

Introduction to computer programming



จุดประสงค์

- แนะนำให้รู้จัก **Java**
- แนะนำให้รู้จักขั้นตอน **compile** และ **run Program**
- แนะนำให้รู้จัก **Program**
- เขียนโปรแกรม **Java** อย่างง่าย ๆ ได้
- สร้าง **project** และ **run Program Java** ได้

ทำไมต้องเรียน **Java**

- ง่าย: แม้ว่าบางส่วนของ **Java** จะมีต้นแบบมาจาก **C++** แต่ก็ผู้สร้าง **Java** ก็ออกแบบและปรับปรุงให้มันง่ายและดีขึ้น
- **Object-Oriented: Java** ถูกออกแบบมาสำหรับการเขียนโปรแกรมแบบ **Object** ไม่ใช่แบบกระบวนการ (**Procedural**) ทำให้มีความยืดหยุ่น (**Flexibility**), มีความเป็นหน่วย (**Modularity**), มีความชัดเจนของตัวโปรแกรม (**Clarity**), และ มีการนำกลับมาใช้ได้ใหม่ (**Reusability**) มากขึ้น
- โปรแกรมแบบกระจาย (**Distributed**): **Java** ได้ถูกออกแบบมาให้การติดต่อระหว่างโปรแกรมผ่านเครือข่ายทำได้ง่ายขึ้น

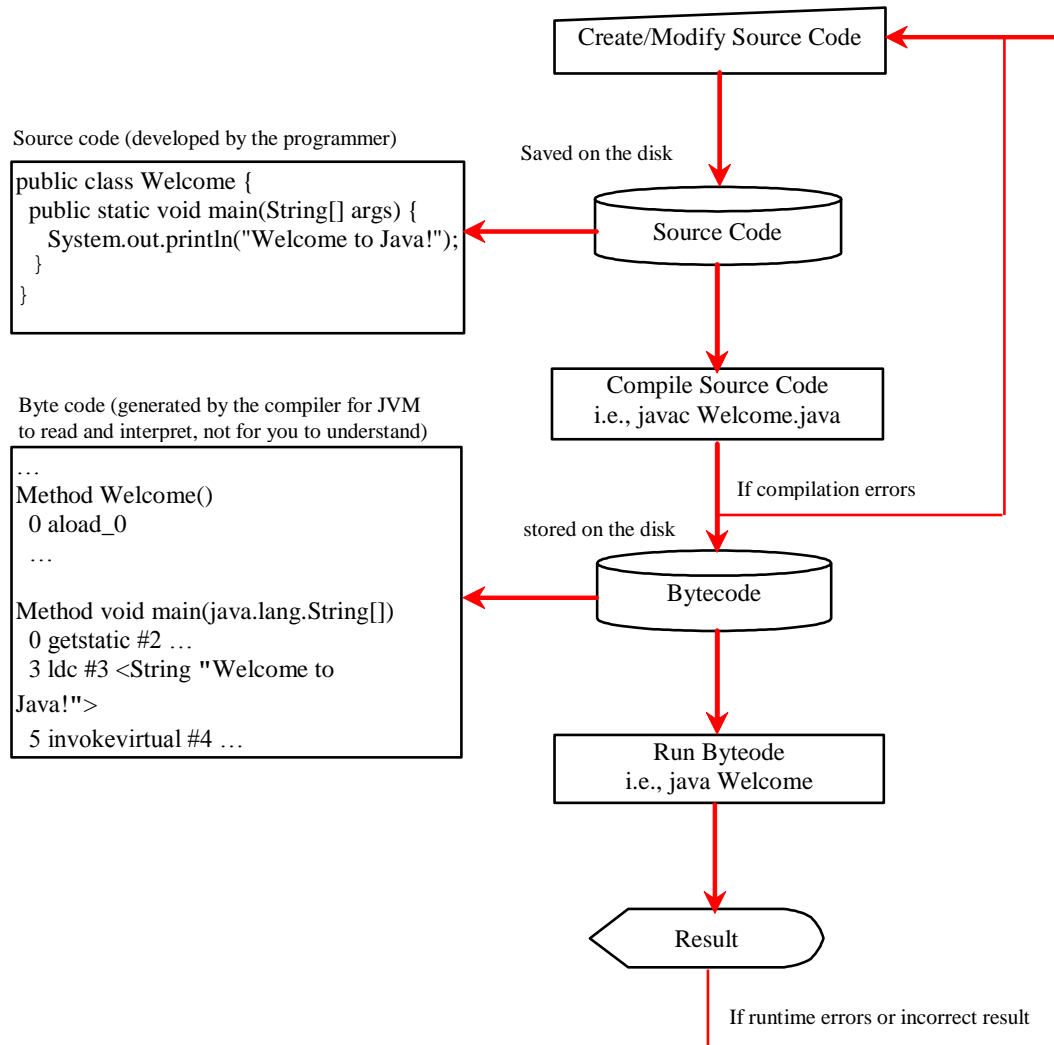
- ผ่านการแปลมาแล้ว (**Interpreted**): แม้การพัฒนาโปรแกรมภาษา **Java** นั้นเหมือนกับการพัฒนาภาษาอื่นๆ คือ ผู้พัฒนาต้องเขียนโปรแกรม (**source code**) ก่อน แล้วแปลโปรแกรมเป็นโปรแกรมที่ **CPU** จะสามารถ **execute** ได้ แต่ ภาษา **Java** ต่างกับภาษาอื่นๆ ตรงที่ **source code** ของ **Java** จะถูกแปลเป็นรูปแบบ (**format**) ที่สามารถปฏิบัติงานได้ทุก **platform** (**Java Virtual Machine** มีหน้าที่เป็น **Interpreter** รับผิดชอบในการแปลนี้)
- ทนทานต่อความผิดพลาด (**Robust**): **Java** สามารถพบปัญหาหรือ **bug** ได้หลายชนิดก่อนที่จะ **run** โปรแกรม นอกจากนี้ **Java** ยังมีกลไกจัดการกับ **exception** เวลามีปัญหาเกิดขึ้นในระบบ

