

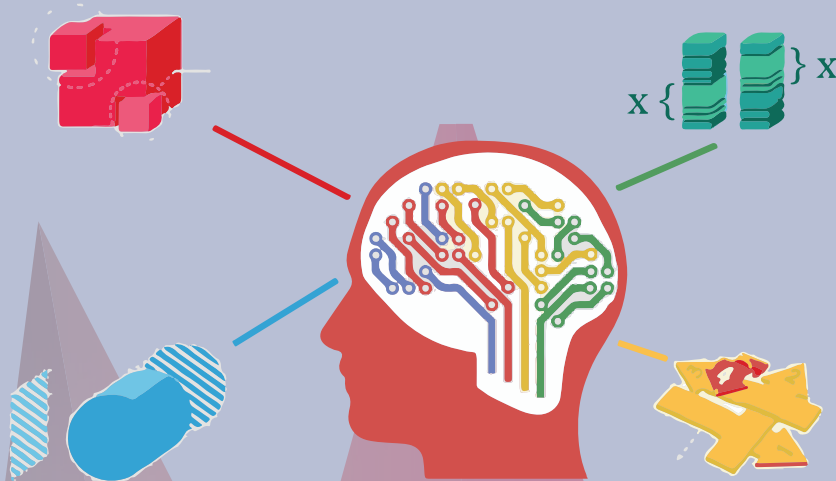
หนังสือเรียน รายวิชาพื้นฐานวิทยาศาสตร์และเทคโนโลยี

เทคโนโลยี

(วิทยาการคำนวณ)

ชั้น ม.๑

ตามมาตรฐานการเรียนรู้และตัวชี้วัด กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี
(ฉบับปรับปรุง พ.ศ. ๒๕๖๐) ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช ๒๕๕๑



ผู้เรียบเรียง

นายพัฒพงษ์ อมรวงศ์

ผู้ตรวจ

นายภูซังค์ จันทร์เปล่ง

นายวิวัฒน์ สุนันทวงศ์

นายสุทิน โรจน์ประเสริฐ

บรรณาธิการ

นายวิเชียร วิสูงเส



บริษัท มีเดีย อินเทลลิเจนซ์ เทคโนโลยี จำกัด
Media Intelligence Technology Co.,Ltd.

หนังสือเรียน รายวิชาพื้นฐานวิทยาศาสตร์และเทคโนโลยี

เทคโนโลยี (วิทยาการคำนวณ) ชั้น ม.๑

ตามมาตรฐานการเรียนรู้และตัวชี้วัด กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี
(ฉบับปรับปรุง พ.ศ. ๒๕๖๐) ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช ๒๕๕๑

พัฒพงษ์ อมรวงศ์

ราคา ๙๐ บาท

สงวนลิขสิทธิ์ © พ.ศ. ๒๕๖๒ โดย พัฒพงษ์ อมรวงศ์
ห้ามการลอกเลียนไม่ว่าส่วนหนึ่งส่วนใดของหนังสือเล่มนี้
นอกจากจะได้รับอนุญาต
พิมพ์ครั้งที่ ๑ พ.ศ. ๒๕๖๒
จำนวน ๓,๐๐๐ เล่ม

ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ
พัฒพงษ์ อมรวงศ์

หนังสือเรียน รายวิชาพื้นฐานวิทยาศาสตร์และเทคโนโลยี เทคโนโลยี
(วิทยาการคำนวณ) ชั้น ม.๑ -- ปทุมธานี: มีเดีย อินเทลลิเจนซ์ เทคโนโลยี,
๒๕๖๒.

๑๔๐ หน้า.

1. วิทยาศาสตร์ (วิทยาการคำนวณ) I. ชื่อเรื่อง

ISBN 978-616-7319-77-3

จัดพิมพ์และจัดจำหน่ายโดย



บริษัท มีเดีย อินเทลลิเจนซ์ เทคโนโลยี จำกัด

Media Intelligence Technology Co.,Ltd.

๔๔/๘๓ หมู่ ๗ ต.คลองสอง อ.คลองหลวง จ.ปทุมธานี ๑๒๑๒๐

โทรศัพท์ ๐-๒๕๑๖-๕๖๙๐-๑ โทรสาร ๐-๒๕๑๖-๕๖๙๒

www.mitmedia.com e-mail : mediamit@yahoo.com



ออกแบบ : KanomDesign Studio

พิมพ์ที่ : บริษัท เยลโล่การพิมพ์ (1988) จำกัด



คำนำ

หนังสือเรียนรายวิชาพื้นฐานวิทยาศาสตร์และเทคโนโลยี เทคโนโลยี
(วิทยาการคำนวณ) ชั้น ม.๑ นี้ มีเนื้อหาครอบคลุมตั้งแต่ การออกแบบอัลกอริทึม
การออกแบบโปรแกรมอย่างง่าย การรวบรวมและประมวลผลข้อมูล การนำเสนอข้อมูล
และสารสนเทศ และการใช้เทคโนโลยีสารสนเทศอย่างปลอดภัย มีสาระการเรียนรู้
ทางเทคโนโลยีตรงตามผลการเรียนรู้ คือ “เข้าใจและใช้แนวคิดเชิงคำนวณในการแก้ปัญหา
ที่พบในชีวิตจริงอย่างเป็นขั้นตอนและเป็นระบบ ใช้เทคโนโลยีสารสนเทศและการสื่อสาร
ในการเรียนรู้ การทำงาน และการแก้ปัญหาได้อย่างมีประสิทธิภาพ รู้เท่าทันและมีจริยธรรม”
ซึ่งทั้งหมดนี้เป็นพื้นฐานที่สำคัญในพัฒนาโปรแกรม การนำเสนอข้อมูลและสารสนเทศ
และความปลอดภัยในการใช้เทคโนโลยีสารสนเทศ

ในการจัดทำหนังสือรายวิชาพื้นฐานวิทยาศาสตร์และเทคโนโลยี เทคโนโลยี
(วิทยาการคำนวณ) ชั้น ม.๑ นี้ ผู้เขียนได้เน้นเนื้อหาในเชิงการฝึกปฏิบัติในการออกแบบ
โปรแกรมอย่างง่ายด้วยภาษาไพทอนด้วย โดยออกแบบขั้นตอนการให้เนื้อหาจากง่ายไปยาก
ให้ผู้เรียนปฏิบัติตาม และมีภาพประกอบเพื่อเสริมการเรียนรู้ ผู้เรียนจึงสามารถใช้หนังสือนี้
เป็นคู่มือการเรียนรู้ได้ทั้งการเรียนรู้ด้วยตนเอง และฝึกปฏิบัติในชั้นเรียน แต่ละบทมี
กิจกรรมและการประเมินผลการเรียนรู้ให้ผู้เรียนได้ฝึกทักษะและทบทวนความรู้ อันจะทำให้
ผู้เรียนมีความรู้ ความเข้าใจ และมีทักษะการเขียนโปรแกรมภาษาไพทอนเพิ่มขึ้น

ผู้เขียนหวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะเป็นแนวทางให้ผู้เรียนได้พัฒนา
ศักยภาพในการออกแบบอัลกอริทึม การออกแบบโปรแกรมภาษาไพทอน สามารถ
รวบรวมและนำเสนอข้อมูล และใช้เทคโนโลยีสารสนเทศอย่างปลอดภัยต่อไป

