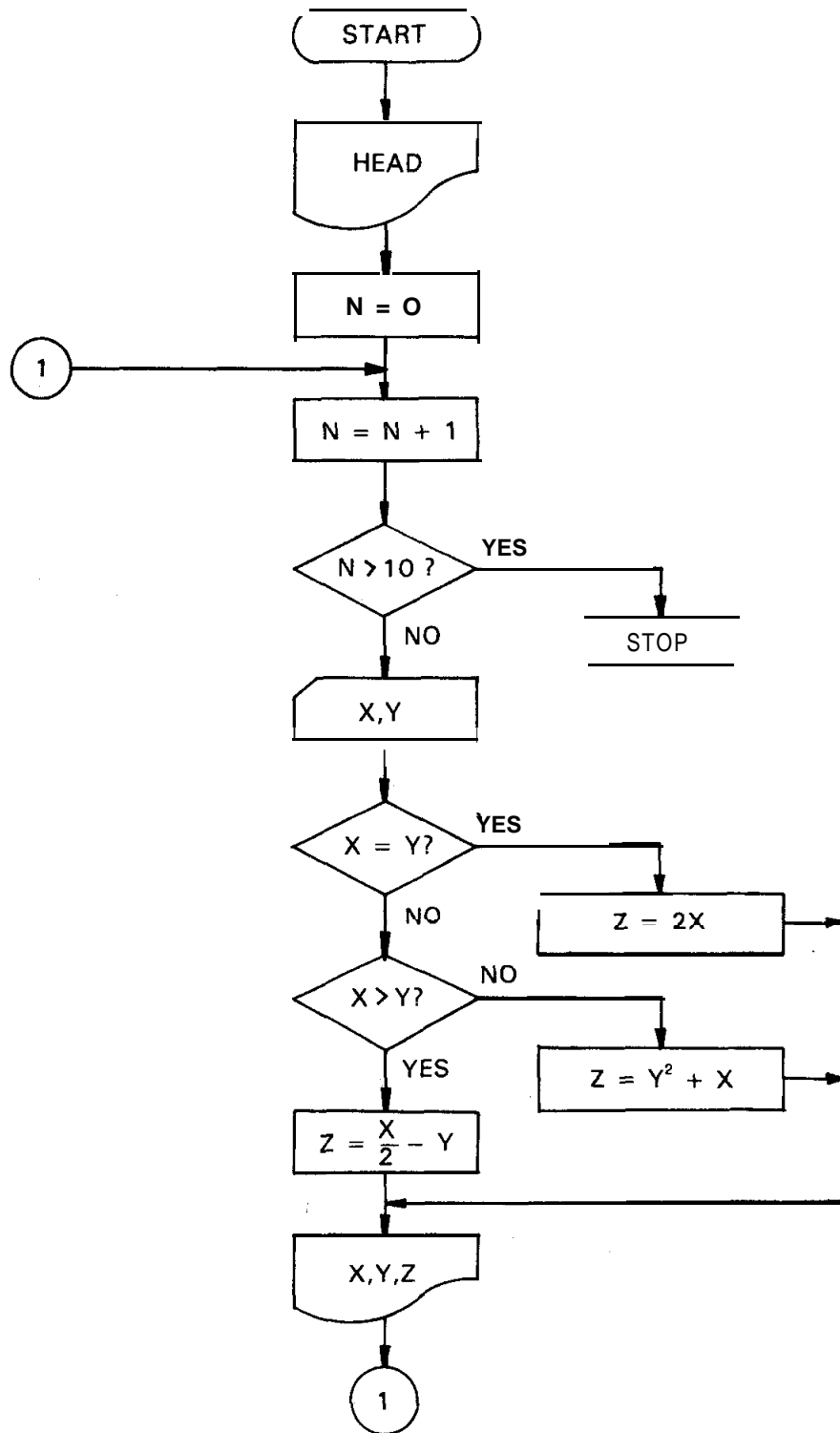
### วิเคราะห์โจทย์

1. สิ่งที่โจทย์ต้องการ : ค่า Z
2. ข้อมูลเข้า : ค่า X และ Y จากบัตร 10 บัตร
3. การประมวลผล : สูตรและเงื่อนไขของการคำนวณค่า Z จากค่า X และ Y

ที่อ่านเข้าไปเก็บใน memory

- ก. ถ้า  $X < Y$     หาค่า  $Z = Y^2 + X$
  - ข. ถ้า  $X > Y$     "     $Z = \frac{X}{2} - Y$
  - ค. ถ้า  $X = Y$     "     $Z = 2 X$
- : N ใช้ในการนับ input cards
4. ข้อมูลที่ต้องแสดง : ค่า X, Y และ Z ของแต่ละชุดของข้อมูล





4. ห้างสรรพสินค้าไทย ต้องการทำบัญชีเงินเดือนของพนักงานของห้าง 100 คน พนักงาน 1 คน มีข้อมูลบันทึกอยู่ในบัตร 1 บัตร

กำหนด Input card format

field ที่	Column	รายการข้อมูล
1	1-6	รหัสประจำตัวพนักงาน
2	8-12	เงินเดือน (Salary)
3	13-14	จำนวนชั่วโมงที่ทำล่วงเวลา
4	15-17	อัตราค่าล่วงเวลา (บาท/ชม.)

