

ประโยชน์ของผังงาน

1. ทำให้มองเห็นรูปแบบของงานได้ทั้งหมด โดยใช้เวลาไม่มาก
2. การเขียนผังงานเป็นสากลสามารถนำไปเขียนโปรแกรมได้ทุกภาษา
3. สามารถตรวจสอบข้อผิดพลาดของโปรแกรมได้อย่างรวดเร็ว
4. หากมีการพัฒนาระบบงานในลำดับต่อไป สามารถทำได้อย่างรวดเร็ว โดยศึกษาจากผังงาน จะสามารถศึกษาได้อย่างรวดเร็ว และเข้าใจง่ายกว่าการศึกษาจากโปรแกรม

หลักการเขียนผังงานที่ดี

1. มีทางเข้าหรือจุดเริ่มต้น และทางออกหรือจุดสิ้นสุดเพียงทางเดียวเท่านั้น
2. ลำดับขั้นตอนการทำงานควรจะเริ่มจากบนลงล่าง หรือจากซ้ายไปขวา
3. ในสัญลักษณ์ใด ๆ มีทางออกเพียงทางเดียว ยกเว้นสัญลักษณ์แสดงการตัดสินใจ หรือ ทางเลือก

สามารถมีทางออกได้อย่างน้อยสองทาง

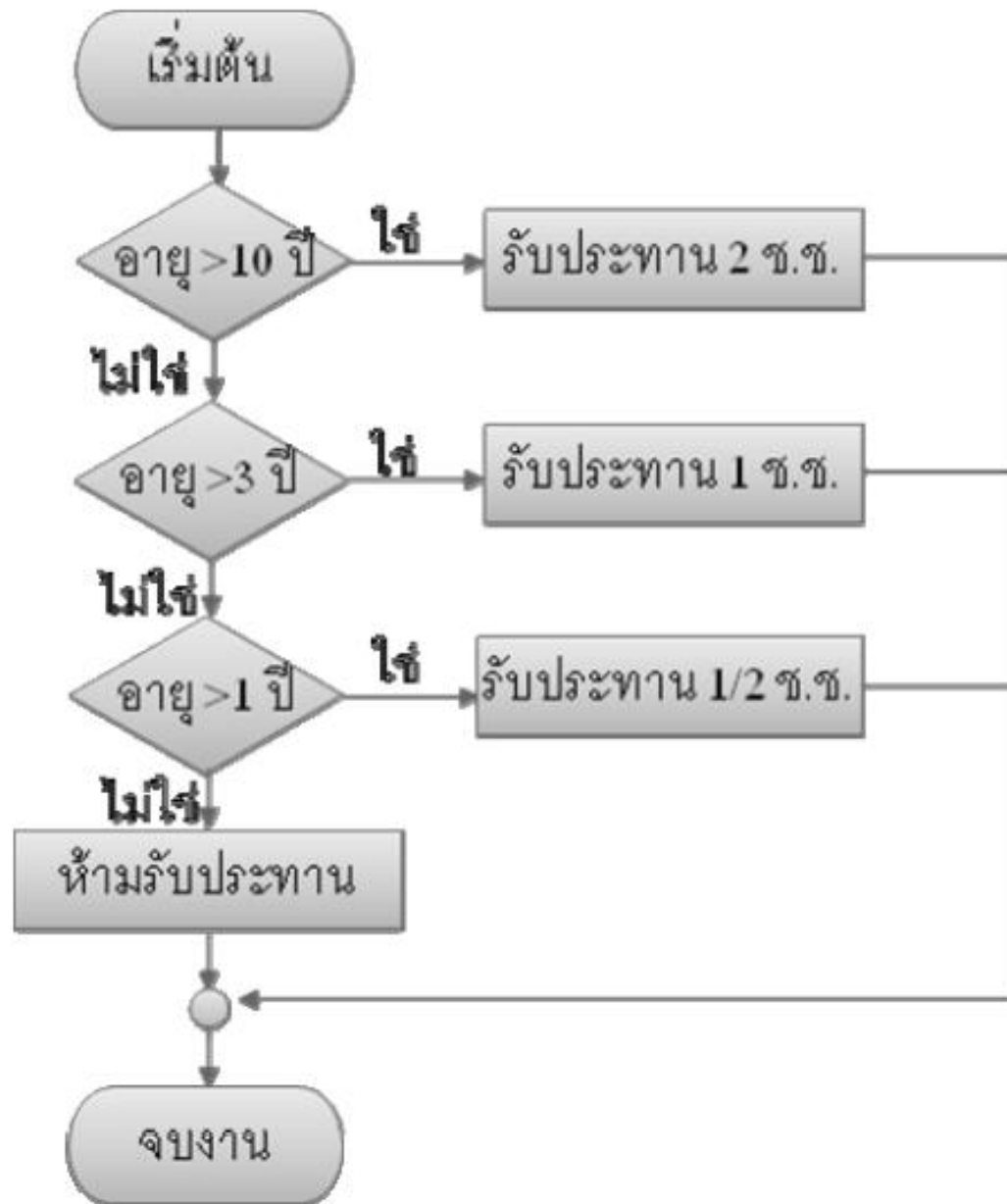
4. เส้นทางเดินในผังงานควรชัดเจน เป็นระเบียบ

