บทที่ 1

1. การออกแบบและพัฒนาโปรแกรมด้วยผังงาน (Flowchart)

(วิธีการเขียนขั้นตอนแก้ปัญหาผังงาน (Flowchart))

สื่อสไลด์นี้ใช้รูปแบบฟอนต์ สารบรรณ รุ่นปรับปรุงใหม่ “Sarabun New”
สามารถดาวน์โหลดฟอนต์ได้ที่ URL : https://www.f0nt.com/release/th-sarabun-new/

1 การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรม

คอมพิวเตอร์กับการแก้ปัญหา

การทำงานของเครื่องคอมพิวเตอร์ จะทำตามโปรแกรมที่เขียนขึ้น ดังนั้นการนำเครื่องคอมพิวเตอร์มาช่วยสำหรับการแก้ปัญหา จึงต้องมีโปรแกรมสำหรับการแก้ปัญหาในงานนั้น ๆ

ขั้นตอนของการวิเคราะห์ปัญหา คือ การนำปัญหาของงาน มาสร้างเป็นขั้นตอนวิธีในรูปแบบผังงาน หรือซูโดโค้ด โดยเรียงลำดับการทำงาน เพื่อนำไปใช้ในการเขียนโปรแกรมด้วยภาษาต่าง ๆ ต่อไป

1 การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรม

ขั้นตอนของการแก้ปัญหา (Problem Solving) สำหรับเตรียมการก่อนลงมือเขียนโปรแกรมคอมพิวเตอร์ มี 5 ขั้นตอนดังนี้

ขั้นตอนการแก้ปัญหา

Problem Specification Phase

การทำความเข้าใจกับปัญหา

Input and Output Specification Phase

พิจารณาลักษณะของข้อมูลเข้าและข้อมูลออก

Hand Example Phase

การทดลองแก้ไขปัญหาด้วยตนเอง

Algorithm Deverlopment Phase

การเขียนขั้นตอนวิธีการแก้ปัญหา

Testing Phase

การทดสอบขั้นตอนวิธีการแก้ปัญหา

1 การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรม

- การเขียนขั้นตอนวิธีการแก้ปัญหา (Algorithm Development Phase)

นำขั้นตอนของการทดลองการแก้ไขปัญหาด้วยตนเอง มาทำการเรียบเรียงเป็นลำดับขั้นตอนวิธีการทำงาน โดยเขียนเป็นข้อ ๆ ตั้งแต่ขั้นตอนแรกจนถึงขั้นตอนสุดท้าย

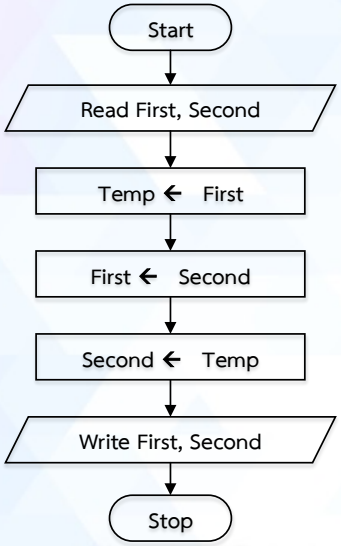
  - เขียนลำดับขั้นตอนวิธีการทำงานทั้งหมด**อย่างย่อ**
  - เขียนลำดับขั้นตอนวิธีการทำงาน**อย่างละเอียด**
  - วิธีการเขียนลำดับขั้นตอนวิธี
    - Flowcharts เขียนขั้นตอนวิธีด้วยรูปภาพผังงาน
    - Pseudo code (อ่านว่า ซูโดโค้ด) เขียนขั้นตอนวิธีด้วยภาษาที่ใกล้เคียงภาษาคอมพิวเตอร์

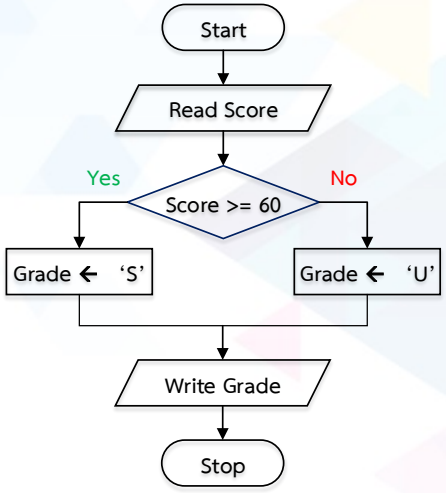
วิธีการเขียนขั้นตอนแก้ปัญหาผังงาน (Flowchart)

ผังงาน คือ การเขียนอธิบายขั้นตอนวิธีการทำงาน ในลักษณะของรูปภาพ ลักษณะของผังงานแต่ละรูปแบบก็จะมีความหมายแตกต่างกัน

ประโยชน์ของผังงาน

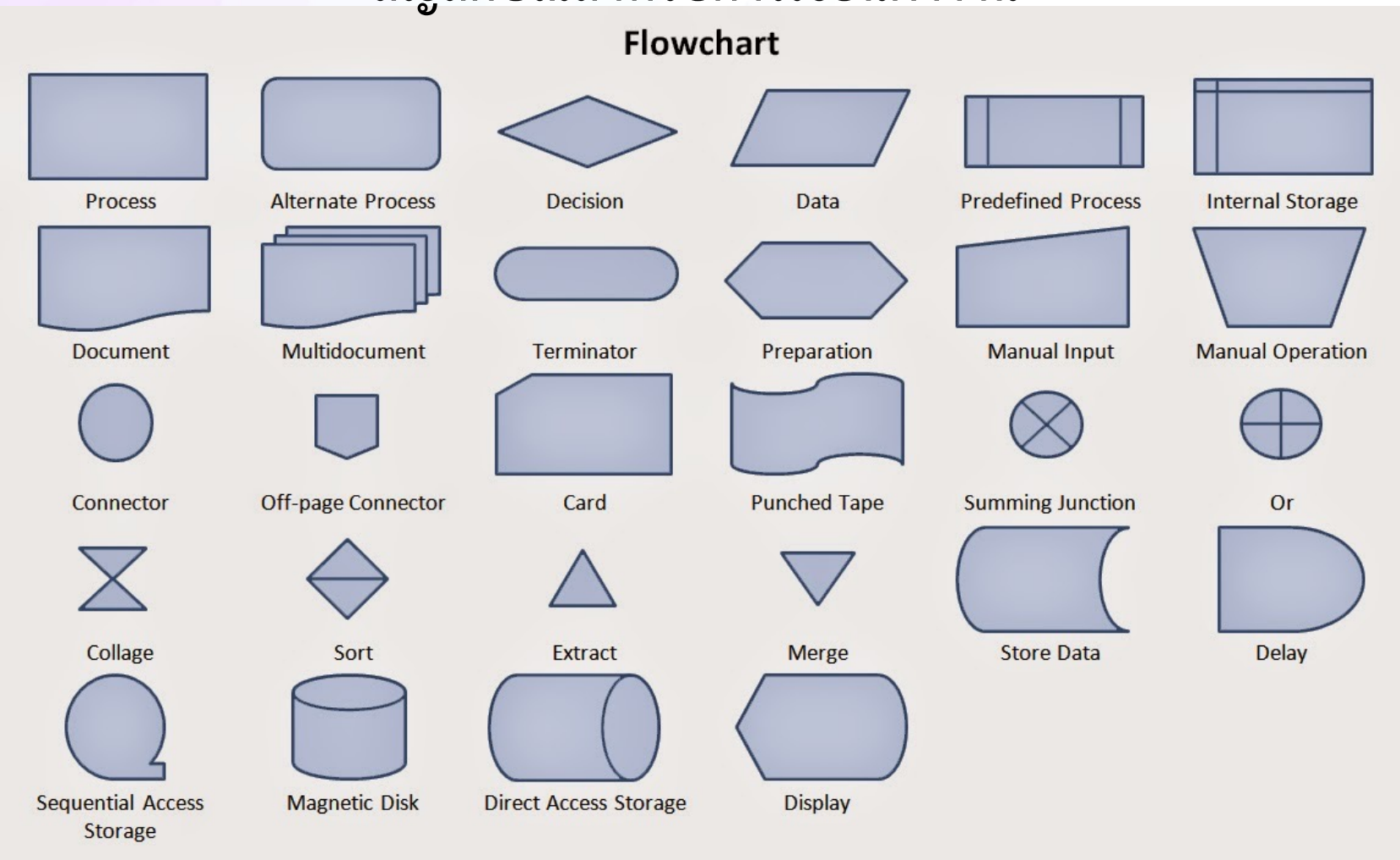
- อธิบายขั้นตอนวิธีการทำงานในลักษณะของรูปภาพ
- เห็นลำดับและทิศทางของขั้นตอนวิธีการทำงานได้ชัดเจนกว่าข้อความ





2. ผังงาน

สัญลักษณ์สำหรับการเขียนผังงาน

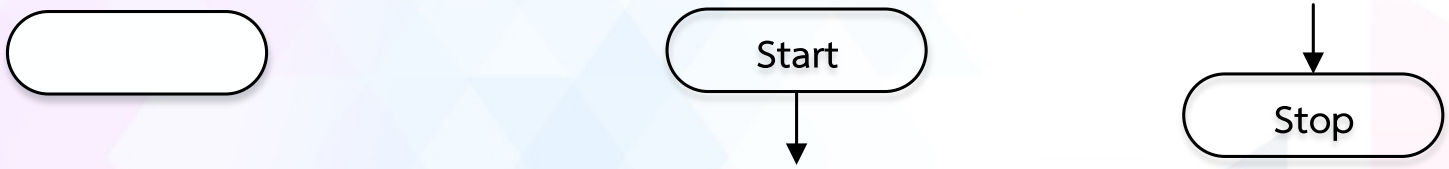


http://meteeblog.blogspot.com/2015/01/flowchart.html

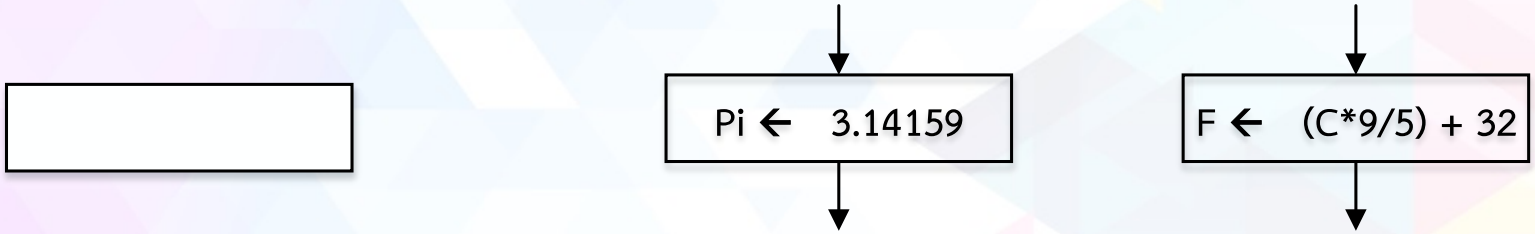
2. ผังงาน

สัญลักษณ์สำหรับการเขียนผังงาน

- Terminator จุดเริ่มต้นและจุดสิ้นสุดของผังงาน



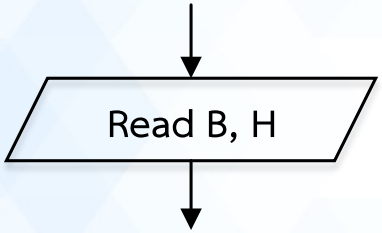
- Process การกำหนดค่า **การคำนวณ** และการประมวลผลทั่วไป