สมมติว่าพนักงานทุกคนต้องเสียภาษีรายเดือนตามอัตราต่อไปนี้  
เงินรายได้ (INCOME) น้อยกว่าหรือเท่ากับ 1000 บาท ยกเว้นภาษี

1001 - 5000 บาท เสีย 3% ของรายได้

มากกว่าหรือเท่ากับ 5001 บาท เสีย 5% ของรายได้

(เงินรายได้ = เงินเดือน + เงินล่วงเวลา)

จงเขียนผังโปรแกรม เพื่อคำนวณเงินที่ห้างจะต้องจ่ายแก่พนักงานแต่ละคนภายหลังจากหักภาษีไว้แล้ว

กำหนด Output format

```

                THAI DEPARTMENT STORE
                THE PAYROLL
EMPLOYEE NUMBER  INCOME(BAHT)  TAX(BAHT)  NETPAY(BAHT)
    :                :                :                :
    } head
    
```

วิเคราะห์โจทย์

1. สิ่งที่โจทย์ต้องการ : คำนวณเงินที่ห้างจะต้องจ่ายแก่พนักงานแต่ละคนภายหลังจากหักภาษีไว้แล้ว

2. ข้อมูลเข้า : ข้อมูล 4 รายการให้ตั้ง Data name ว่า ECODE, SAL, HOUR และ RATE ตามลำดับ field

: ที่มาของข้อมูล อ่านจากบัตร

3. วิธีการประมวลผล : สูตร ก) เงินรายได้ = เงินเดือน + เงินล่วงเวลา

ข) เงินล่วงเวลา = HOUR \* RATE

: Variable ที่เกิดขึ้นระหว่างการประมวลผล

ก) INCOME = เงินรายได้

= SAL + HOUR \* RATE

(ในที่นี้เราไม่ต้องการพิมพ์จำนวนเงินล่วงเวลาใน output จึงไม่จำเป็นต้องเก็บผลลัพธ์ของ HOUR \* RATE ไว้)

ข) TAX = เงินภาษีที่จะหัก ณ ที่จ่าย

ค) NETPAY = เงินรายได้สุทธิ

= INCOME - TAX

: เงื่อนไขในการคำนวณภาษี

n) ถ้า  $INCOME \leq 1000$ , TAX = 0

ข) ถ้า  $1001 \leq INCOME \leq 5000$ , TAX = INCOME \* .03

ค) ถ้า  $INCOME \geq 5001$ , TAX = INCOME \* .05

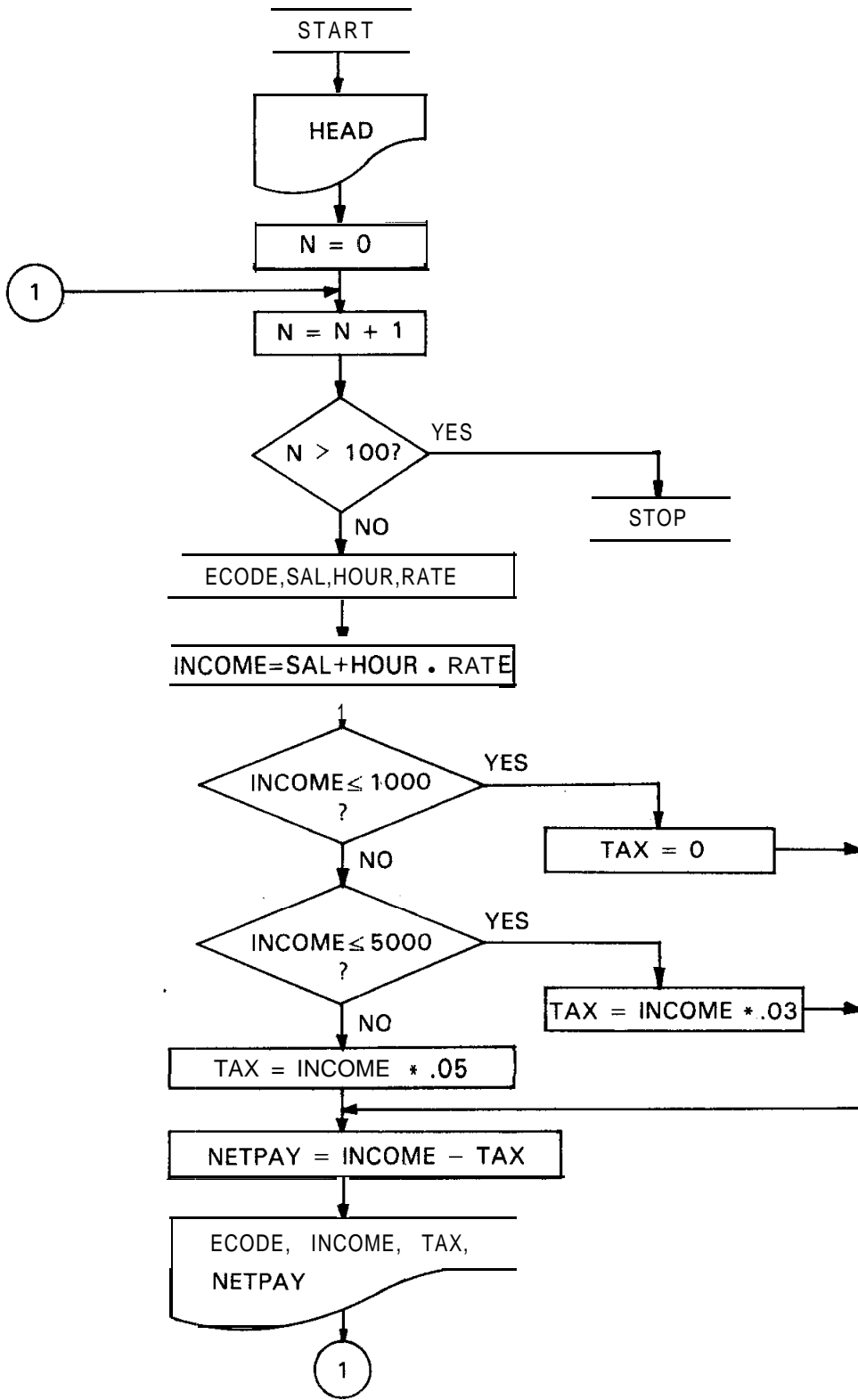
: การนับ นับบัตรที่ต้องทำการประมวลผล ซึ่งมีอยู่ 100 บัตร