- **ปลอดภัย (secure): Java** มีกลไกเกี่ยวกับความปลอดภัยเพื่อป้องกันระบบจากการทำงานของโปรแกรมแปลกหน้า
- เป็นกลางไม่ขึ้นกับสถาปัตยกรรมตัวเครื่อง (**Architecture-neutral and portable**): โปรแกรมที่แปลด้วย **Java Virtual Machine** แล้ว (กลายเป็น **Java bytecode**) จะสามารถปฏิบัติงานได้ในทุกสถาปัตยกรรมคอมพิวเตอร์
- เคลื่อนย้ายได้ (**Portable**): **bytecode** (โปรแกรมที่ถูกแปลแล้ว) สามารถถูกนำไป **run** ที่เครื่องแบบใดก็ได้ โดยไม่ต้องถูกแปลใหม่

โปรแกรม

- **โปรแกรม** คือ รูปแบบทางตรรกะของคำสั่งต่างๆ ที่นำมารวมกัน เพื่อแก้ปัญหาเฉพาะบางอย่าง และนำมาใช้สั่งให้คอมพิวเตอร์ทำงาน ผู้เขียน โปรแกรมจะเขียนโปรแกรมตามความต้องการของ นักวิเคราะห์ระบบ หรืออาจกล่าวได้สั้นๆ ว่า โปรแกรมหมายถึง ชุดคำสั่งซึ่งถูกเขียนขึ้นเพื่อสั่งให้คอมพิวเตอร์ทำงานตามต้องการ

ภาษาคอมพิวเตอร์และการพัฒนาโปรแกรม



ภาษาคอมพิวเตอร์และการพัฒนาโปรแกรม

- กำหนดและวิเคราะห์ปัญหา (Problem Definition and Problem Analysis)
- เขียนผังงานและซูโดโค้ด (Pseudocoding)
- เขียนโปรแกรม (Programming)
- ทดสอบและแก้ไขโปรแกรม (Program Testing and Debugging)
- ทำเอกสารและบำรุงรักษาโปรแกรม (Program Documentation and Maintenance)

การกำหนดและวิเคราะห์ปัญหา

- กำหนดขอบเขตของปัญหา
- กำหนดลักษณะของข้อมูลเข้าและออกจากระบบ (Input/Output Specification)
- กำหนดวิธีการประมวลผล (Process Specification)

ตัวอย่าง

ตัวอย่าง

ถ้าหากต้องการออกแบบโปรแกรมให้คอมพิวเตอร์รับค่าข้อมูล 3 ค่า และแสดงค่าเฉลี่ยทางจอภาพ เราอาจกำหนดและวิเคราะห์ปัญหาได้ ดังนี้

1. รับข้อมูลจากคีย์บอร์ด

- รับข้อมูลเฉพาะที่เป็นตัวเลขมาเก็บในตัวแปร
- ถ้าข้อมูลเท่ากับ 0 ให้รับใหม่

ตัวอย่าง (ต่อ)

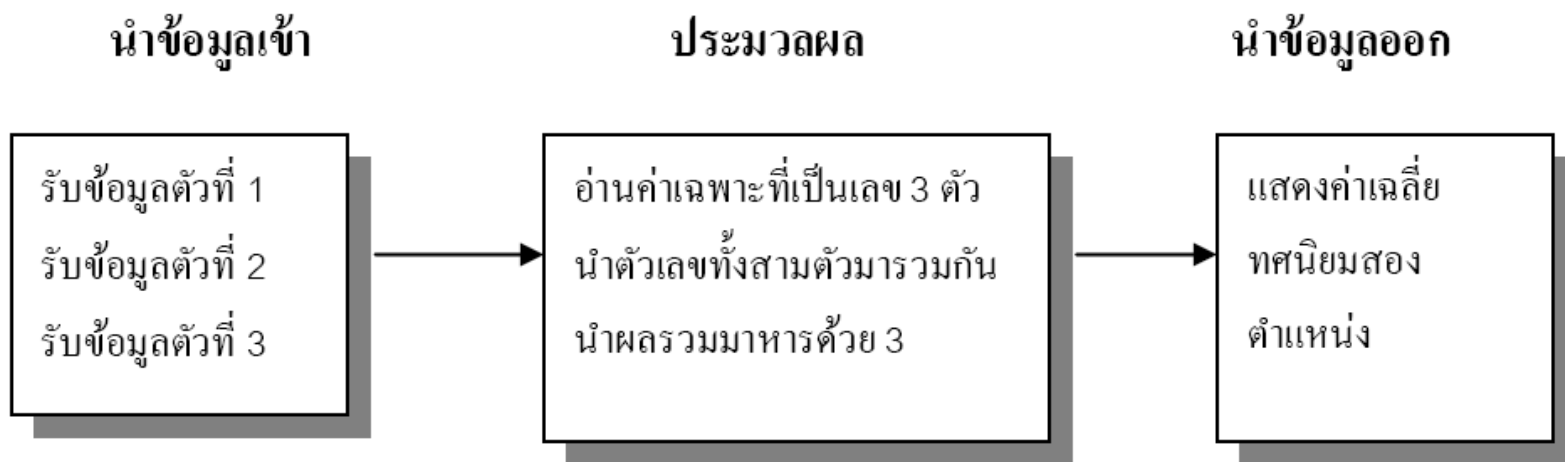
2. หาค่าเฉลี่ย

- รวมค่าทุกค่าที่รับมาเข้าด้วยกัน
- นำค่าผลรวมที่ได้หารด้วย 3
- นำค่าผลลัพธ์ไปเก็บในตัวแปร