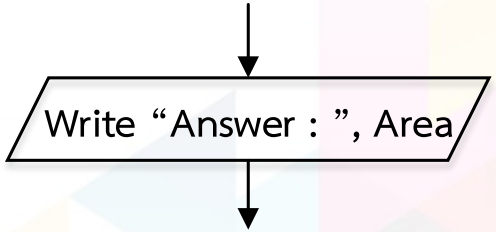


2. ผังงาน

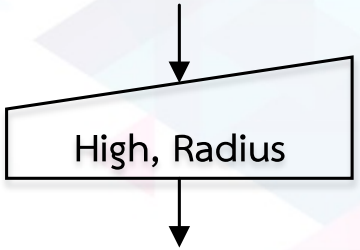
สัญลักษณ์สำหรับการเขียนผังงาน

- Input/Output (Data) การรับข้อมูลเข้าและการแสดงผลข้อมูล โดยไม่ระบุชนิดอุปกรณ์





- Manual Input (Keyboard) การรับข้อมูลเข้าทางแป้นพิมพ์



2. ผังงาน

สัญลักษณ์สำหรับการเขียนผังงาน

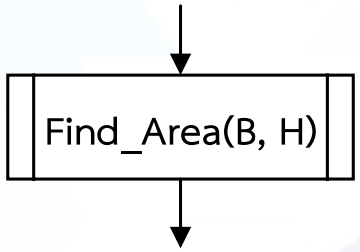
- Display (Monitor) การแสดงผลข้อมูลออกทาง**จอภาพ**

- Document (Printer) การแสดงผลเอกสาร หรือการแสดงผลข้อมูลออกทาง**เครื่องพิมพ์**

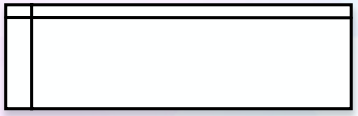
2. ผังงาน

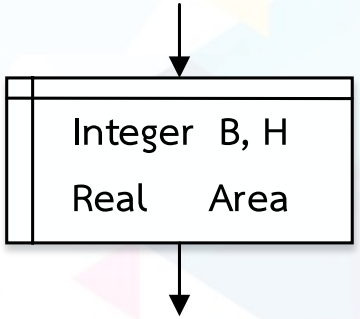
สัญลักษณ์สำหรับการเขียนผังงาน

- Predefined Process เรียกใช้โปรแกรมย่อย ฟังก์ชัน หรือโมดูล



- Internal Storage การประกาศตัวแปร

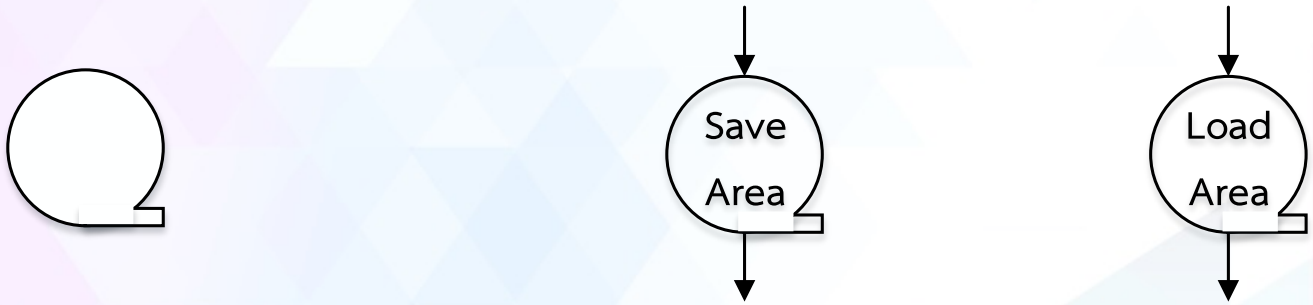




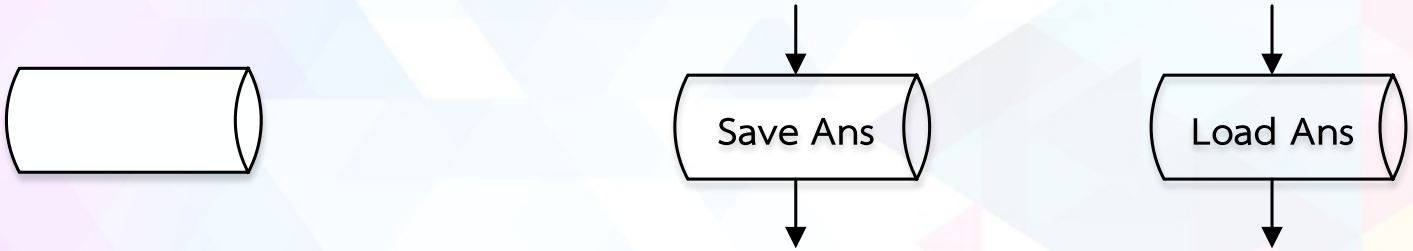
2. ผังงาน

สัญลักษณ์สำหรับการเขียนผังงาน

- Sequential Access Storage การติดต่อกับอุปกรณ์ที่เป็นการเข้าถึงแบบลำดับ



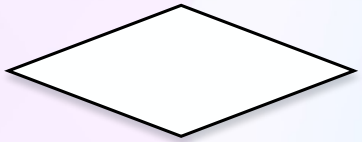
- Direct Access Storage การติดต่อกับอุปกรณ์ที่เป็นการเข้าถึงข้อมูลแบบตรง

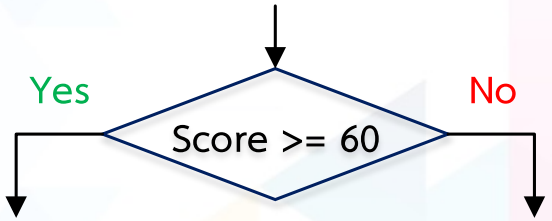


2. ผังงาน

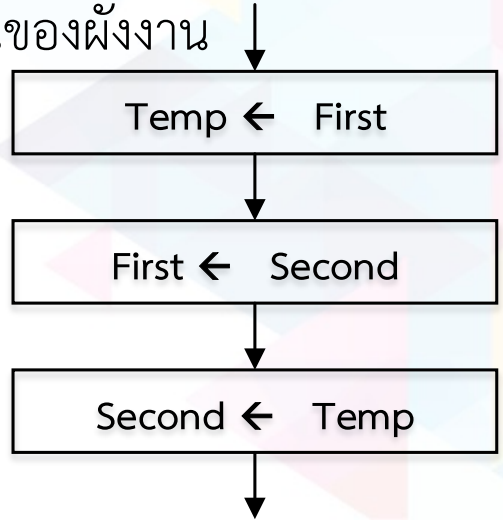
สัญลักษณ์สำหรับการเขียนผังงาน

- Decision การตัดสินใจ การเปรียบเทียบ จะมีผลใน 2 ทิศทาง คือ กรณีผลตรวจสอบเงื่อนไข เป็นเท็จ และเป็นจริง





- Flow line/Direction การแสดงทิศทางการทำงานของผังงาน

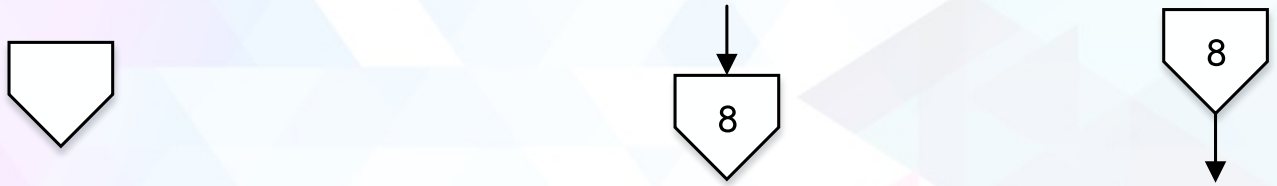


2. ผังงาน

สัญลักษณ์สำหรับการเขียนผังงาน

- On-page Connector จุดต่อภายในหน้าเดียวกัน

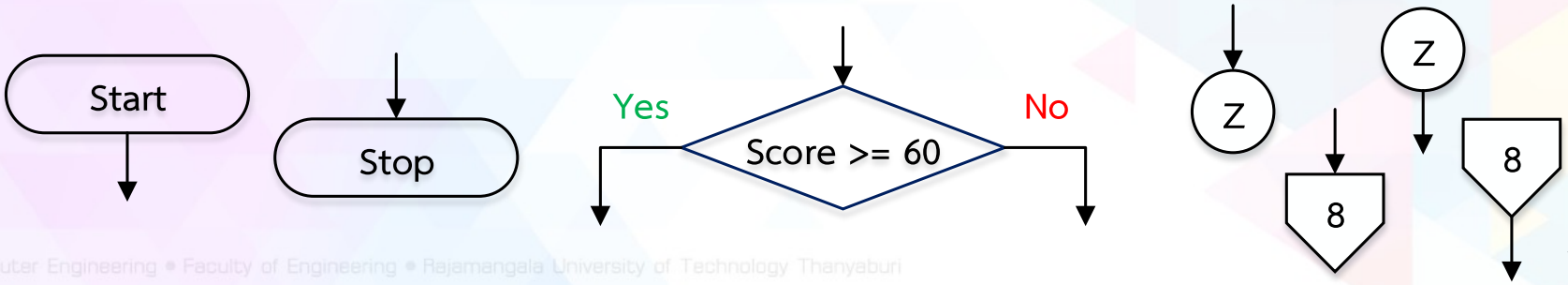
- Off-page Connector จุดต่อระหว่างหน้า หรือคนละหน้า



2. ผังงาน

ลักษณะการเขียนผังงาน

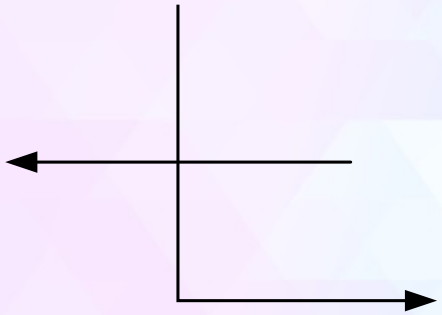
- ทุกผังงานต้องมีจุดเริ่มต้นและจุดสิ้นสุดเพียงหนึ่งแห่งเท่านั้น
- ทุกสัญลักษณ์ของผังงานต้องมีลูกศรชี้ทิศทางเข้า และลูกศรชี้ทิศทางออกอย่างละหนึ่งลูกศร ยกเว้นสัญลักษณ์จุดเริ่มต้น จุดสิ้นสุด การตัดสินใจ และ จุดต่อ
  - จุดเริ่มต้นมีเฉพาะทิศทางออก
  - จุดสิ้นสุดมีเฉพาะทิศทางเข้า
  - การตัดสินใจมีทิศทางเข้า 1 ทิศทาง มีทิศทางออก 2 ทิศทาง
  - จุดต่อมีแบบทิศทางเข้าอย่างเดียว และ ทิศทางออกอย่างเดียว



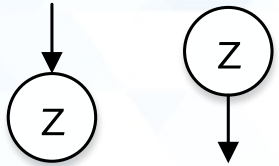
2. ผังงาน

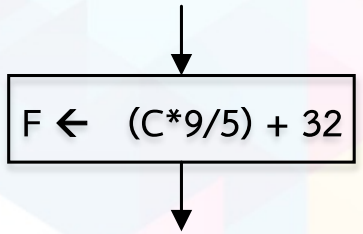
ลักษณะการเขียนผังงาน

- ทิศทางการทำงาน นิยมเขียนจากบนลงล่างหรือจากซ้ายไปขวา
- เส้นของลูกศรที่ใช้บอกทิศทางไม่ควรเขียนตัดกันหรือทับกัน
- ไม่ควรเขียนเส้นเชื่อมโยงที่อยู่ห่างกันมาก ควรใช้สัญลักษณ์จุดต่อแทน
- นิยมใช้เครื่อง หมายลูกศร (←) แทนการใช้เครื่องหมายเท่ากับ (=)