หลักการเขียนผังงานที่ดี

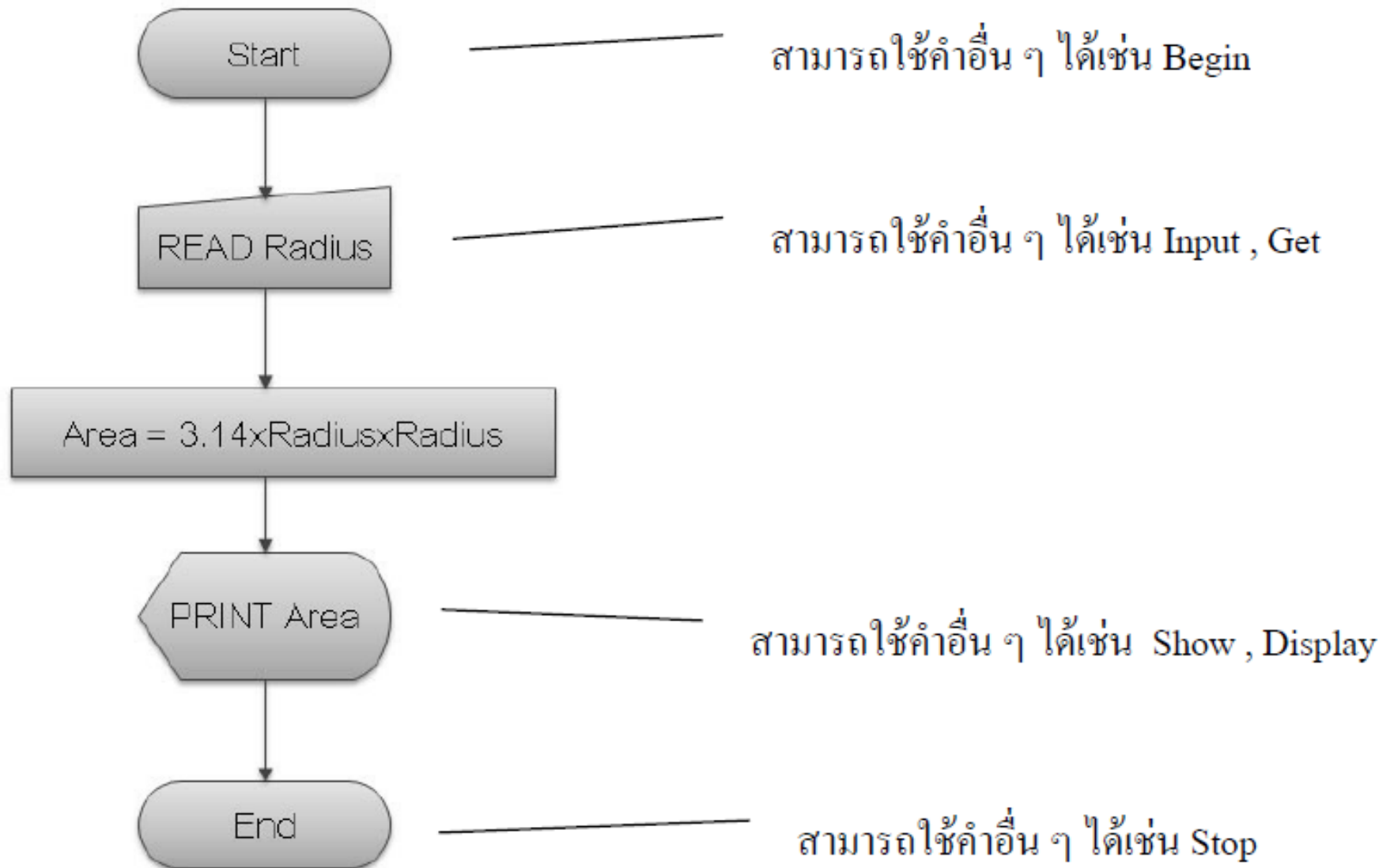
5. ข้อความหรือคำสั่งใด ๆ ที่อยู่ในสัญลักษณ์ควรสั้น กระชับ ได้ใจความ และสามารถเข้าใจได้ง่าย
6. ใช้สัญลักษณ์ที่มีขนาดเหมาะสมกับคำสั่ง
7. การกำหนดทิศทางการทำงานด้วยลูกศร ควรจะมีทิศทางจากบนลงล่าง หรือ ขวาไปซ้ายเท่านั้น
8. ในกระบวนการทำงานที่ต้องการเพิ่มคำอธิบายเข้าไปเพื่อให้เกิดความเข้าใจ ก็สามารถทำได้โดยการใช้สัญลักษณ์หมายเหตุประกอบ

ตัวอย่าง Flowchart

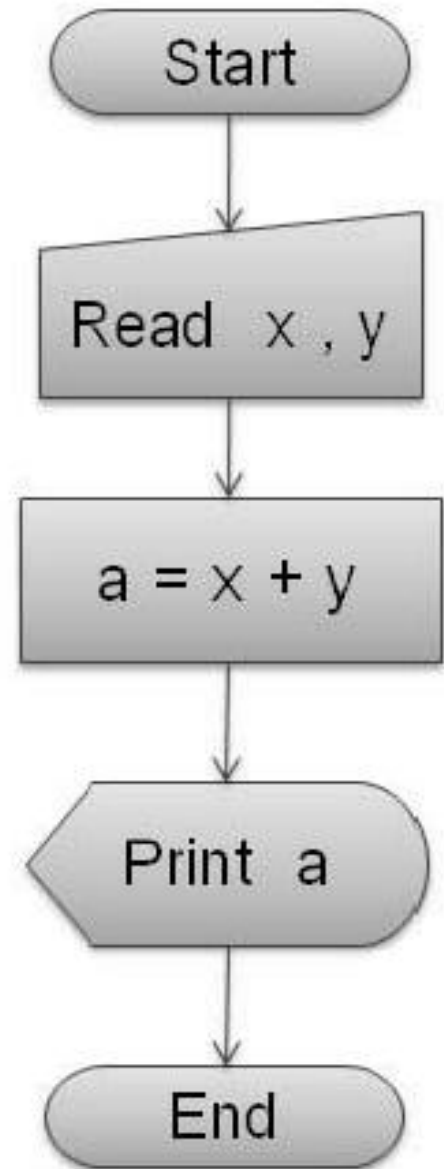
- ตัวอย่าง การเขียนผังงานแสดงเงื่อนไขการรับประทานยา โดยแบ่งขนาดรับประทานตามอายุ ดังนี้
 - อายุมากกว่า 10 ปี รับประทานครั้งละ 2 ช้อนชา
 - อายุ 3 - 10 ปี รับประทานครั้งละ 1 ช้อนชา
 - อายุ 1 -3 ปี รับประทานครั้งละ 1/2 ช้อนชา
 - เด็กแรกเกิด ห้ามรับประทาน



