หน่วยที่ 4
การเขียนผังงานและการวิเคราะห์งาน

ผังงาน (Flowchart)

ผังงาน (Flowchart) คือ แผนภาพที่ใช้แสดงลำดับขั้นตอนในการทำงานของโปรแกรม ซึ่งจะใช้ภาพสัญลักษณ์สื่อความหมายแทนแต่ละขั้นตอนของการทำงาน และจะใช้ลูกศรสื่อถึงลำดับขั้นตอนในการทำงาน ซึ่งจะทำให้เราทราบขั้นตอนและลำดับการทำงานของโปรแกรมได้อย่างถูกต้อง โดยทั่วไปแล้วผังงานมักจะเขียนหลังจากที่เรามั่นใจแล้วว่าวิธีการประมวลผลที่คิดขึ้นนั้นถูกต้องแล้ว งานบางประเภทอาจไม่ต้องมีผังงานก็สามารถพัฒนาโปรแกรมให้สำเร็จได้เช่นกัน โดยแบ่งออกเป็น 2 ประเภท

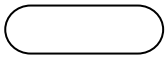
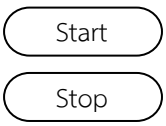
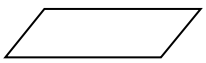
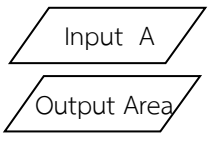
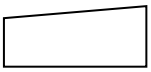
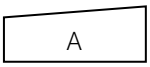
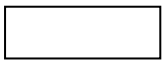
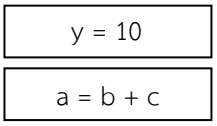
1. **ผังงานระบบ** (System Flowchart) ใช้แสดงขั้นตอนการทำงานภายในระบบงานหนึ่ง ๆ โดยกล่าวถึงข้อมูลต่าง ๆ ที่เกี่ยวข้องทั้งหมด เช่น เอกสารเบื้องต้นคืออะไร วัสดุที่ใช้คืออะไร ใช้หน่วยความจำประเภทใด จะต้องส่งผ่านไปยังหน่วยงานใด วิธีการประมวลผลและการแสดงผลลัพธ์ โดยจะกล่าวอย่างกว้าง ๆ ไม่สามารถนำมาเขียนเป็นโปรแกรมได้
2. **ผังงานโปรแกรม** (Program Flowchart) ผังงานประเภทนี้จะแสดงถึงขั้นตอนของคำสั่งที่ใช้ในโปรแกรม การรับข้อมูล การประมวลผล และการแสดงข้อมูล บางครั้งจะเรียกว่า ผังการเขียนโปรแกรม

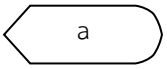
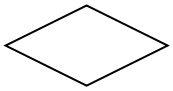
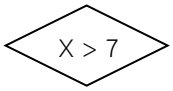
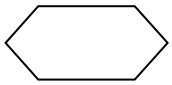
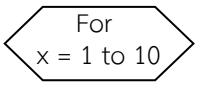
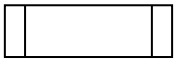
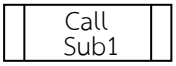
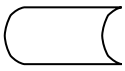
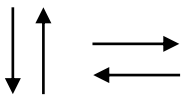
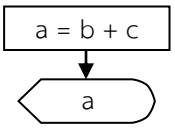
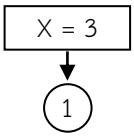
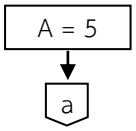
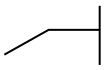
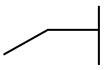
ประโยชน์ของผังงาน

1. ช่วยลำดับขั้นตอนการทำงานของโปรแกรม และสามารถนำไปเขียนโปรแกรมได้โดยไม่สับสน
2. ช่วยในการตรวจสอบ และแก้ไขโปรแกรมได้ง่าย เมื่อเกิดข้อผิดพลาด
3. ช่วยให้การดัดแปลง แก้ไข ทำได้อย่างสะดวกและรวดเร็ว
4. ช่วยให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมได้ง่าย และรวดเร็วมากขึ้น

สัญลักษณ์ที่ใช้ในการเขียนผังงาน

สัญลักษณ์ที่ใช้ในการเขียนผังงาน หมายถึง รูปภาพ ที่หน่วยงาน American National Standard Institute (ANSI) และ International Standard Organization (ISO) ได้กำหนดให้สัญลักษณ์มาตรฐานที่ใช้ในการเขียนผังงาน ดังนี้