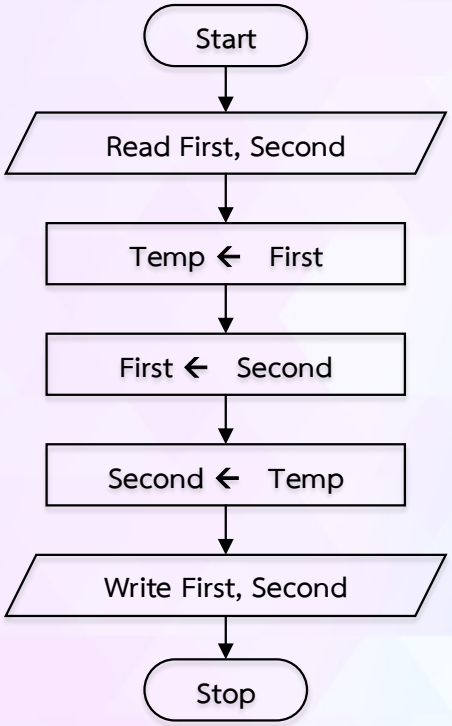
ห้ามวาดเส้นทับกันลักษณะนี้

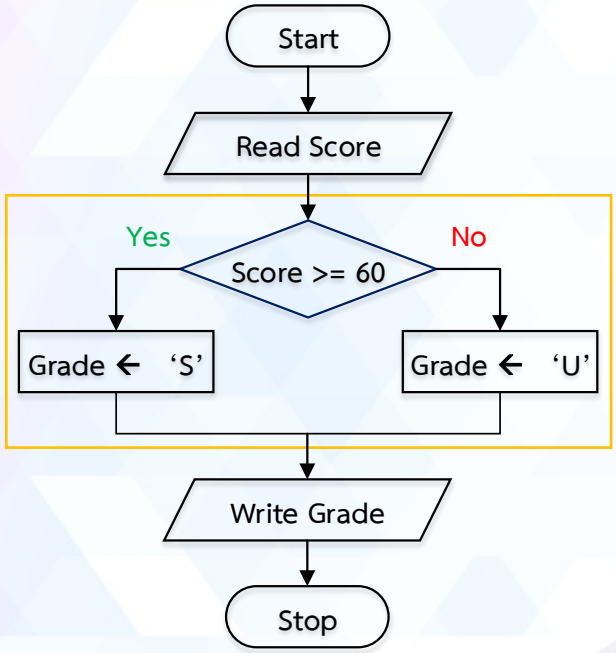


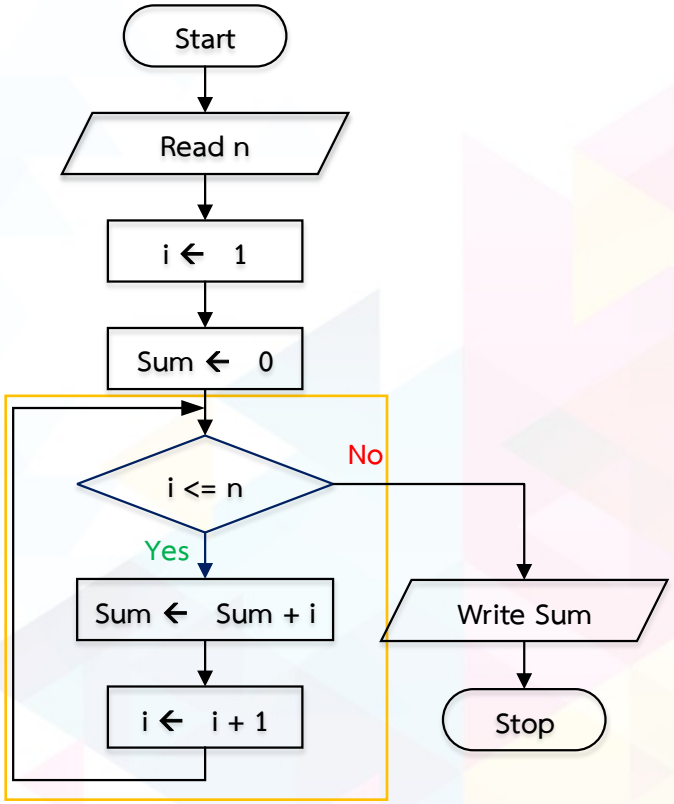


2. ผังงาน

ตัวอย่าง ลักษณะผังงาน มี 3 แบบ





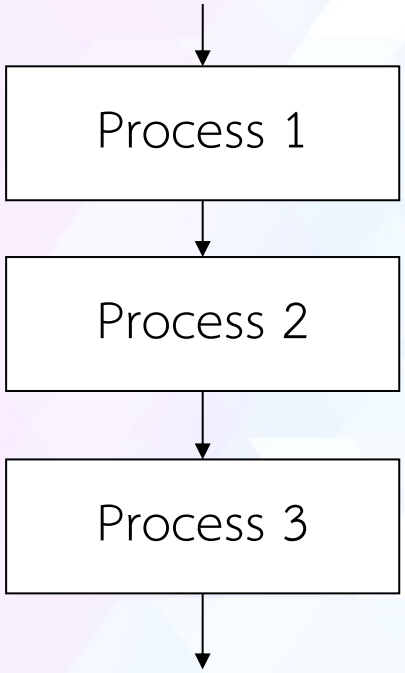


1. การทำงานแบบลำดับ

2. การทำงานแบบเลือกทำ

3. การทำงานแบบทำซ้ำ

ผังงานแบบลำดับ

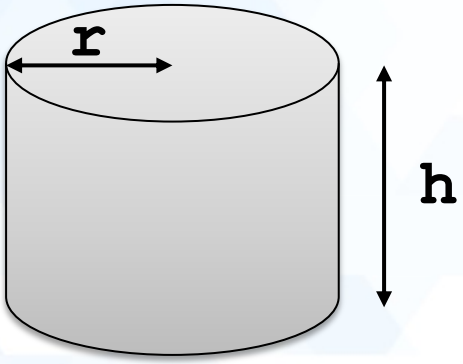


- เป็นโครงสร้างพื้นฐานของทุกผังงาน
- ทำงานทีละขั้นตอน
- ทำงานตามทิศทางของลูกศร

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบลำดับ

ตัวอย่างที่1 ขั้นตอนวิธีการแก้ปัญหาด้วยคอมพิวเตอร์

จงเขียนขั้นตอนวิธีการแก้ปัญหาสำหรับการหาค่าปริมาตรของรูปทรงกระบอก



สูตรการคำนวณ ปริมาตรทรงกระบอก = $\pi \times r^2 \times h$

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบลำดับ

- **ขั้นตอนที่ 1** ทำความเข้าใจกับปัญหา

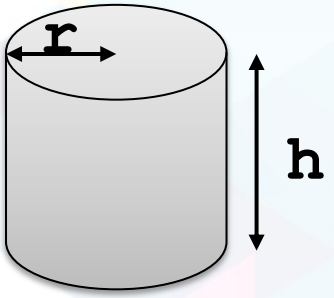
  การคำนวณหาปริมาตรของรูปทรงกระบอก

- **ขั้นตอนที่ 2** ลักษณะของข้อมูลเข้าและข้อมูลออก

  ข้อมูลเข้า คือ ความสูงของทรงกระบอก และรัศมีของวงกลมที่เป็นฐานของทรงกระบอก

  ข้อมูลออก คือ ค่าของปริมาตรทรงกระบอก

⚠ คำถาม ข้อมูลเข้าและออกควรเป็นจำนวนเต็ม หรือ จำนวนจริง ?



ปริมาตรทรงกระบอก = $\pi \times r^2 \times h$

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบลำดับ

- **ขั้นตอนที่ 3** ทดลองแก้ไขปัญหาด้วยตนเอง

  การคำนวณหาปริมาตรทรงกระบอก หาได้จากสูตร

  ปริมาตรทรงกระบอก = $\pi \times r^2 \times h$

  ถ้า รัศมีของฐาน = 5 หน่วย

  ความสูง = 12 หน่วย

  π คือ ค่าคงที่ มีค่า 3.14159 (เป็นเลขจำนวนจริง)

  ปริมาตรทรงกระบอก = $3.14159 \times 5^2 \times 12$

  = 942.477 หน่วย^3

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบลำดับ

- ขั้นตอนที่ 4 การพัฒนาลำดับขั้นตอนวิธีการแก้ปัญหา

วิธีการเขียนขั้นตอนแก้ปัญหา

1) รับข้อมูล

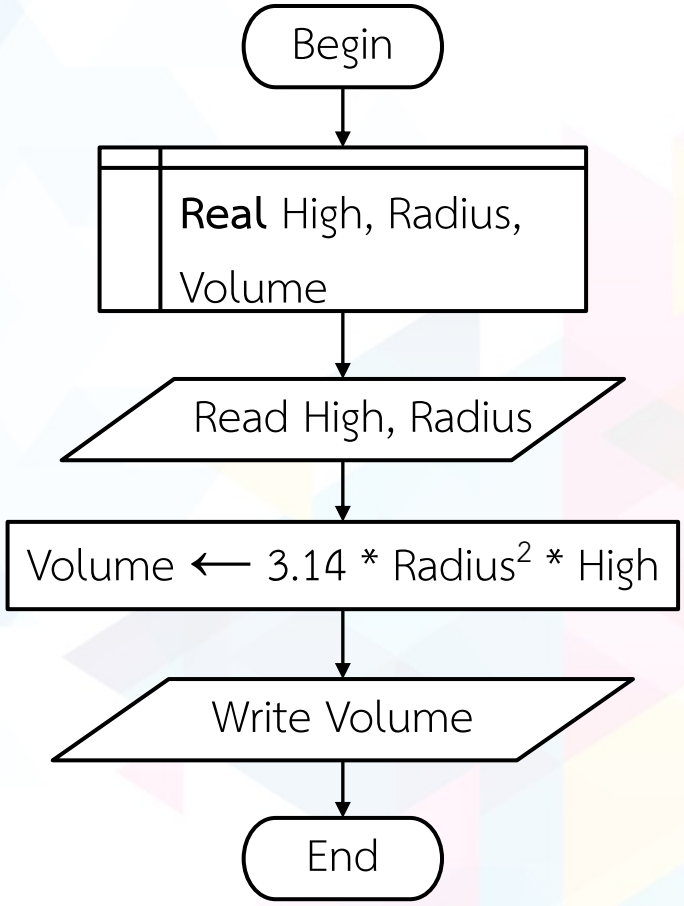
High และ Radius

2) คำนวณ

Volume = 3.14 * Radius^2 * High

3) เขียนข้อมูล Volume

ผังงาน (Flowchart)



2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบลำดับ

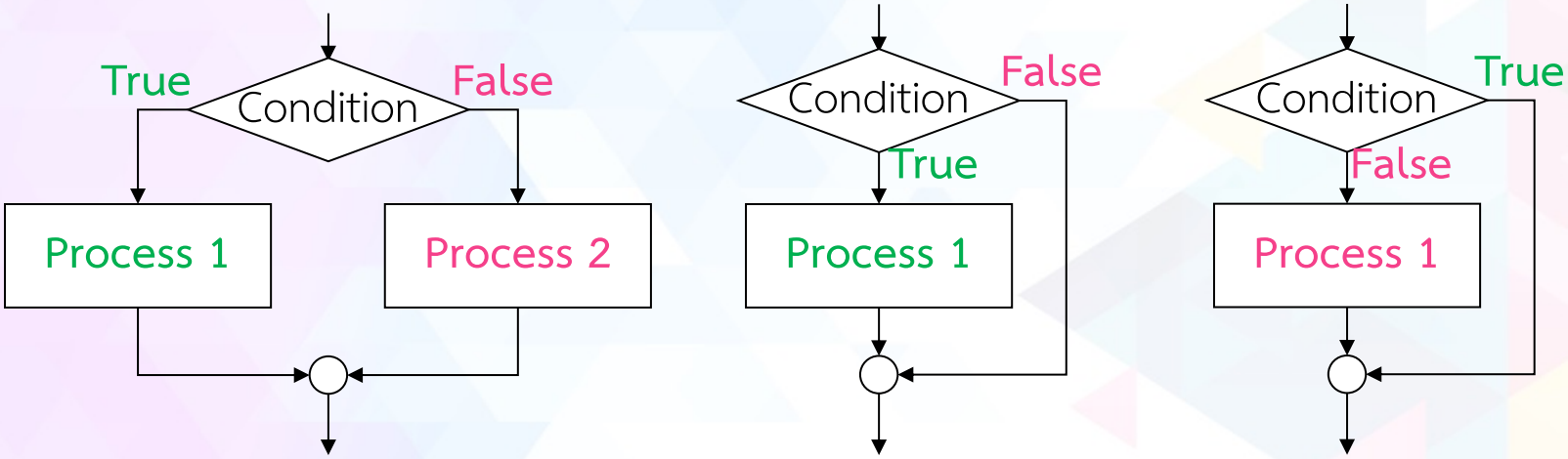
- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา
  - เตรียมข้อมูล หรือ ผลลัพธ์ที่คาดไว้ สำหรับการทดสอบโปรแกรมที่พัฒนาขึ้นในขั้นตอนที่ 4