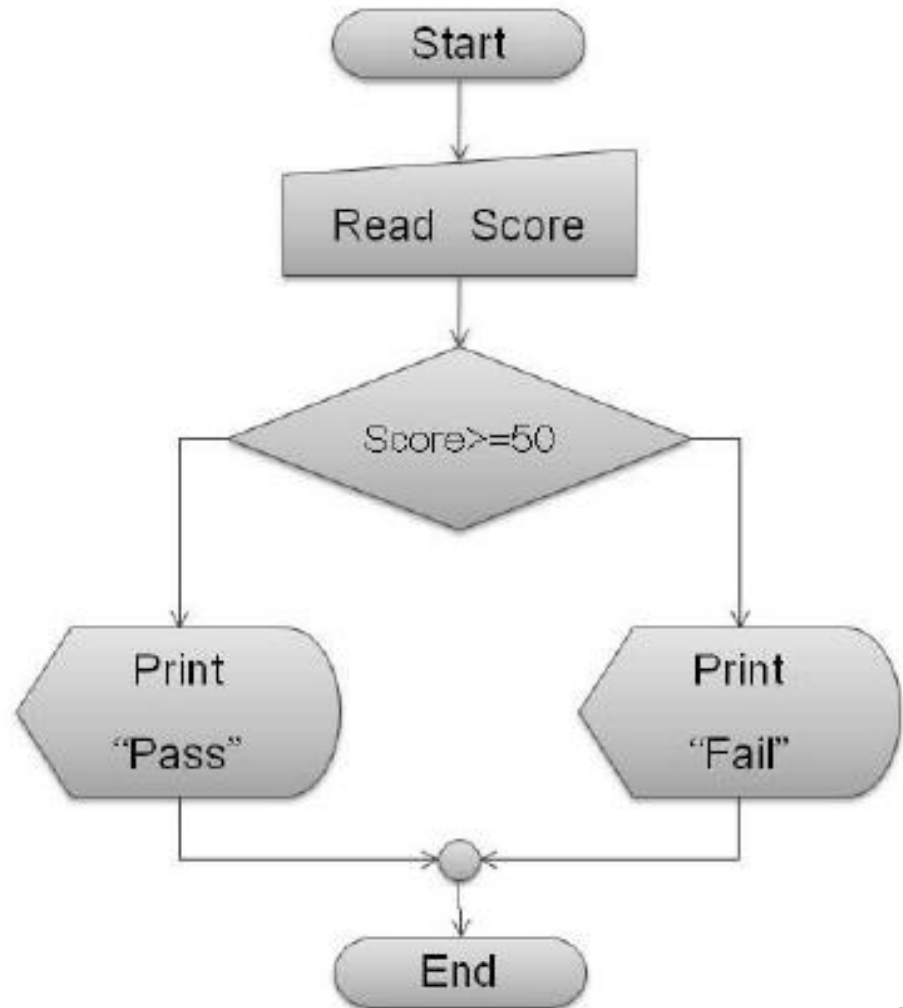
- ตัวอย่างการเขียน **Flowchart** หาพื้นที่วงกลม



ผังงานแสดงโปรแกรมการ
คำนวณค่า a จากสูตร $a = x + y$
โดยรับค่า x และ y ทางแป้นพิมพ์ และ
แสดงผลลัพธ์ a ออกทางจอภาพ



ฟังก์ชันแสดงโปรแกรมการประเมินผลการเรียน โดยรับคะแนน
นักศึกษาเข้ามาทางแป้นพิมพ์ ถ้าคะแนนมากกว่าหรือเท่ากับ 50 ให้
แสดงคำว่า “Pass” ที่หน้าจอ แต่ถ้าน้อยกว่า
50 ให้แสดงคำว่า Fail



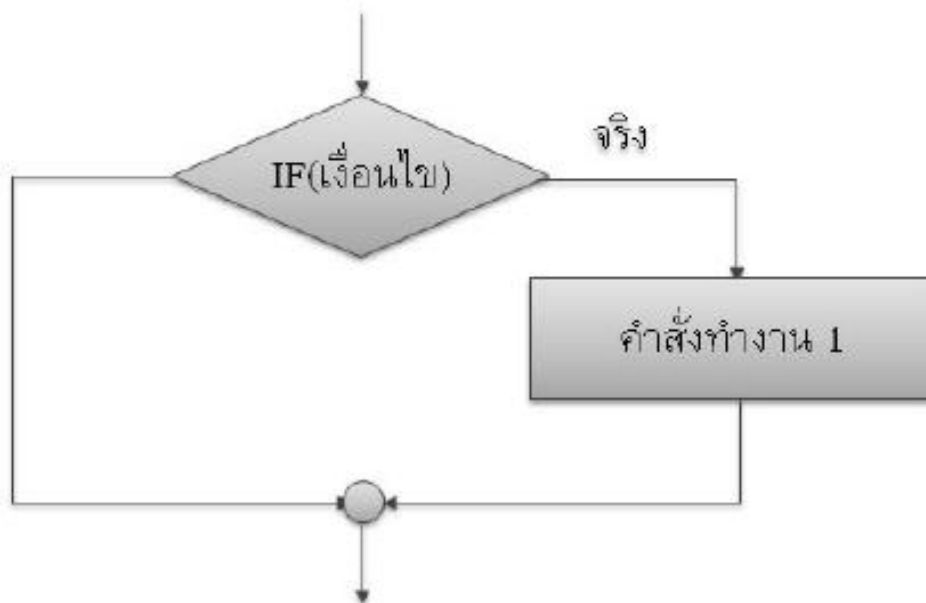
ผังงานแบบมีเงื่อนไข

- การเขียนผังงานแบบมีทางเลือกสามารถแบ่งออกเป็นลักษณะการเขียนดังนี้
 - ผังงานแบบมีทางเลือกโดยใช้คำสั่ง **IF**
 - ผังงานที่มีมากกว่า 2 ทางเลือกจากการใช้คำสั่ง **CASE**

ผังงานแบบมีเงื่อนไข

- ผังงานแบบมี 2 ทางเลือกจากการใช้คำสั่ง **IF**
- **IF** แบบทางเลือกเดียว **Single Selection**