สัญลักษณ์รูปภาพ	ความหมาย	ตัวอย่าง	ความหมาย
	แสดงการเริ่มต้น (Strat) หรือ สิ้นสุดของผังงาน (Stop)		1. เริ่มผังงาน 2. จบผังงาน
	รับข้อมูล (Input) หรือ แสดงผลข้อมูล (Output) โดยไม่ระบุชนิดของอุปกรณ์		1. รับค่า A โดยไม่ระบุอุปกรณ์ 2. แสดงผลค่า Area โดยไม่ระบุอุปกรณ์
	รับข้อมูลทางแป้นพิมพ์ (Input from Keyboard)		รับค่า A ทางแป้นพิมพ์
	การกำหนดค่า (Assignment) หรือ การคำนวณ (Process)		1. กำหนดให้ตัวแปร y เท่ากับ 10 2. คำนวณค่า b + c แล้วเก็บผลลัพธ์ไว้ที่ตัวแปร a

สัญลักษณ์รูปภาพ	ความหมาย	ตัวอย่าง	ความหมาย
	การแสดงผลลัพธ์ทางจอภาพ (Display)		แสดงค่า a ทางจอภาพ
	การแสดงผลลัพธ์ทางเครื่องพิมพ์ (Printer)		แสดงค่า Area ทางเครื่องพิมพ์
	การเปรียบเทียบ (Compare) หรือ การตัดสินใจ (Decision)		เปรียบเทียบ X > 7 หรือไม่
	การกำหนดค่าล่วงหน้า หรือ การเตรียมการ		กำหนดให้ x = 1 แล้วเพิ่มขึ้นทีละ 1 จนถึง 10
	โปรแกรมย่อย (Subprogram)		เรียกโปรแกรมย่อยชื่อ Sub1
	รับข้อมูลหรือแสดงผลข้อมูลโดยใช้เทปแม่เหล็ก	–	–
	รับข้อมูลหรือแสดงผลข้อมูลโดยใช้จานแม่เหล็ก	–	–
	การเก็บข้อมูล	–	–
	เส้นแสดงทิศทางการทำงาน (Flow)		ประมวลผลแล้วต่อด้วยแสดงผลลัพธ์ทางจอภาพ
	จุดเชื่อมต่อในหน้าเดียวกัน		หลังกำหนด X = 3 แล้วให้ไปทำงานต่อที่จุดต่อเนื่องที่จุดต่อเนื่อง 1 ในหน้าเดียวกัน
	จุดเชื่อมต่อไปหน้าอื่น		หลังกำหนด A = 5 แล้วให้ไปทำงานต่อที่จุดต่อเนื่อง a ซึ่งอยู่คนละหน้ากัน
	หมายเหตุหรือคำอธิบาย		–

หลักการเขียนผังงานที่ดี

1. มีทางเข้าหรือจุดเริ่มต้น และทางออกหรือจุดสิ้นสุดเพียงทางเดียวเท่านั้น
2. ลำดับขั้นตอนการทำงานควรเริ่มจากบนลงล่าง หรือจากซ้ายไปขวา
3. ในสัญลักษณ์ใด ๆ มีทางออกเพียงทางเดียว ยกเว้นสัญลักษณ์แสดงการตัดสินใจ หรือทางเลือก สามารถมีทางออกได้อย่างน้อยสองทาง
4. เส้นทางเดินในผังงานควรชัดเจน เป็นระเบียบ

5. ข้อความหรือคำสั่งใด ๆ ที่อยู่ในสัญลักษณ์ควรสั้น กระชับ ได้ใจความและสามารถเข้าใจได้ง่าย
6. ใช้สัญลักษณ์ที่มีขนาดเหมาะสมกับคำสั่ง
7. การกำหนดทิศทางการทำงานด้วยลูกศร ควรจะมีทิศทางจากบนลงล่าง หรือขวาไปซ้ายเท่านั้น
8. ในกระบวนการทำงานที่ต้องการเพื่อคำอธิบายเข้าไปเพื่อให้เกิดความเข้าใจ ก็สามารถทำได้โดยการใช้สัญลักษณ์หมายเหตุประกอบ

ขั้นตอนการพัฒนาโปรแกรม

การเขียนโปรแกรมคอมพิวเตอร์ให้ทำงานตามที่เราต้องการนั้น ผู้เขียนโปรแกรมจะต้องรู้ว่าจะให้โปรแกรมทำอะไร มีข้อมูลอะไรที่ต้องให้กับโปรแกรมบ้าง และต้องการอะไรจากโปรแกรมรวมทั้งรูปแบบการแสดงผลด้วย โดยทั่วไปแล้วขั้นตอนการพัฒนาโปรแกรมแบ่งได้ ดังนี้

1. กำหนดและวิเคราะห์ปัญหา (Problem Definition and Problem Analysis)
2. เขียนผังงานและซูโดโค้ด (Flowchart and Pseudocoding)
3. เขียนโปรแกรม (Programming)
4. ทดสอบและแก้ไขโปรแกรม (Program Testing & Debugging)
5. ทำเอกสารและบำรุงรักษาโปรแกรม (Program Documentation and Maintenance)

1. กำหนดและวิเคราะห์ปัญหา (Problem Definition and Problem Analysis)