3. แสดงผลลัพธ์ทางจอภาพ

- แสดงค่าว่าค่าเฉลี่ยเท่ากับ
- แสดงผลลัพธ์โดยมีทศนิยมสองตำแหน่ง

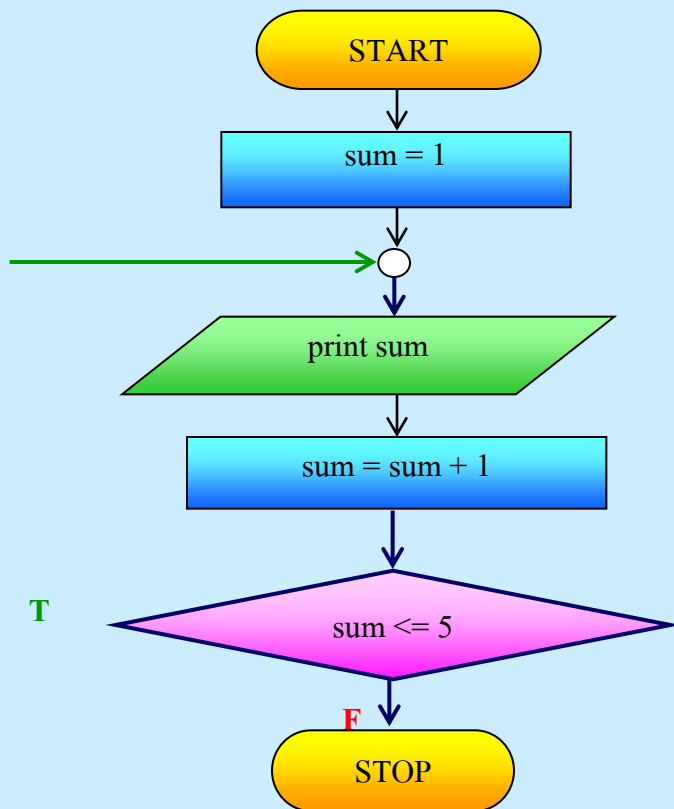
ขั้นตอนการทำงาน



การเขียนผังงานและซูดโค้ด

หลังจากที่ได้วิเคราะห์ปัญหาแล้ว ขั้นตอนต่อไปจะใช้เครื่องมือช่วยในการออกแบบโปรแกรม โดยเขียนเป็นลำดับขั้นตอนการทำงานของโปรแกรมที่เรียกว่าอัลกอริทึม (Algorithm) โดยจะเขียนแสดงขั้นตอนการแก้ปัญหา ด้วยประโยคที่ชัดเจน ไม่คลุมเครือ และมีรายละเอียดการทำงานพอสมควรเพียงพอที่จะนำไปเขียนเป็นโปรแกรมให้ทำงานจริง โดยอัลกอริทึมนั้นอาจเขียนให้อยู่ในรูปของรหัสจำลองหรือซูดโค้ด (Pseudo-code) หรือเขียนเป็นผังงาน (Flowchart) ก็ได้

ผังงาน หมายถึงการแทนขั้นตอนวิธีการทำงานโดยใช้สัญลักษณ์แทนความหมายของการทำงานบางอย่าง และมีเส้นลูกศรแสดงลำดับหรือทิศทางการทำงาน



ชุดโคด หมายถึงการแทนขั้นตอนวิธีการทำงานโดยใช้คำในภาษาอังกฤษ มาเรียงต่อกันเพื่ออธิบายวิธีการทำงานแล้วได้ความหมายเป็นขั้นตอนการทำงานที่ต้องการ

Procedure PR Tone2five

sum =1

do

print sum

sum = sum + 1

while sum <= 5

end

กระบวนการแก้ปัญหาด้วยคอมพิวเตอร์

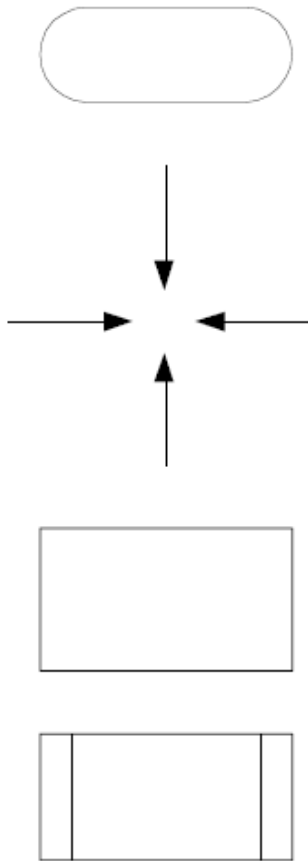
วิเคราะห์ปัญหา

- ต้องการอะไร
- ต้องการเอาต์พุตอย่างไร
- ข้อมูลเข้าเป็นอย่างไร
- ตัวแปรที่ใช้
- วิธีการประมวลผลเป็นอย่างไร

การเขียนผังงาน

ผังงานเป็นแผนภาพที่แสดงลำดับขั้นตอนในการทำงานของโปรแกรม โดยมีการลงรายละเอียด ใ้วิธีการ และจัดลำดับขั้นตอนของโปรแกรม ตั้งแต่เริ่มต้นจากการรับข้อมูล การประมวลผล ไปจนถึงการแสดงผลลัพธ์การทำงาน

สัญลักษณ์ที่ใช้ในการเขียนผังงาน



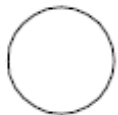
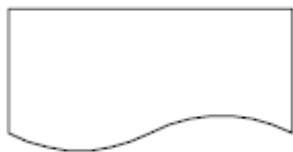
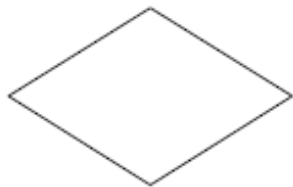
การเริ่มต้นและการสิ้นสุดการทำงานของโปรแกรม

ลูกศรแสดงทิศทางการทำงานของโปรแกรม และการไหลของข้อมูล

การประมวลผลการคำนวณต่าง ๆ

การทำงานย่อย

สัญลักษณ์ที่ใช้ในการเขียนผังงาน



การรับหรือแสดงข้อมูล โดยไม่ระบุชนิดของ
อุปกรณ์

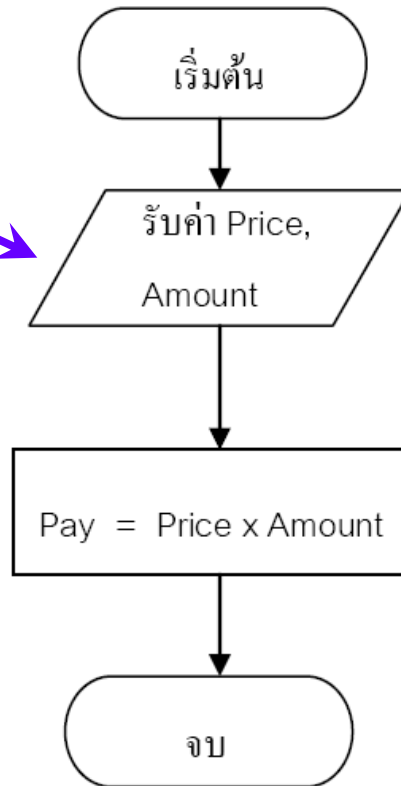
การตรวจสอบเงื่อนไข เพื่อเลือกทำอย่างใด
อย่างหนึ่ง โดยจะมีเส้นลูกศรแสดงทิศ
ทางการทำงานต่อไป

