การเขียนผังงาน (Flowchart)

รายวิชา 232301 การเขียนโปรแกรม 1
คณะบริหารศาสตร์ สาขาคอมพิวเตอร์ธุรกิจ
อ.อติโรจน์ วงษ์วัชรศักดิ์

บทนำ

การเขียนผังงาน (Flowchart) เป็นเทคนิคหรือวิธีการอย่างหนึ่งสำหรับใช้เขียนแสดงอัลกอริทึม การเขียนผังงานจะใช้รูปภาพที่เป็นสัญลักษณ์มาตรฐานสากลนำไปเขียนแทนกิจกรรมต่างๆ ในแต่ละขั้นตอนของอัลกอริทึม ซึ่งกิจกรรมที่แตกต่างกันก็จะใช้สัญลักษณ์รูปภาพที่แตกต่างกัน โดยสัญลักษณ์ภาพนี้กำหนดตามมาตรฐานของ ANSI (American National Standards Institute) และ ISO (International Standard Organization)

การเขียนแสดงอัลกอริทึมด้วยผังงานสามารถออกแบบได้ง่าย ผู้อื่นสามารถเข้าใจผังงานได้ง่าย เพราะเป็นมาตรฐานสากลและเมื่อนำผังงานไปเขียนโปรแกรมก็จะทำได้สะดวกรวดเร็ว

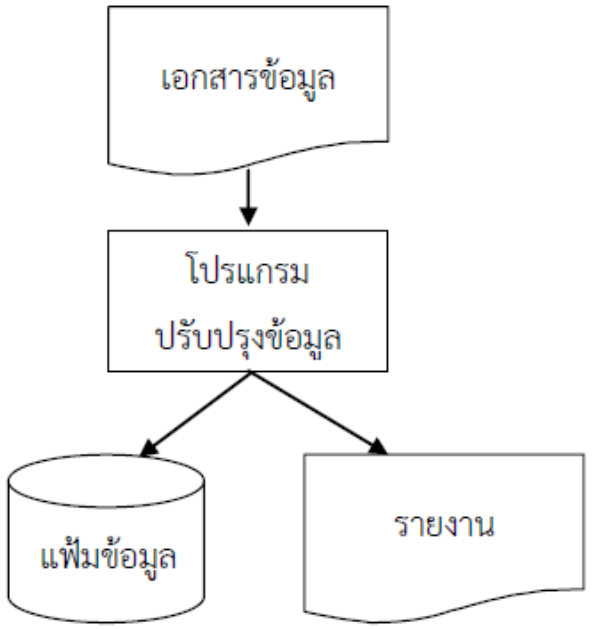
ประเภทของผังงาน

ผังงานทางคอมพิวเตอร์มี 2 ประเภทคือ

- ผังงานระบบ (System Flowchart)
- ผังงานโปรแกรม(Program Flowchart)

ผังงานระบบ (System Flowchart)

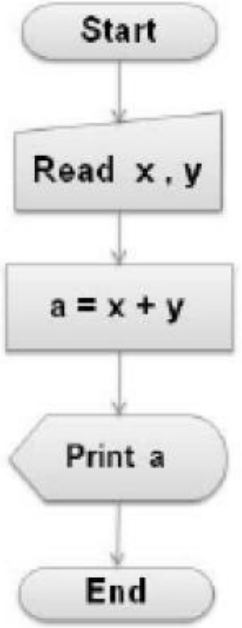
ผังงานระบบจะเป็นการแสดงให้เราเห็นว่าในระบบหนึ่งๆ นั้นมีขั้นตอนในการทางานอย่างไร ซึ่งจะมองเห็นในลักษณะภาพกว้าง ๆ ของระบบ แต่จะไม่เจาะลึกลงไปว่าในระบบว่าในแต่ละงานนั้นมีการทำงานอย่างไร คือจะให้เห็นว่าจุดเริ่มต้นของงานเริ่มจากส่วนใด เป็นข้อมูลแบบใด มีการประมวลผลอย่างไร และจะได้ผลลัพธ์เป็นอย่างไรและเก็บอยู่ที่ใด ดังรูป



ภาพที่ 4-1 ผังงานระบบ

ผังงานโปรแกรม(Program Flowchart)

ผังงานโปรแกรมหรือเรียกสั้นๆ ว่า ผังงาน จะเป็นผังงานที่แสดงให้เห็นถึงลำดับขั้นตอนในการทำงานของ โปรแกรม ตั้งแต่การรับข้อมูล การประมวลผล ตลอดจนผลลัพธ์ที่ได้ จะทำให้เขียนโปรแกรมได้สะดวกขึ้น ซึ่งผังงานชนิดนี้อาจสร้างมาจากผังงานระบบ โดยดึงเอาจุดที่เกี่ยวข้องกับคอมพิวเตอร์มาวิเคราะห์ว่าจะใช้ทำงานส่วนใด เพื่อที่จะให้ได้มาซึ่งผลลัพธ์ที่ต้องการ ดังรูป

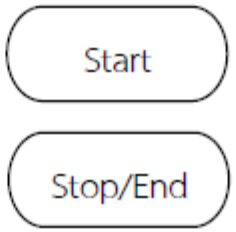
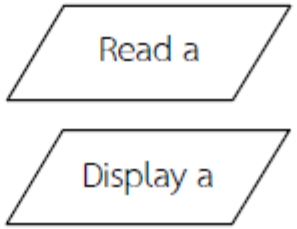
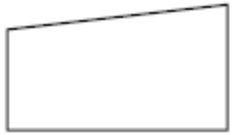
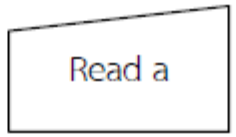
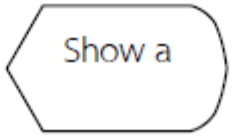


ภาพที่ 4-2 ผังงานโปรแกรม

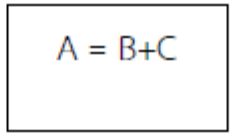
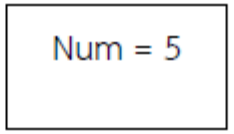
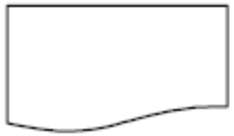
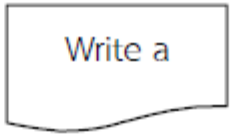
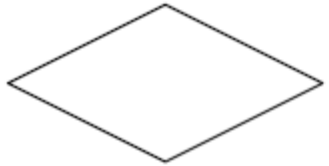
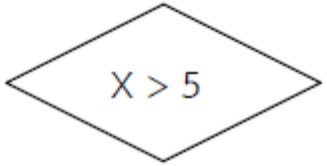
ประโยชน์ของผังงาน

1. ทำให้มองเห็นรูปแบบของงานได้ทั้งหมด โดยใช้เวลาไม่มาก
2. การเขียนผังงานเป็นสากลสามารถนาไปเขียนโปรแกรมได้ทุกภาษา
3. สามารถตรวจสอบข้อผิดพลาดของโปรแกรมได้อย่างรวดเร็ว
4. หากมีการพัฒนาระบบงานในลำดับต่อไป สามารถทำได้อย่างรวดเร็ว โดยศึกษาจากผังงานจะสามารถ ศึกษาได้อย่างรวดเร็วและเข้าใจง่ายกว่าการศึกษาจากโปรแกรม

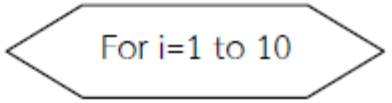
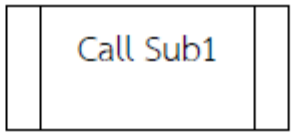
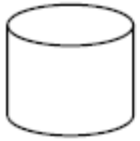
สัญลักษณ์ของผังงาน