ขั้นตอนนี้เป็นขั้นตอนแรกสุดที่นักเขียนโปรแกรมจะต้องทำ การให้คอมพิวเตอร์แก้ปัญหาต่าง ๆ ให้เรานั้น จะต้องมีแนวทางที่แก้ไขปัญหาที่เหมาะสมให้กับคอมพิวเตอร์ เพื่อให้การทำงานเป็นไปอย่างมีประสิทธิภาพ โดยมีขั้นตอนย่อย ๆ ดังนี้

1. กำหนดขอบเขตของปัญหา โดยกำหนดรายละเอียดให้ชัดเจนว่าจะให้คอมพิวเตอร์ทำอะไร ตัวแปร ค่าคงที่ที่ต้องใช้เป็นลักษณะใด ถาหากเราไม่กำหนดขอบเขตของปัญหาจะทำให้คอมพิวเตอร์ตัดสินใจได้ยากว่าข้อมูลต่าง ๆ ที่เกิดขึ้นนั้นถูกหรือผิด

2. กำหนดลักษณะของข้อมูลเข้าและข้อมูลออกจากระบบ (Input/Output Specification) โดยต้องรู้ว่า ข้อมูลที่จะส่งเข้าไปเป็นอย่างไร มีอะไรบ้าง เพื่อให้โปรแกรมทำการประมวลผลและแสดงผลลัพธ์ เช่น การรับค่าจากคีย์บอร์ด การใช้เมาส์ การกำหนดปุ่มต่าง ๆ ลักษณะการแสดงผลทางหน้าจอว่า จะให้มีรูปร่างอย่างไร โดยคำนึงถึงผู้ใช้เป็นหลักในการออกแบบโปรแกรม

3. การกำหนดวิธีการประมวลผล (Process Specification) โดยต้องรู้ว่าจะให้คอมพิวเตอร์ทำการประมวลผลอย่างไร จึงจะได้ผลลัพธ์ตามที่ต้องการ

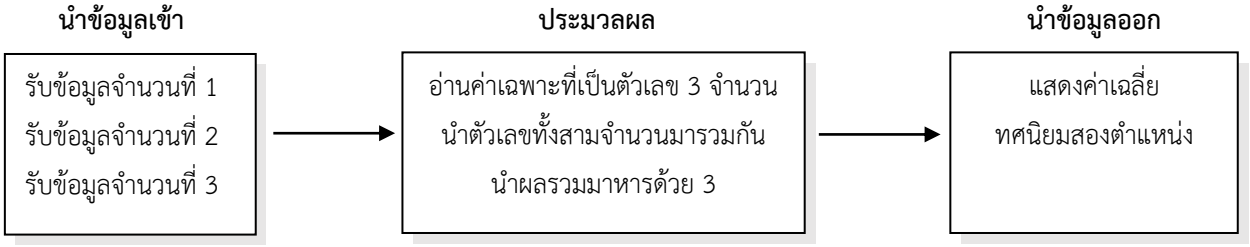
ตัวอย่าง

ถ้าต้องการออกแบบโปรแกรมให้คอมพิวเตอร์รับค่าข้อมูล 3 ค่า และให้แสดงค่าเฉลี่ยทางจอภาพ อาจกำหนดและวิเคราะห์ปัญหาได้ดังนี้

1. รับข้อมูลจากคีย์บอร์ด **<== Input**
    1.1 รับข้อมูลเฉพาะที่เป็นตัวเลขมาเก็บในตัวแปร 3 จำนวน
    1.2 ถ้าข้อมูลเป็น 0 ให้รับค่าใหม่

2. หาค่าเฉลี่ย **<= Process**
    2.1 รวมค่าทุกค่าที่รับมาเข้าด้วยกัน
    2.2 นำค่าผลรวมที่ได้ไปหารด้วย 3
    2.3 นำค่าผลลัพธ์ทีได้ไปเก็บในตัวแปร
3. แสดงผลลัพธ์ทางจอภาพ **<= Output**
    3.1 แสดงคำว่า “ค่าเฉลี่ย เท่ากับ ”
    3.2 แสดงผลลัพธ์ที่ได้ โดยมีทศนิยมสองตำแหน่ง

จะเห็นว่า จะมีการนำปัญหามาแจกแจงย่อยว่าจะต้องทำอะไรบ้าง โดยข้อมูลที่รับเข้าไป คือ ตัวเลขสามจำนวน การประมวลผล คือ การหาค่าเฉลี่ย ส่วนเอาต์พุต คือ การพิมพ์ผลลัพธ์ สามารถเขียนการทำงานของระบบได้ดังแผนภาพดังนี้



กำหนดและวิเคราะห์ปัญหา (Problem Definition and Problem Analysis)

กำหนดขอบเขตของปัญหา ให้ชัดเจนว่าจะให้คอมพิวเตอร์ทำอะไร (What?)