แสดงผลลัพธ์หรือรายงานทางเครื่องพิมพ์

จุดเชื่อมต่อของผังงาน

ตัวอย่างการเขียนผังงาน

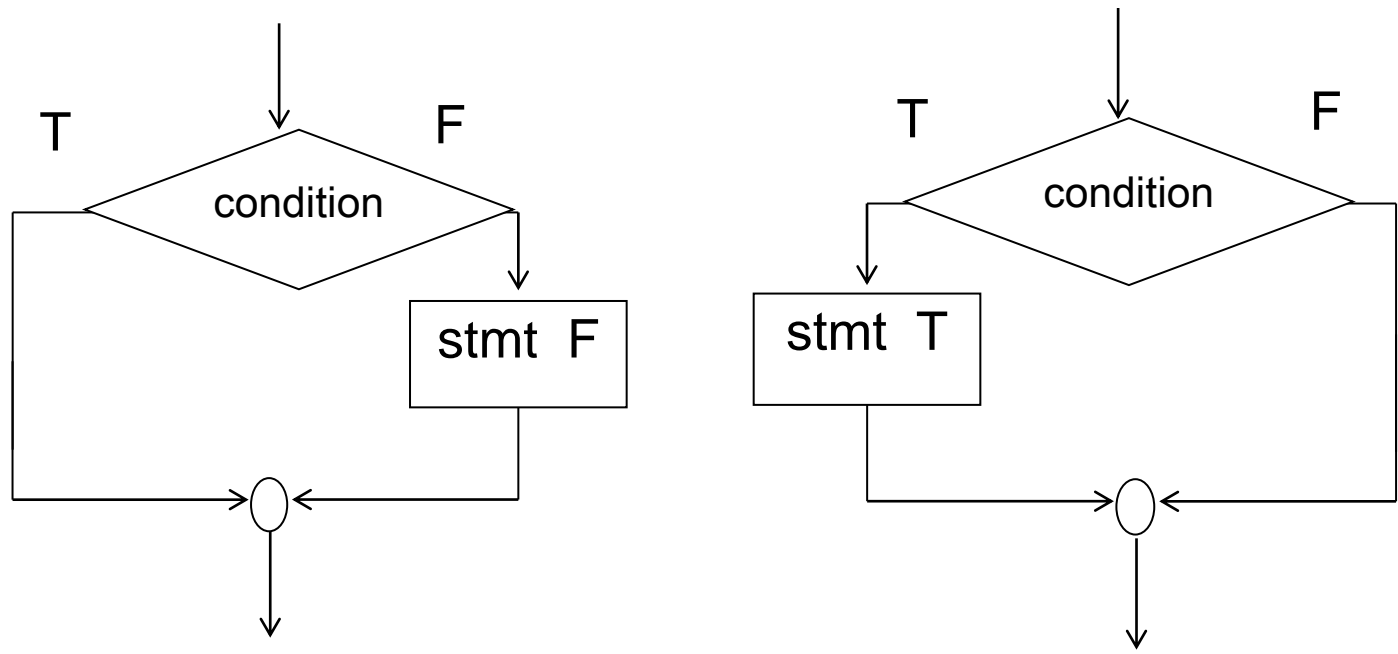
สัญลักษณ์ของ
การรับค่า



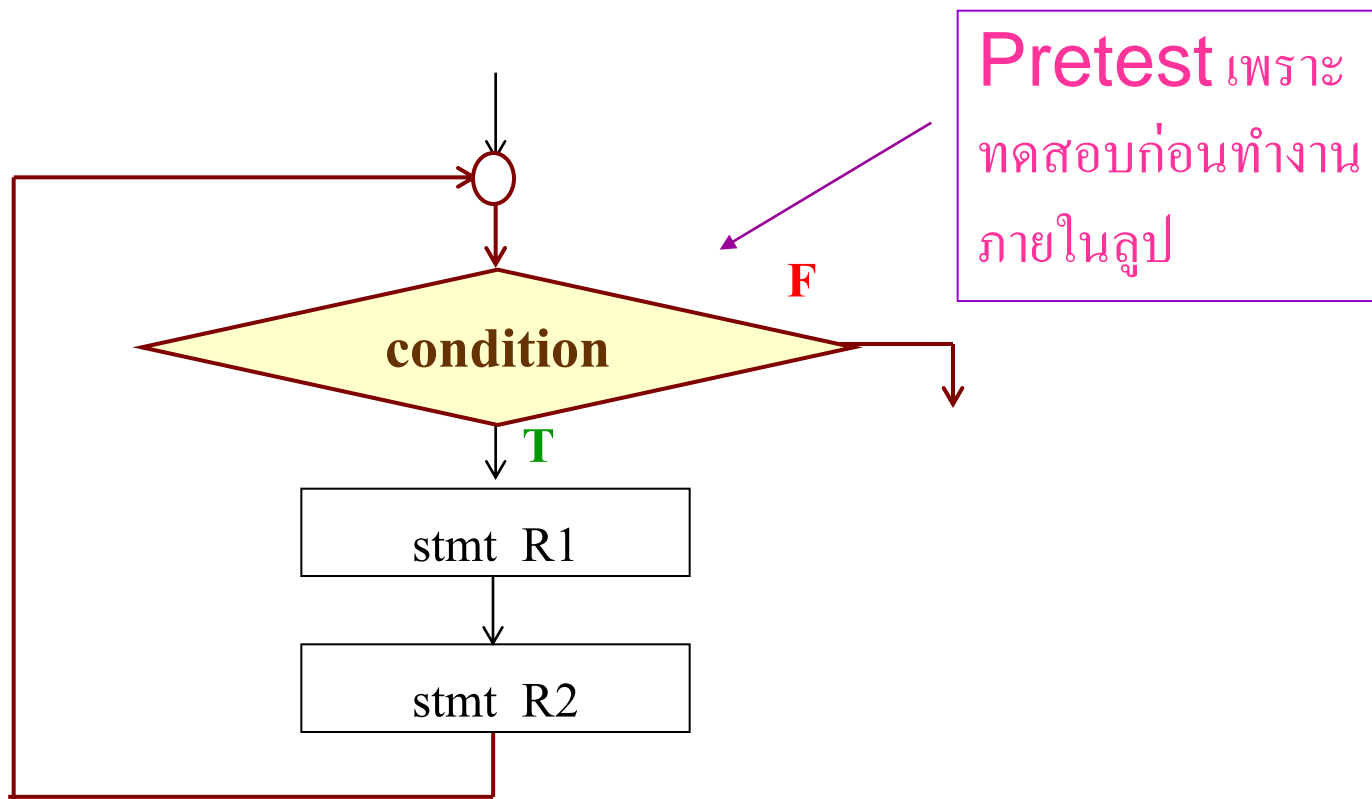
สัญลักษณ์ของ
การคำนวณ