ใช้ Variable N ในการนับบัตร

4. ข้อมูลที่ต้องแสดง

พิมพ์ ECODE, INCOME, TAX และ NETPAY ของพนักงานแต่ละคน





5. จงเขียนผังโปรแกรมเพื่อ

1. นับจำนวนนักเรียนที่สอบผ่านทั้ง 3 วิชา (คะแนน : คะแนน  $\geq 60$ ) และได้คะแนนเฉลี่ยมากกว่า 80 คะแนน

2. พิมพ์รายการของนักเรียนที่มีลักษณะตามข้อ 1. พร้อมทั้งคะแนนรวม 3 วิชา และคะแนนเฉลี่ยของแต่ละคนด้วย

กำหนดให้มีนักเรียนทั้งหมด 1000 คน (นักเรียน 1 คนมีข้อมูลบันทึกอยู่ในบัตร 1 บัตร)

กำหนด Card format

field ที่	Column	รายการข้อมูล
1	1-5	รหัสประจำตัว
2	7-11	คะแนนคณิตศาสตร์
3	13-17	คะแนนสังคม
4	18-23	คะแนนภาษาอังกฤษ

กำหนด Output format

```
SCORE REPORT  
STUDENT NUMBER MATH SOCIAL ENGLISH TOTAL AVERAGE  
.  
.  
TOTAL = XXXX PERSONS
```

} head

วิเคราะห์โจทย์

1. สิ่งที่โจทย์ต้องการ : นับและพิมพ์รายการของนักเรียนที่สอบผ่านทั้ง 3 วิชาและได้คะแนนเฉลี่ยเกิน 80 คะแนน

2. ข้อมูลเข้า : ตั้ง Variable สำหรับข้อมูล 4 รายการตามลำดับ field ดังนี้ SCODE, MATH, SOC, และ ENG