พัฒพงษ์ อมรวงศ์

apatpong@gmail.com

สารบัญ

หน่วยที่ 1 การออกแบบอัลกอริทึม

1.1 ความรู้พื้นฐานเกี่ยวกับอัลกอริทึม	2
กิจกรรม 1-1 เขียนอัลกอริทึมทางกลับบ้านหลังเลิกเรียน	8
1.2 การออกแบบอัลกอริทึมโดยใช้ผังงาน (Flowchart)	8
กิจกรรม 1-2 การเขียนผังงานการเดินทางกลับบ้าน	13
1.3 การออกแบบอัลกอริทึมโดยใช้รหัสเทียม (Pseudo-code)	14
กิจกรรม 1-3 การเขียนรหัสเทียมของการหาพื้นที่สามเหลี่ยม	16
1.4 การออกแบบอัลกอริทึมเพื่อแก้ปัญหาทางคณิตศาสตร์	
วิทยาศาสตร์อย่างง่าย	18
กิจกรรม 1-4 การเขียนรหัสเทียมคำนวณหาพื้นที่แผ่นหญ้า	23
สรุป	23
การวัดและประเมินผลการเรียนรู้ หน่วยที่ 1	26

หน่วยที่ 2 การออกแบบโปรแกรมอย่างง่าย

2.1 ประเภทของตัวแปลภาษา	28
2.2 คุณลักษณะของภาษาไพทอน	29
2.3 การดาวน์โหลดและติดตั้งชุดพัฒนาโปรแกรมภาษาไพทอน	30
2.4 การใช้งานแบบโต้ตอบ	35
2.5 การสร้างโปรแกรม และบันทึกโปรแกรม	37
2.6 การทดสอบโปรแกรม และการแก้ไขที่ผิด	43
กิจกรรม 2-1 การสร้างโปรแกรมนามบัตร	46
2.7 การออกแบบและเขียนโปรแกรมที่มีการใช้ตัวแปร	46
กิจกรรม 2-2 การสร้างโปรแกรมเพื่อกำหนดข้อมูลแบบจำนวนเต็ม	50
กิจกรรม 2-3 การเพิ่มข้อความประกอบการแสดงค่าของตัวแปร	51
กิจกรรม 2-4 การเพิ่มข้อความประกอบการแสดงค่าของตัวแปร	52
2.8 การออกแบบและเขียนโปรแกรมที่มีการใช้เงื่อนไข	53
กิจกรรม 2-5 การเขียนผังงานและโปรแกรมเพื่อกำหนดค่าของตัวแปร	57

กิจกรรม 2-6 การเขียนผังงานและโปรแกรมเพื่อกำหนดค่าของตัวแปร	58
กิจกรรม 2-7 สร้างโปรแกรมเพื่อป้อนค่า	61
กิจกรรม 2-8 เขียนผังงานและสร้างโปรแกรมเพื่อป้อนค่า	62
กิจกรรม 2-9 เขียนผังงานและสร้างโปรแกรม mygrade.py	62
2.9 การออกแบบและเขียนโปรแกรมที่มีการวนซ้ำ	65
กิจกรรม 2-10 สร้างโปรแกรม count10.py	68
กิจกรรม 2-11 ออกแบบโปรแกรมสำหรับวนซ้ำพิมพ์ข้อความ	68
กิจกรรม 2-12 โปรแกรม forCount.py	71
กิจกรรม 2-13 โปรแกรม forDown.py	72
2.10 การออกแบบและเขียนโปรแกรม เพื่อแก้ปัญหาทางคณิตศาสตร์	
วิทยาศาสตร์อย่างง่าย	72
กิจกรรม 2-14 โปรแกรมคำนวณหาพื้นที่สี่เหลี่ยมคางหมู	76
กิจกรรม 2-15 โปรแกรมคำนวณหาความเร็วในการวิ่งของนักเรียน	76
กิจกรรม 2-16 โปรแกรมคำนวณดัชนีมวลกาย	77
สรุป	78
การวัดและประเมินผลการเรียนรู้ หน่วยที่ 2	80

หน่วยที่ 3 การรวบรวมและประมวลผลข้อมูล

3.1 ข้อมูล และประเภทของข้อมูล	82
กิจกรรม 3-1 ข้อมูลและประเภทของข้อมูล	83
3.2 การรวบรวมข้อมูลปฐมภูมิ	83
กิจกรรม 3-2 การรวบรวมข้อมูล	86
3.3 การประมวลผลข้อมูล	86
กิจกรรม 3-3 การประมวลผลข้อมูล	87
3.4 ขั้นตอนการประมวลผลข้อมูล	87
กิจกรรม 3-4 การสืบค้นข้อมูล	88
3.5 การใช้โปรแกรมประมวลผลข้อมูลอย่างง่าย	89
กิจกรรม 3-5 การคำนวณหาค่าเฉลี่ย	95
กิจกรรม 3-6 การหาค่า BMI ทศนิยม 2 ตำแหน่ง	98
กิจกรรม 3-7 การหาค่าคำนวณหาดัชนีมวลกายของนักเรียน	99

สรุป	99
การวัดและประเมินผลการเรียนรู้ หน่วยที่ 3	101

หน่วยที่ 4 การนำเสนอข้อมูลและสารสนเทศ 102

4.1 ลักษณะของการนำเสนอข้อมูลและสารสนเทศ	103
กิจกรรม 4-1 การนำเสนอข้อมูลในรูปข้อความกึ่งตาราง	104
4.2 การนำเสนอโดยใช้ตาราง	104
กิจกรรม 4-2 การนำเสนอข้อมูลในรูปตาราง	105
4.3 การนำเสนอข้อมูลโดยใช้แผนภูมิ	105
กิจกรรม 4-3 การนำเสนอข้อมูลเป็นแผนภูมิรูปภาพ	106
กิจกรรม 4-4 การนำเสนอข้อมูลในรูปแผนภูมิแท่ง	109
กิจกรรม 4-5 การนำเสนอข้อมูลในรูปแผนภูมิเส้น	111
4.4 การสร้างกราฟผลการทดลอง และวิเคราะห์แนวโน้ม	112
กิจกรรม 4-6 การสร้างงานนำเสนอข้อมูลด้วยแผนภูมิ	126
สรุป	126
การวัดและประเมินผลการเรียนรู้ หน่วยที่ 4	128

หน่วยที่ 5 การใช้เทคโนโลยีสารสนเทศอย่างปลอดภัย 129

5.1 การใช้เทคโนโลยีสารสนเทศอย่างปลอดภัย	130
กิจกรรม 5-1 ค้นหาเว็บไซต์ที่มีเครื่องหมาย Secured site	132
5.2 การจัดการอัตลักษณ์	133
กิจกรรม 5-2 การห้ามใช้รหัสผ่านเดียวกับทุกบัญชี	134
5.3 การพิจารณาความเหมาะสมของเนื้อหา	134
กิจกรรม 5-3 เว็บไซต์ที่ไม่เหมาะสม	135
5.4 ข้อตกลง ข้อกำหนดในการใช้สื่อ หรือแหล่งข้อมูลต่างๆ	135
กิจกรรม 5-4 ค้นหาเว็บไซต์ที่มีสัญลักษณ์ข้อตกลงของครีเอทีฟคอมมอนส์	138
สรุป	138
การวัดและประเมินผลการเรียนรู้ หน่วยที่ 5	139

บรรณานุกรม 140

หน่วยที่ 1

การออกแบบอัลกอริทึม

มาตรฐานการเรียนรู้

1. บอกความรู้พื้นฐานเกี่ยวกับอัลกอริทึมได้
2. ออกแบบอัลกอริทึมโดยใช้ผังงานได้
3. ออกแบบอัลกอริทึมโดยใช้รหัสเทียมได้
4. ออกแบบอัลกอริทึมเพื่อแก้ปัญหาทางคณิตศาสตร์ วิทยาศาสตร์อย่างง่ายได้

ตัวชี้วัดและสาระแกนกลาง

1. ความรู้พื้นฐานเกี่ยวกับอัลกอริทึม
2. การออกแบบอัลกอริทึมโดยใช้ผังงาน (Flowchart)
3. การออกแบบอัลกอริทึมโดยใช้รหัสเทียม (Pseudo-code)
4. การออกแบบอัลกอริทึมเพื่อแก้ปัญหาทางคณิตศาสตร์ วิทยาศาสตร์อย่างง่าย