ผังงานแบบทางเลือก

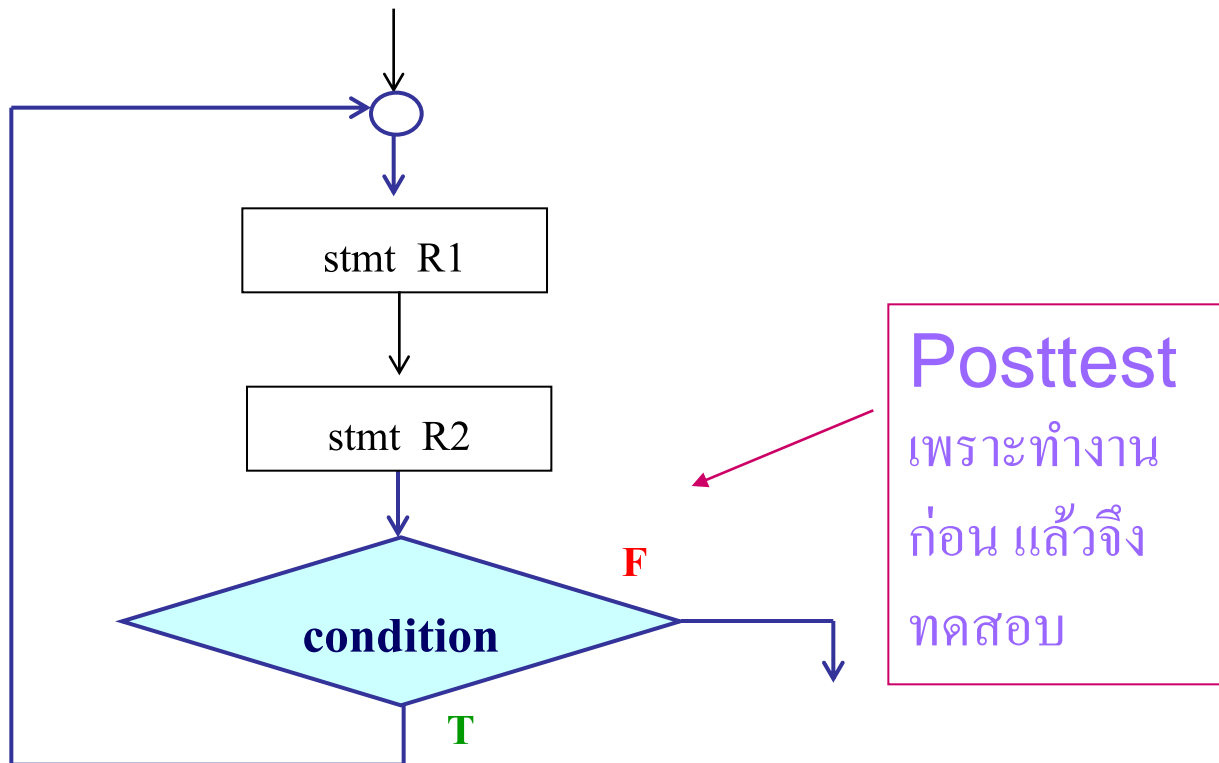
เป็นการทำงานที่ต้องมีการตัดสินใจ เพื่อเลือกการทำงาน โดยใช้
สัญลักษณ์การเปรียบเทียบ



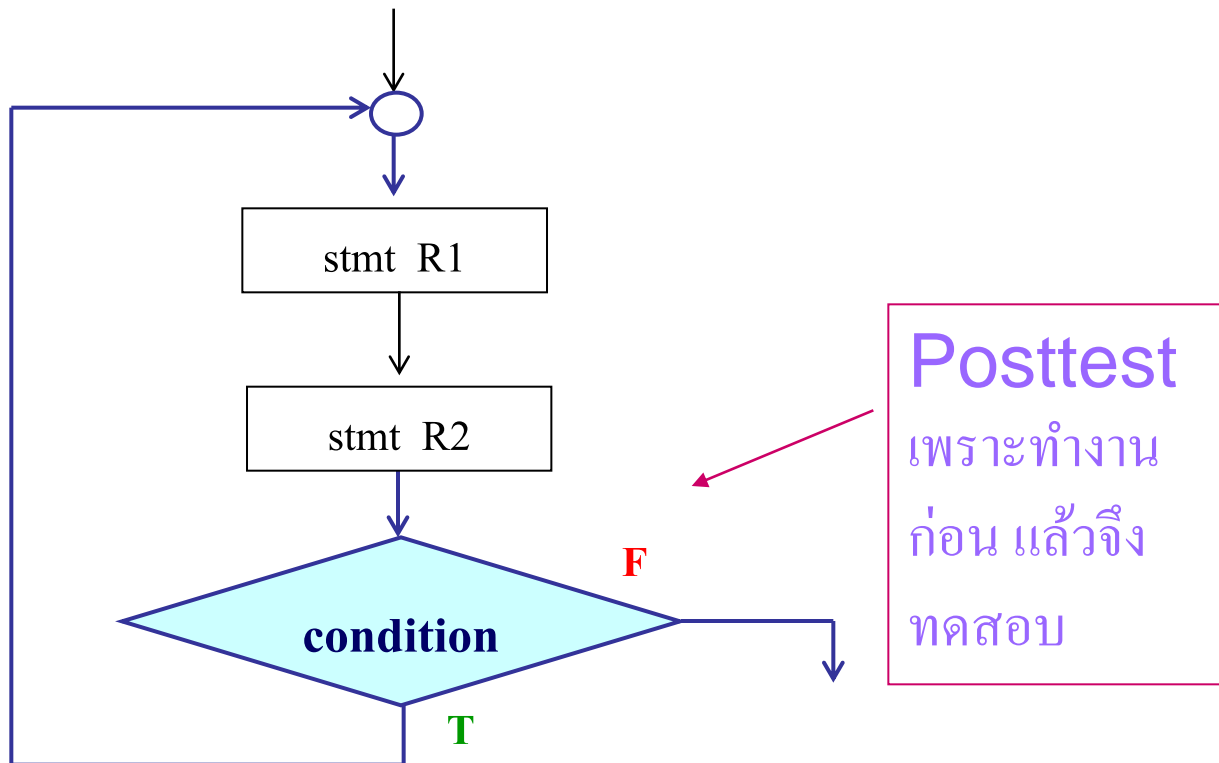
ผังงานแบบวนซ้ำ (while-do)