สัญลักษณ์	ความหมาย	ตัวอย่าง	ความหมาย
	แสดงการเริ่มต้น หรือ การสิ้นสุดของผังงาน		1. เริ่มผังงาน 2. จบผังงาน
	รับหรือแสดงข้อมูลโดย ไม่ระบุชนิดของอุปกรณ์		1. รับค่า a โดยไม่ระบุ อุปกรณ์ 2. แสดงค่า a โดยไม่ระบุ อุปกรณ์
	ป้อนข้อมูลด้วยคีย์บอร์ด		ป้อนค่า a ทางคีย์บอร์ด
	แสดงผลลัพธ์ทางจอภาพ		แสดงค่า a ทางจอภาพ

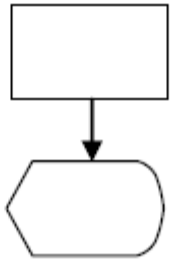
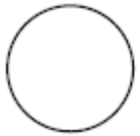
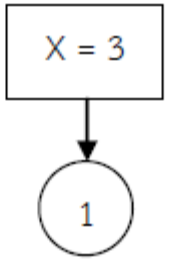
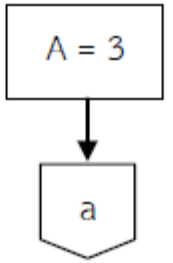
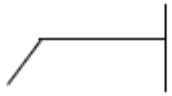
สัญลักษณ์ของผังงาน

สัญลักษณ์	ความหมาย	ตัวอย่าง	ความหมาย
	การคำนวณหรือการกำหนดค่า	 	1. คำนวณค่า B+C แล้วเก็บผลลัพธ์ไว้ที่ A 2. กำหนดค่า Num = 5
	แสดงข้อมูลหรือผลลัพธ์ทางเครื่องพิมพ์ลงบนกระดาษ		พิมพ์ค่า a ทางกระดาษ
	การเปรียบเทียบการตัดสินใจ		เปรียบเทียบ X > 5 หรือไม่

สัญลักษณ์ของผังงาน

สัญลักษณ์	ความหมาย	ตัวอย่าง	ความหมาย
	การเตรียมการ การกำหนดค่าล่วงหน้า		กำหนดให้ i=1 แล้ว เพิ่มขึ้นทีละ 1 จนถึง 10
	โปรแกรมย่อย		เรียกโปรแกรมย่อย ชื่อ sub1
	รับหรือแสดงข้อมูลโดย ใช้เทปแม่เหล็ก	-	-
	รับหรือแสดงข้อมูลโดย ใช้จานแม่เหล็ก	-	-
	การเก็บข้อมูล	-	-

สัญลักษณ์ของผังงาน

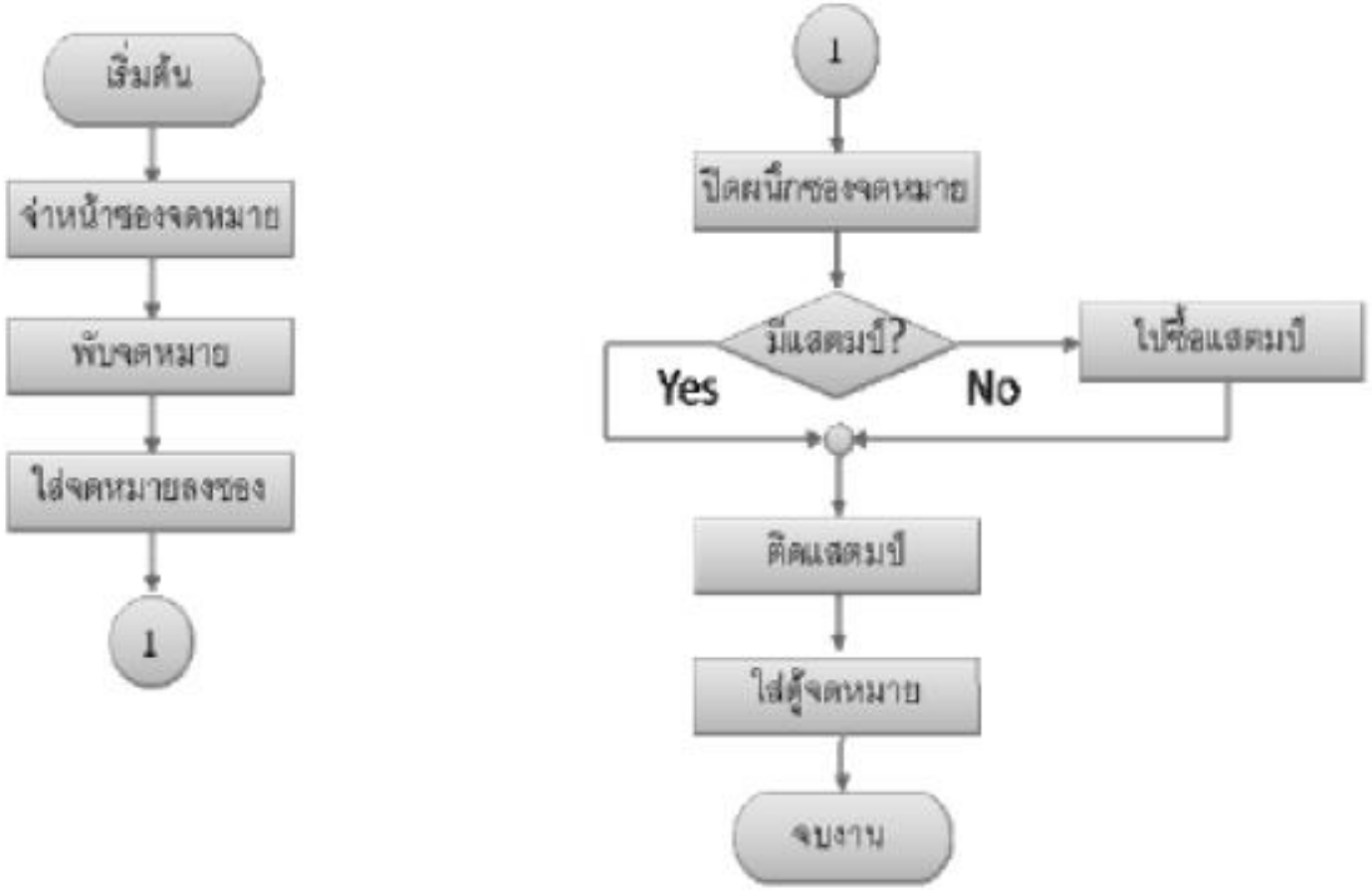
สัญลักษณ์	ความหมาย	ตัวอย่าง	ความหมาย
	เส้นแสดงทิศทางการทำงาน		ประมวลผลแล้วต่อด้วยการแสดงผล
	จุดเชื่อมต่อในหน้าเดียวกัน		หลังกำหนด a=3 แล้วให้ไปทำงานต่อที่จุดต่อเนื่อง 1 หน้าเดียวกัน
	จุดเชื่อมต่อไปหน้าอื่น		หลังกำหนด b=5 แล้วให้ไปทำงานต่อที่จุดต่อเนื่อง a ซึ่งอยู่คนละหน้ากัน
	หมายเหตุหรือคำอธิบาย		-