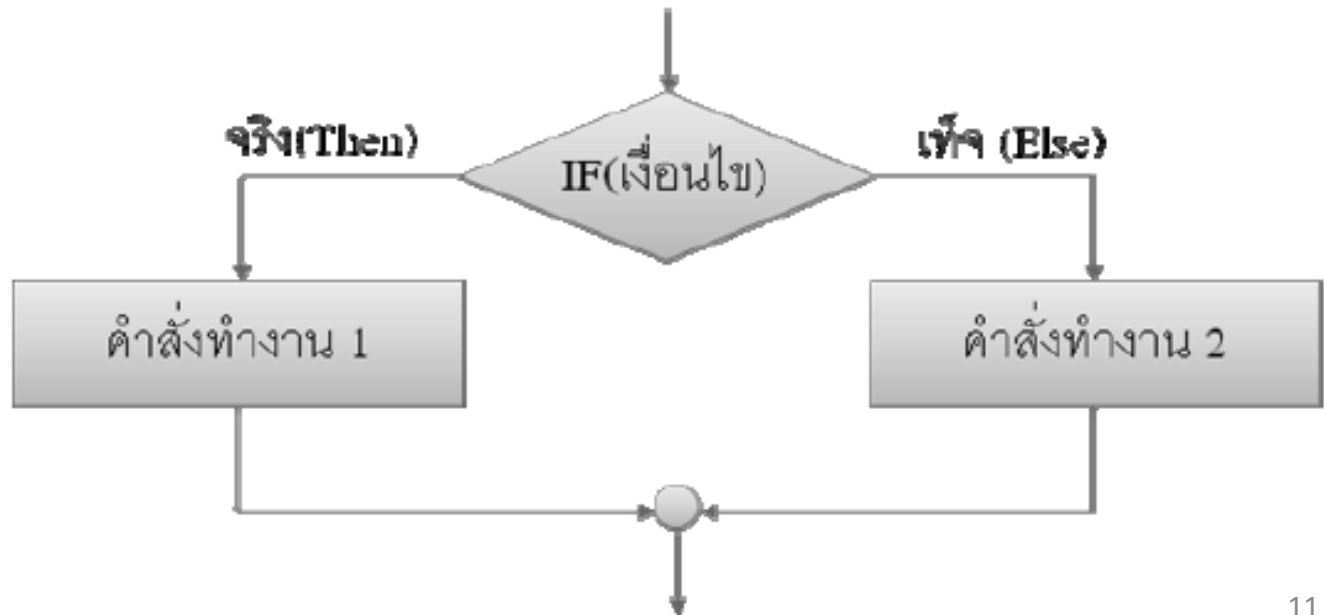
จะทำการตรวจสอบเงื่อนไข ถ้าเป็นจริง จะไปทำคำสั่งทำงาน ถ้าเป็นเท็จ ก็จะออกจากโครงสร้างโดยมีรูปแบบการเขียนดังนี้



ผังงานแบบมีเงื่อนไข

- **IF** แบบสองทางเลือก **Double Selection**

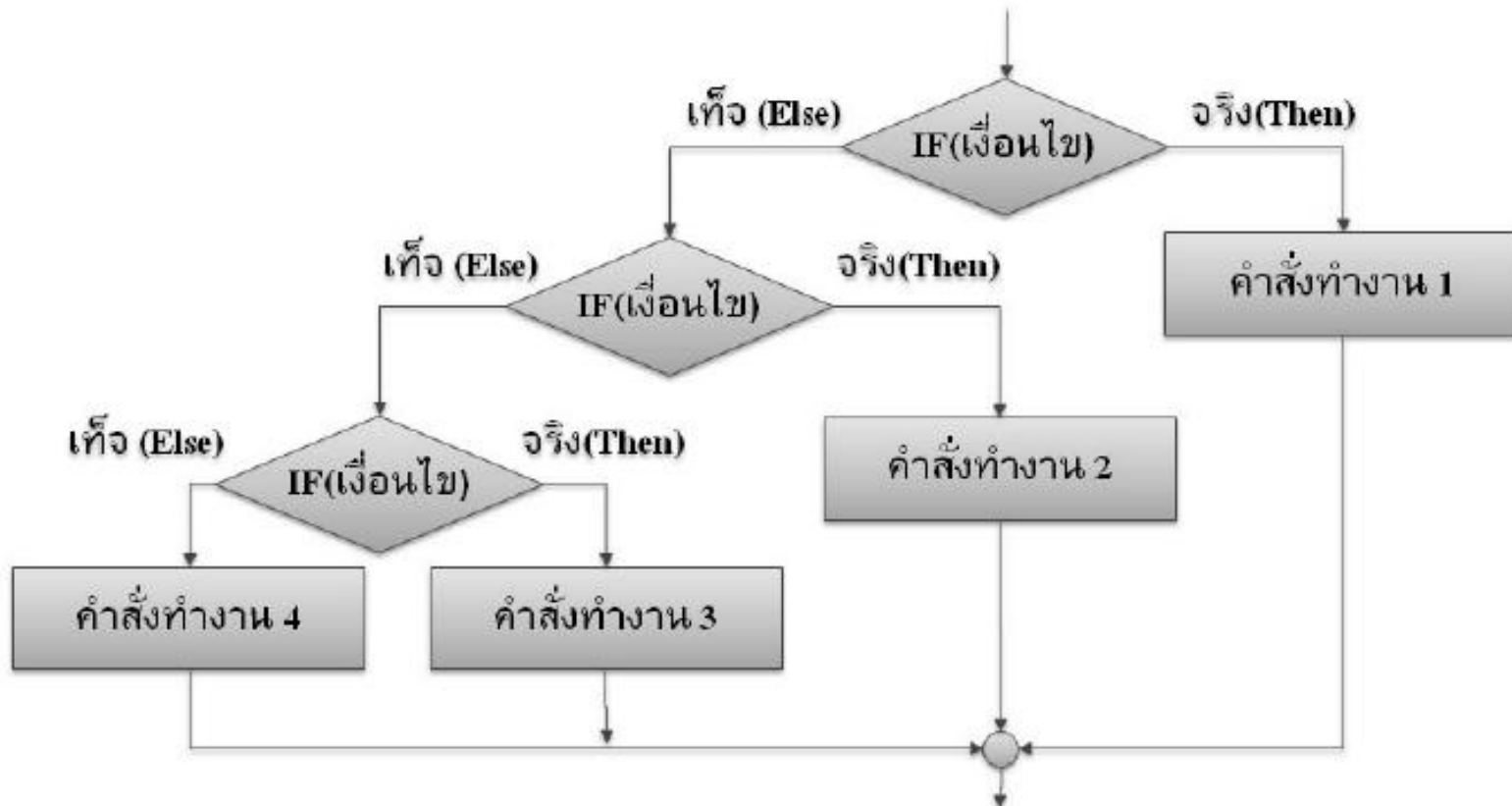
จะทำการตรวจสอบเงื่อนไข ถ้าเป็นจริง จะไปทำคำสั่งทำงาน 1 แล้วออกจากโครงสร้าง ถ้าเป็นเท็จจะไปทำคำสั่งทำงาน 2 แล้วออกจากโครงสร้าง ซึ่งมีรูปแบบการเขียนดังนี้