ผังงานแบบวนซ้ำ (do-while)



ผังงานแบบวนซ้ำ (do-while)

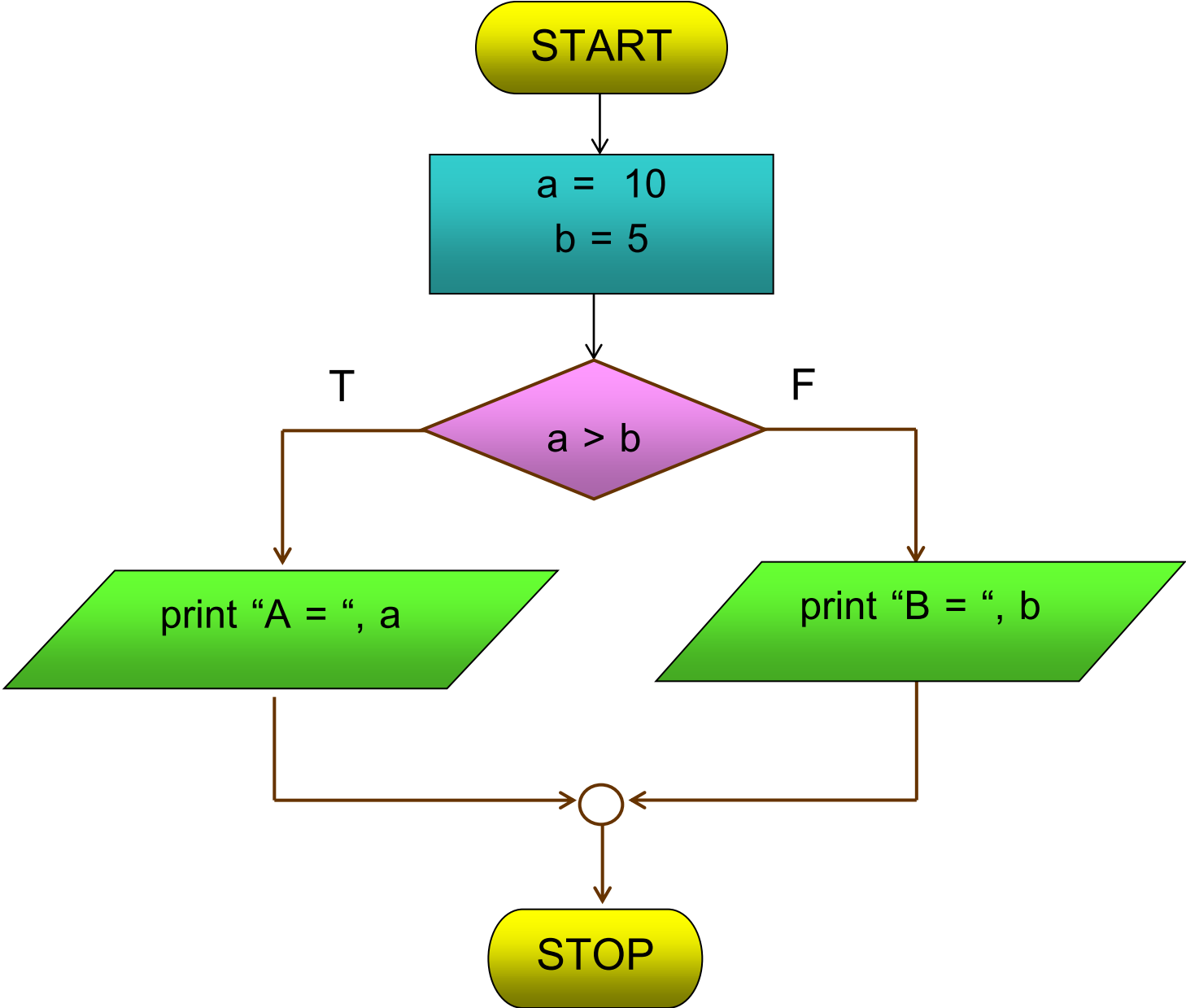


ตัวอย่าง

ตัวอย่าง จงเขียนผังงานเพื่อกำหนดค่าให้กับตัวแปร **a** และ **b** แล้วเปรียบเทียบเพื่อแสดงผล

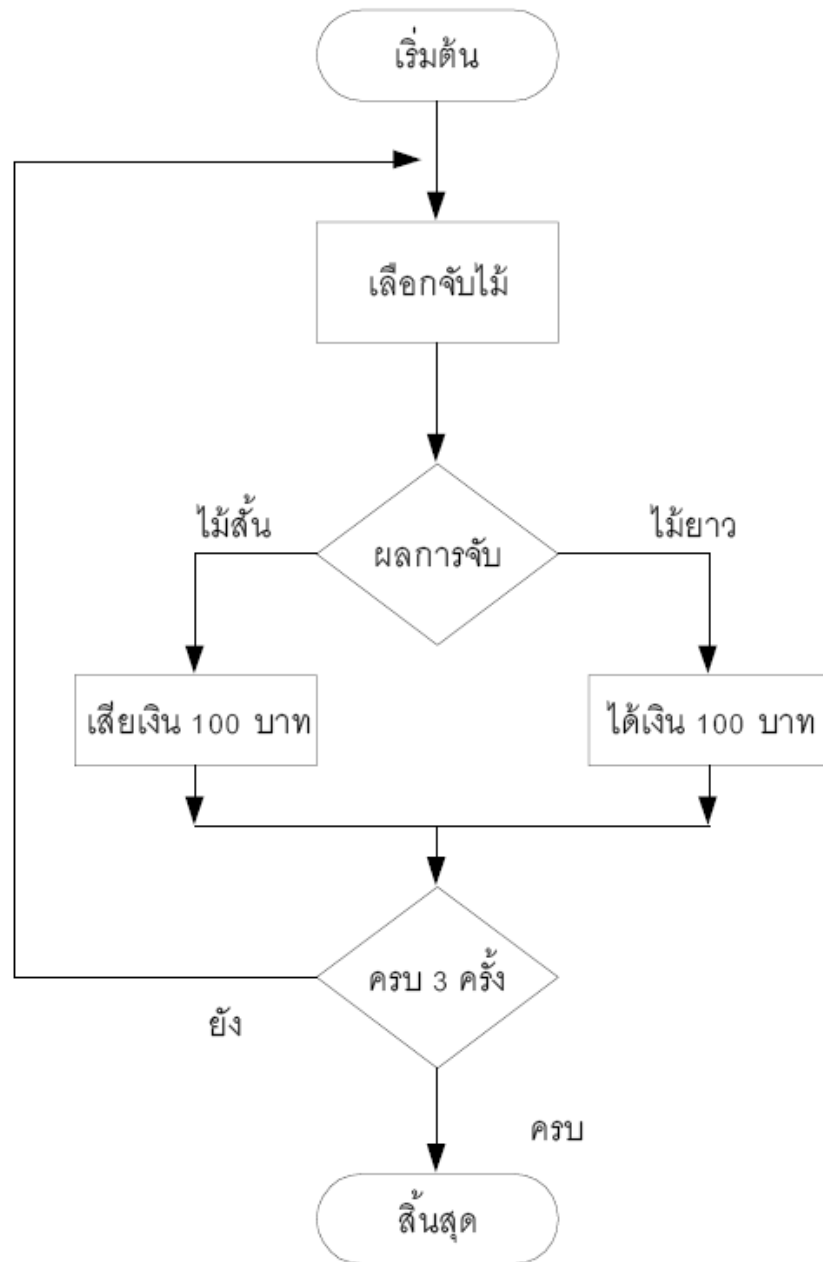
ถ้าตัวแปร **a** มีค่ามากกว่าตัวแปร **b** ให้พิมพ์ค่าตัวแปร **a** ออกมาแสดงบนจอภาพ