หลักการเขียนผังงานที่ดี

1. มีทางเข้าหรือจุดเริ่มต้น และทางออกหรือจุดสิ้นสุดเพียงทางเดียวเท่านั้น
2. ลำดับขั้นตอนการทำงานควรจะเริ่มจากบนลงล่าง หรือจากซ้ายไปขวา
3. ในสัญลักษณ์ใด ๆ มีทางออกเพียงทางเดียว ยกเว้นสัญลักษณ์แสดงการตัดสินใจหรือทางเลือกสามารถมีทางออกได้อย่างน้อยสองทาง
4. เส้นทางเดินในผังงานควรชัดเจน เป็นระเบียบ
5. ข้อความหรือคำสั่งใดๆ ที่อยู่ในสัญลักษณ์ควรสั้น กระชับ ได้ใจความและสามารถเข้าใจได้ง่าย
6. ใช้สัญลักษณ์ที่มีขนาดเหมาะสมกับคำสั่ง
7. การกาหนดทิศทางการทำงานด้วยลูกศร ควรจะมีทิศทางจากบนลงล่างหรือขวาไปซ้ายเท่านั้น
8. ในกระบวนการทำงานที่ต้องการเพิ่มคำอธิบายเข้าไปเพื่อให้เกิดความเข้าใจ ก็สามารถทำได้โดยการใช้สัญลักษณ์หมายเหตุประกอบ

ตัวอย่างการเขียนผังงานในชีวิตประจำวัน

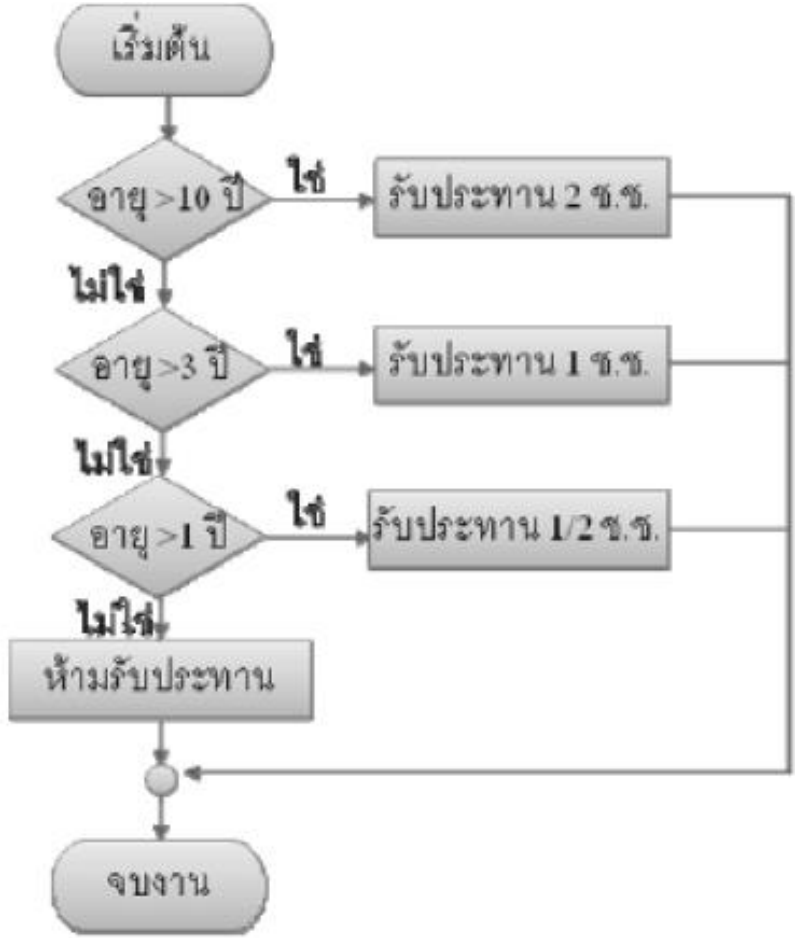
<u>ตัวอย่างที่ 1</u> การเขียนผังงานขั้นตอนการส่งจดหมาย



ตัวอย่างการเขียนผังงานในชีวิตประจำวัน

<u>ตัวอย่างที่ 2</u> การเขียนผังงานแสดงเงื่อนไขการรับประทานยา โดยแบ่งขนาดรับประทานตามอายุ ดังนี้

- อายุมากกว่า 10 ปี รับประทานครั้งละ 2 ช้อนชา
- อายุ 3 - 10 ปี รับประทานครั้งละ 1 ช้อนชา
- อายุ 1 -3 ปี รับประทานครั้งละ 1/2 ช้อนชา
- เด็กแรกเกิด ห้ามรับประทาน



ตัวอย่างการเขียนผังงานในชีวิตประจำวัน

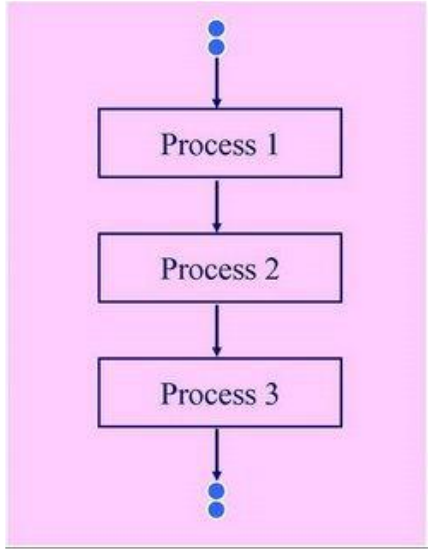
ตัวอย่างที่ 3 การเขียนผังงานการหาพื้นที่วงกลม

รูปแบบการเขียนผังงาน

1. การเขียนผังงานแบบเรียงลำดับการทำงาน (Sequential Structure)
2. การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)
    - ผังงานแบบมี 2 ทางเลือก IF
    - ผังงานที่มีมากกว่า 2 ทางเลือก CASE
3. การเขียนผังงานแบบมีการทำงานวนซ้ำ (Iteration Structure)
    - While
    - Do while
    - Repeat until (Do until)
    - For

การเขียนผังงานแบบเรียงลำดับการทำงาน (Sequential Structure)

โครงสร้างการทำงานแบบลำดับ (Sequence) เป็นลักษณะโครงสร้างพื้นฐานของผังงานทุกผังงาน ลักษณะการทำงานของโครงสร้างผังงานแบบลำดับจะทำงานทีละขั้นตอน ตั้งแต่ขั้นตอนแรกจนถึงขั้นตอนสุดท้าย ลำดับการทำงานของผังงานจะทำงานตามทิศทางของลูกศร โครงสร้างของผังงานแบบลำดับ มีลักษณะโครงสร้างดังนี้

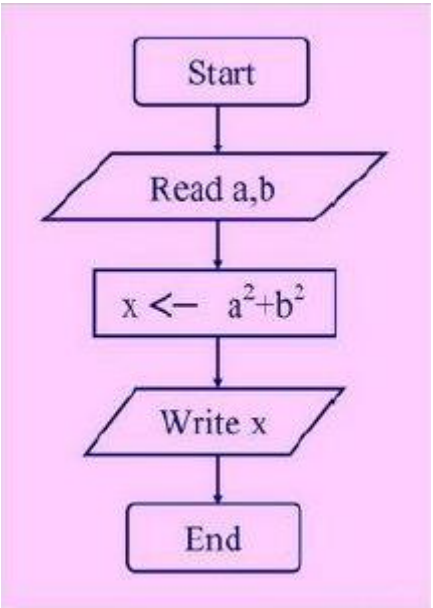


การเขียนผังงานแบบเรียงลำดับการทำงาน (Sequential Structure)

ตัวอย่าง โครงสร้างผังงานการทำงานแบบลำดับ

จากโครงสร้างผังงานแบบลำดับตามรูป สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

1. เริ่มต้นการทำงาน
2. รับค่าข้อมูลเข้ามาเก็บไว้ในตัวแปร a และตัวแปร b
3. คำนวณค่า a2 + b2 แล้วไปเก็บไว้ในตัวแปร x
4. แสดงค่าในตัวแปร x 5. สิ้นสุดการทำงาน



การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)