: ที่มาของข้อมูล อ่านจากบัตร

3. การประมวลผล : คำนวณหาคะแนนรวมและคะแนนเฉลี่ย

Variable ที่เกิดขึ้นระหว่างการประมวลผล

ก) TOTAL = คะแนนรวม 3 วิชา

ข) AVER = คะแนนเฉลี่ยของ 3 วิชา

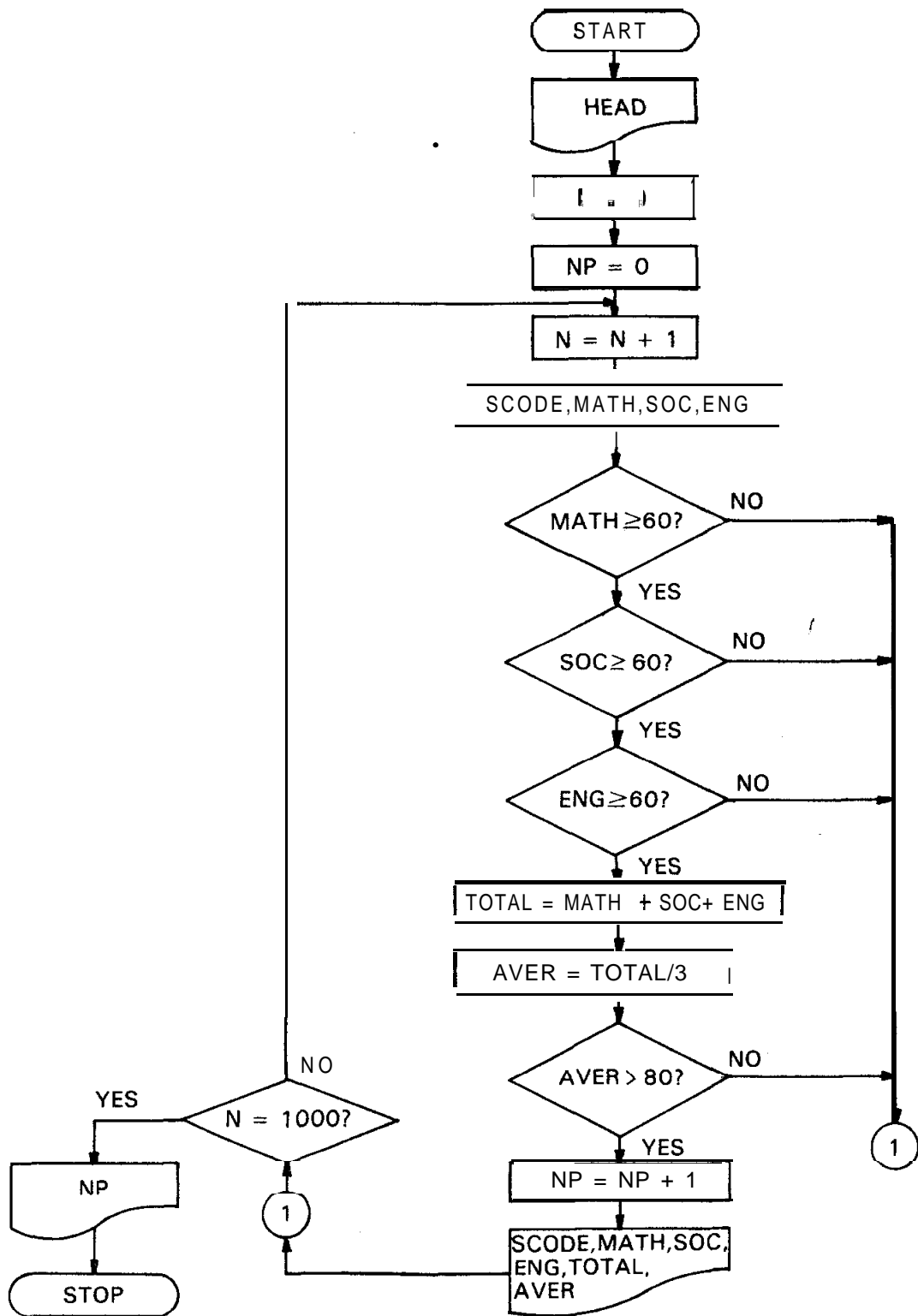
: เงื่อนไขในการประมวลผล

ผ่านทั้ง 3 วิชา (คะแนนแต่ละวิชา  $\geq 60$ ) และคะแนนรวมเกิน 80 คะแนน แล้วจึงจะนับ ใช้ NP นับจำนวนนักเรียนตามเงื่อนไขนี้

: การนับ ใช้ N ในการนับบัตรที่มีอยู่ 1000 บัตร

4. ข้อมูลที่ต้องแสดง

พิมพ์ SCORE, MATH, SOC, ENG, TOTAL และ AVER ของนักเรียนที่สอบผ่านทั้ง 3 วิชา และได้คะแนนเฉลี่ยเกิน 80 คะแนน และพิมพ์จำนวนนักเรียนที่นับได้ใน NP ด้วย



6. โรงพยาบาลแห่งหนึ่งเก็บข้อมูลของคนที่ยืนดีจะบริจาคโลหิตให้เมื่อโรงพยาบาลต้องการคนทั้งหมด มี 1500 คน แต่ละคนมีข้อมูลอยู่ในบัตร 1 บัตร โดยการกำหนดรหัสของกลุ่มเลือดดังนี้

Group A = 1                      Group B = 2  
 Group AB = 3                    Group O = 4

กำหนด Card format

field ที่	Column	รายการข้อมูล
1	1-6	รหัสประจำตัว (Code)
2	6	Group เลือด
3	10-11	อายุ
4	13-17	น้ำหนัก
5	19-24	ความสูง
6	26-32	เบอร์โทรศัพท์ที่ติดต่อได้ ทุกเวลา (สมมุติว่าทุกคนมี)

วันหนึ่งโรงพยาบาลต้องการเลือด Group O (โอ) จึงเขียนผังโปรแกรมเพื่อเลือกพิมพ์-record ของคนที่มีเลือด Group O และนับจำนวนคนที่มีเลือด Group O ทั้งหมดด้วย

กำหนด Output format

```

                                HOSPITAL XY
                                BLOOD TYPE O
คนที่
↓
NUMBER CODE AGE WEIGHT HEIGHT TELEPHONE—NUMBER
:         :         :         :         :         :
TOTAL = XXXX PERSONS
    
```