ถ้าไม่ใช่ ให้พิมพ์ค่าตัวแปร **b** ออกมาแสดงบนจอภาพ



ตัวอย่าง

จงพิจารณาในการจับไม้สั้น ไม้ยาวสามครั้ง ถ้าได้ไม้ยาวให้ได้เงิน 100 บาท ถ้าได้ไม้สั้นให้เสียเงิน 100 บาท



ชูโดโค้ด

ชูโดโค้ดเป็นคำอธิบายขั้นตอนการทำงานของโปรแกรมโดยใช้
ถ้อยคำผสมระหว่างภาษาอังกฤษและภาษาการเขียนโปรแกรมแบบ
โครงสร้างที่เข้าใจง่ายมาแสดงลำดับการทำงานของโปรแกรม หรือ
อาจใช้ภาษาไทยก็ได้ โดยให้ผู้เขียนโปรแกรมสามารถพัฒนาขั้นตอน
ต่าง ๆ ให้เป็นโปรแกรมได้ง่ายขึ้น

ตัวอย่างเช่นในการเขียนชุดโค้ดสำหรับให้คอมพิวเตอร์หาค่าเฉลี่ย จากข้อมูลที่รับเข้าทางแป้นพิมพ์อาจเขียนได้ดังนี้

Algorithm การหาค่าเฉลี่ย

1. ตัวนับ = 0
2. ผลรวม = 0
3. รับค่าทางแป้นพิมพ์เก็บไว้ใน (ข้อมูล)
4. ถ้า ข้อมูล มากกว่า 0
 เพิ่มค่าตัวนับขึ้นหนึ่งค่า
 ผลรวม = ผลรวม + ค่าข้อมูล
 ย้อนกลับไปทำขั้นตอนที่ 3
 ถ้าไม่มากกว่าไปทำขั้นตอนที่ 5
5. ค่าเฉลี่ย = ผลรวมหารด้วยตัวนับ
6. แสดงค่าเฉลี่ยทางจอภาพ โดยมีทศนิยมสองตำแหน่ง
7. จบ

การวิเคราะห์ปัญหา

ตัวอย่างที่ 1

จงเขียนแนวทางการแก้ปัญหาด้วยคอมพิวเตอร์ สำหรับให้
คอมพิวเตอร์คำนวณหาค่าจ้างพนักงานเป็นรายชั่วโมง จากนั้นแสดง
ค่าจ้างที่คำนวณได้

ตัวอย่างการวิเคราะห์

วิธีทำ

ต้องการอะไร

ต้องการทราบค่าจ้างของพนักงานแต่ละคน

ต้องการเอาต์พุตอย่างไร

ต้องการเอาต์พุตเป็นค่าจ้างสุทธิของพนักงานทางจอภาพ

ตัวอย่างการวิเคราะห์

ข้อมูลเข้า

- รหัสพนักงาน
- ชื่อพนักงาน
- จำนวนชั่วโมงทำงานเก็บในตัวแปรชื่อ Hours
- ค่าจ้างรายชั่วโมงเก็บในตัวแปรชื่อ PayRate

ตัวอย่างการวิเคราะห์

กำหนดวิธีการคำนวณ

ค่าจ้างสุทธิ = จำนวนชั่วโมง x อัตราต่อชั่วโมง

ขั้นตอนการประมวลผล

1. เริ่มต้น
2. รับรหัสพนักงาน, ชื่อพนักงาน, จำนวนชั่วโมงทำงาน, ค่าจ้างรายชั่วโมง
3. คำนวณ ค่าจ้างสุทธิ = Hours x PayRate
4. แสดงผลลัพธ์ เป็นรหัสพนักงาน ชื่อ และค่าจ้างสุทธิ
5. จบการทำงาน

ตัวอย่างที่ 2

จงเขียนแนวทางการแก้ปัญหาด้วยคอมพิวเตอร์ในการหา
ปริมาตรทรงกระบอก และแสดงค่าปริมาตรที่คำนวณได้

ต้องการอะไร

ต้องการคำนวณปริมาตรของรูปทรงกระบอก

ต้องการเอาต์พุตอย่างไร

ค่าปริมาตรทรงกระบอกเป็นรูปแบบเลขทศนิยมทางจอภาพ

การวิเคราะห์ปัญหา

ข้อมูลเข้า ความสูงของทรงกระบอก (h) และรัศมีของทรงกระบอก(r) เป็นรูปแบบเลขทศนิยม

วิธีการคำนวณ ปริมาตรทรงกระบอก = $\pi \times \text{รัศมี}^2 \times \text{ความสูง}$
ขั้นตอนการประมวลผล