เป็นการเขียนผังงานที่มีลักษณะการทำงานแบบมีเงื่อนไขทางตรรกะ โดยใช้ประโยชน์จากพีชคณิตบูลีน เพื่อให้เครื่องประมวลผลลักษณะตัดสินใจ เลือกทิศทางการทำงานตามคาสั่งที่กาหนดไว้

การเขียนผังงานแบบมีทางเลือกสามารถแบ่งออกเป็นลักษณะการเขียนดังนี้

- ผังงานแบบมีทางเลือกโดยใช้คำสั่ง IF
- ผังงานที่มีมากกว่า 2 ทางเลือกจากการใช้คำสั่ง CASE

การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)

เป็นการเขียนผังงานที่มีลักษณะการทำงานแบบมีเงื่อนไขทางตรรกะ โดยใช้ประโยชน์จากพีชคณิตบูลีน เพื่อให้เครื่องประมวลผลลักษณะตัดสินใจ เลือกทิศทางการทำงานตามคาสั่งที่กาหนดไว้

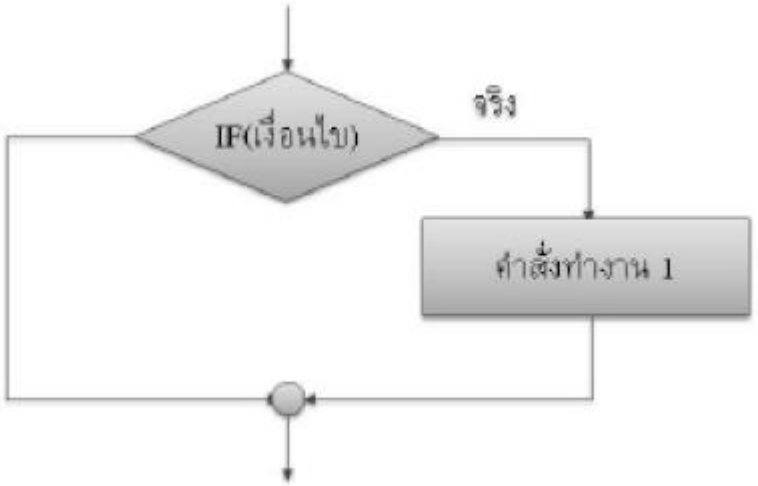
การเขียนผังงานแบบมีทางเลือกสามารถแบ่งออกเป็นลักษณะการเขียนดังนี้

- ผังงานแบบมีทางเลือกโดยใช้คำสั่ง IF
- ผังงานที่มีมากกว่า 2 ทางเลือกจากการใช้คำสั่ง CASE

การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)

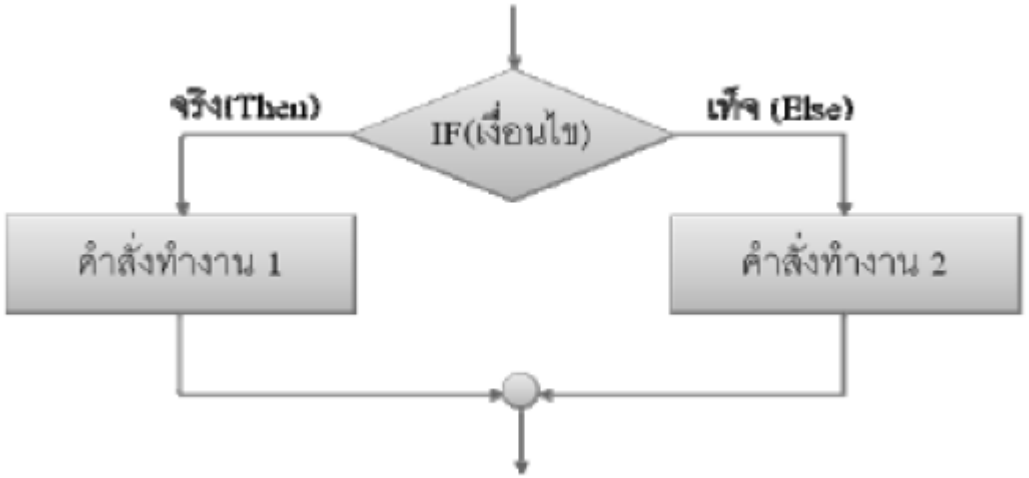
ผังงานแบบมี 2 ทางเลือกจากการใช้คำสั่ง IF

o **IF แบบทางเลือกเดียว Single Selection** จะทำการตรวจสอบเงื่อนไข ถ้าเป็นจริงจะไปทำคำสั่งทำงาน ถ้าเป็นเท็จก็จะออกจากโครงสร้างโดยมีรูปแบบการเขียนดังนี้



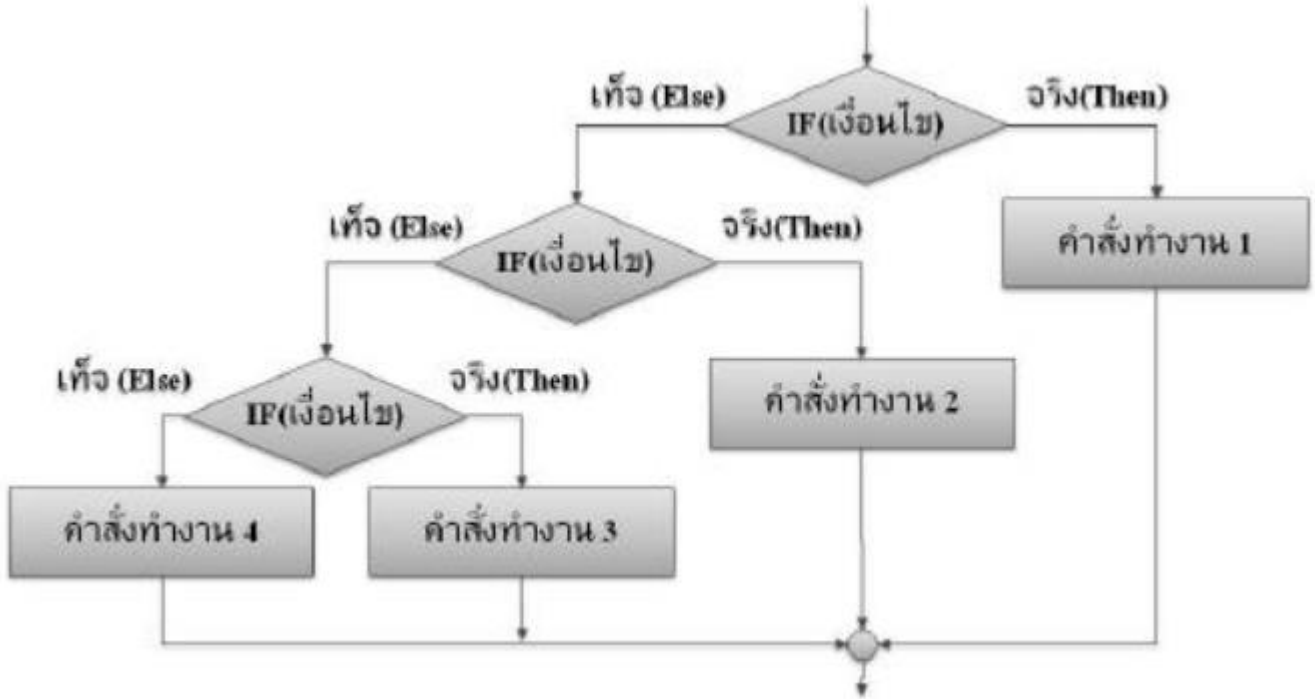
การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)

o **IF แบบสองทางเลือก Double Selection** จะทำการตรวจสอบเงื่อนไข ถ้าเป็นจริงจะไปทำคำสั่งทำงาน 1 แล้วออกจากโครงสร้าง ถ้าเป็นเท็จจะไปทำคำสั่งทางาน 2 แล้วออกจากโครงสร้าง ซึ่งมีรูปแบบการ เขียนดังนี