ผังงานแบบมีเงื่อนไข

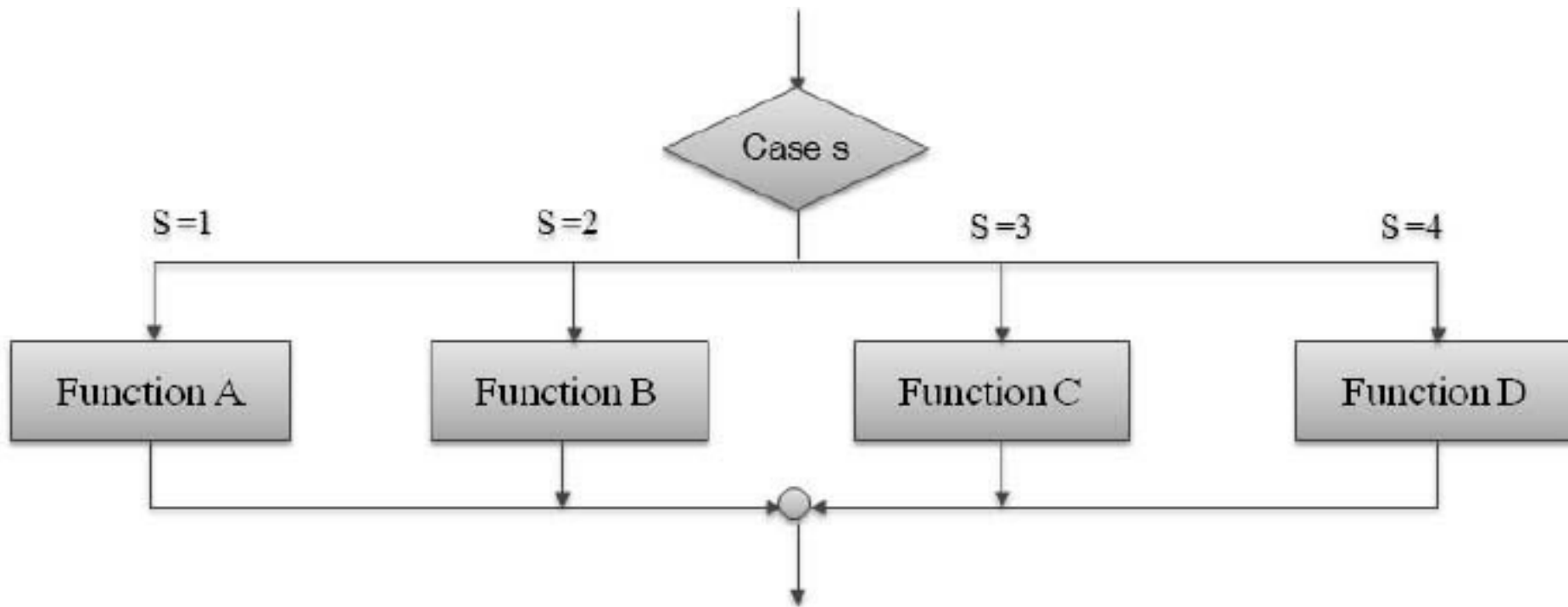
- **IF** แบบหลายทางเลือก **Multi Selection**

เป็นการตรวจสอบแบบหลายเงื่อนไข ทำให้มีทางออกได้หลายทาง ซึ่งเป็นการนำโครงสร้างของ **IF** มาซ้อนกันเพื่อให้ได้เงื่อนไขมากขึ้นตามความต้องการ โดยมีรูปแบบการเขียนดังนี้



ผังงานที่มีมากกว่า 2 ทางเลือก CASE

เป็นการตรวจสอบเงื่อนไขของตัวแปรตรวจสอบ ว่ามีค่าตรงกับค่าใด ค่าดังกล่าวจะเป็นตัวบอกกิจกรรมที่ต้องทำ โดยมีรูปแบบการเขียนดังนี้



ผังงานแบบวนซ้ำ

- การเขียนผังงานแบบมีการทำงานวนซ้ำ (**Iteration Structure**)