1. *เครื่องคอมพิวเตอร์จะรอรับข้อมูล 2 ค่า ป้อนค่า ความสูง = 12 และรัศมีของฐาน = 5*
2. *คำนวณหาค่าปริมาตรตามสูตร = $3.14159 * 5^2 * 12 = 942.477$ ลูกบาศก์หน่วย*
3. *แสดงค่าของปริมาตรทรงกระบอกที่คำนวณได้ คือ 942.477 ลูกบาศก์หน่วย*

การทำงานแบบเลือกทำ

ผังงานแบบเลือกทำ

- ใช้เลือกทำงานให้เหมาะสมกับข้อมูลที่ประมวลผลอยู่ในขณะนั้น
- ผังงานแบบเลือกทำประกอบด้วย **สัญลักษณ์การตัดสินใจ** 1 สัญลักษณ์ ส่วนการทำงานหลังการตัดสินใจ จะอยู่เป็นลำดับถัดไป



2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบเลือกทำ

ตัวอย่างที่2 ขั้นตอนวิธีการแก้ปัญหาด้วยคอมพิวเตอร์

จงเขียนขั้นตอนวิธีการแก้ปัญหาสำหรับการตัดเกรดในวิชาฝึกงานโดยเกรดมีสองเกรดคือ ผ่าน (S) และไม่ผ่าน (U)

เกณฑ์การตัดเกรด	คะแนน 60-100	ได้เกรด S
	คะแนน 0-59	ได้เกรด U

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบเลือกทำ

- ขั้นตอนที่ 1 ทำความเข้าใจกับปัญหา

  ตัดเกรด ผ่าน(S) และไม่ผ่าน(U)

- ขั้นตอนที่ 2 ลักษณะของข้อมูลเข้าและข้อมูลออก

  ข้อมูลเข้า คือ คะแนนประเมินเป็นข้อมูลชนิดตัวเลข

  ข้อมูลออก คือ เกรด S และ U เป็นข้อมูลชนิดตัวอักษร

- ขั้นตอนที่ 3 ทดลองแก้ไขปัญหาด้วยตนเอง

  การตัดเกรดผ่านและไม่ผ่าน

  ถ้าได้คะแนนประเมิน = 70

  คะแนนอยู่ในช่วง 60-100 ได้เกรดเป็น S

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบเลือกทำ

- **ขั้นตอนที่ 4** การพัฒนาลำดับขั้นตอนวิธีการแก้ปัญหา

  เขียนอธิบายขั้นตอนวิธีการทำงาน**อย่างละเอียด**

1. เริ่มต้น
2. รับค่าคะแนนประเมิน

3. ถ้า คะแนนประเมินมากกว่าหรือเท่ากับ (>=) 60 แล้ว

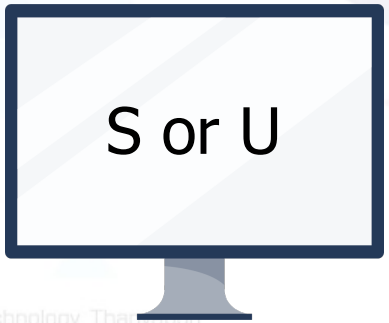
   3.1 ให้เกรด S

   มิฉะนั้น

   3.2 ให้เกรด U

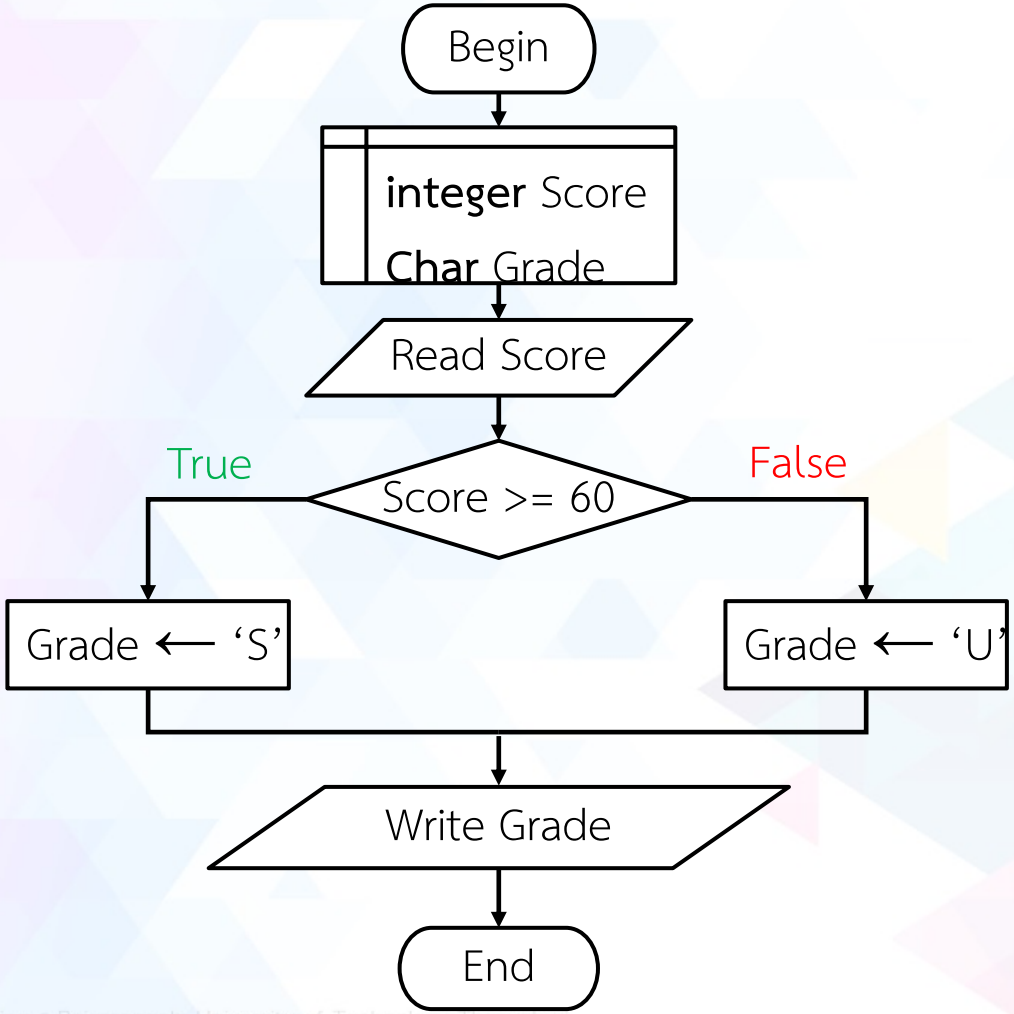
   คำนวณคิดเกรด

4. แสดงค่าเกรด
5. จบการทำงาน



2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบเลือกทำ

- ขั้นตอนที่ 4 การพัฒนาลำดับขั้นตอนวิธีการแก้ปัญหาผังงาน (Flowchart)



2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบเลือกทำ

- **ขั้นตอนที่ 5** การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case :

Case 1 : Score = 120 ได้ผลลัพธ์ S

Case 2 : Score = 69 ได้ผลลัพธ์ S

Case 3 : Score = 45 ได้ผลลัพธ์ U

Case 4 : Score = -10 ได้ผลลัพธ์ U

คำถาม : พิจารณาแต่ละ Case ว่าถูกต้องตามโจทย์งานหรือไม่

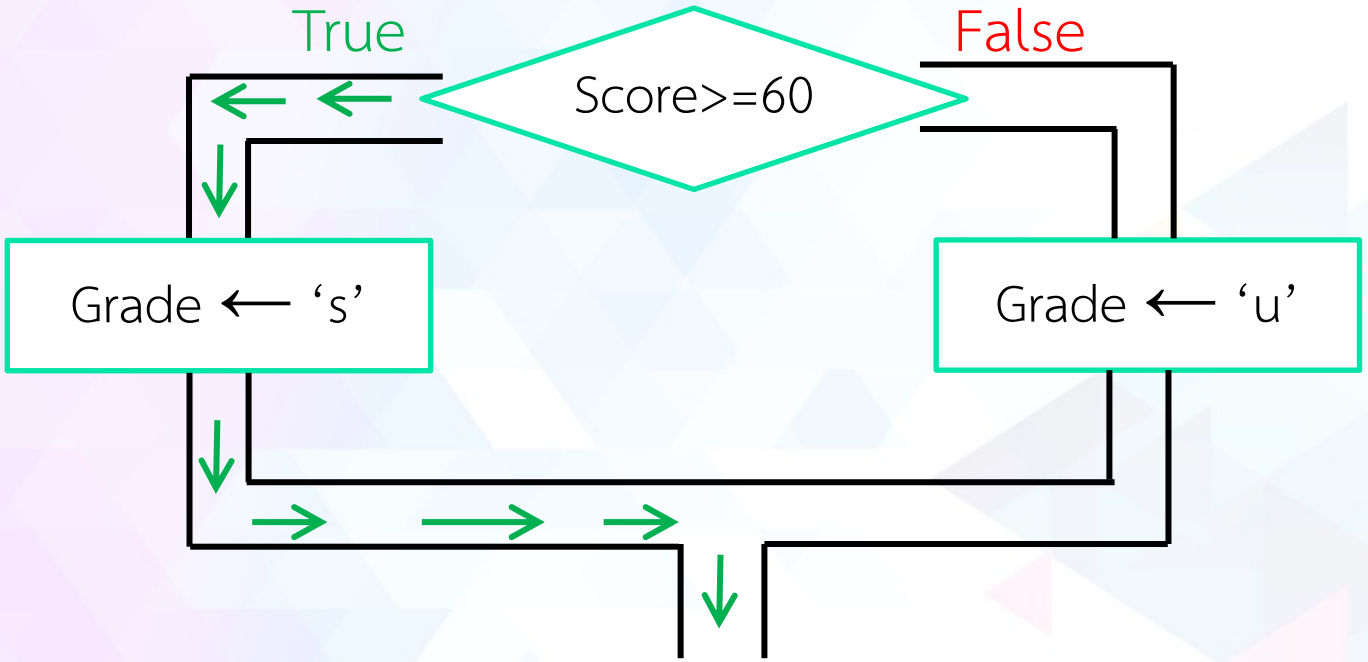
จะเห็นได้ว่าในกรณีที่ป้อนคะแนนไม่อยู่ในช่วง 0-100 ก็สามารถออกมาเป็นเกรดได้ เพราะจากขั้นตอนวิธี เราใช้ 60 เป็นตัวแบ่งช่วงเกรด

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบเลือกทำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case : Case 1 : Score = 120 ได้ผลลัพธ์ S