วิเคราะห์ปัญหา (Problem analysis) เป็นขั้นตอนการศึกษาถึงปัญหา

Input : ข้อมูลที่ต้องนำเข้า

Process : วิธีการประมวลผล (How?) ข้อมูลที่นำเข้าเพื่อให้ได้ผลลัพธ์ที่ต้องการ

Output : ผลลัพธ์ที่ต้องการ

ตัวอย่าง เช่น

EX1. ต้องการสร้างโปรแกรมที่มีการนำตัวเลขจำนวนเต็มเข้ามา 5 จำนวน และให้แสดงผลเป็นผลรวมทั้งหมด

- **ข้อมูลนำเข้า (Input)** คือ ตัวเลขจำนวนเต็ม 5 จำนวน เช่น 3 , 5 , 10 , 2 , 4
- **ประมวลผล (Process)** คือ คำนวณหาค่าผลรวม โดยการนำตัวเลขทั้ง 5 จำนวนมารวมกัน
  เช่น 3 + 5 + 10 + 2 + 4
- **ข้อมูลนำออก (Output)** คือ แสดงค่าผลรวมออกทางจอภาพ เช่น 24

EX2. ต้องการสร้างโปรแกรมที่มีการนำตัวเลขจำนวนเต็มเข้ามา 3 จำนวน และให้แสดงผลเป็นค่าเฉลี่ยออกทางจอภาพ

- **ข้อมูลนำเข้า (Input)** คือ ตัวเลขจำนวนเต็ม 3 จำนวน เช่น 3 , 5 , 10
- **ประมวลผล (Process)** คือ คำนวณหาค่าเฉลี่ย โดยนำตัวเลขทั้งสามจำนวนมารวมกันแล้วหารด้วย 3
  เช่น (3 + 5 + 10) / 3
- **ข้อมูลนำออก (Output)** คือ แสดงค่าเฉลี่ยออกทางจอภาพ เช่น 6

2. เขียนผังงานและซูโดโค้ด (Flowchart and Pseudo-coding)

หลังจากที่ได้วิเคราะห์ปัญหาแล้ว ขั้นตอนต่อไปจะต้องใช้เครื่องมือช่วยในการออกแบบโปรแกรม ซึ่งยังไม่ได้เขียนเป็นโปรแกรมจริง ๆ แต่จะช่วยให้เขียนโปรแกรมได้ง่ายขึ้น และทำให้ผู้อื่นนำโปรแกรมไปพัฒนาต่อได้ง่ายขึ้น โดยเขียนเป็นลำดับขั้นตอนการทำงานของโปรแกรมที่เรียกว่า **อัลกอริทึม** (Algorithm) ซึ่งจะแสดงขั้นตอนการแก้ปัญหา โดยใช้ประโยคที่ชัดเจนไม่คลุมเครือ และมีรายละเอียดการทำงานพอสมควรเพียงพอที่จะนำไปเขียนเป็นโปรแกรมให้ทำงานจริง โดยอัลกอลิทึมนั้นอาจเขียนให้อยู่ในรูปของรหัสจำลองหรือซูโดโค้ด (Pseudo-code) หรือเขียนเป็นผังงานก็ได้ โดย**ซูโดโค้ด** จะเป็นคำอธิบายขั้นตอนการทำงานของโปรแกรม เป็นคำย่อไม่มีรูปแบบเฉพาะตัว โดยแต่ละส่วนจะเป็นแนวทางในการเขียนโปรแกรมซึ่งทำให้เขียนโปรแกรมเป็นภาษาต่าง ๆ ได้ง่ายขึ้น ส่วน**ผังงาน**จะใช้สัญลักษณ์ต่าง ๆ แทนการทำงานและทิศทางของโปรแกรม

3. เขียนโปรแกรม (Programming)

หลังจากที่ผ่านขั้นตอนทั้งสองแล้ว ขั้นต่อไปจะต้องเขียนโปรแกรมเพื่อให้คอมพิวเตอร์สามารถประมวลผลได้ โดยเปลี่ยนขั้นตอนการทำงานให้อยู่ในรูปรหัสภาษาคอมพิวเตอร์ การเขียนโปรแกรมจะต้องเขียนตามภาษาที่คอมพิวเตอร์เข้าใจโดยอาจใช้ภาษาระดับสูง หรือระดับต่ำซึ่งสามารถเลือกได้หลายภาษา การเขียนโปรแกรมแต่ละภาษาจะต้องทำตามหลักไวยากรณ์ (Syntax) ที่กำหนดไว้ในภาษานั้น ๆ นอกจากนี้การเลือกใช้ภาษาจะต้องพิจารณาถึงความถนัดของผู้เขียนโปรแกรมด้วย

4. ทดสอบและแก้ไขโปรแกรม (Program Testing & Debugging)

หลังจากเขียนโปรแกรมจะต้องทดสอบความถูกต้องของโปรแกรมที่เขียนขึ้น หาจุดผิดพลาดของโปรแกรมว่ามีหรือไม่ และตรวจสอบจนไม่พบที่ผิดอีก จุดผิดพลาดของโปรแกรมนี้ เรียกว่า **บัก** (Bug) ส่วนการแก้ไขข้อผิดพลาดให้ถูกต้อง เรียกว่า **ดีบัก** (Debug) โดยทั่วไปแล้วข้อผิดพลาดจากการเขียนโปรแกรมจะมี 2 ประเภท คือ