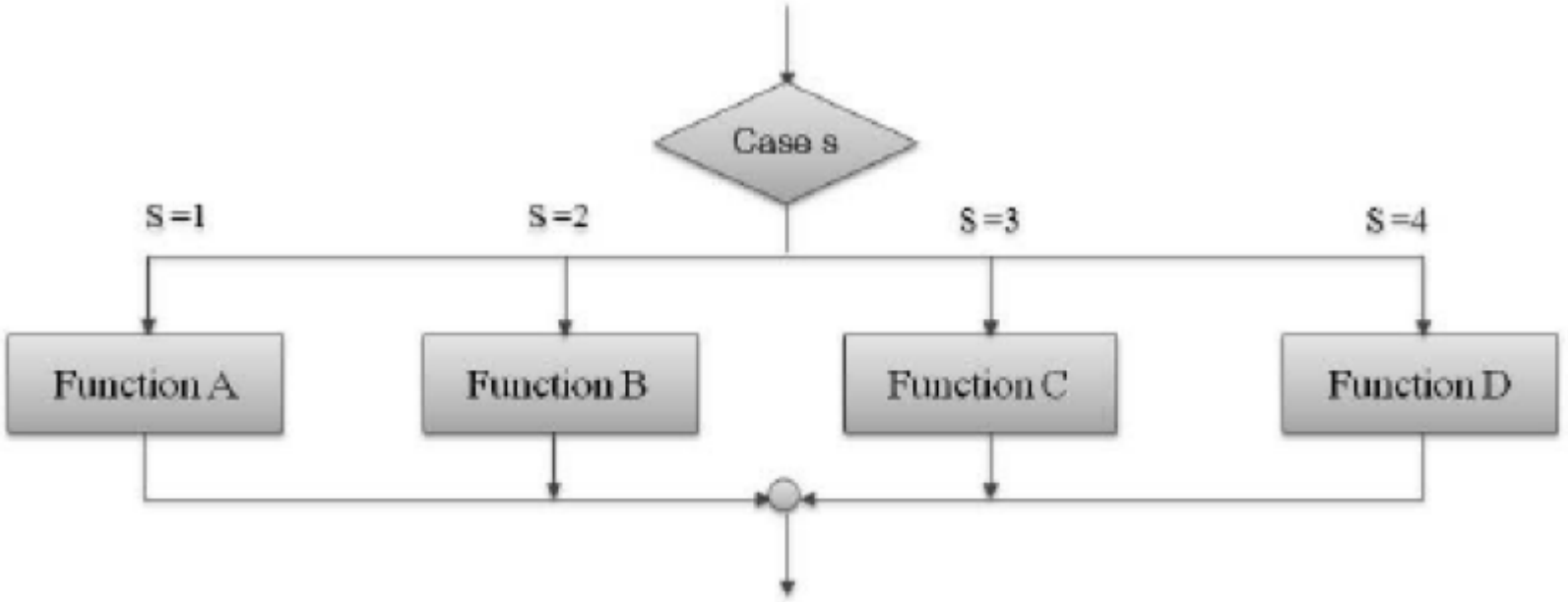
การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)

o **IF แบบหลายทางเลือก Multi Selection** เป็นการตรวจสอบแบบหลายเงื่อนไข ทำให้มีทางออกได้หลายทาง ซึ่งเป็นการนำโครงสร้างของ IF มาซ้อนกันเพื่อให้ได้เงื่อนไขมากขึ้นตามความต้องการ โดยมีรูปแบบการเขียนดังนี้



การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)

ผังงานที่มีมากกว่า 2 ทางเลือก CASE เป็นการตรวจสอบเงื่อนไขของตัวแปรตรวจสอบว่ามีค่าตรงกับค่าใด ค่าดังกล่าวจะเป็นตัวบอกกิจกรรมที่ต้องทำ โดยมีรูปแบบการเขียนดังนี้



การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)

ตัวอย่าง โครงสร้างผังงานแบบเลือกทำหรือมีเงื่อนไข

ผังงานแสดงโปรแกรมการประเมินผลการเรียน โดยรับคะแนนนักศึกษาเข้ามาทางแป้นพิมพ์ ถ้าคะแนนมากกว่าหรือเท่ากับ 50 ให้แสดงคาว่า “Pass” ที่หน้าจอ แต่ถ้าน้อยกว่า 50 ให้แสดงคำว่า “Fail” ที่หน้าจอ

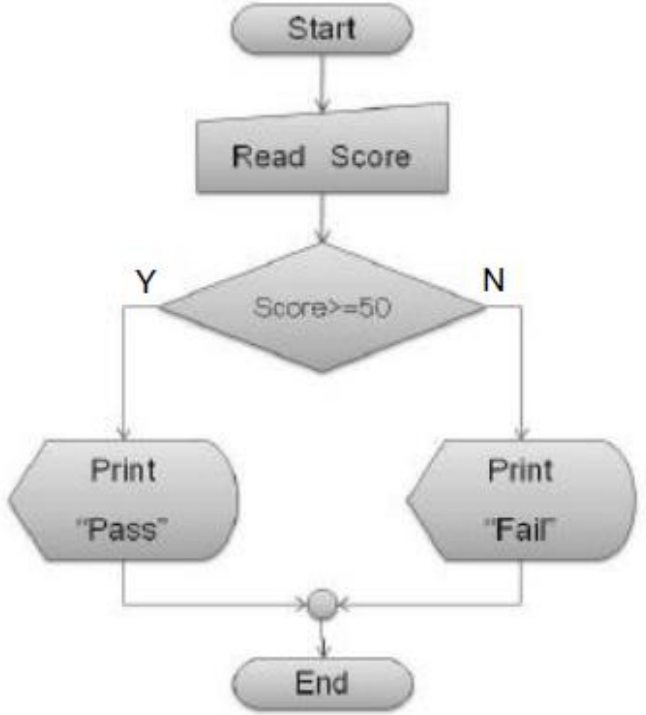
จากโครงสร้างผังงานแบบเลือกทำหรือมีเงื่อนไขตามรูป สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

1. เริ่มต้นการทำงาน
2. รับค่าข้อมูลเข้ามาเก็บไว้ในตัวแปร score
3. ถ้าคะแนนที่รับ (ตัวแปร score) มากกว่าหรือเท่ากับ 50 ทำ

   3.1 แสดงคาว่า “Pass” ที่หน้าจอ

   มิฉะนั้นแล้ว

   3.2 แสดงคาว่า “Fail” ที่หน้าจอ
4. สิ้นสุดการทำงาน



การเขียนผังงานแบบมีทางเลือกการทำงาน (Decision Structure)