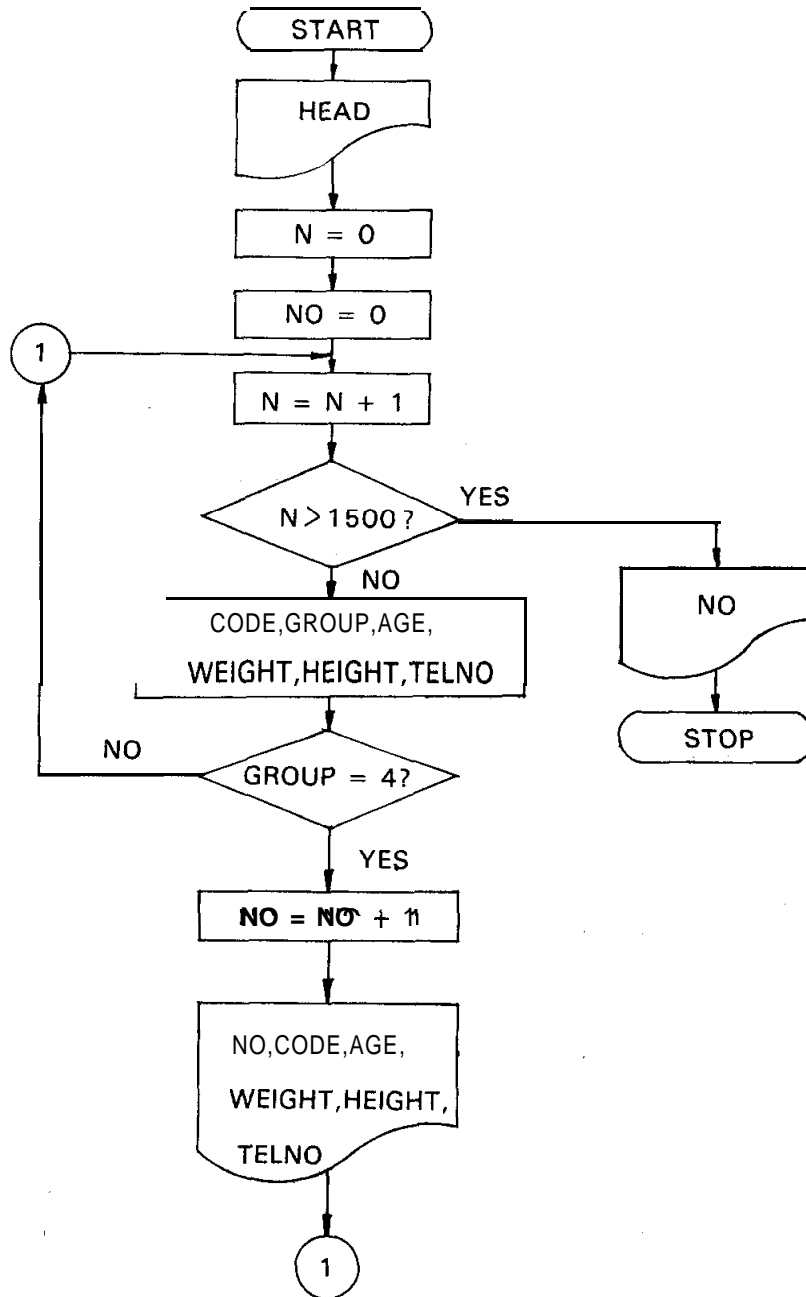
} head

วิเคราะห์โจทย์

1. สิ่งที่โจทย์ต้องการ : จำนวนและรายชื่อของคนที่มีเลือด Group O
2. ข้อมูลเข้า : ข้อมูล 6 รายการจากบัตร ตั้ง Variable ตามลำดับ field ดังนี้ CODE, GROUP, AGE, WEIGHT, HEIGHT, และ TELNO

3. การประมวลผล : จำแนกคนที่มึเลือด Group O นั่นคือ record ที่มี GROUP = 4 ใช้ Variable NO ในการนับคนกลุ่มนี้ และใช้ N ในการนับบัตรทั้งหมด

4. ข้อมูลที่ต้องแสดง : พิมพ์คนที่ (NO), CODE, AGE, WEIGHT, HEIGHT และ TELNO ของคนที่มึเลือด Group O และพิมพ์จำนวนคนที่มึเลือด Group O ด้วย



7. จงเขียนผังโปรแกรมเพื่อที่จะคำนวณพื้นที่ของสามเหลี่ยมและพิมพ์ผลลัพธ์ออกมา โดยที่ให้เครื่องอ่านข้อมูลเข้าคือด้านทั้ง 3 ด้านของสามเหลี่ยมเข้าไปเก็บไว้ที่ชื่อ A, B, C ก่อนหาพื้นที่ของสามเหลี่ยมต้องตรวจก่อนว่าด้านทั้ง 3 ที่กำหนดให้เป็นด้านทั้ง 3 ของสามเหลี่ยมจริง (นั่นคือ 2 ด้านใด ๆ รวมกันย่อมยาวกว่าด้านที่เหลือ)

$$\text{กำหนดสูตร : พื้นที่ของสามเหลี่ยม} = \sqrt{S(S - A)(S - B)(S - C)}$$

$$\text{โดยที่ } S = \frac{A + B + C}{2}$$

สมมุติบัตรใบสุดท้ายมีค่า A = 0 และถ้าด้านทั้ง 3 ไม่เป็นด้านของสามเหลี่ยม ให้พิมพ์