4.1 การเขียนคำสั่งไม่ถูกต้องตามหลักการเขียนโปรแกรมภาษานั้น ๆ ซึ่งเรียกว่า Syntax Error หรือ Coding Error ซึ่งข้อผิดพลาดประเภทนี้มักพบตอนแปลภาษาโปรแกรมเป็นรหัสภาษาเครื่อง

4.2 ข้อผิดพลาดทางตรรกะ หรือ Logic Error เป็นข้อผิดพลาดที่โปรแกรมทำงานได้ แต่ผลลัพธ์ออกมาไม่ถูกต้อง

5. ทำเอกสารและบำรุงรักษาโปรแกรม (Program Documentation and Maintenance)

ขั้นตอนนี้จะทำให้ผู้ใช้สามารถใช้งานโปรแกรมได้อย่างมีประสิทธิภาพ และสะดวกในการตรวจสอบข้อผิดพลาด โดยเขียนเป็นเอกสารประกอบโปรแกรมขึ้นมา โดยทั่วไปแล้วแบ่งออกเป็น 2 ประเภท คือ

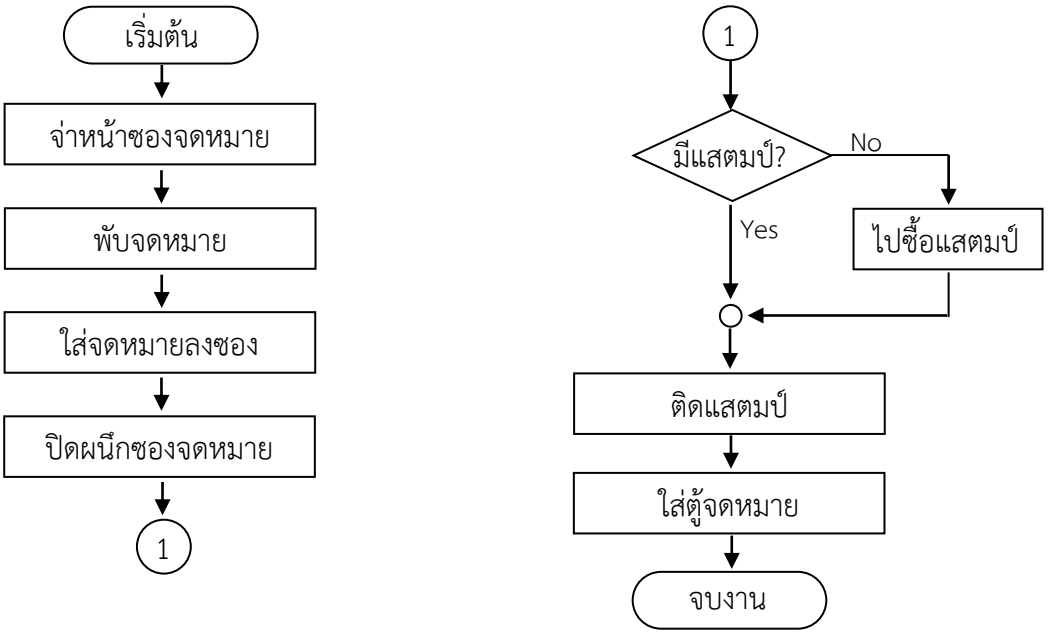
5.1 คู่มือการใช้ หรือ User Document หรือ User Guide ซึ่งจะอธิบายการใช้โปรแกรม

5.2 คู่มือโปรแกรมเมอร์ หรือ Program Document หรือ Technical Reference ซึ่งจะอำนวยความสะดวกในการแก้ไขโปรแกรม และพัฒนาโปรแกรมในอนาคต โดยจะมีรายละเอียดต่าง ๆ เกี่ยวกับโปรแกรม เช่น ชื่อโปรแกรม การรับข้อมูล การพิมพ์ผลลัพธ์ ขั้นตอนต่าง ๆ ในโปรแกรม เป็นต้น

ส่วนการบำรุงรักษาโปรแกรม (Maintenance) เป็นการที่ผู้เขียนโปรแกรมจะต้องคอยตรวจสอบการใช้โปรแกรมจริง เพื่อแก้ไขข้อผิดพลาดซึ่งอาจเกิดขึ้นในภายหลัง รวมทั้งพัฒนาโปแรกรมให้ทันสมัยอยู่เสมอเมื่อเวลาผ่านไป

ตัวอย่างการเขียนผังงานในชีวิตประจำวัน

<u>ตัวอย่างที่ 1</u> การเขียนผังงานขั้นตอนการส่งจดหมาย



<u>ตัวอย่างที่ 2</u> การเขียนผังงานตรวจสอบเงื่อนไขการรับประทานยา โดยแบ่งการรับประทานยาตามอายุ ดังนี้