Case 2 : Score = 69 ได้ผลลัพธ์ S

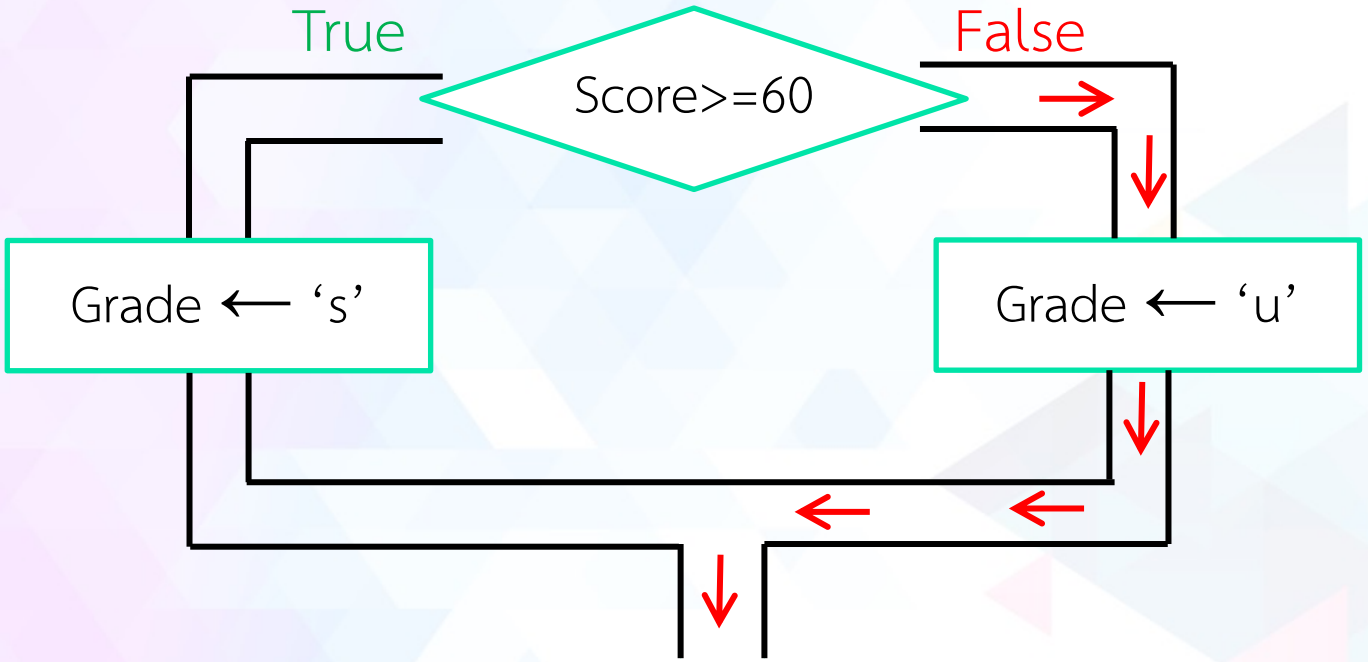


2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบเลือกทำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case : Case 3 : Score = 45. ได้ผลลัพธ์ U

Case 4 : Score = -10. ได้ผลลัพธ์ U

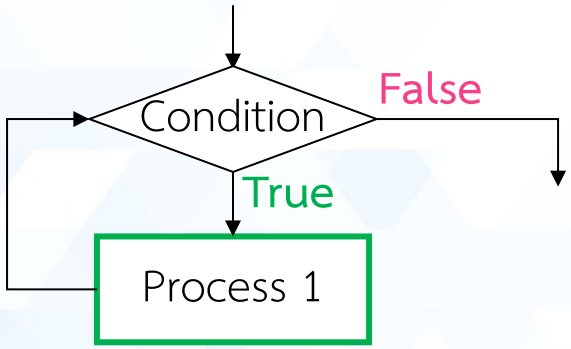


การทำงานแบบทำซ้ำ

ผังงานแบบทำซ้ำ

- ใช้ในกรณีที่ต้องการทำงานใดงานหนึ่งมากกว่า 1 ครั้ง
- โครงสร้างของผังงานแบบทำซ้ำ ต้องมี **สัญลักษณ์การตัดสินใจ** เพื่อตรวจสอบเงื่อนไขว่า จะเข้าสู่ขั้นตอนการทำซ้ำหรือไม่
- การทำซ้ำ แบ่งได้เป็น 2 ลักษณะ คือ
  - ทำซ้ำในขณะที่ (While-Do)
  - ทำซ้ำจนกระทั่ง (Repeat-Until)

ผังงานแบบทำซ้ำในขณะที่



- ขั้นตอนแรกของการทำงาน คือ การตัดสินใจ
- หากเงื่อนไขเป็นจริง จะทำ Process 1 ซึ่งเรียกว่า **“ขั้นตอนการทำซ้ำ”**
- เมื่อทำเสร็จ จะกลับไปตรวจสอบเงื่อนไขอีกครั้ง
- หากเงื่อนไขเป็นเท็จ จะ**ออกจากขั้นตอนการทำซ้ำ** เพื่อทำงานอื่นต่อไป

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

ตัวอย่างที่3 ขั้นตอนวิธีการแก้ปัญหาด้วยคอมพิวเตอร์

จงเขียนขั้นตอนวิธีการแก้ปัญหาสำหรับการหาผลรวมของตัวเลขจำนวนเต็มตั้งแต่เลข 1 ถึง ค่า n

สูตรการคำนวณ

$$\sum_{i=1}^{n} i = 1 + 2 + \cdots + n$$

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- **ขั้นตอนที่ 1** ทำความเข้าใจกับปัญหา

  คำนวณผลรวมตัวเลขจำนวนเต็ม ตั้งแต่ 1 ถึง ค่า n

- **ขั้นตอนที่ 2** ลักษณะของข้อมูลเข้าและข้อมูลออก

  ข้อมูลเข้า คือ ตัวเลขค่าสุดท้ายที่จะคำนวณถึง เป็นข้อมูลชนิดตัวเลขจำนวนเต็ม

  ข้อมูลออก คือ ตัวเลขผลรวมที่คำนวณได้ เป็นข้อมูลชนิดตัวเลขจำนวนเต็ม

- **ขั้นตอนที่ 3** ทดลองแก้ไขปัญหาด้วยตนเอง

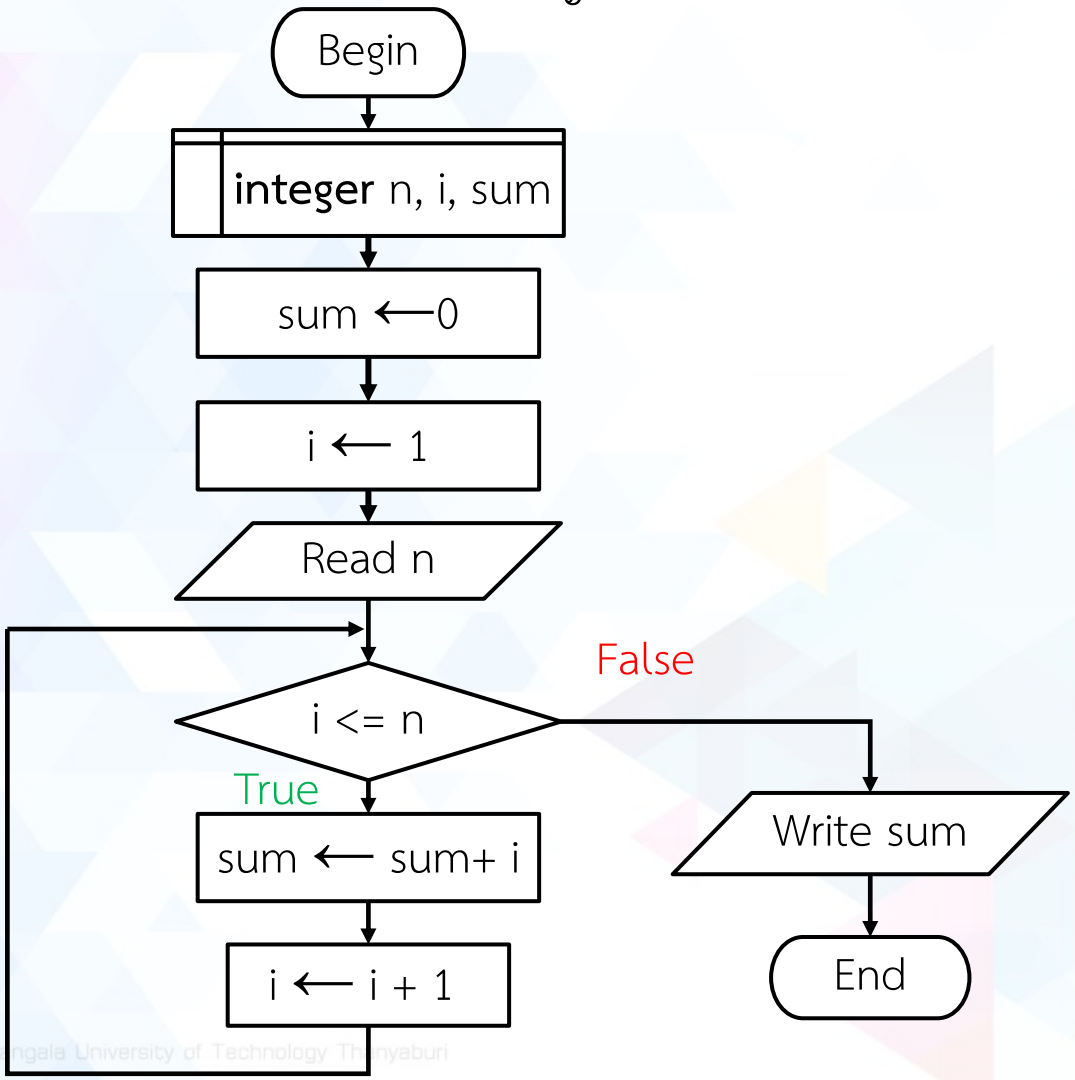
  ถ้าให้ n มีค่าเท่ากับ 6

  ผลรวม = 1 + 2 + 3 + 4 + 5 + 6 = 21

  คำตอบ ผลรวม มีค่าเท่ากับ 21

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 4 การพัฒนาลำดับขั้นตอนวิธีการแก้ปัญหาผังงาน (Flowchart)

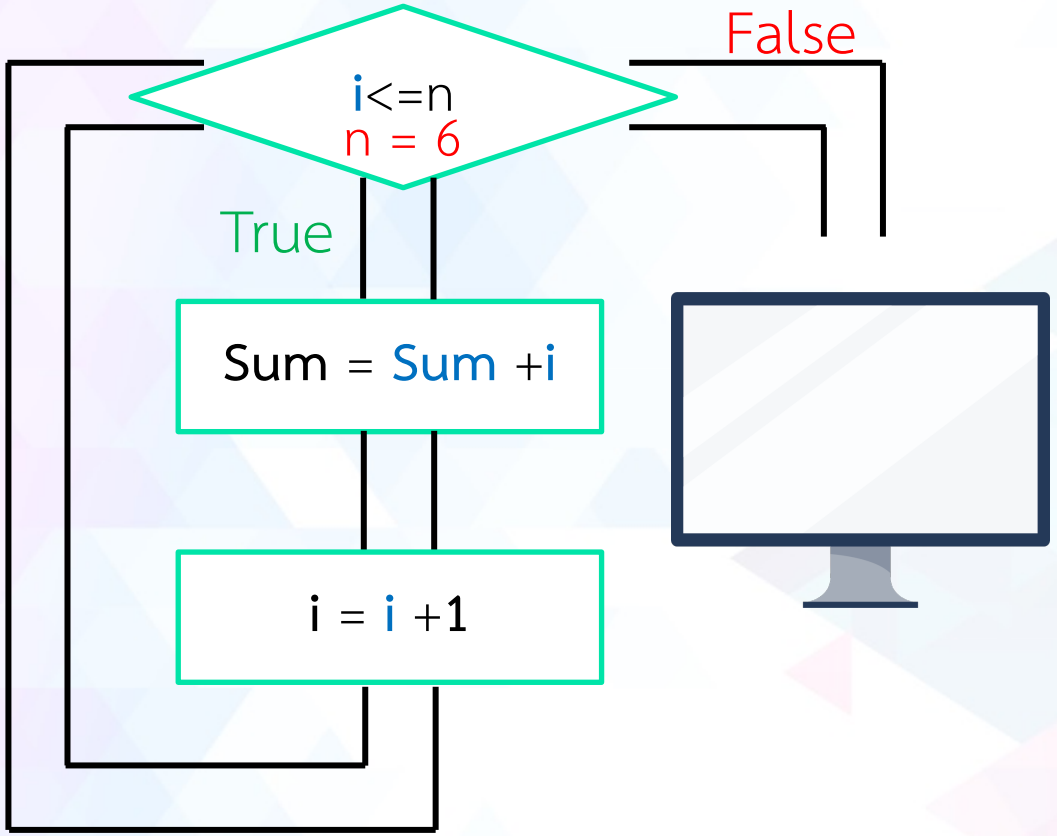


Computer Engineering • Faculty of Engineering • Rajamangala University of Technology Thanyaburi