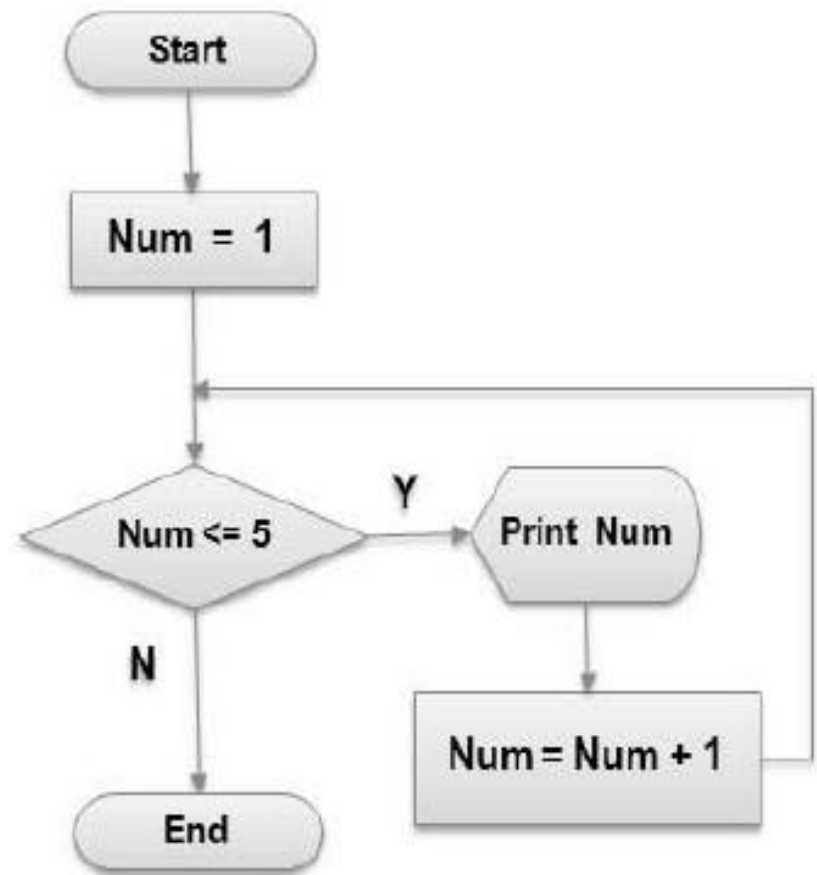
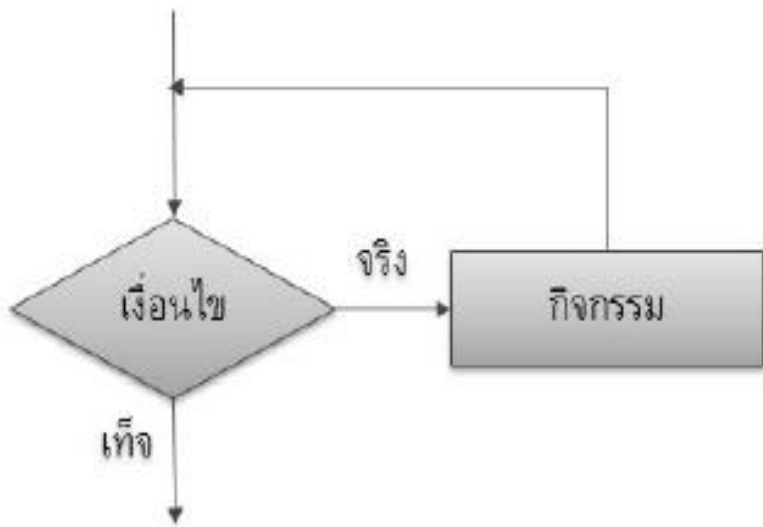
โปรแกรมส่วนใหญ่จะมีคำสั่งสำหรับการทำงานซ้ำหรือเรียกว่า ลูป (**Loop**) โดยการซ้ำของโปรแกรมจะอยู่ภายใต้เงื่อนไข จริง หรือ เท็จ ตามที่ผู้เขียนโปรแกรมได้ออกแบบไว้ รูปแบบของการวนซ้ำมีดังนี้

- **While**

โครงสร้างคำสั่งจะทำการตรวจสอบเงื่อนไขก่อน ถ้าเงื่อนไขเป็น จริง ก็จะทำกิจกรรมนั้นซ้ำไปเรื่อยๆ แต่เมื่อเงื่อนไขเป็น เท็จ จะหยุดทำซ้ำแล้วออกจากลูปไป มีรูปแบบการเขียนดังนี้

ผังงานแบบวนซ้ำ

ตัวอย่าง โปรแกรมแสดงตัวเลข 1 - 5 โดยใช้ While



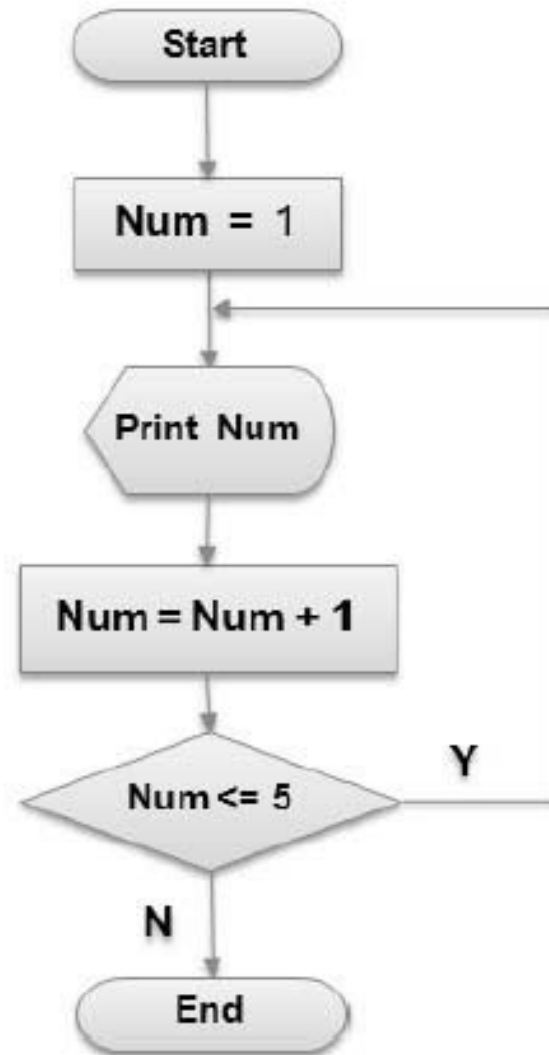
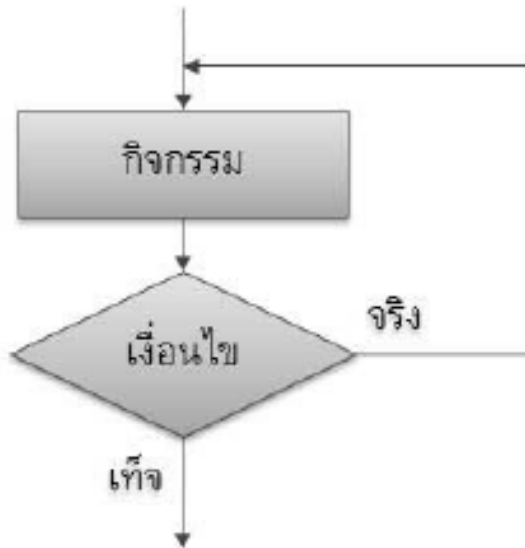
ผังงานแบบวนซ้ำ

- **Do while**

ทำกิจกรรมที่ต้องการก่อน แล้วจึงตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริง ก็จะทำกิจกรรมนั้นซ้ำไปเรื่อยๆ แต่เมื่อเงื่อนไขเป็นเท็จ จะหยุดทำซ้ำแล้วออกจากลูปไป มีรูปแบบการเขียนดังนี้