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

ข้อมูลนำเข้า (Input)

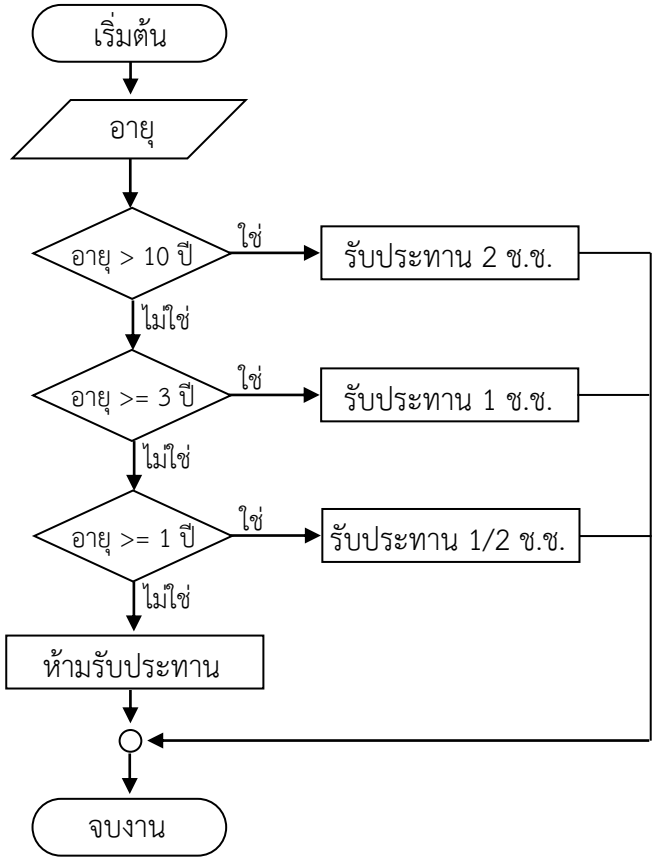
1. อายุ

ประมวลผล (Process)

2. เช็คอายุมากกว่า 10 ปี
3. เช็คอายุ 3 – 10 ปี
4. เช็คอายุ 1 – 3 ปี

ข้อมูลนำออก (Output)

5. แสดงจำนวนการรับประทานยาตามอายุ



ชื่อตัวแปรที่นิยมใช้			
Triangle = รูปสามเหลี่ยม	Base = ฐาน	Long = ยาว	Number = ตัวเลข, จำนวน
Square = รูปสี่เหลี่ยม	High = ความสูง	Area = พื้นที่	PI = π
Circle = รูปวงกลม	Width = ความกว้าง	Radius = รัศมี	Name = ชื่อ

ตัวอย่างที่ 3 ผังงานขั้นตอนการคำนวณหาพื้นที่รูปสี่เหลี่ยม

ข้อมูลนำเข้า (Input)

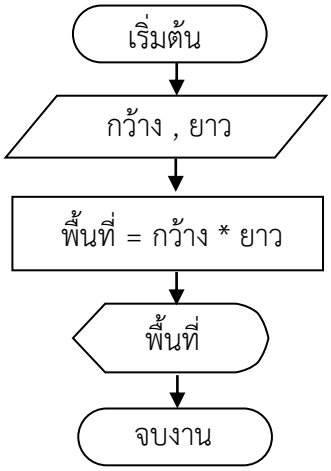
1. ความกว้าง
2. ความยาว

ประมวลผล (Process)

3. คำนวณหาพื้นที่รูปสี่เหลี่ยม
   พื้นที่ = กว้าง * ยาว

ข้อมูลนำออก (Output)

4. แสดงผลลัพธ์พื้นที่ทางจอภาพ



ตัวอย่างที่ 4 ผังงานขั้นตอนการคำนวณหาพื้นที่รูปสามเหลี่ยม

ข้อมูลนำเข้า (Input)

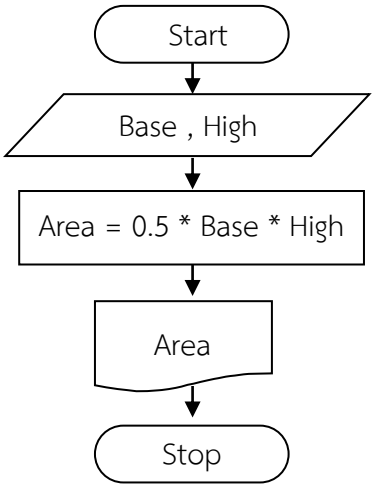
1. ความยาวฐาน (Base)
2. ความสูง (High)

ประมวลผล (Process)

3. คำนวณหาพื้นที่รูปสามเหลี่ยม
   Area = 0.5 * Base * High

ข้อมูลนำออก (Output)