1.1 ความรู้พื้นฐานเกี่ยวกับอัลกอริทึม

คำว่า “อัลกอริทึม (Algorithm)” หลายคนอาจจะไม่เคยรู้จัก แต่บางคนที่ใช้อินเทอร์เน็ตสำหรับค้นหาข้อมูลหรือภาพด้วยกูเกิล อาจเคยเห็นข่าวการปรับปรุงอัลกอริทึมของกูเกิลดังภาพที่ 1.1



ภาพที่ 1.1 การปรับปรุงอัลกอริทึมของกูเกิล

หรือคนที่ใช้เฟซบุ๊กก็อาจเคยเห็นการปรับปรุงอัลกอริทึมดังภาพที่ 1.2



ภาพที่ 1.2 การปรับปรุงอัลกอริทึมของเฟซบุ๊ก

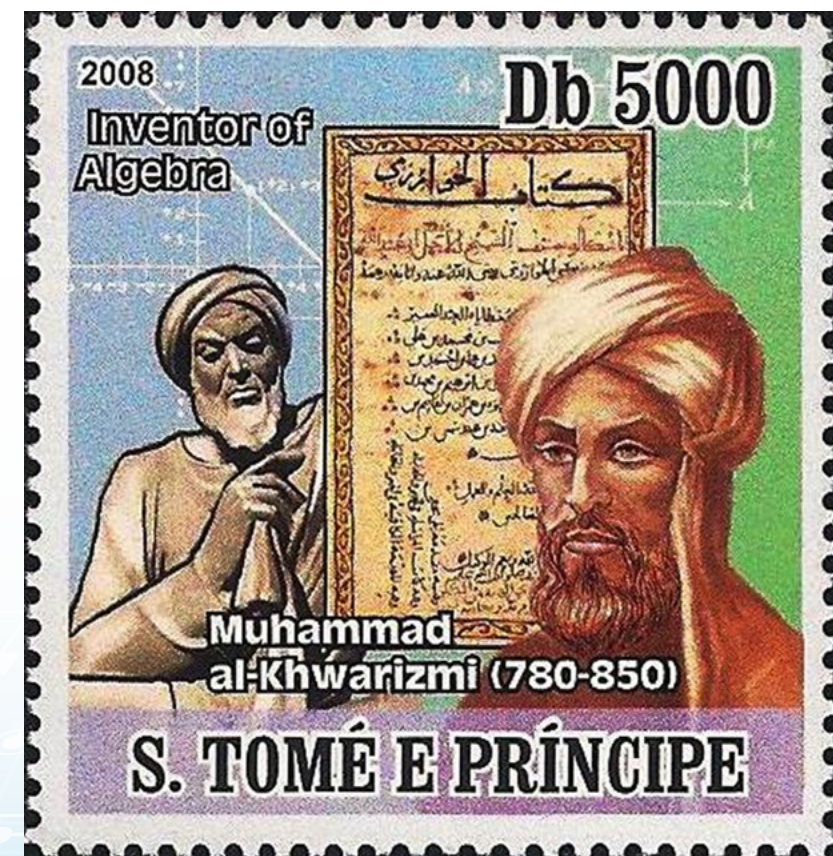
ในหน่วยที่ 1 นี้ นักเรียนจะได้ศึกษาเกี่ยวกับอัลกอริทึมหมายถึงอะไร มีความสำคัญ และมีประโยชน์อย่างไรในชีวิตจริงของเรา การออกแบบและปรับปรุงอัลกอริทึมทำได้อย่างไร ลองพิจารณาศึกษาดูในบทนี้

1.1.1 ความหมายของอัลกอริทึม

อัลกอริทึม หมายถึง ชุดขั้นตอนการทำงานตามลำดับที่คอมพิวเตอร์ และอุปกรณ์อัจฉริยะอื่นๆ (เช่น สมาร์ทโฟน) สามารถทำได้ เพื่อทำการคำนวณ ประมวลผลข้อมูล และทำงานที่ต้องใช้เหตุผลอย่างอัตโนมัติ (Association for Computing Machinery, 2017)

โดยทั่วไปแล้ว อัลกอริทึม หมายถึง ลำดับขั้นตอนที่ถูกออกแบบไว้อย่างชัดเจนในการแก้ปัญหาใดปัญหาหนึ่ง หรือขั้นตอนวิธีการทำงานของงานใดงานหนึ่งให้สำเร็จ ปัญหาที่ต้องแก้ อาจเป็นปัญหาในเรื่องต่างๆ หรือการทำงานในชีวิตจริง เช่น ขั้นตอนการปรุงอาหาร ขั้นตอนการเดินทางไปเที่ยวต่างจังหวัด ขั้นตอนการแก้ปัญหาทางคณิตศาสตร์ด้วยคอมพิวเตอร์ ฯลฯ

คำว่า “อัลกอริทึม” มาจากชื่อของ al-Kwarizmi นักคณิตศาสตร์ชาวเปอร์เซียในศตวรรษที่ 8 ในภาพที่ 1.3 ซึ่งเขียนหนังสือเกี่ยวกับพีชคณิตของการคำนวณเชิงตัวเลข จนได้รับการยกย่องให้เป็น “บิดาแห่งพีชคณิต” และเป็นผู้ริเริ่มใช้ตัวเลขแบบฮินดู-อารบิก 1, 2, 3, ... ซึ่งทำให้การคำนวณง่ายกว่าการใช้ตัวเลขแบบโรมัน I, II, III, ...

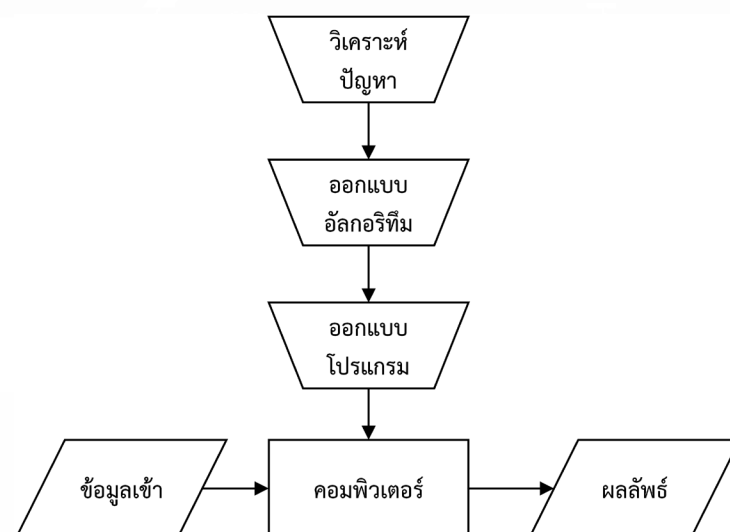


ภาพที่ 1.3 Muhammad ibn Musa al-Khwarizmi (Wobbe Vegter, April 2005)

1.1.2 ความสำคัญของอัลกอริทึม

ในชีวิตประจำวัน ทุกคนต้องเคยพบกับปัญหาต่างๆ ไม่ว่าจะเป็นปัญหาด้านการเงิน การงาน การเรียน หรือแม้แต่การเล่นเกม เมื่อพบกับปัญหา แต่ละคนมีวิธีที่จะจัดการ หรือ คิดอัลกอริทึมหรือขั้นตอนในการแก้ปัญหาเหล่านั้นแตกต่างกันไป ซึ่งแต่ละวิธีการอาจให้ผลลัพธ์ ที่เหมือนกัน หรือแตกต่างกันบ้าง ทั้งนี้ขึ้นอยู่กับความรู้ ความสามารถ และประสบการณ์ของ บุคคลผู้นั้น