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

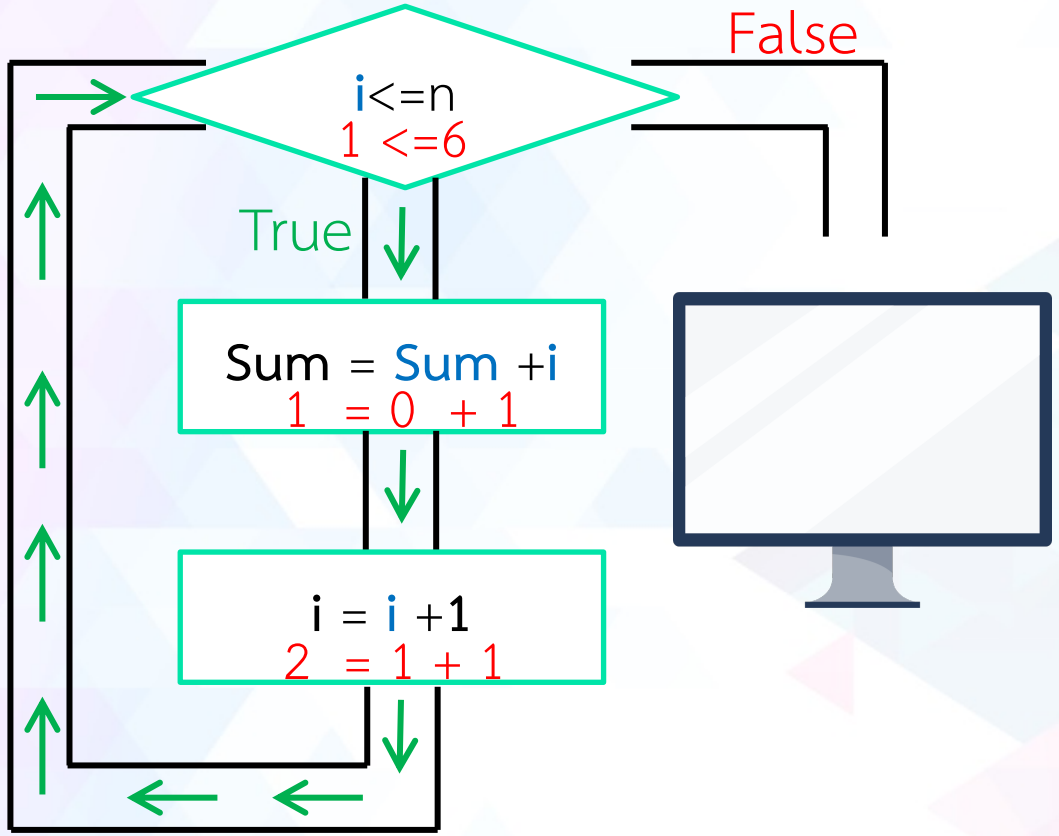
Test Case: n = 6



2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case: n = 6



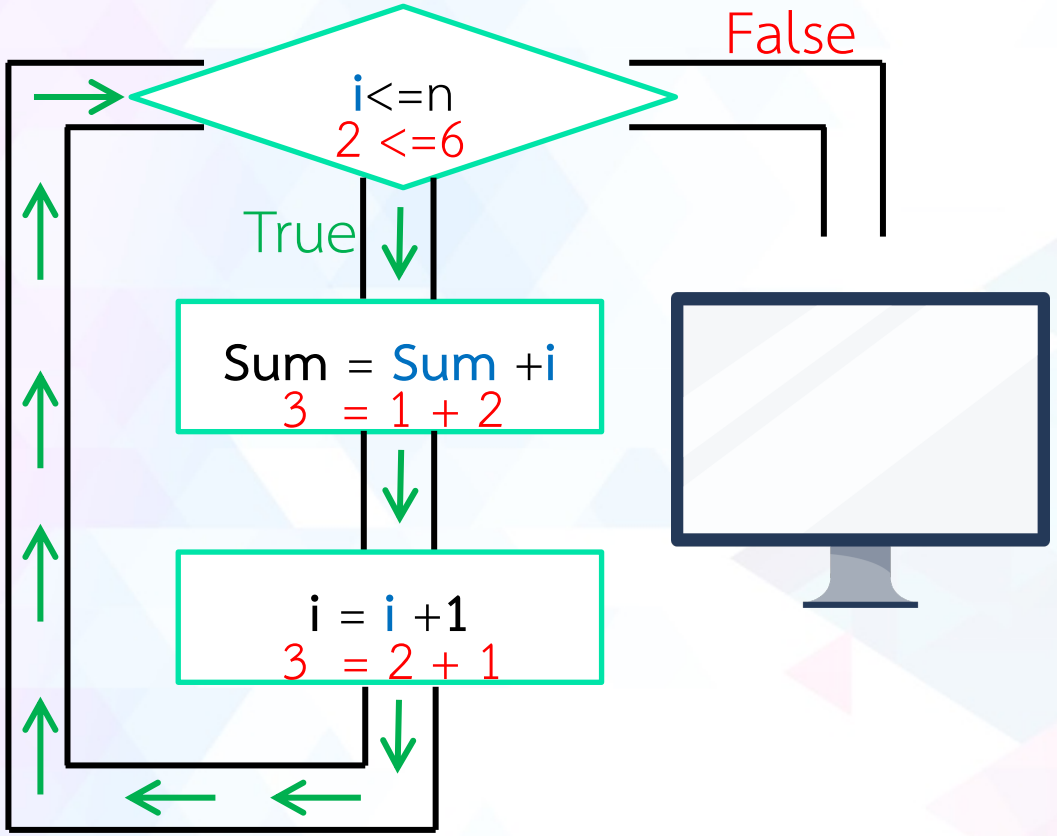
2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case: n = 6

i = 2

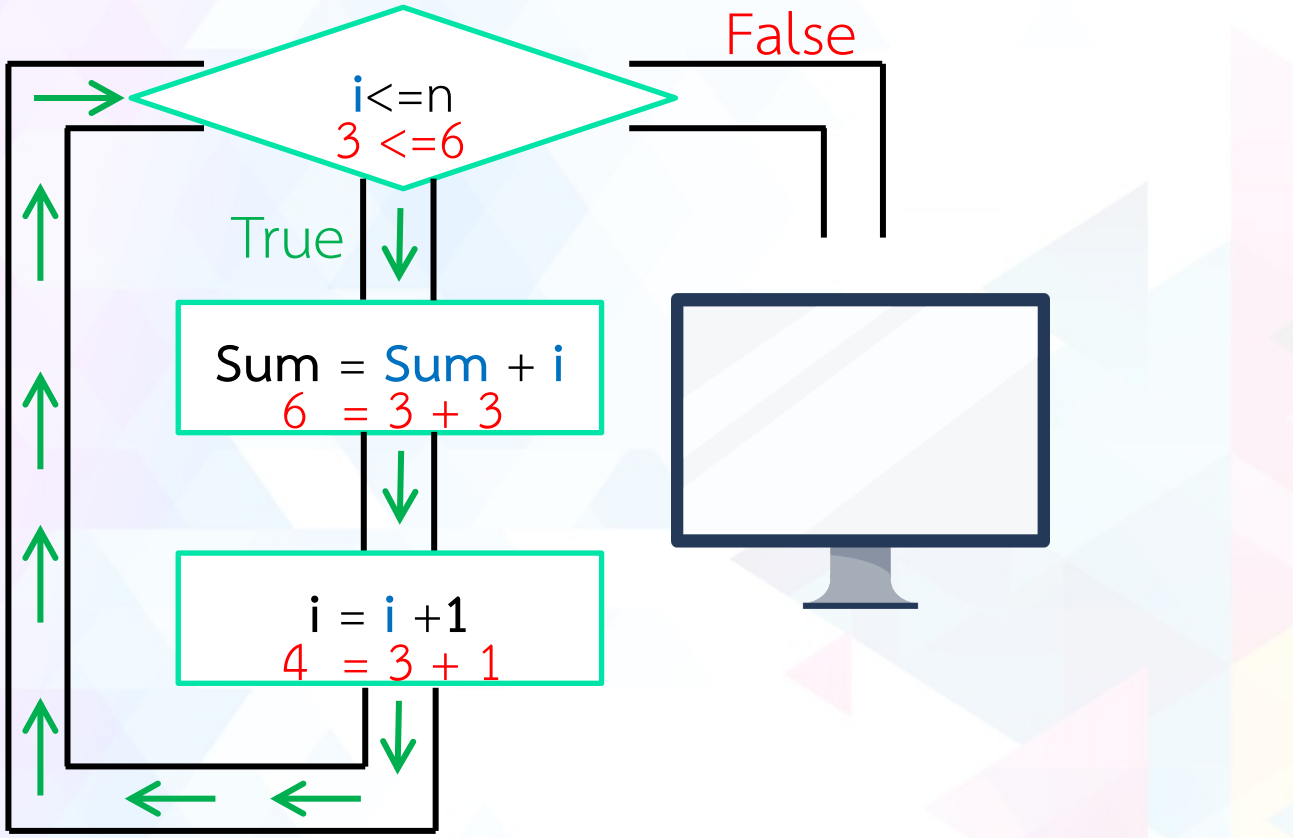
Sum = 1



2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case: n = 6



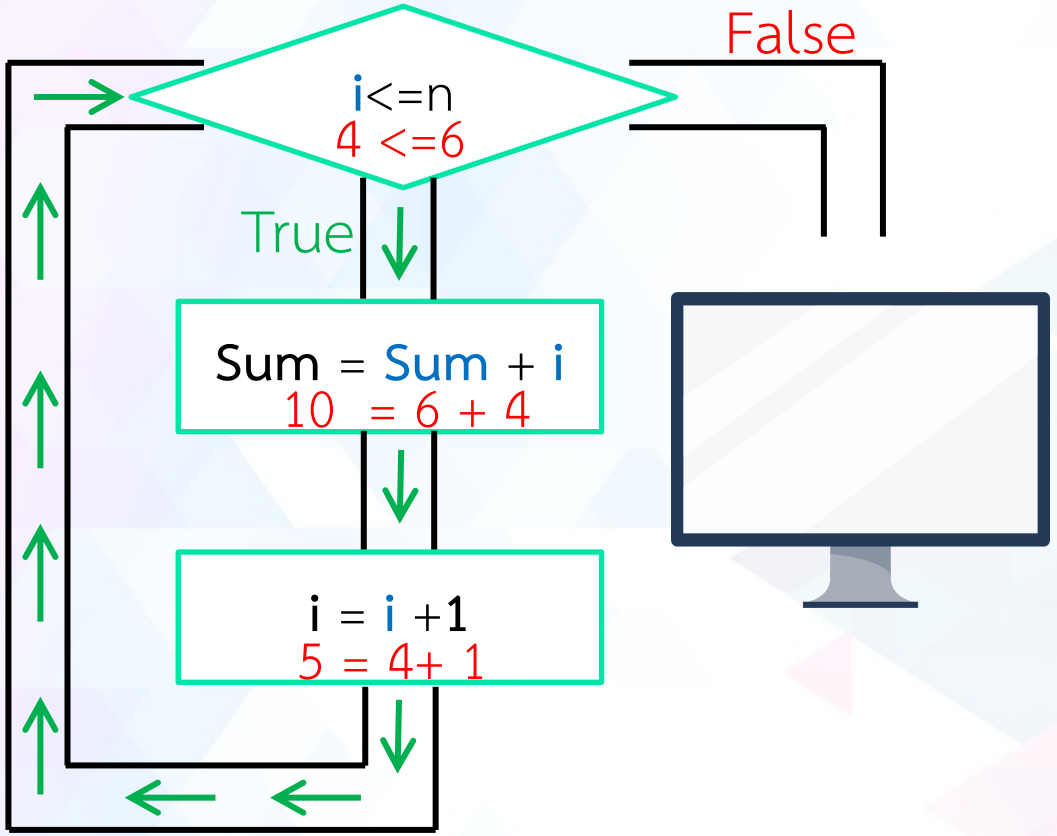
2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case: n = 6