4. แสดงผลลัพธ์พื้นที่ (Area)
   ทางกระดาษ



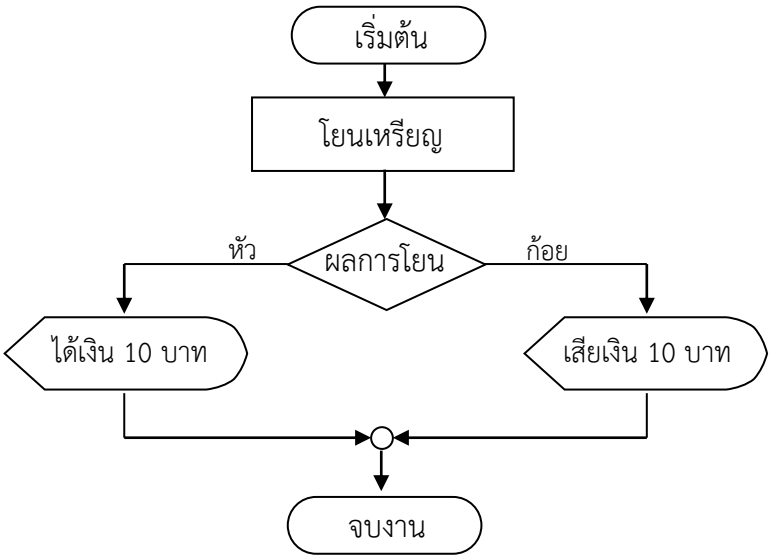
ตัวอย่างที่ 5 ผังงานขั้นตอนการโยนเหรียญ โดยให้แสดงการโยนเหรียญ ถ้าออกหัวผู้โยนจะได้เงิน 10 บาท ถ้าออกก้อยผู้โยนจะเสียเงิน 10 บาท

ข้อมูลนำเข้า (Input)

1. โยนเหรียญ

ประมวลผล (Process)/ข้อมูลนำออก (Output)

2. ตรวจสอบเงื่อนไข ถ้าผลการโยนเหรียญออกหัว แล้วทำ
   2.1 แสดงผลได้เงิน 10 บาททางจอภาพ มิฉะนั้นแล้ว
   2.2 แสดงผลเสียเงิน 10 บาททางจอภาพ



<u>ตัวอย่างที่ 6</u> ผังงานขั้นตอนการหาค่าเฉลี่ย โดยรับค่าตัวเลขจำนวนเต็มไปเก็บในตัวแปร A , B และ C จากนั้นให้คำนวณแล้วให้แสดงผลเป็นค่าเฉลี่ยออกทางจอภาพ

ข้อมูลนำเข้า (Input)

1. รับค่า A
2. รับค่า B
3. รับค่า C

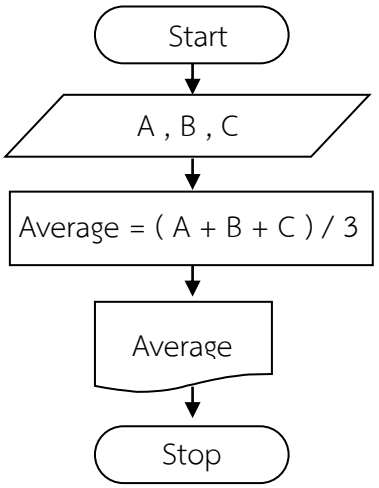
ประมวลผล (Process)

4. คำนวณหาเฉลี่ย

Average = (A + B + C) / 3

ข้อมูลนำออก (Output)

5. แสดงผลลัพธ์ค่าเฉลี่ย (Average) ทางกระดาษ



<u>ตัวอย่างที่ 7</u> ผังงานขั้นตอนการจับไม้สั้นไม้ยาวสามครั้ง ถ้าได้ไม้ยาวให้ได้คะแนน 100 คะแนน ถ้าได้ไม้สั้นให้เสียคะแนน 100 คะแนน แล้วให้แสดงผลคะแนนออกทางจอภาพ

ข้อมูลนำเข้า (Input)

1. กำหนดคะแนน ค่าเริ่มต้น Total = 0
2. กำหนดตัวนับรอบ ค่าเริ่มต้น i = 0
3. เลือกจับไม้

ประมวลผล (Process)/ข้อมูลนำออก (Output)