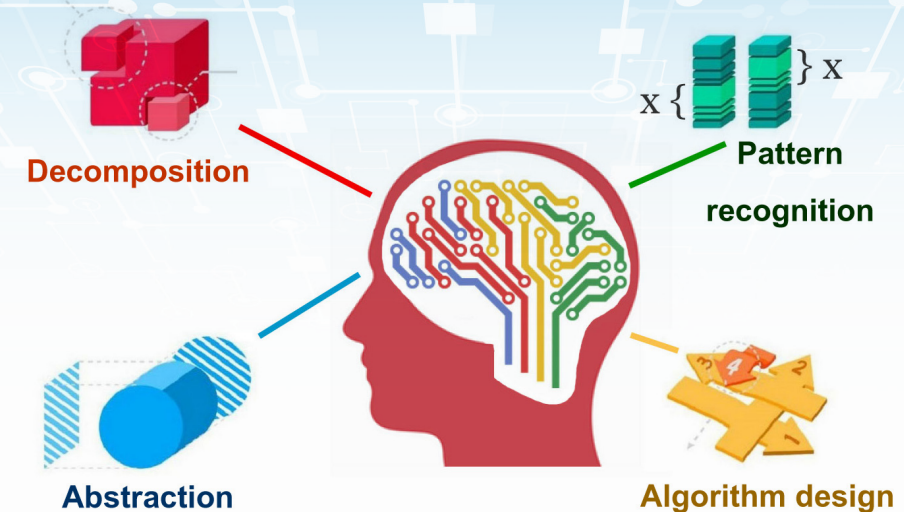
การนำคอมพิวเตอร์มาช่วยในการแก้ปัญหา ก็ต้องมีขั้นตอนเช่นกัน คือ วิเคราะห์ปัญหา แล้วออกแบบขั้นตอนการทำงาน หรืออัลกอริทึมก่อน เมื่อตรวจสอบความถูกต้องของอัลกอริทึม ดีแล้ว จึงจะนำอัลกอริทึมไปออกแบบและเขียนโปรแกรมสั่งการให้คอมพิวเตอร์ทำงานตาม ที่ต้องการต่อไป ดังภาพที่ 1.4



ภาพที่ 1.4 ขั้นตอนการแก้ปัญหาด้วยคอมพิวเตอร์

ถ้าการออกแบบอัลกอริทึมทำได้ไม่ดี หรือไม่สามารถทำได้ การออกแบบโปรแกรม ก็ทำได้ไม่ดี หรือไม่สามารถทำได้เช่นกัน แต่ถ้าออกแบบอัลกอริทึมดี การออกแบบและเขียน โปรแกรมก็ทำได้อย่างรวดเร็วและง่ายขึ้น ดังนั้น นักคอมพิวเตอร์ จะต้องหาวิธีการที่จะให้ได้มา ซึ่งอัลกอริทึมที่ดีและมีประสิทธิภาพมากที่สุด นั่นคือ สามารถแก้ปัญหาได้ในเวลาอันรวดเร็วที่สุด และใช้ทรัพยากรน้อยที่สุด

การออกแบบอัลกอริทึมจึงถูกจัดให้เป็นส่วนสำคัญอีกส่วนหนึ่งที่นักคอมพิวเตอร์จะต้อง สามารถทำได้ โดยนักคอมพิวเตอร์ที่ดีจะต้องมีวิธีการคิดเชิงคำนวณ (Computational thinking) ซึ่งประกอบด้วยแนวคิดแบบต่างๆ ดังภาพที่ 1.5



ภาพที่ 1.5 การคิดเชิงคำนวณ (Computational thinking)

1.1.3 การคิดเชิงคำนวณ

การคิดเชิงคำนวณ เป็นกระบวนการคิดที่ต้องใช้ทักษะและเทคนิคเพื่อแก้ปัญหาในเชิง นามธรรม

การคิดเชิงคำนวณ ประกอบด้วย

- 1) การแบ่งย่อยปัญหา (Decomposition) เมื่อมีปัญหา สามารถวิเคราะห์หรือ แบ่งปัญหาออกเป็นปัญหาย่อยๆ ได้ แล้วทำการแก้ปัญหาทีละปัญหาย่อย
- 2) การมองรูปแบบของปัญหา (Pattern recognition) สามารถรู้จำรูปแบบ (Pattern) ของปัญหา หรือสถานการณ์ที่เกิดขึ้นซ้ำๆ กันได้ เพื่อหาวิธีแก้ปัญหาที่คล้ายกัน
- 3) การมองภาพรวม หรือแนวคิดเชิงนามธรรม (Abstraction) เป็นการประเมิน ความสำคัญในรายละเอียดของปัญหา แยกแยะส่วนที่เป็นสาระสำคัญออกจากส่วนที่ไม่ใช่สาระ สำคัญ
- 4) การออกแบบอัลกอริทึม (Algorithm design) เป็นส่วนที่สำคัญที่สุด นั่นคือ การออกแบบลำดับการทำงานที่สามารถกำหนดขั้นตอนวิธีการแก้ปัญหา คิดหาวิธีการทำงานให้ เร็วและสั้นที่สุดจนได้ผลลัพธ์ตามที่กำหนด

1.1.4 คุณสมบัติของอัลกอริทึม

ในการออกแบบอัลกอริทึมไม่ว่าจะเป็นอัลกอริทึมที่ใช้กับชีวิตจริงหรืออัลกอริทึมที่ใช้กับ คอมพิวเตอร์จะต้องมีคุณสมบัติดังต่อไปนี้

- อะไรก็ได้
- 1) มีการทำงานที่ชัดเจน และทำได้จริง ไม่คลุมเครือ เช่น รับค่าตัวเลข 2 ค่า ไม่ใช่รับอะไรก็ได้
 - 2) มีลำดับขั้นตอนการทำงานที่ชัดเจน จนจบการทำงาน ไม่ใช่ทำงานจนไม่มีการหยุด
 - 3) ง่าย หรือตรวจสอบแก้ไขได้
 - 4) มีความถูกต้อง ทั้งการทำงาน และผลลัพธ์
 - 5) ทำงานได้ครบทุกเงื่อนไข
 - 6) มีประสิทธิภาพ คือ ใช้เวลา และหน่วยความจำที่เหมาะสม

1.1.5 ลักษณะของอัลกอริทึม

อัลกอริทึมเขียนได้หลายรูปแบบดังต่อไปนี้

- 1) แบบบรรยาย (Narrative description)
- 2) แบบขั้นตอน (Step by step)
- 3) แบบผังงาน (Flowchart)
- 4) แบบรหัสเทียม (Pseudo-code)
- 5) แบบโปรแกรม (Program)

การเขียนอัลกอริทึมแบบบรรยายและแบบขั้นตอนเหมาะสำหรับใช้กับการทำงานทั่วไป แบบที่นิยมใช้กับคอมพิวเตอร์คือ ผังงาน และรหัสเทียม เนื่องจากดูง่ายกว่า และไม่ขึ้นอยู่กับภาษาคอมพิวเตอร์ ส่วนแบบโปรแกรมขึ้นอยู่กับภาษาที่ใช้

ตัวอย่างที่ 1 การเขียนอัลกอริทึมแบบบรรยาย และแบบขั้นตอน

ปัญหา ต้องการหาคะแนนเฉลี่ยของการสอบ 7 วิชา ถ้าคะแนนเฉลี่ยต่ำกว่า 50 สอบไม่ผ่าน แต่ถ้าคะแนนเฉลี่ยตั้งแต่ 50 ขึ้นไป สอบผ่าน

การวิเคราะห์ปัญหา

จากปัญหา สามารถวิเคราะห์ปัญหาได้โดยแบ่งการทำงานออกเป็น 3 ส่วนตามลักษณะการทำงานของคอมพิวเตอร์ คือ ส่วนนำเข้าข้อมูล (Data input) ส่วนประมวลผล (Processing) และส่วนแสดงผลข้อมูล (Data output) โดยกำหนดรายละเอียดของข้อมูลเข้า วิธีการประมวลผล และการแสดงผล ดังต่อไปนี้