ผังงานแบบวนซ้ำ

ตัวอย่าง โปรแกรมแสดงตัวเลข 1 – 5 โดยใช้ Do While

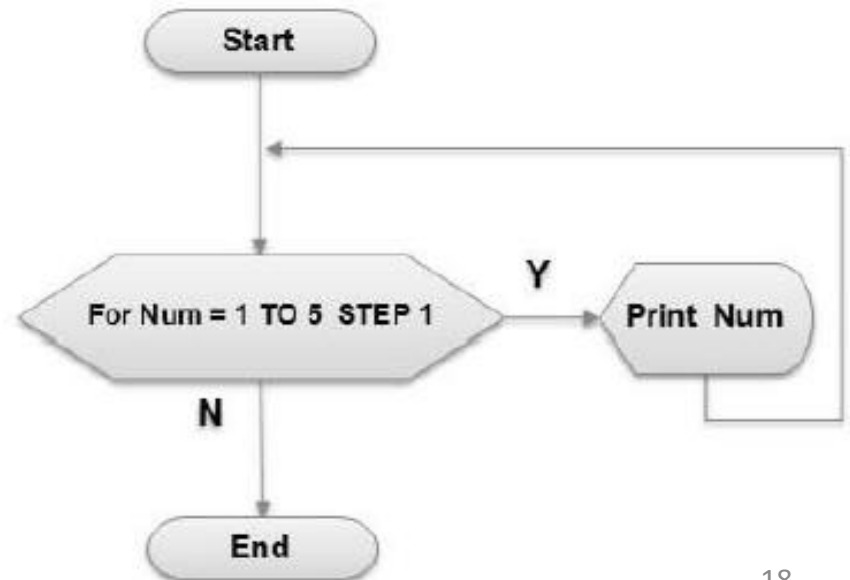
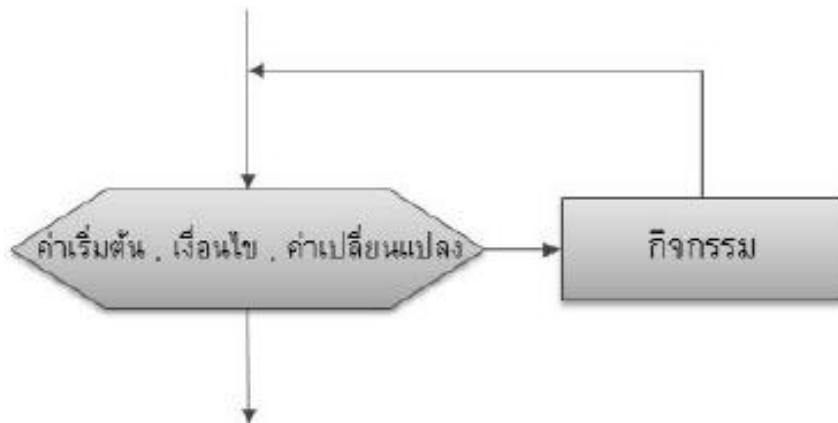


ผังงานแบบวนซ้ำ

- **For**

โครงสร้างคำสั่งจะทำการวนซ้ำโดยรู้จำนวนแน่นอน โดยมีการกำหนดค่าเริ่มต้น ค่าสิ้นสุด ของตัวแปรนับรอบ มีรูปแบบการเขียนดังนี้

ตัวอย่าง โปรแกรมแสดงตัวเลข 1 – 5 โดยใช้ For



ตัวอย่างการเขียนผังงานจากโจทย์ปัญหา

- จากโจทย์ปัญหาวิเคราะห์หาตัวแปรที่ใช้ในอัลกอริทึม นำมาเขียน **Pesudo Code** และเขียนผังงาน

ตัวอย่าง 1 จงเขียน **Flowchart** รับข้อมูลจำนวนมะม่วงที่ซื้อ เพื่อคำนวณหาราคามะม่วงที่ต้องจ่าย โดยมีเงื่อนไขว่า ถ้าซื้อมะม่วงน้อยกว่าหรือเท่ากับ 5 กิโลกรัม คิดราคากิโลกรัมละ 35 บาท แต่ถ้าซื้อมะม่วงน้อยกว่า 5 กิโลกรัมขึ้นไป คิดราคากิโลกรัมละ 30 บาท

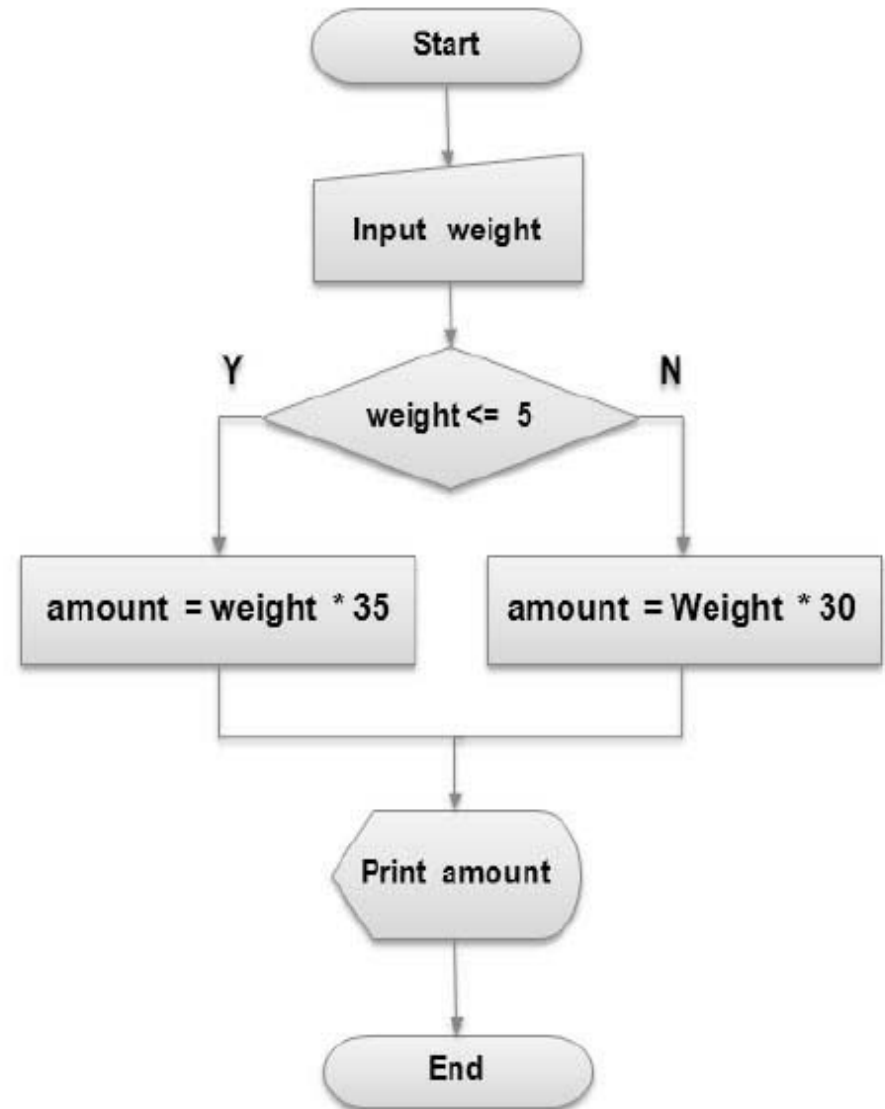
- กำหนดตัวแปรที่ใช้
- นำหนักของส้มที่ซื้อ ใช้ชื่อตัวแปร **weight**
- ราคาส้มที่ต้องจ่าย ใช้ชื่อตัวแปร **amount**