ตัวอย่าง โครงสร้างผังงานแบบเลือกทำหรือมีเงื่อนไข

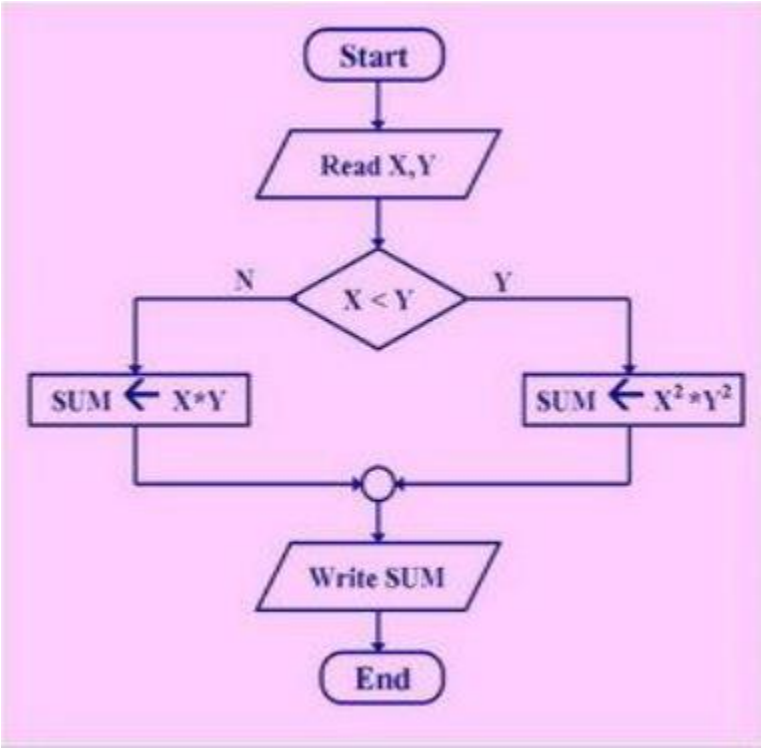
จากตัวอย่าง ประกอบด้วยโครงสร้างผังงาน 2 โครงสร้าง คือการทำงานแบบลำดับและการเลือกทำ การทำงานของผังงาน สามารถอธิบายลำดับขั้นตอนการทำงานได้ดังนี้

1. เริ่มต้นการทำงาน
2. รับค่าข้อมูล จำนวน 2 ค่า มาเก็บไว้ในตัวแปร X และ Y
3. ตรวจสอบเงื่อนไข ถ้า X น้อยกว่า Y แล้วทำ

   3.1 คานวณค่า SUM = X2*Y2

   มิฉะนั้นแล้ว

   3.2 คานวณค่า SUM = X * Y 4. แสดงค่า SUM
5. จบการทำงาน



การเขียนผังงานแบบมีการทำงานวนซ้ำ (Iteration Structure)

โปรแกรมส่วนใหญ่จะมีคำสั่งสำหรับการทำงานซ้ำหรือเรียกว่า ลูป (Loop) โดยการทำซ้ำของโปรแกรมจะอยู่ภายใต้เงื่อนไข จริง หรือ เท็จ ตามที่ผู้เขียนโปรแกรมได้ออกแบบไว้ รูปแบบของการวนซ้ำมีดังนี้

- **คำสั่ง While (ขณะที่)**

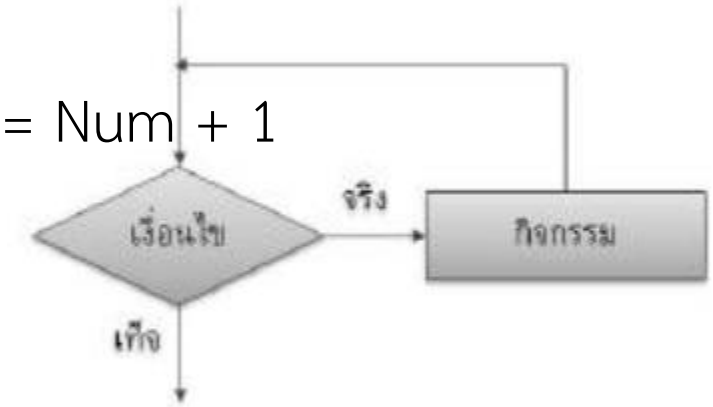
โครงสร้างคำสั่งจะทำการตรวจสอบเงื่อนไขก่อน ถ้าเงื่อนไขเป็น “จริง” ก็จะทำกิจกรรมนั้นซ้ำไปเรื่อยๆ แต่เมื่อเงื่อนไขเป็น “เท็จ” จะหยุดทำซ้ำแล้วออกจากลูปไป มีรูปแบบการเขียนดังนี้

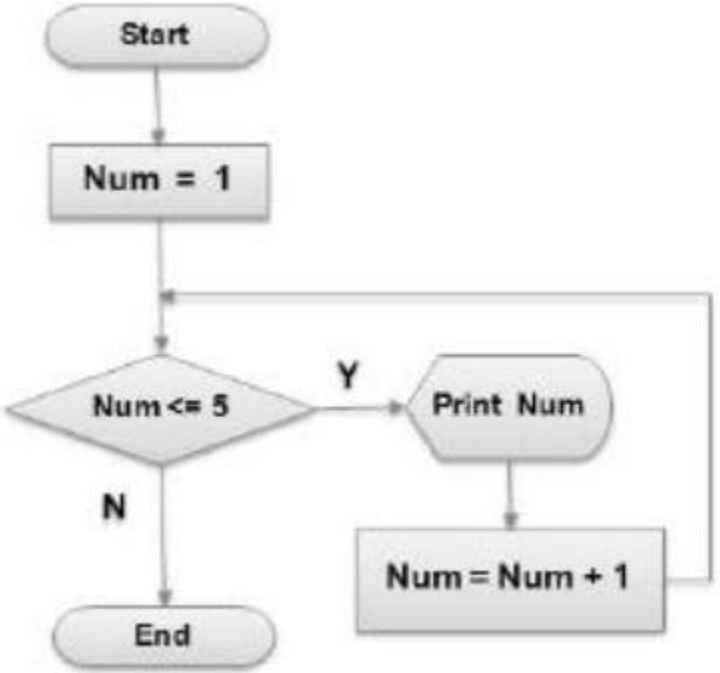
การเขียนผังงานแบบมีการทำงานวนซ้ำ (Iteration Structure)

ตัวอย่าง โปรแกรมแสดงตัวเลข 1 – 5 โดยใช้คำสั่ง While

ลำดับขั้นตอนการทำงานของผังงานนี้ คือ

1. เริ่มต้นการทำงาน
2. กำหนดค่าให้ Num = 1
3. ในขณะที่ Num น้อยกว่าหรือเท่ากับ 5 ทำ
   3.1 แสดงค่า Num
   3.2 คำนวณค่า Num = Num + 1
4. จบการทำงาน





การเขียนผังงานแบบมีการทำงานวนซ้ำ (Iteration Structure)

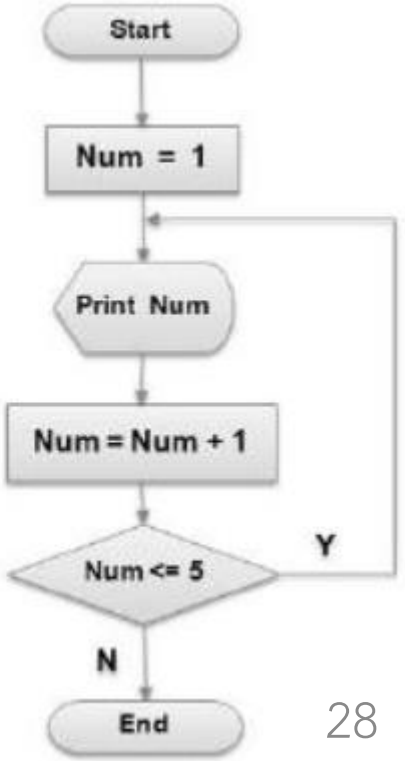
- **คำสั่ง Do while (ทำ-ขณะที่)**

ทำกิจกรรมที่ต้องการก่อน แล้วจึงตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็น “จริง” ก็จะทำกิจกรรมนั้นซ้ำไปเรื่อยๆ แต่เมื่อเงื่อนไขเป็น “เท็จ” จะหยุดทำซ้ำแล้วออกจากลูปไป มีรูปแบบการเขียนดังนี้