1. เริ่มต้น
2. รับค่าความสูงและค่ารัศมีของทรงกระบอก
3. คำนวณค่าปริมาตร จาก ปริมาตร = $\pi \times r^2 \times h$
4. แสดงค่าปริมาตรทรงกระบอกทางจอภาพ
5. จบการทำงาน

แบบฝึกหัดการวิเคราะห์ปัญหา

- จงวิเคราะห์ปัญหาและเขียนอัลกอริทึมเพื่อรายงานผลสอบของนักศึกษาวิชาคอมพิวเตอร์ โดยให้แสดงคะแนนรวมและเกรดออกมา
- จงวิเคราะห์ปัญหาและเขียนอัลกอริทึมของการบวกตัวเลขจำนวนสามค่าที่อ่านเข้ามาและแสดงผลออกทางเครื่องพิมพ์
- จงวิเคราะห์ปัญหาและเขียนอัลกอริทึมสำหรับหาค่าเฉลี่ยของอุณหภูมิประจำวัน โดยรับค่าอุณหภูมิสูงสุดและอุณหภูมิต่ำสุดเป็นเลขจำนวนเต็มเข้าไปและให้แสดงค่าอุณหภูมิเฉลี่ยออกทางจอภาพ

การทดสอบขั้นตอนวิธีการแก้ปัญหา

ถ้าหากต้องการพัฒนาโปรแกรมให้รับข้อมูลตัวเลขจำนวนเต็มทางแป้นพิมพ์จำนวน 9 ตัว แล้วให้ระบบนับว่าตัวเลขเหล่านั้นมีค่าที่เป็นเลขบวกกี่ตัว เป็นเลขลบกี่ตัว และเป็นศูนย์กี่ตัว โดยให้แสดงผลลัพธ์ทางจอภาพ

จะทดสอบขั้นตอนการแก้ปัญหายังไง ?

วิเคราะห์ขั้นตอนการทำงาน

สิ่งที่ต้องการ	ต้องการทราบจำนวนตัวเลขที่เป็นบวก เป็นลบ และเป็นศูนย์
ผลลัพธ์ที่ต้องการ	แสดงจำนวนตัวเลขที่เป็นบวก, ลบ และศูนย์ทางจอภาพ
ข้อมูลนำเข้า	รับข้อมูลเข้าทางแป้นพิมพ์ทีละตัว
ตัวแปรที่ใช้	ให้ X แทนตัวเลขที่รับเข้าทางแป้นพิมพ์ทีละตัว num_p แทนตัวนับจำนวนที่เป็นบวก num_n แทนตัวนับจำนวนที่เป็นลบ num_z แทนตัวนับจำนวนที่เป็นศูนย์

วิธีการประมวลผล

วิธีการคิด

ถ้าตัวเลขเป็นจำนวนบวก ให้บวก 1 เข้ากับตัวนับจำนวนบวก

ถ้าตัวเลขเป็นจำนวนลบ ให้บวก 1 เข้ากับตัวนับจำนวนลบ

ถ้าตัวเลขเป็นศูนย์ ให้บวก 1 เข้ากับตัวนับจำนวนศูนย์

กรรมวิธีการประมวลผล

1. กำหนดให้ตัวนับ num_p , num_n และ num_z มีค่าเป็นศูนย์
2. อ่านค่าตัวเลขทางแป้นพิมพ์มาเก็บในตัวแปร X
3. ถ้าอ่านไม่ได้ ไปข้อ 7
4. ถ้า X เป็นจำนวนบวก ให้บวก 1 กับ num_p แล้วไปข้อ 2
5. ถ้า X เป็นจำนวนลบ ให้บวก 1 กับ num_n แล้วไปข้อ 2
6. ถ้า X เป็นค่าศูนย์ ให้บวก 1 กับ num_z แล้วไปข้อ 2
7. พิมพ์ค่าตัวนับทางจอภาพทั้งสามตัว
8. หยุด

ตรวจสอบผลลัพธ์

สำหรับกรรมวิธีการประมวลผล ถ้าหากในข้อ 6 ตัดคำว่า

“ถ้า X เป็นค่าศูนย์” ทิ้งไป คำตอบก็จะได้เท่าเดิม

เนื่องจากถ้าหากผ่านข้อ 4 และ 5 มาได้ก็แสดงว่าตัวเลขนั้นมีค่าเป็นศูนย์อยู่แล้ว
สำหรับการตรวจสอบวิธีการประมวลผลสามารถทำได้ดังนี้

1. ลองยกตัวอย่างข้อมูลขึ้นมา 1 ชุด ถ้าหากตัวเลขทั้ง 9 ตัวมีค่าเป็น

13,7,-4,0,-5,2,6,-2 และ 35

2. คิดคำตอบการประมวลผลด้วยตนเอง

จะต้องได้จำนวนบวก 5 ตัว จำนวนลบ 3 ตัว และศูนย์ 1 ตัว

การเขียนโปรแกรม

เปลี่ยนขั้นตอนการทำงานให้อยู่ในรูปรหัสภาษาคอมพิวเตอร์
การเขียนโปรแกรมจะต้องเขียนตามภาษาที่คอมพิวเตอร์เข้าใจโดย
อาจใช้ภาษาระดับสูง หรือระดับต่ำซึ่งสามารถเลือกได้หลายภาษา
การเขียนโปรแกรมแต่ละภาษาจะต้องทำตามหลักไวยากรณ์ (syntax)
ที่กำหนดไว้ในภาษานั้น

โปรแกรมภาษาจาวาอย่างง่าย

```
1: // This program prints Welcome to Java!  
2: public class Welcome {  
3:     public static void main(String[] args) {  
4:         System.out.println("Welcome to  
           Java!");  
5:     }  
6: }
```

เครื่องหมายต่าง ๆ

- { } - เป็นตัวกำหนดขอบเขตหรือบล็อกของฟังก์ชัน
- () - เป็นการระบุตัวผ่านค่าหรืออาร์กิวเมนต์ให้กับฟังก์ชัน
ถ้าภายในวงเล็บไม่มีข้อความใด ๆ แสดงว่าไม่มีตัวผ่านค่าที่ต้องการระบุสำหรับฟังก์ชันนั้น ๆ
- /* */ - เป็นการกำหนด comment หรือข้อความ ที่ไม่ต้องการให้คอมไพเลอร์ปฏิบัติงาน ซึ่งข้อความที่อยู่ภายในเครื่องหมายนี้จะถือว่า ไม่ใช่คำสั่งปฏิบัติงาน