ข้อมูลเข้า : คะแนนสอบ 7 วิชา

การประมวลผล : การหาค่าเฉลี่ย คือการรวมคะแนนสอบทั้ง 7 วิชา แล้วหารด้วยจำนวนวิชา

การแสดงผล : แสดงตามเงื่อนไข ดังนี้ ถ้าค่าคะแนนเฉลี่ยต่ำกว่า 50 แสดงข้อความ “สอบไม่ผ่าน” แต่ถ้าค่าคะแนนเฉลี่ย มากกว่า หรือเท่ากับ 50 แสดงข้อความ “สอบผ่าน”

การเขียนอัลกอริทึม

เมื่อทำการวิเคราะห์ปัญหาเสร็จแล้ว เขียนเป็นอัลกอริทึมแบบบรรยาย และแบบขั้นตอนได้ดังนี้

อัลกอริทึมแบบบรรยาย ใส่คะแนนสอบทั้ง 7 วิชา คำนวณค่าคะแนนเฉลี่ยโดยการบวกคะแนนทั้ง 7 วิชา แล้วหารด้วย 7 ถ้าค่าคะแนนเฉลี่ยต่ำกว่า 50 แสดงข้อความ “สอบไม่ผ่าน” แต่ถ้าค่าคะแนนเฉลี่ยมากกว่า หรือเท่ากับ 50 แสดงข้อความ “สอบผ่าน”

อัลกอริทึมแบบขั้นตอน

ขั้นตอนที่ 1 ใส่คะแนนสอบทั้ง 7 วิชา

ขั้นตอนที่ 2 บวกคะแนนทั้ง 7 วิชา

ขั้นตอนที่ 3 คำนวณค่าคะแนนเฉลี่ย โดยหารด้วย 7

ขั้นตอนที่ 4 ถ้าค่าคะแนนเฉลี่ยต่ำกว่า 50 แสดงข้อความ “สอบไม่ผ่าน”

ขั้นตอนที่ 5 ถ้าค่าคะแนนเฉลี่ยมากกว่า หรือเท่ากับ 50 แสดงข้อความ “สอบผ่าน”

สำหรับตัวอย่างของอัลกอริทึมแบบผังงาน และแบบรหัสเทียม โปรดดูในหัวข้อที่เกี่ยวข้องต่อไป

1.1.6 ประโยชน์ของอัลกอริทึม

ประโยชน์ของอัลกอริทึม มีดังนี้

- 1) ทำให้การทำงาน หรือการแก้ปัญหาชัดเจนขึ้น เพราะทุกอย่างจะถูกแยกแยะกิจกรรมจัดเรียงเป็นขั้นตอน มีวิธีการ และทางเลือกไว้
- 2) ช่วยในการแก้ปัญหา
- 3) ทำให้การทำงานสำเร็จอย่างรวดเร็ว
- 4) อัลกอริทึมที่ออกแบบไว้ดีแล้ว สามารถนำไปใช้ได้อีก หรืออาจนำไปใช้กับงานอื่นได้

1.1.7 ตัวอย่างของการใช้อัลกอริทึมในปัจจุบัน

การควบคุมไฟจราจร

การแปลงเลขฐานสิบ เป็นเลขฐานสองในคอมพิวเตอร์

การเข้ารหัส และการถอดรหัสข้อมูล

การหาเส้นทางที่ดีที่สุด

การเรียงลำดับข้อมูล

การติดต่อสื่อสาร

การบีบอัดข้อมูล

ฯลฯ

กิจกรรม 1-1 เขียนอัลกอริทึมทางกลับบ้านหลังเลิกเรียน

ในการเดินทางกลับบ้านของนักเรียนหลังเลิกเรียน ต้องทำอะไรบ้างจึงจะถึงบ้าน ให้เขียนเป็นอัลกอริทึมแบบขั้นตอนตามที่สามารถทำได้จริง

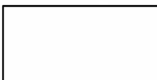
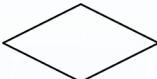
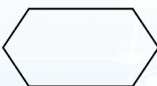
1.2 การออกแบบอัลกอริทึมโดยใช้ผังงาน (Flowchart)

การเขียนผังงานเป็นการออกแบบอัลกอริทึมที่ใช้รูปภาพสัญลักษณ์แทนขั้นตอนการทำงานต่างๆ ในแต่ละขั้น และใช้เส้นหัวลูกศรเชื่อมต่อ เพื่อแสดงเส้นทางการทำงาน

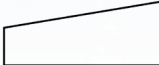
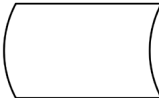
1.2.1 สัญลักษณ์ที่ใช้เขียนผังงาน และความหมาย

สัญลักษณ์ที่ใช้ในการเขียนผังงาน ตามมาตรฐาน ISO (International Organization for Standardization) บางส่วนที่ควรทราบ มีดังตารางที่ 1.1 และตารางที่ 1.2

ตารางที่ 1.1 สัญลักษณ์ที่ใช้ในการเขียนผังงาน (บางส่วน)

สัญลักษณ์	ความหมาย
	จุดเริ่มต้น (Start) หรือจุดสิ้นสุด (End)
	เส้นทาง (Flow direction)
	การประมวลผล (Process)
	การทำงานตามกระบวนการคำสั่ง หรือฟังก์ชันที่กำหนดไว้ (Predefined process)
	การตัดสินใจ (Decision)
	การเตรียมการ (Preparation) หรือการกำหนดค่าเริ่มต้น (Initialization)
	จุดต่อในหน้าเดียวกัน (On-page connector)
	จุดต่อระหว่างหน้า (Off-page connector)

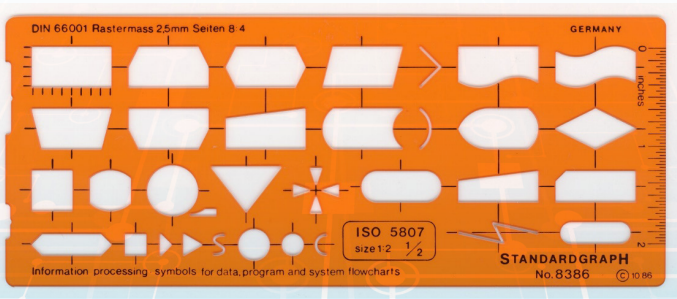
ตารางที่ 1.2 สัญลักษณ์ที่เกี่ยวข้องกับข้อมูลเข้า/ออก

สัญลักษณ์	ความหมาย
	ข้อมูลเข้า/ข้อมูลออก (Data input/Data output)
	การป้อนข้อมูล (Manual input)
	การแสดงผลที่จอภาพ (Display)
	การพิมพ์เอกสาร (Document)
	ข้อมูลที่บันทึกไว้ (Stored data)
	ฐานข้อมูล (Database)