<u>ตัวอย่าง</u> โปรแกรมแสดงตัวเลข 1 – 5 โดยใช้คำสั่ง Do While

ลำดับขั้นตอนการทำงานของผังงานนี้ คือ

1. เริ่มต้นการทำงาน
2. กำหนดค่าให้ Num = 1
3. แสดงค่า Num ออกทางจอภาพ
4. คำนวณ Num + 1 เก็บไว้ในตัวแปร Num
5. ขณะที่ Num น้อยกว่าหรือเท่ากับ 5 ทำข้อ 3-4 มิฉะนั้น
6. จบการทางาน



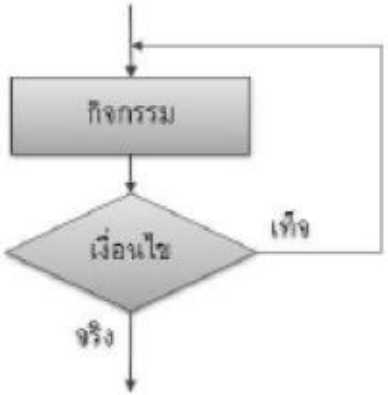
การเขียนผังงานแบบมีการทำงานวนซ้ำ (Iteration Structure)

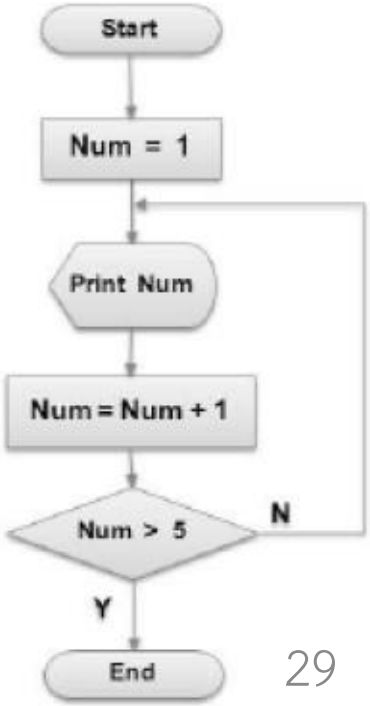
- **คำสั่ง Repeat until (Do until)**

ทำกิจกรรมที่ต้องการก่อนแล้วจึงตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็น “เท็จ” ก็จะทากิจกรรมนั้นซ้ำไปเรื่อยๆ แต่ถ้าเงื่อนไขเป็น “จริง” จะหยุดทำซ้ำแล้วออกจากลูปไป มีรูปแบบการเขียนดังนี้

<u>ตัวอย่าง</u> โปรแกรมแสดงตัวเลข 1 – 5 โดยใช้คำสั่ง Repeat until (Do until)

1. เริ่มต้นการทำงาน
2. กำหนดให้ Num = 1
3. ทำซ้ำจนกระทั่ง Num > 5
   3.1 แสดงค่า Num
   3.2 คำนวณค่า Num เท่ากับ Num + 1
4. จบการทางาน





ข้อแตกต่างระหว่างการทำซ้ำลักษณะทำในขณะที่ (Do While) กับทำจนกระทั่ง (Do Unit)

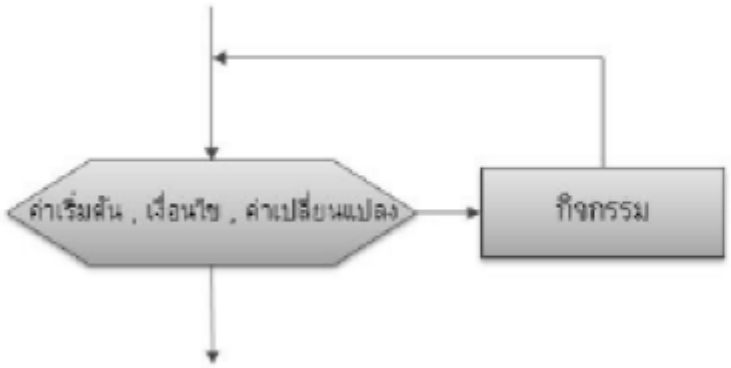
ลักษณะการทำงาน	ทำในขณะที่	ทำจนกระทั่ง
การตรวจสอบเงื่อนไข	ตรวจสอบเงื่อนไขก่อนการทำงานส่วนที่ต้องการทำซ้ำ	ทำงานส่วนที่ต้องการทำซ้ำก่อนแล้วจึงตรวจสอบเงื่อนไข
การตัดสินใจเพื่อทำซ้ำ	เมื่อผลจากการตรวจสอบเงื่อนไขเป็นจริง	เมื่อผลจากการตรวจสอบเงื่อนไขเป็นเท็จ
จำนวนการทำซ้ำ	อาจไม่มีการทางานในส่วนของการทำงานต้องการทำซ้ำ ถ้าผลที่ได้จากการตรวจสอบเงื่อนไขครั้งแรกเป็นเท็จ	ทำขั้นตอนการทางานที่ต้องการทำซ้ำอย่างน้อย 1 ครั้ง

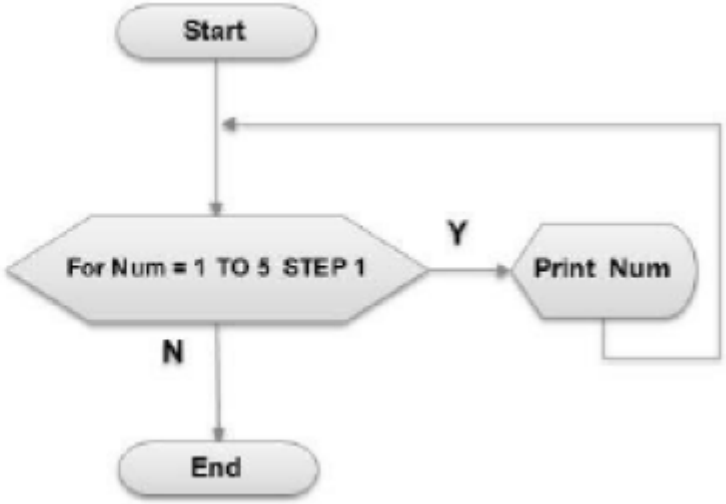
การเขียนผังงานแบบมีการทำงานวนซ้ำ (Iteration Structure)

- **คำสั่ง For**

โครงสร้างคำสั่งจะทำการวนซ้ำโดยรู้จำนวนแน่นอน โดยมีการกำหนดค่าเริ่มต้น ค่าสิ้นสุดของ ตัวแปรนับรอบ มีรูปแบบการเขียนดังนี้