i = 4

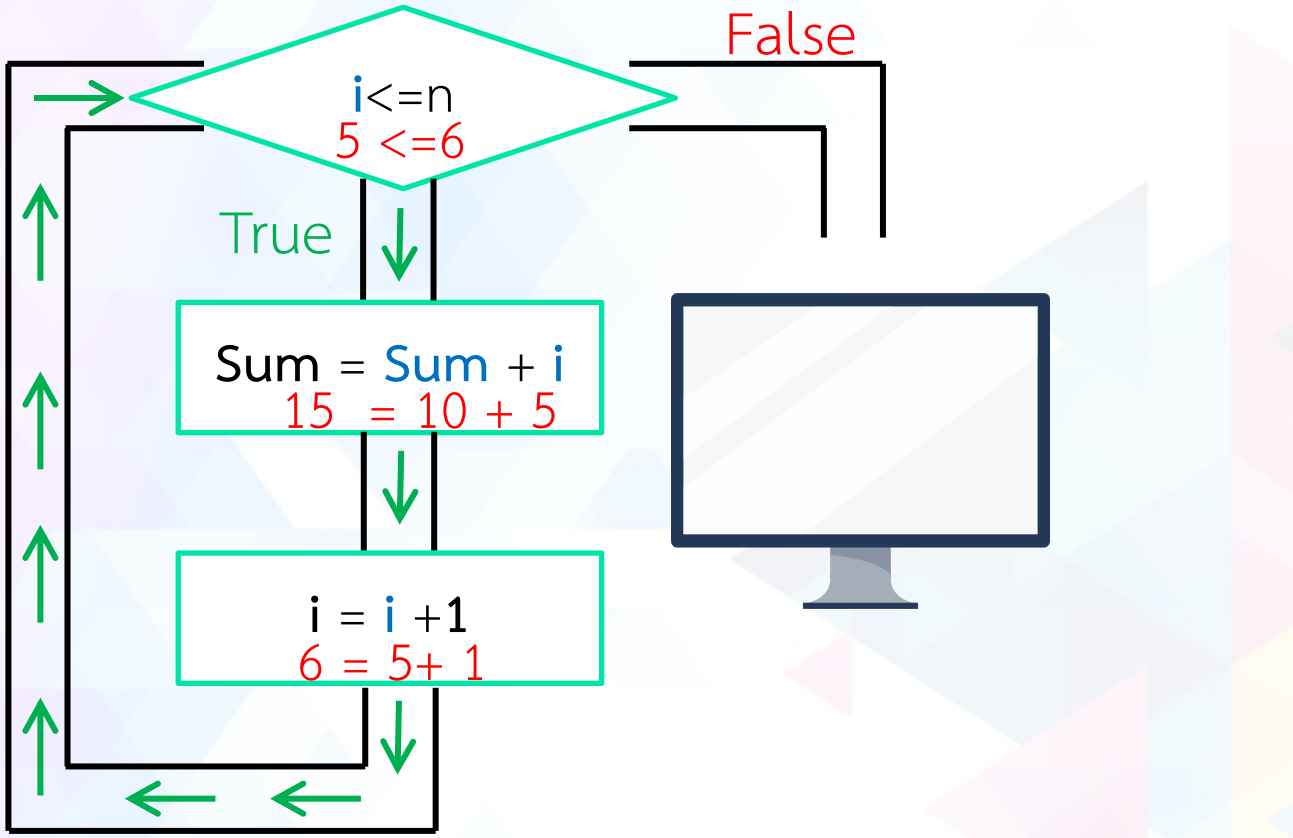
Sum = 6



2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case: n = 6



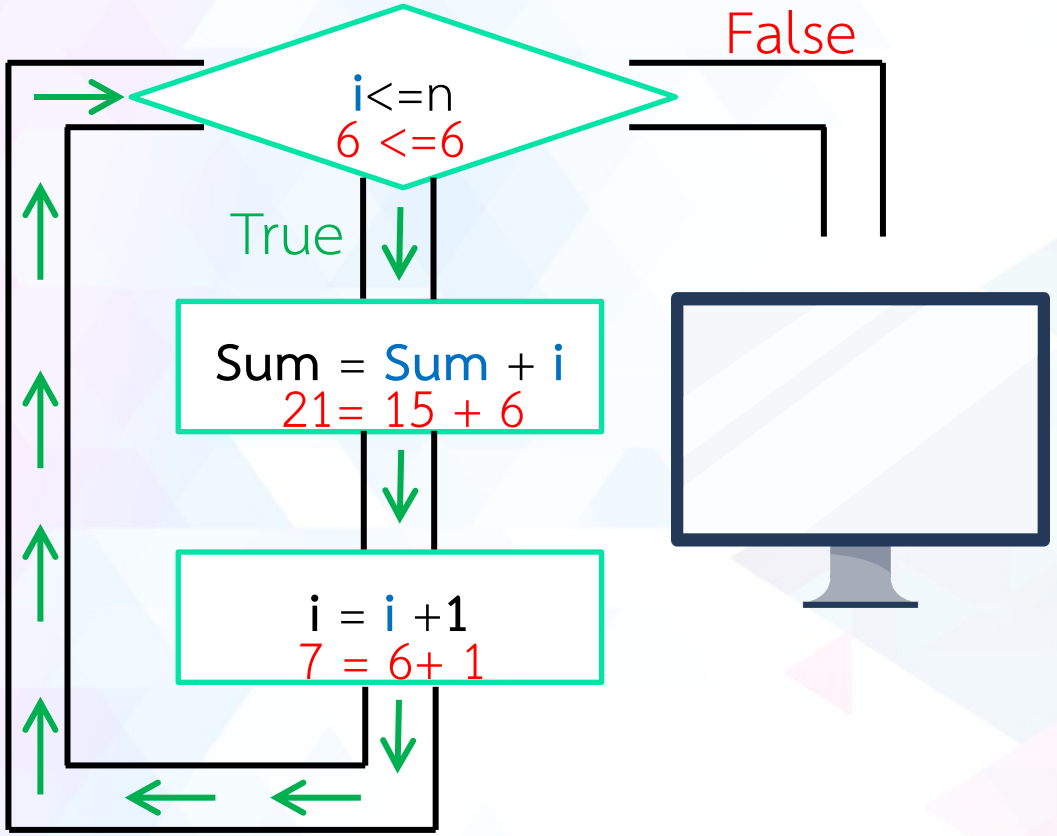
2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case: n = 6

i = 6

Sum = 15



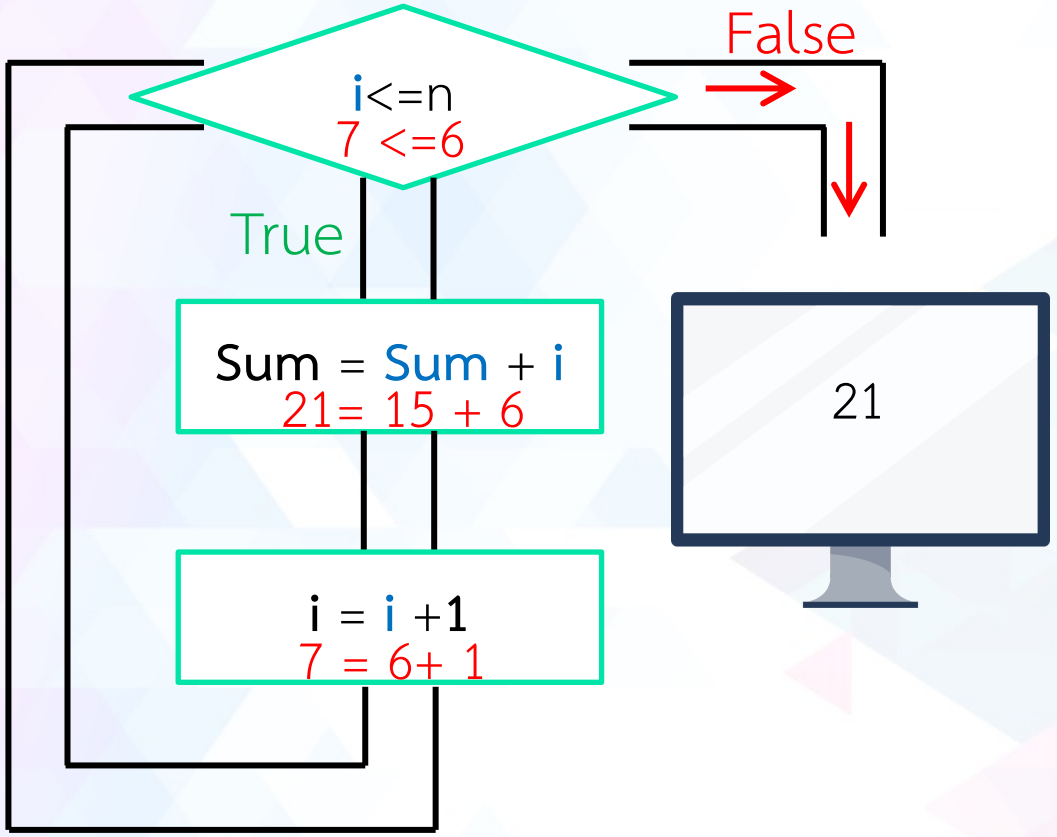
2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

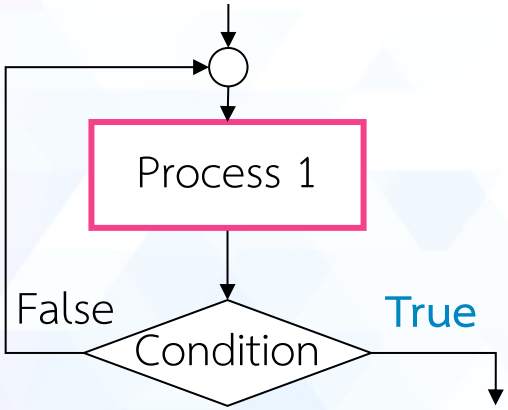
Test Case: n = 6

i = 7

Sum = 21



ผังงานแบบทำซ้ำจนกระทั่ง



- ขั้นตอนแรกของการทำงาน คือ Process 1 ซึ่งเรียกว่า **“ขั้นตอนการทำซ้ำ”**
- ถัดมาจะเป็นการตรวจสอบเงื่อนไข
  - หากเงื่อนไขยังไม่เป็นจริง จะกลับไปทำ**ขั้นตอนการทำซ้ำ**อีก
  - แต่หากเป็นจริงแล้ว จะ**ออกจากขั้นตอนการทำซ้ำ** เพื่อทำงานอื่นต่อไป