1.2.2 เครื่องมือที่ใช้ในการเขียนผังงาน

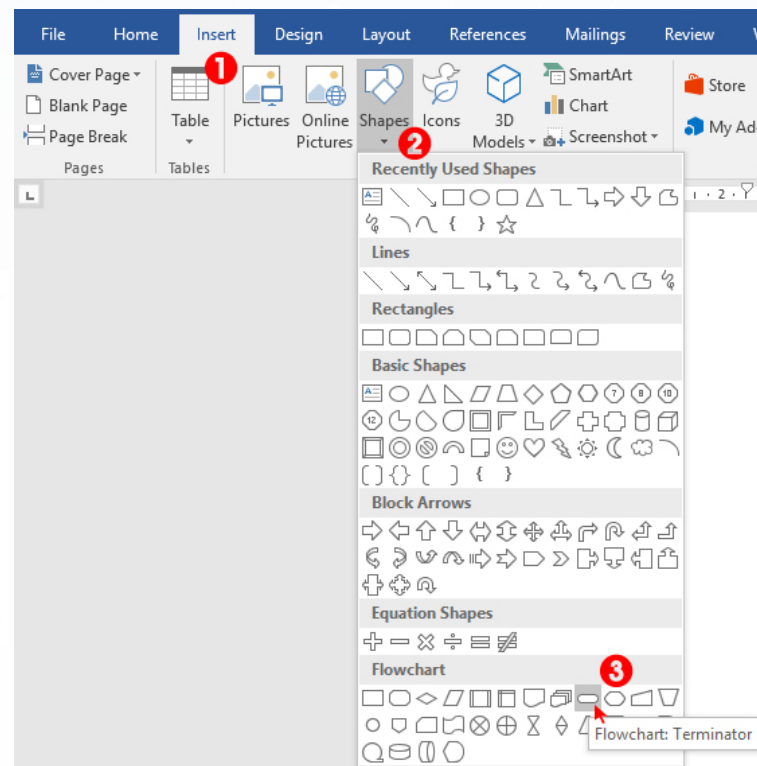
การเขียนผังงานอาจเขียนด้วยมือหรือใช้เครื่องมือช่วยดังต่อไปนี้

- 1) บรรทัดแบบสำหรับเขียนผังงาน (Flowchart template ruler) การเขียนผังงานด้วยมือใช้สำหรับการร่างแบบ หรือออกแบบตามวิธีคิดแบบคร่าวๆ อาจได้ผังงานที่ไม่สวย แต่ถ้าใช้บรรทัดแบบสำหรับเขียนผังงานดังภาพที่ 1.6 ช่วยก็จะทำให้เขียนได้เร็ว และสวยงามขึ้น



ภาพที่ 1.6 บรรทัดแบบสำหรับเขียนผังงาน

2) ซอฟต์แวร์ หรือโปรแกรมสำหรับเขียนผังงาน (Flowchart software) ในปัจจุบันมีโปรแกรมสำหรับเขียนผังงานจำนวนมาก ทั้งแบบไม่เสียค่าใช้จ่าย เช่น draw.io ซึ่งเป็นแบบออนไลน์และแบบที่ต้องซื้อมาติดตั้งสำหรับมืออาชีพ เช่น ไมโครซอฟต์วิซิโอ (Microsoft visio) หรือถ้ามีโปรแกรมไมโครซอฟต์เวิร์ดหรือเอ็กเซลอยู่แล้ว ก็สามารถเขียนผังงานได้เช่นกัน โดยเลือกรูปร่าง (Shapes) ตามขั้นตอนดังภาพที่ 1.7



ภาพที่ 1.7 การเลือกรูปร่างสำหรับเขียนผังงาน

1.2.3 หลักการเขียนผังงานที่ดี

หลักการเขียนผังงานที่ดี ควรทำดังต่อไปนี้

- 1) ใช้สัญลักษณ์ตรงตามที่กำหนดไว้ มีขนาด และระยะห่างที่เหมาะสม
- 2) มีจุดเริ่มต้น (Start) และจุดสิ้นสุด (End) เพียงจุดเดียว
- 3) ใช้เส้นหัวลูกศรแสดงทิศทางการไหลของข้อมูล หรือลำดับการทำงาน จากบนลงล่าง และซ้ายไปขวา ยกเว้นการย้อนกลับหรือวนซ้ำ
- 4) เส้นลูกศรต้องต่อเชื่อมกับสัญลักษณ์ โดยต่อเข้าทางเดียว และออกทางเดียวตรงกลางสัญลักษณ์ ยกเว้นสัญลักษณ์การตัดสินใจที่เข้าทางเดียว แต่ออกได้หลายทาง (ดูตัวอย่างในภาพที่ 1.9)

5) ใช้คำอธิบายการทำงานในแต่ละขั้นตอนด้วยข้อความที่สั้น กระชับ ชัดเจน และเข้าใจได้ง่าย ถ้าต้องการเขียนหมายเหตุเพิ่มเติมให้ใช้สัญลักษณ์ของหมายเหตุ ดังตัวอย่างในภาพที่ 1.8

6) ใช้สัญลักษณ์จุดต่อในหน้าเดียวกัน แทนการโยงเส้นยาวๆ หรือเส้นที่ตัดกัน

7) ถ้าไม่สามารถเขียนในหน้าเดียวได้ ให้ใช้สัญลักษณ์จุดต่อระหว่างหน้า

8) ทดสอบความถูกต้องของการทำงานก่อน จึงนำไปใช้เขียนโปรแกรมต่อไป

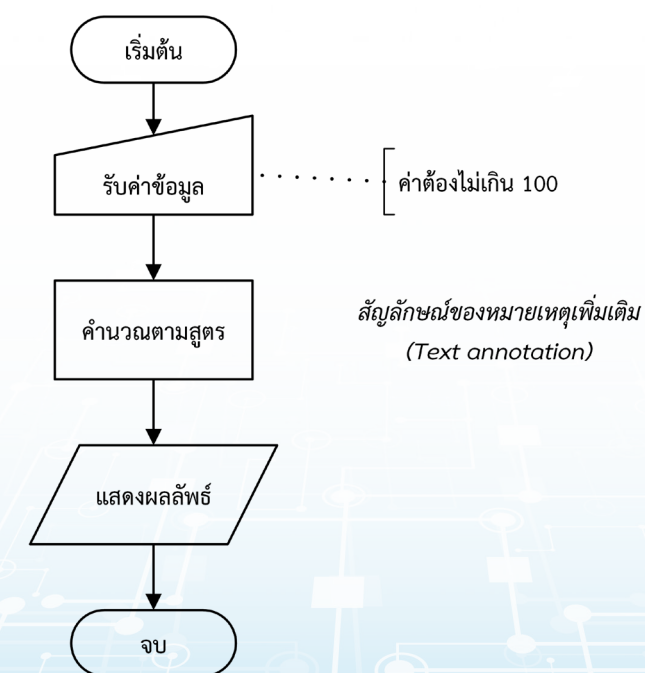
9) เขียนให้ครบทุกฟังก์ชันที่ใช้ และกำหนดขั้นตอนและรายละเอียดให้ชัดเจน เพื่อที่จะได้ตรวจสอบการทำงานของโปรแกรมได้ถูกต้อง เช่น ค่าที่รับเข้าต้องมีหน่วยเป็นอะไร สูตรที่ใช้ในการคำนวณเป็นอย่างไร ผลลัพธ์ที่ได้มีอะไรบ้าง และจะแสดงที่จอ หรือพิมพ์เป็นเอกสาร

10) กำหนดชื่อผังงาน เพื่อให้ทราบว่าเป็นอัลกอริทึมสำหรับทำอะไร

1.2.4 โครงสร้างของผังงาน

ในการออกแบบอัลกอริทึมโดยใช้ผังงาน สามารถเขียนโดยใช้โครงสร้างหลัก 3 แบบ ซึ่งอาจเขียนแบบใดแบบหนึ่ง หรือผสมกันหลายแบบได้ ดังต่อไปนี้

1) โครงสร้างแบบตามลำดับ (Sequence) เป็นการแสดงขั้นตอนการทำงานตามลำดับ ดังตัวอย่างดังภาพที่ 1.8