4. ตรวจสอบเงื่อนไข ถ้าผลการจับไม้ได้ไม้ยาว แล้วทำ

4.1 คำนวณคะแนน

Total = Total + 100

มิฉะนั้นแล้ว

4.2 คำนวณคะแนน

Total = Total – 100

5. คำนวณเพิ่มค่าการนับรอบ

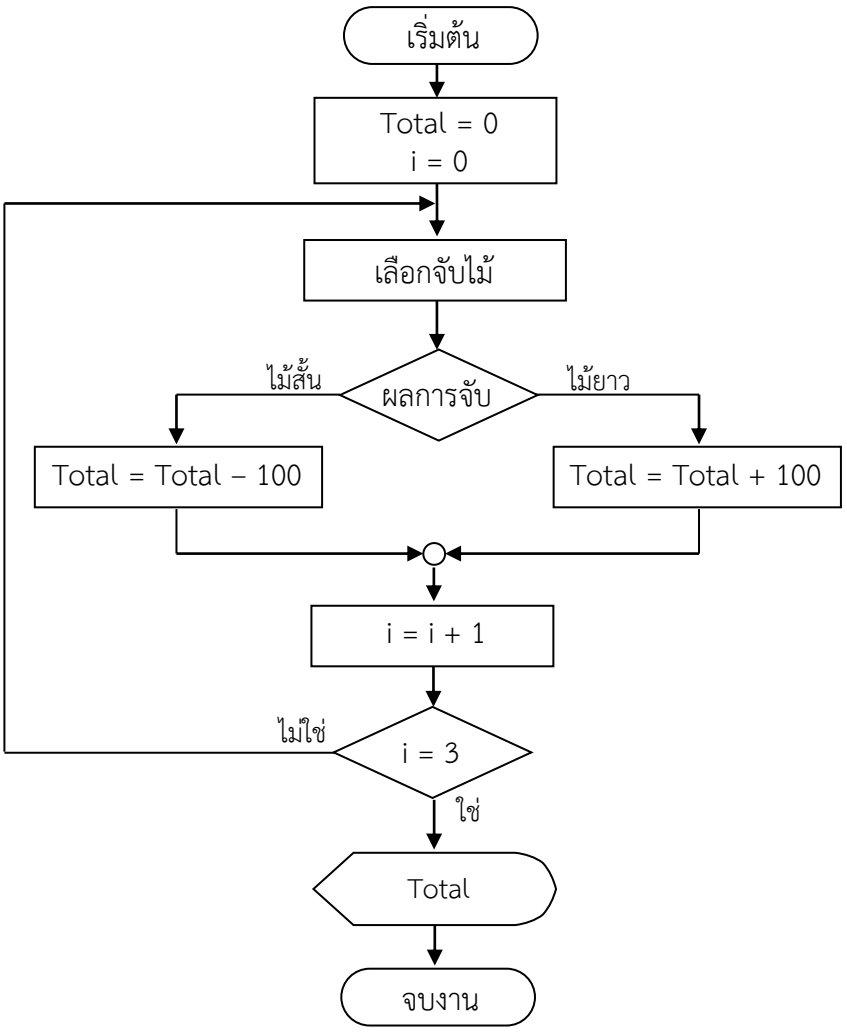
i = i + 1

6. ตรวจสอบเงื่อนไข ถ้าการจับไม้ยังไม่ครบ 3 ครั้ง แล้วทำ

6.1 กลับไปทำข้อ 3

มิฉะนั้นแล้ว

6.2 แสดงผลลัพธ์คะแนน (Total) ทางจอภาพ



ใบงานหน่วยที่ 4

1. ให้นักเรียนเขียนผังงานขั้นตอนการคำนวณหาพื้นที่วงกลมจากข้อมูลที่กำหนดให้ (π = 3.14)

สูตร การหาพื้นที่วงกลม = πr^2

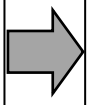
ข้อมูลนำเข้า (Input Unit)
- รัศมี (Radius)

ประมวลผล (Process)
- คำนวณหาพื้นที่รูปวงกลม

Area = (22.0/7.0) * Radius * Radius

หรือ Area = 3.14 * Radius * Radius

ข้อมูลนำออก (Output Unit)
- แสดงผลพื้นที่วงกลมทางจอภาพ (Area)



2. ให้นักเรียนเขียนผังงานขั้นตอนการคำนวณหาความยาวเส้นรอบวงกลม (π = 3.14)

สูตร การหาความยาวเส้นรอบวง = $2\pi r$

ข้อมูลนำเข้า (Input Unit)

.....

.....

.....

ประมวลผล (Process)

.....

.....

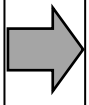
.....

ข้อมูลนำออก (Output Unit)

.....

.....

.....



3. ให้นักเรียนเขียนผังงานขั้นตอนการรับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วแสดงผลว่าตัวเลขนั้นเป็นเลขคู่ (Even Number) หรือไม่

ข้อมูลนำเข้า (Input Unit)

.....

.....

.....

ประมวลผล (Process)

.....

.....

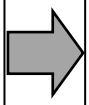
.....

ข้อมูลนำออก (Output Unit)

.....

.....

.....



4. ให้นักเรียนเขียนผังงานขั้นตอนการรับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วแสดงผลว่าเป็นเลขคู่ (Even Number) หรือเลขคี่ (Odd Number)

ข้อมูลนำเข้า (Input Unit)

.....

.....

.....

ประมวลผล (Process)

.....

.....

.....

ข้อมูลนำออก (Output Unit)

.....

.....

.....