ตัวอย่าง โปรแกรมแสดงตัวเลข 1 – 5 โดยใช้คำสั่ง For





ตัวอย่าง การเขียนผังงานจากโจทย์ปัญหา

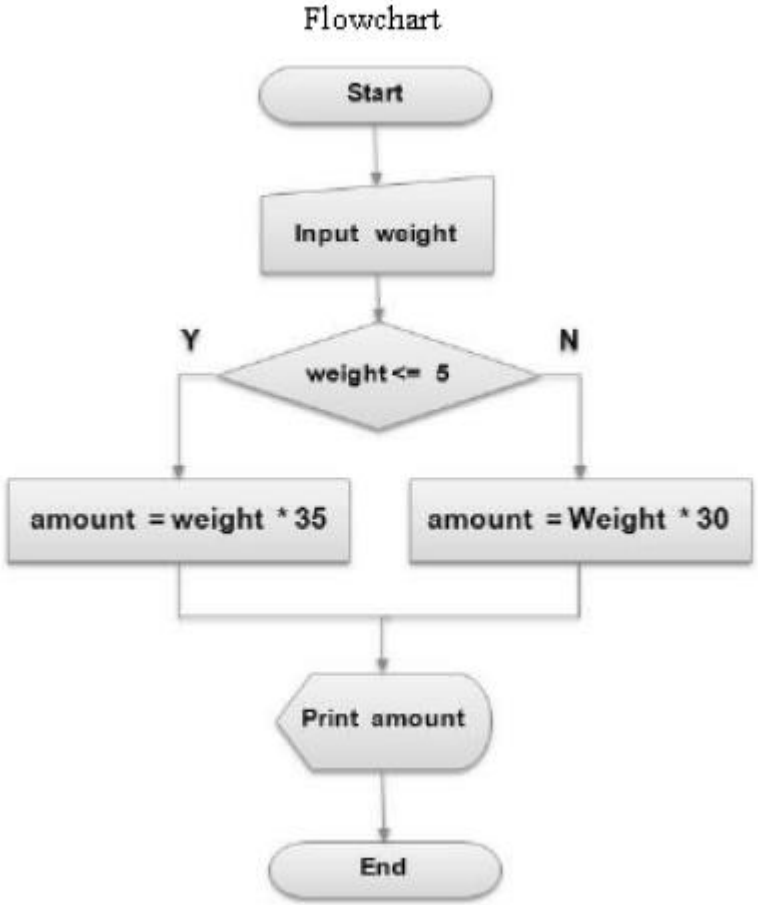
<u>ตัวอย่างที่ 1</u> จงเขียน Flowchart รับข้อมูลจำนวนส้มที่ซื้อ เพื่อคำนวณหาราคาส้มที่ต้องจ่าย โดยมีเงื่อนไขว่า ถ้าซื้อส้มน้อยกว่าหรือเท่ากับ 5 กิโลกรัม คิดราคากิโลกรัมละ 35 บาท แต่ถ้าซื้อส้มน้อยกว่า 5 กิโลกรัมขึ้นไป คิดราคา กิโลกรัมละ 30 บาท

กำหนดตัวแปรที่ใช้ น้ำหนักของส้มที่ซื้อ ใช้ชื่อตัวแปร weight

ราคาส้มที่ต้องจ่าย ใช้ชื่อตัวแปร amount

Pseudo code

```
Compute_amount
        INIT weight , amount  : INTEGER
        READ weight
        IF (weight<=5) THEN
                amount = weight * 35
        ELSE
                amount = weight * 30
        END IF
END
```



ตัวอย่าง การเขียนผังงานจากโจทย์ปัญหา

<u>ตัวอย่างที่ 2</u> จงเขียนผังงานเพื่อรับตัวเลขจำนวน n ตัว ตามจำนวนที่ผู้ใช้ต้องการ และหาผลรวมของตัวเลขทั้งหมดนั้น

กำหนดตัวแปรที่ใช้ในผังงาน จำนวนตัวเลข ใช้ชื่อตัวแปร n, ค่าของตัวเลข ใช้ชื่อตัวแปร number

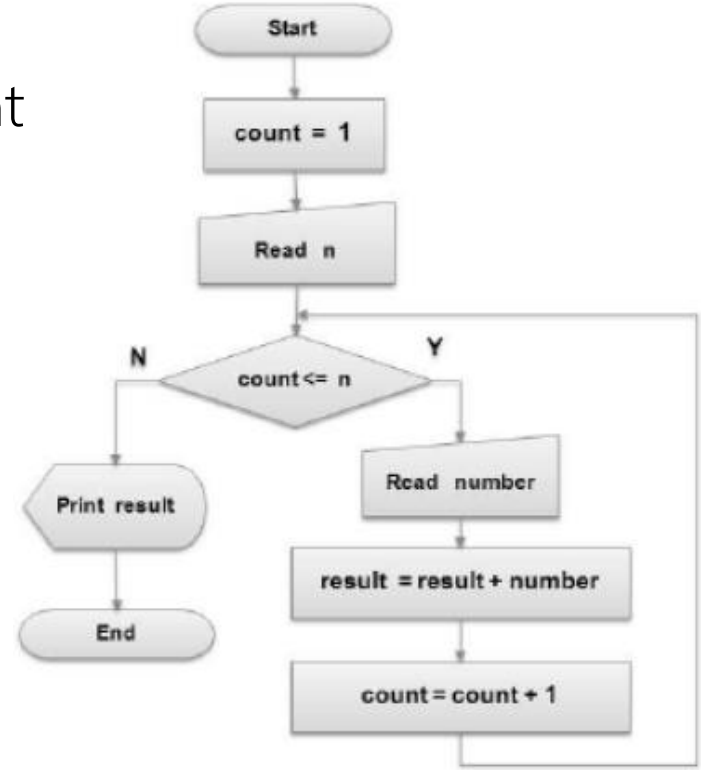
ผลรวม ใช้ชื่อตัวแปร result

ตัวนับจำนวน “ครั้งที่ของการป้อนตัวเลข” ใช้ชื่อตัวแปร count

Pseudo code

```
Sum_Number
        INIT n,number,result,count  : INTEGER
        count = 1
        READ n
        WHILE (count<=n)
                READ number
                result = result + number
                count = count + 1
        END WHILE
        PRINT result
END
```

Flowchart



ตัวอย่าง การเขียนผังงานจากโจทย์ปัญหา

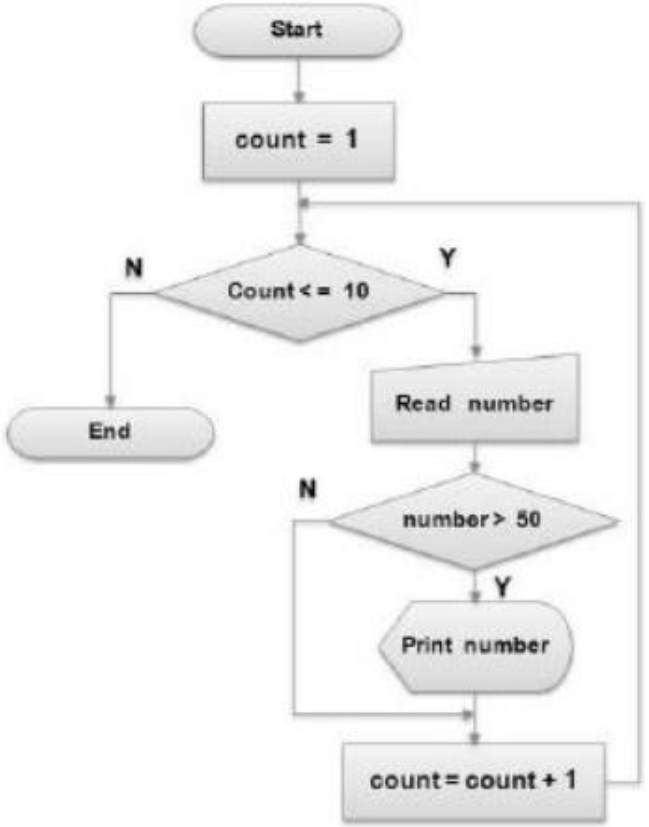
<u>ตัวอย่างที่ 3</u> จงเขียนผังงานเพื่อรับตัวเลขครั้งละ 1 จำนวน และให้ตรวจสอบ ถ้าตัวเลขที่รับเข้ามามีค่ามากกว่า 50 ให้แสดงตัวเลขนั้นบนหน้าจอ จากนั้นวนรับตัวเลขแล้วทำซ้ำจนครบ 10 จำนวน

กำหนดตัวแปรที่ใช้ในผังงาน

ค่าของตัวเลข ใช้ชื่อตัวแปร number

ตัวนับจำนวน “ครั้งที่ของการป้อนตัวเลข” ใช้ชื่อตัวแปร count

Pseudo code

```
Print_Number>50
        INIT number, count  : INTEGER
        count = 1
        WHILE (count<=5)
                Read number
                IF(number>50)THEN
                        Print number
                END IF
                count = count + 1
        END WHILE
END
```

Flowchart