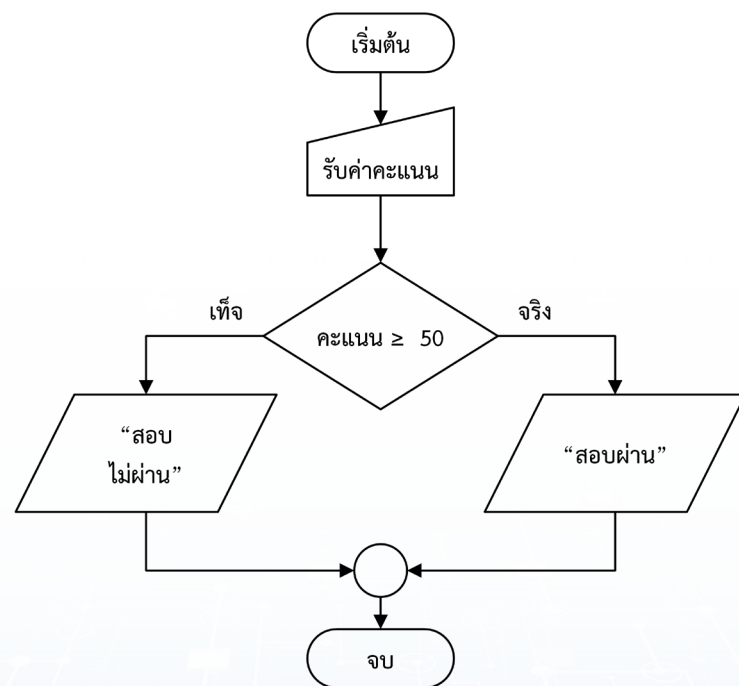
ภาพที่ 1.8 โครงสร้างแบบตามลำดับ

ผังงานตัวอย่างตามภาพที่ 1.8 เป็นการทำงานแบบตามลำดับสำหรับงานการคำนวณทั่วไป โดยเริ่มต้น (ใช้สัญลักษณ์เริ่มต้น หรือ Start) จากรับค่าข้อมูล (ใช้สัญลักษณ์การป้อนข้อมูล) โดยค่าข้อมูลเป็นตัวเลขไม่เกิน 100 ลำดับต่อไปคือ ทำการคำนวณตามสูตรที่กำหนด (ใช้สัญลักษณ์การประมวลผล) ลำดับสุดท้ายคือ แสดงผลลัพธ์ที่ได้ (ใช้สัญลักษณ์ข้อมูลออก) แล้วสิ้นสุดการทำงานที่ จบ หรือ End (ใช้สัญลักษณ์สิ้นสุด หรือ End)

หมายเหตุ

ในการเขียนผังงานสำหรับการทำงานต่างๆ ไป เช่น การกำหนดค่า ให้ใช้สัญลักษณ์การประมวลผล

2) โครงสร้างแบบทางเลือก (Selection) เป็นการแสดงขั้นตอนการทำงานตามเงื่อนไขให้เลือกทำ ซึ่งจะมีการทดสอบเงื่อนไขเพื่อแยกแยะการเลือกทำงานให้เป็นไปตามกรณีที่ต้องการ ถ้าเงื่อนไขเป็นจริงจะให้ทำอะไร และถ้าเงื่อนไขเป็นเท็จจะให้ทำอะไร ตัวอย่างดังภาพที่ 1.9



ภาพที่ 1.9 โครงสร้างแบบทางเลือกตามเงื่อนไขกรณี

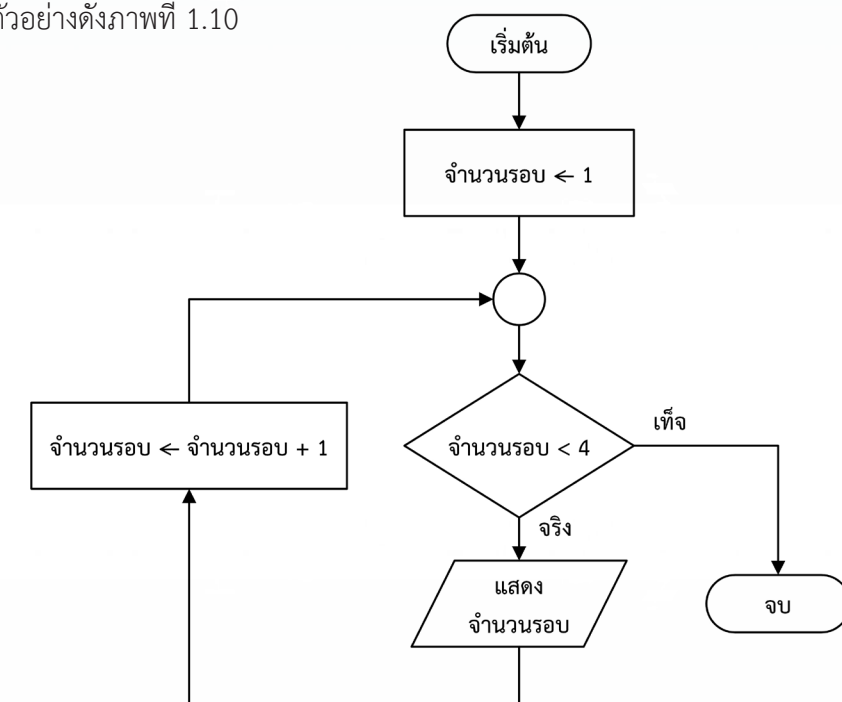
ผังงานตัวอย่างตามภาพที่ 1.9 แสดงการทำงานแบบทางเลือก (ใช้สัญลักษณ์การตัดสินใจ หรือ Decision) ต่อจากการรับค่าคะแนน โดยเปรียบเทียบคะแนนว่า มากกว่า หรือ เท่ากับ 50 จริงหรือไม่ ถ้าเป็นจริง ให้แสดงข้อความ “สอบผ่าน” แล้วจบการทำงาน แต่ถ้าเป็นเท็จ ให้แสดงข้อความ “สอบไม่ผ่าน” แล้วจบการทำงาน

หมายเหตุ

ในการเขียนเงื่อนไข เครื่องหมาย $\geq$ หมายถึง “มากกว่าหรือเท่ากับ”

ในการเขียนทางเลือก คำว่า “จริง” อาจเขียนด้วย True หรือ T หรืออาจใช้ Yes หรือ Y ก็ได้ ส่วนคำว่า “เท็จ” อาจเขียนด้วย False หรือ F หรืออาจใช้ No หรือ N ก็ได้

3) โครงสร้างแบบวนซ้ำ (Iteration) เป็นการแสดงขั้นตอนการทำงานแบบวนซ้ำ ซึ่งจะมีการทดสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริงให้วนซ้ำการทำงานอีก ถ้าเป็นเท็จให้หยุดการวนซ้ำ ตัวอย่างดังภาพที่ 1.10



ภาพที่ 1.10 โครงสร้างแบบวนซ้ำ

1.2.5 ประโยชน์ของผังงาน

การเขียนผังงานมีประโยชน์ดังนี้

- 1) ช่วยจัดลำดับขั้นตอนการทำงานของโปรแกรม และสามารถนำไปเขียนโปรแกรมได้ง่ายขึ้น
- 2) ช่วยในการตรวจสอบ และแก้ไขโปรแกรมได้ง่าย เมื่อเกิดข้อผิดพลาด
- 3) ช่วยให้การดัดแปลง แก้ไข ทำได้อย่างสะดวกและรวดเร็ว
- 4) ช่วยให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมได้อย่างสะดวก และรวดเร็ว

มากขึ้น

กิจกรรม 1-2 การเขียนผังงานการเดินทางกลับบ้าน

จากอัลกอริทึมแบบขั้นตอนของการเดินทางกลับบ้านของกิจกรรม 1-1 ให้นักเรียนเขียนผังงานของการเดินทางกลับบ้าน โดยเลือกใช้สัญลักษณ์การทำงานให้ถูกต้อง

1.3 การออกแบบอัลกอริทึมโดยใช้รหัสเทียม (Pseudo-code)

การเขียนรหัสเทียม เป็นการเขียนขั้นตอนการทำงานโดยใช้รหัสเทียม หรือคำสั่งเทียมที่ไม่ใช่คำสั่งของภาษาคอมพิวเตอร์ แทนการทำงานในแต่ละขั้น (โดยไม่ต้องใช้สัญลักษณ์เหมือนผังงาน) โดยทั่วไปเขียนเป็นภาษาอังกฤษ อาจใช้ภาษาไทยก็ได้ขึ้นอยู่กับความสะดวกของผู้เขียน และกิจกรรมที่จะทำ