'NOT A TRIANGLE' ที่ column REMARK

กำหนด Output format

#### AREA OF TRIANGLES

NO	A	B	C	AREA	REMARK
⋮	.	⋮	⋮	⋮	⋮

#### วิเคราะห์โจทย์

1. สิ่งที่โจทย์ต้องการ : หาพื้นที่ของสามเหลี่ยมที่กำหนดด้านทั้ง 3 ให้

2. ข้อมูลเข้า : ด้าน 3 ด้าน คือ A, B และ C

: อ่านข้อมูลจากบัตร

3. การประมวลผล : Variable ที่เกิดขึ้นระหว่างประมวลผล และสูตร

$$\text{ก) } S = \frac{A + B + C}{2}$$

$$\text{ข) } \text{AREA} = \sqrt{S(S - A)(S - B)(S - C)}$$

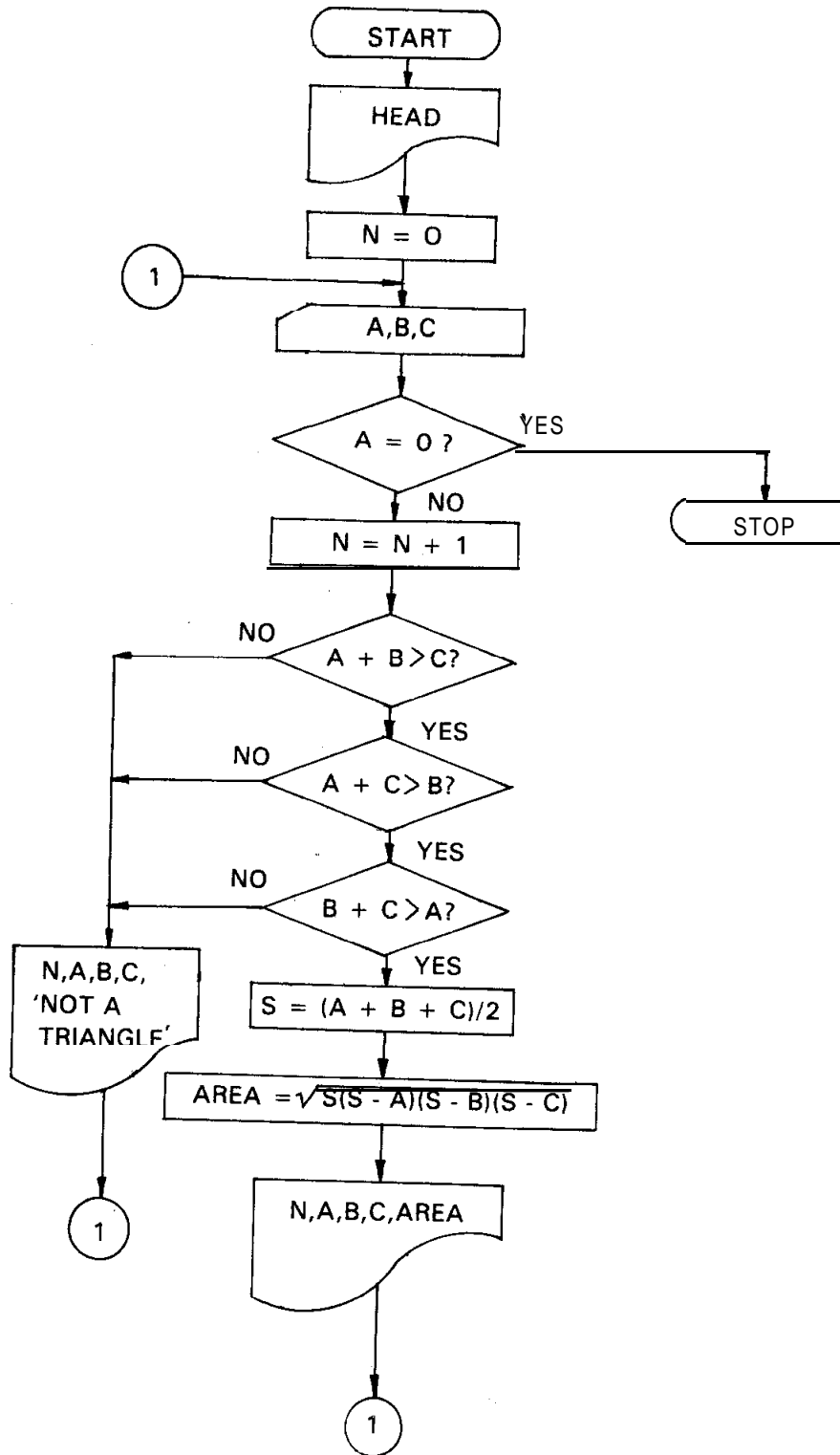
: เงื่อนไขก่อนคำนวณหาพื้นที่ของสามเหลี่ยม คือตรวจ

สอบก่อนว่าด้านทั้ง 3 เป็นด้านของสามเหลี่ยม นั่นคือ  $A + B > C$  และ  $B + C > A$  และ  $A + C > B$

: N ใช้นับบัตร

4. ข้อมูลที่ต้องแสดง

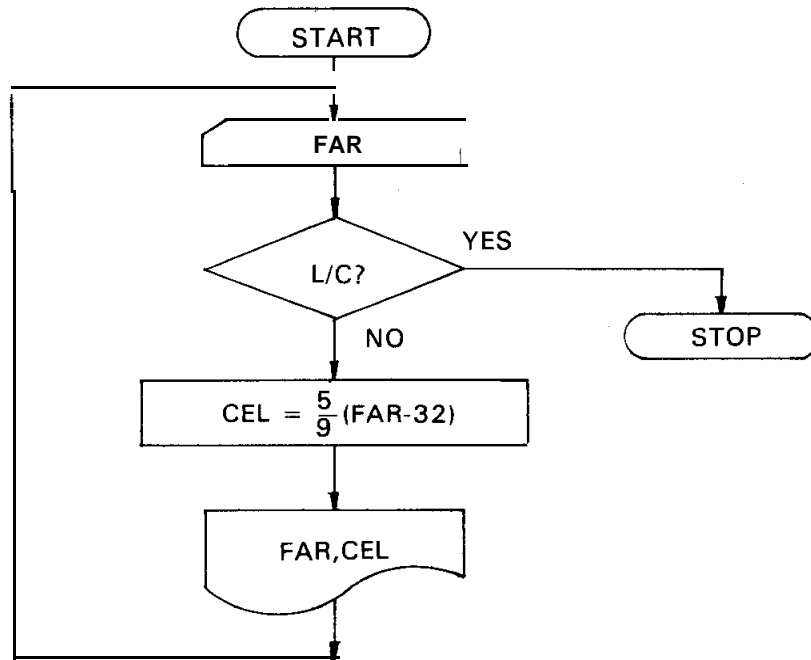
พิมพ์ชุดที่ (N), A, B, C, AREA หรือคำอธิบายใน REMARK นั่นคือ ถ้าด้านทั้ง 3 เป็นด้านของสามเหลี่ยม คำนวณหา AREA และพิมพ์ค่าของ AREA แต่ถ้าด้านทั้ง 3 ไม่เป็นด้านของสามเหลี่ยม ไม่คำนวณหา AREA แต่พิมพ์ 'NOT A TRIANGLE' ที่ REMARK





8. จงเขียนผังโปรแกรมในการที่จะอ่านอนุกรมเป็นองศาฟาเรนไฮต์และแปลงผลลัพธ์ที่ได้ออกมาเป็นองศาเซลเซียส และพิมพ์ผลออกมา