ตัวอย่างการเขียนผังงานจากโจทย์ปัญหา

Pseudo code

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

Flowchart



ตัวอย่างการเขียนผังงานจากโจทย์ปัญหา

ตัวอย่างที่ 2 จงเขียนผังงานเพื่อรับตัวเลขจำนวน n ตัว ตามจำนวนที่ผู้ใช้ต้องการ และหาผลรวมของตัวเลขทั้งหมดนั้นกำหนดตัวแปรที่ใช้ในผังงาน

จำนวนตัวเลข ใช้ชื่อตัวแปร n

ค่าของตัวเลข ใช้ชื่อตัวแปร **number**

ผลรวม ใช้ชื่อตัวแปร **result**

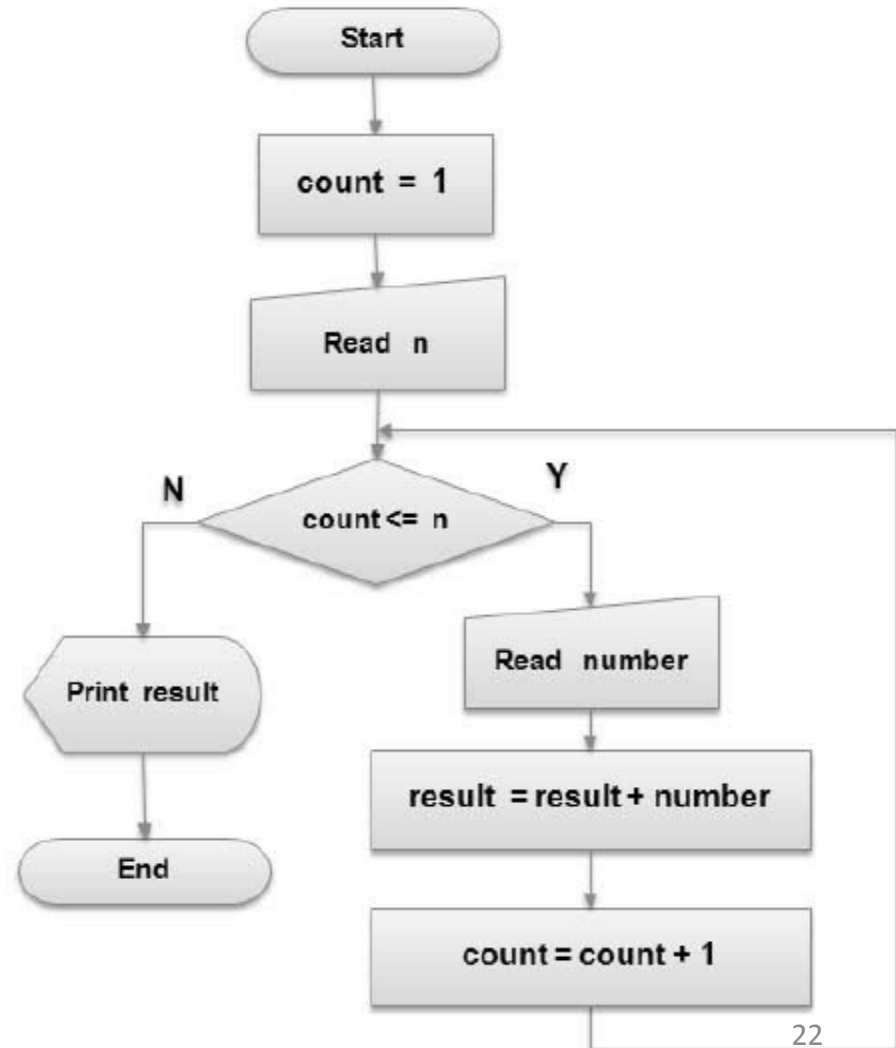
ตัวนับจำนวน “ครั้งที่ของการป้อนตัวเลข” ใช้ชื่อตัวแปร **count**

ตัวอย่างการเขียนผังงานจากโจทย์ปัญหา

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



ตัวอย่างการเขียนผังงานจากโจทย์ปัญหา

ตัวอย่างที่ 3 จงเขียนผังงานเพื่อรับตัวเลขครั้งละ 1 จำนวน และให้ตรวจสอบ ถ้าตัวเลขที่รับเข้ามามีค่ามากกว่า 50 ให้แสดงตัวเลขนั้นบนหน้าจอ จากนั้นวนรับตัวเลขแล้วทำซ้ำจนครบ 10 จำนวน กำหนดตัวแปรที่ใช้ในผังงาน

ค่าของตัวเลข ใช้ชื่อตัวแปร **number**

ตัวนับจำนวน “ครั้งที่ของการป้อนตัวเลข” ใช้ชื่อตัวแปร **count**

ตัวอย่างการเขียนผังงานจากโจทย์ปัญหา

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
```