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

ตัวอย่างที่4 ตรวจสอบการรับเลขเป็นจำนวนเต็มและมีค่าตั้งแต่ 2 ถึง 12 หรือไม่

เป้าหมาย: ตรวจสอบการรับเลขเป็นจำนวนเต็มและมีค่าตั้งแต่ 2 ถึง 12

2.2. การวิเคราะห์ปัญหาสำหรับการออกแบบโปรแกรมผังงานแบบทำซ้ำ

- **ขั้นตอนที่ 1** ทำความเข้าใจกับปัญหา

  ตรวจสอบการรับเลข เป็นจำนวนเต็มและมีค่าตั้งแต่ 2 ถึง 12

- **ขั้นตอนที่ 2** ลักษณะของข้อมูลเข้าและข้อมูลออก

  ข้อมูลเข้า คือ จำนวนเต็มที่มีค่าตั้งแต่ 2 ถึง 12

  ข้อมูลออก คือ จำนวนเต็มที่มีค่าตั้งแต่ 2 ถึง 12

- **ขั้นตอนที่ 3** ทดลองแก้ไขปัญหาด้วยตนเอง

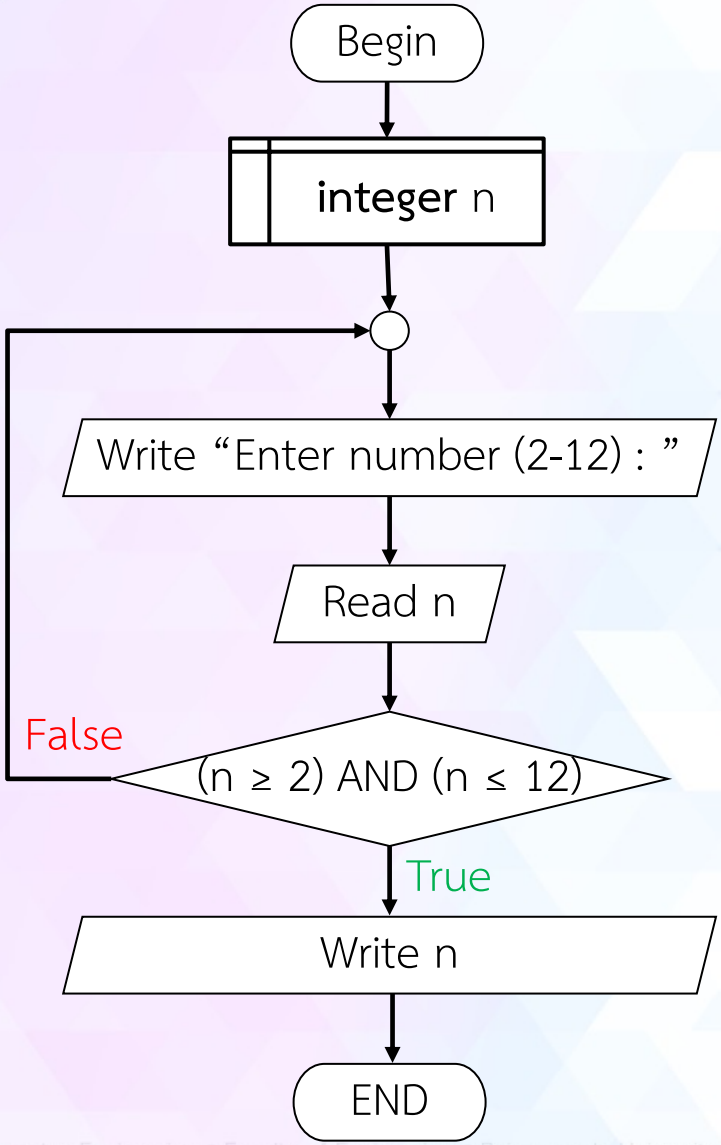
  ถ้าให้ n มีค่าเท่ากับ 1 โปรแกรมจะกลับมาเริ่มต้นรอรับค่าใหม่

  ถ้าให้ n มีค่าเท่ากับ 2 – 12 โปรแกรมจะต้องแสดงผลลัพธ์และไม่กลับมารอรับค่าใหม่

  ถ้าค่า n ไม่ได้อยู่ในช่วงที่ระบุ จะรับค่า n ไปเรื่อย ๆ

ขั้นตอนที่ 4 การพัฒนาลำดับขั้นตอนวิธีการแก้ปัญหา

ผังงาน (Flowchart)

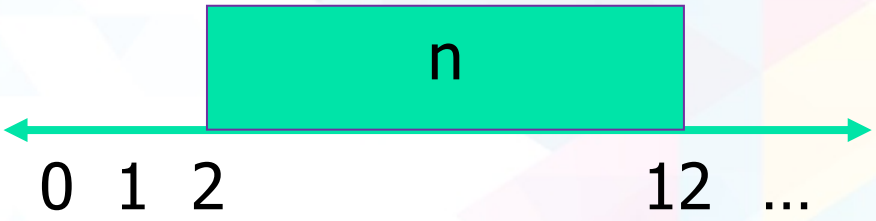


ตั้งชื่อตัวแปร n

แสดงข้อความ “Enter number (2-12) : ”

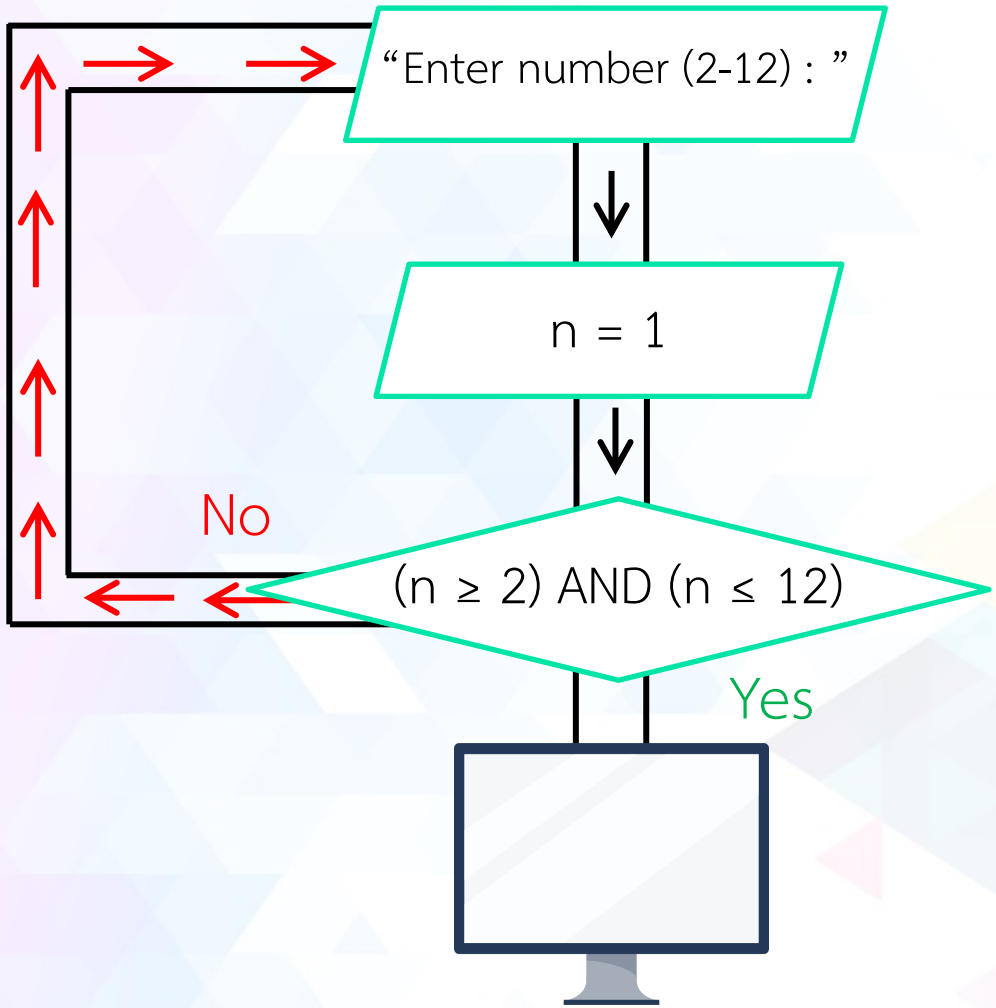
รับข้อมูลเข้า แล้วนำมาเก็บไว้ในตัวแปร n

ตรวจสอบว่า n มีค่าตั้งแต่ 2 ถึง 12 ให้แสดงตัวเลขนั้นๆได้



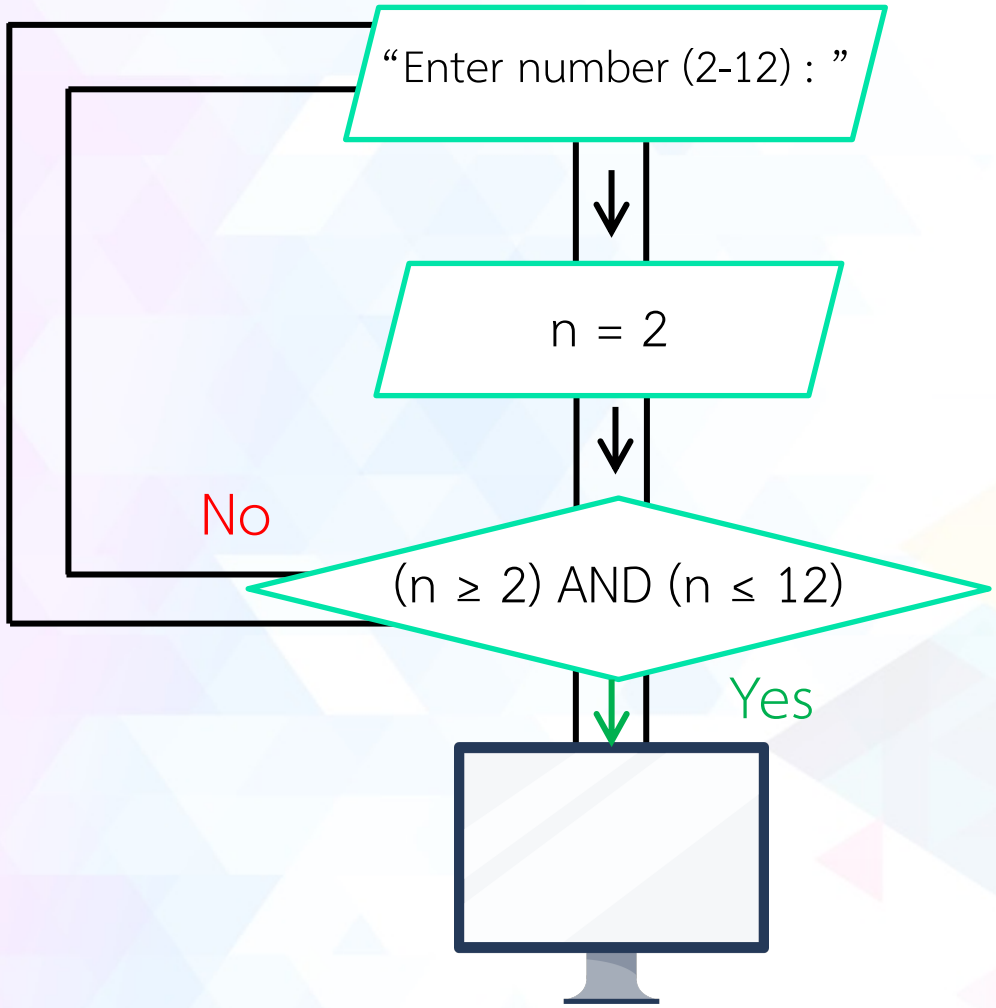
ขั้นตอนที่ 5 การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case: n = 1



- **ขั้นตอนที่ 5** การทดสอบขั้นตอนวิธีการแก้ปัญหา

Test Case: n = 2



ความแตกต่างระหว่าง การทำซ้ำในขณะที่ กับ การทำซ้ำจนกระทั่ง

	การทำซ้ำในขณะที่ (While-Do)	**การทำซ้ำจนกระทั่ง** (Repeat-Until)
การตรวจสอบเงื่อนไข	ตรวจสอบก่อนทำขั้นตอนการทำซ้ำ	ตรวจสอบหลังทำขั้นตอนการทำซ้ำ
การตัดสินใจเพื่อทำซ้ำ	ทำซ้ำเมื่อผลการตรวจสอบเงื่อนไขเป็นจริง	ทำซ้ำเมื่อผลการตรวจสอบเงื่อนไขเป็นเท็จ
จำนวนการทำซ้ำ	อาจไม่มีการทำขั้นตอนการทำซ้ำหากผลการตรวจสอบเงื่อนไขเป็นเท็จตั้งแต่แรก	ทำขั้นตอนการทำซ้ำอย่างน้อย 1 ครั้ง

จบบทที่ 1

สื่อสไลด์นี้ใช้รูปแบบฟอนต์ สารบรรณ รุ่นปรับปรุงใหม่ “Sarabun New”
สามารถดาวน์โหลดฟอนต์ได้ที่ URL : https://www.f0nt.com/release/th-sarabun-new/