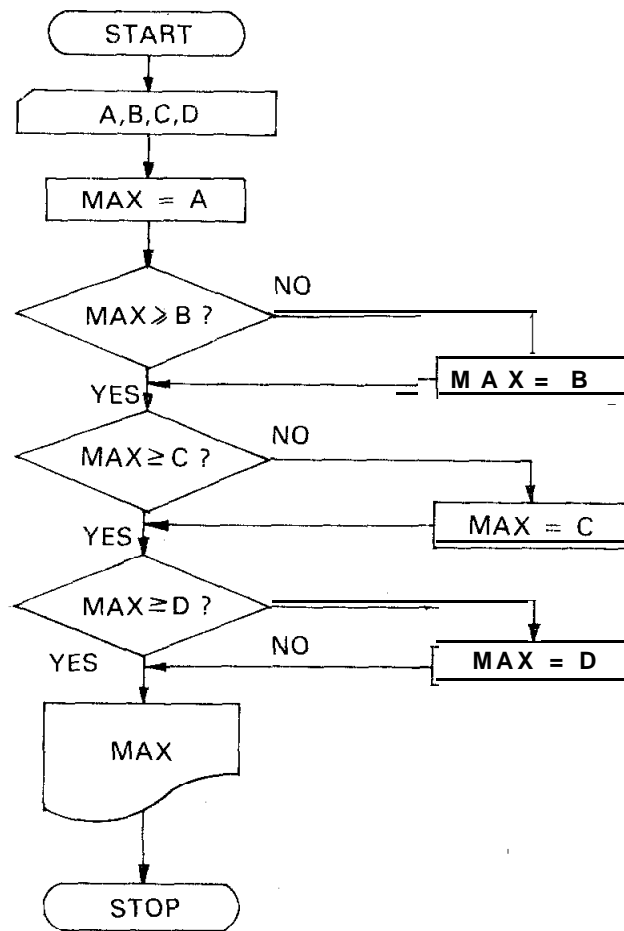
กำหนดให้ FAR = องศาฟาเรนไฮต์  
 CEL = องศาเซลเซียส  
 L/C = last Card



9. จงเขียนผังโปรแกรมในการอ่านข้อมูลเข้า 4 ค่าเข้าไปเก็บที่ A,B,C,D แล้วให้หาค่าสูงสุดของข้อมูลชุดนี้ แล้วพิมพ์ผลลัพธ์ที่ได้ออกมา

วิเคราะห์โจทย์

1. สิ่งที่โจทย์ต้องการ : ค่าสูงสุดของข้อมูลชุดนี้
2. ข้อมูลเข้า : ค่า A,B,C และ D จากบัตร
3. การประมวลผล : MAX = ค่าสูงสุดของข้อมูลชุดนี้  
 : กำหนด MAX = A ก่อนแล้วเปรียบเทียบ MAX กับข้อมูล  
 ตัวอื่น โดยให้ MAX คงค่าสูงอยู่เสมอภายหลังการเปรียบเทียบ
4. ข้อมูลที่ต้องแสดง : ค่าของ MAX



10. บริษัทแห่งหนึ่งจะจ่ายเงินโบนัสให้แก่คนงานโดยคิดจากเงินเดือนและยอดขายในแต่ละปี การจ่ายจะจ่ายเป็นไปตามหลักเกณฑ์ดังนี้

- เงินโบนัสจะเป็น 3% ของยอดขาย ถ้ายอดขาย  $\leq 4$  เท่าของเงินเดือน
- เงินโบนัสจะเป็น 4% ของยอดขาย ถ้ายอดขาย  $> 4$  เท่าของเงินเดือน  
(แต่ไม่ถึง 6 เท่าของเงินเดือน)

- เงินโบนัสจะเป็น 5% ของยอดขาย ถ้ายอดขาย  $\geq 6$  เท่าของเงินเดือน

จงเขียนผังโปรแกรมตั้งแต่เริ่มอ่านข้อมูลจากบัตร (ข้อมูลคือรหัสของพนักงานขาย, เงินเดือน และยอดขายทั้งปีของพนักงานขายแต่ละคน) จนกระทั่งคำนวณหาโบนัสสิ้นปี และพิมพ์ผลออกมา

กำหนดบัตรใบสุดท้าย : ให้ ECODE = 0

กำหนด Output format

BONUS REPORT				
EMPLOYEE	CODE	SALARY	AMOUNT SOLD	BONUS
:	:	:	:	:

} head

### วิเคราะห์โจทย์

1. สิ่งที่โจทย์ต้องการ : เงินโบนัสของพนักงานขายแต่ละคน
2. ข้อมูลเข้า : ข้อมูล 3 รายการจากบัตร ตั้ง Variable ตามลำดับ field

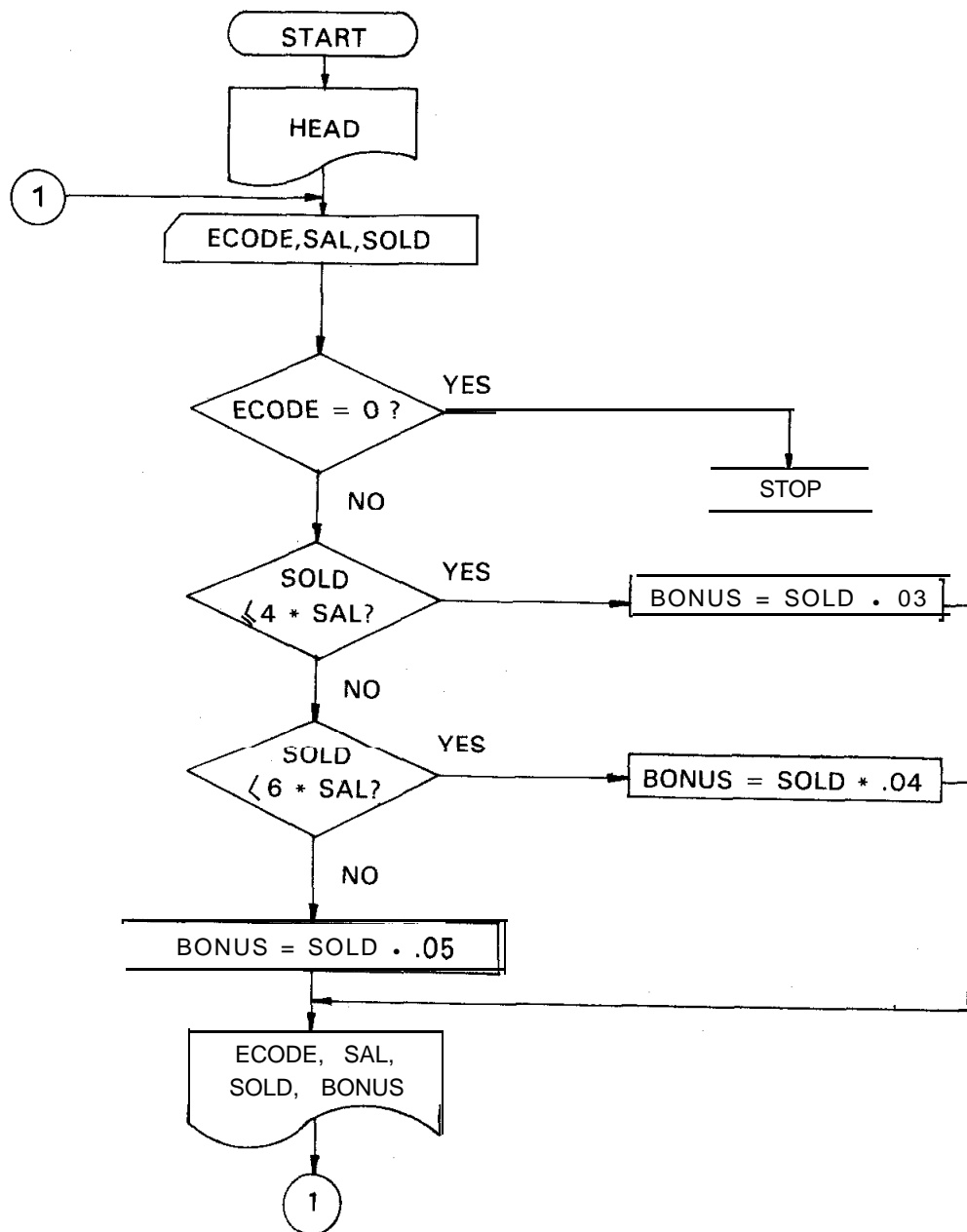
ดังนี้ ECODE, SAL, และ SOLD

3. การประมวลผล : Variable ที่เกิดขึ้นในขั้นตอนการประมวลผล  
BONUS = เงินโบนัส  
: เงื่อนไขในการคำนวณเงินโบนัส

- ก)  $SOLD < 4 * SAL$  คิด  $BONUS = SOLD * .03$   
ข)  $4 * SAL < SOLD < 6 * SAL$  "  $BONUS = SOLD * .04$   
ค)  $SOLD > 6 * SAL$  "  $BONUS = SOLD * .05$

4. ข้อมูลที่ต้องแสดง

พิมพ์ ECODE, SAL, SOLD, BONUS ของพนักงานขายแต่ละคน



11. จาก Quadratic form :  $ax^2 + bx + c = 0$

จงเขียนผังโปรแกรมให้เครื่องอ่านสัมประสิทธิ์  $a, b, c$  แล้วคำนวณหาค่าของ  $X$  และพิมพ์ผลออกมาโดยใช้สูตร

$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

หมายเหตุ ถ้าค่าของ  $(b^2 - 4ac)$  เป็นบวก, X จะมี 2 ค่า

ถ้าค่าของ  $(b^2 - 4ac)$  เป็นลบ ให้พิมพ์ผลว่า 'THE ROOTS ARE COMPLEX' ที่ REMARK  
และ ถ้าค่าของ  $(b^2 - 4ac)$  เป็นศูนย์, X จะมี 2 ค่าที่เท่ากัน คือ  $-\frac{b}{2a}$

กำหนดบัตรใบสุดท้าย : A < 0

กำหนด Output format

QUADRATIC EQUATIONS						
NO	A	B	C	ROOT 1	ROOT 2	REMARK
:	:	:	:	:	:	:

} head

### วิเคราะห์โจทย์

- สิ่งที่โจทย์ต้องการ : รากทั้ง 2 ของสมการกำลังสอง
- ข้อมูลเข้า : อ่านสัมประสิทธิ์ของ X ในสมการกำลังสอง คือ A, B และ C จากบัตร
- การประมวลผล : เงื่อนไขในการหาราก, สูตร และ Variable ที่เกิดขึ้นระหว่างการ

ประมวลผล

$$DEL = B^2 - 4 AC$$

X1 และ X2 เป็นรากทั้ง 2 ของสมการกำลังสอง

n) ถ้า  $DEL > 0$  คำนวณ  $X1 = \frac{-B + \sqrt{DEL}}{2A}$  และ  $X2 = \frac{-B - \sqrt{DEL}}{2A}$

ข) ถ้า  $DEL = 0$  คำนวณ  $x1 = -B/(2A)$  และ  $x2 = x1$

ค) ถ้า  $DEL < 0$  พิมพ์ 'THE ROOTS ARE COMPLEX' ที่ REMARK

: N ใช้นับบัตร

- ข้อมูลที่ต้องแสดง

พิมพ์ N,A,B,C และ (ค่า X1 และ X2) หรือพิมพ์คำอธิบายใน REMARK