1.3.1 คำสั่งที่ใช้ในการเขียนรหัสเทียม

คำสั่งที่ใช้จะเป็นคำสั่งพื้นฐานในการทำงานของคอมพิวเตอร์ เขียนด้วยตัวพิมพ์ใหญ่ ดังตัวอย่างในตารางที่ 1.3

ตารางที่ 1.3 คำสั่งที่ใช้ในการเขียนรหัสเทียม

การทำงาน	คำสั่ง	ตัวอย่าง
การกำหนดค่าให้กับตัวแปร	SET	SET count TO 0 หรือ count , 0
การนำข้อมูลเข้า (Input)	GET INPUT	GET user_name INPUT side
การนำข้อมูลออก (Output)	DISPLAY PRINT	DISPLAY counter PRINT area
การคำนวณ (Compute)	COMPUTE CALCULATE	COMPUTE area = width * length หรือ area , width * length
การเพิ่มค่า (Add one)	INCREMENT	INCREMENT counter
การทำงานตามเงื่อนไข (Selection)	IF ... THEN ... ELSE ... ENDIF	IF score >50 THEN PRINT "Pass" ELSE PRINT "Fail" ENDIF

หมายเหตุ

ตัวแปร ใช้สำหรับเก็บค่าข้อมูลที่ใช้ในการประมวลผล สำหรับรายละเอียดให้ดูในหัวข้อที่เกี่ยวข้องต่อไปในบทที่ 2

ตารางที่ 1.3 คำสั่งที่ใช้ในการเขียนรหัสเทียม (ต่อ)

การทำงาน	คำสั่ง	ตัวอย่าง
การวนซ้ำ (Iteration)	WHILE ... ENDWHILE	a , 1 WHILE a < 4 DISPLAY a a , a + 1 ENDWHILE

หมายเหตุ

ตัวแปร หมายถึงที่เก็บค่าข้อมูล โดยใช้ชื่อแทน เช่น ตัวแปรชื่อ count ในการคำนวณ ใช้เครื่องหมายทางคณิตศาสตร์ดังตารางที่ 1.4

ตารางที่ 1.4 เครื่องหมายทางคณิตศาสตร์ (Arithmetic operators)

การคำนวณ	เครื่องหมาย	ตัวอย่าง
การบวก	+	a , b + c
การลบ	-	z , x - y
การคูณ	*	area , x * y
การหาร	/	average , sum / n

1.3.2 หลักการเขียนรหัสเทียมที่ดี

หลักการเขียนรหัสเทียมที่ดี ควรทำดังต่อไปนี้

1.3.2.1 กำหนดชื่ออัลกอริทึม เพื่อให้ทราบว่าเป็นอัลกอริทึมสำหรับทำอะไร

1.3.2.2 เริ่มจากคำ BEGIN, START หรือ เริ่มต้น

1.3.2.3 สิ้นสุดด้วยคำ END, STOP หรือ จบ หรือหยุด

1.3.2.4 คำสั่งต่างๆ อยู่ระหว่าง BEGIN และ END โดยจัดให้เยื้องกันไป

1.3.2.5 ใช้คำสั่งเดียวในแต่ละบรรทัด

1.3.2.6 คำสั่งหลักใช้ตัวพิมพ์ใหญ่

1.3.2.7 ใช้ Comment หรือหมายเหตุเพื่อให้เข้าใจง่ายขึ้น โดยใช้เครื่องหมาย # นำหน้าตามด้วยข้อความอะไรก็ได้เพื่อเป็นหมายเหตุ ไม่ใช่คำสั่งให้ทำงาน โดยอาจเขียนต่อท้ายบรรทัดคำสั่ง หรือคั่นระหว่างแต่ละคำสั่ง หรือกลุ่มคำสั่งก็ได้

1.3.3 โครงสร้างของรหัสเทียม

ในการออกแบบอัลกอริทึมโดยใช้รหัสเทียม สามารถเขียนโดยใช้โครงสร้างหลัก 3 แบบ ซึ่งอาจเขียนแบบใดแบบหนึ่ง หรือผสมกันหลายแบบได้ ดังต่อไปนี้

1) โครงสร้างแบบตามลำดับ เป็นการแสดงขั้นตอนการทำงานตามลำดับ ดังตัวอย่าง

```
Algorithm_1: Area_of_Rectangle
BEGIN
    GET R_Width
    GET R_Length
    # Area Calculation
    R_Area , R_Width * R_Length
    DISPLAY R_Area
END
```

จากตัวอย่าง เป็นอัลกอริทึมของการหาพื้นที่สี่เหลี่ยมผืนผ้า (Rectangle) โดยทำงานตามลำดับดังต่อไปนี้ (เขียนอัลกอริทึมเป็นภาษาไทย)

```
เริ่มต้น
รับค่า ความกว้าง
รับค่า ความยาว
# คำนวณหาพื้นที่
พื้นที่ , กว้าง * ยาว
แสดง ค่าพื้นที่
จบ
```

จากตัวอย่างดูเหมือนจะเขียนง่าย แต่การที่จะเขียนอัลกอริทึมนี้ได้จะต้องวิเคราะห์ก่อนว่า ในการหาพื้นที่สี่เหลี่ยมผืนผ้าหาได้อย่างไร ใช้สูตรอะไร ใช้ตัวแปรอะไรบ้าง จะต้องกำหนดค่าหรือป้อนค่าอะไรบ้าง และจะแสดงค่าของอะไรบ้าง ที่สำคัญที่สุดคือ การเขียนลำดับการทำงานให้ถูกต้อง เช่น ถ้าแสดงค่าของพื้นที่ก่อนก็ไม่ถูกต้อง หรือคำนวณตามสูตรก่อน แต่ยังไม่ได้อ่านค่า เป็นต้น

2) โครงสร้างแบบทางเลือก เป็นการแสดงขั้นตอนการทำงานตามเงื่อนไขให้เลือกทำ ซึ่งจะมีการทดสอบเงื่อนไขเพื่อแยกแยะการเลือกทำงานให้เป็นไปตามกรณีที่ต้องการ ถ้าเงื่อนไขเป็นจริงจะให้ทำอะไร และถ้าเงื่อนไขเป็นเท็จจะให้ทำอะไร ดังตัวอย่าง

```
Algorithm_2: Check_OT
START
    NormalMax , 10
    GET HoursWorked

    # Decision Making

    IF HoursWorked > NormalMax THEN
        DISPLAY "Overtime"
    ELSE
        DISPLAY "Regular time"
    ENDIF
STOP
```

จากตัวอย่าง เป็นอัลกอริทึมของการตรวจสอบชั่วโมงการทำงานว่ามีการทำงานเกินเวลาหรือไม่ โดยกำหนดให้อัตราชั่วโมงการทำงานปกติ (NormalMax) ไม่เกิน 10 ชั่วโมง ลำดับต่อไปคือ รับค่าชั่วโมงที่ทำงานจริง (HoursWorked) จากนั้นทำการทดสอบเงื่อนไข ถ้าชั่วโมงที่ทำงานจริง มากกว่า อัตราชั่วโมงการทำงานปกติ ให้แสดงข้อความ “เวลาเกิน” แล้วจบการตรวจสอบ แต่ถ้าไม่ใช่ ให้แสดงข้อความ “เวลาปกติ” แล้วจบการตรวจสอบ

3) โครงสร้างแบบวนซ้ำ เป็นการแสดงขั้นตอนการทำงานแบบวนซ้ำ ซึ่งจะมีการทดสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริงให้วนซ้ำการทำงานอีก ถ้าเป็นเท็จให้หยุดการวนซ้ำ ดังตัวอย่าง

```
Algorithm_3: Iteration
START
    a , 1
    WHILE a < 4
        DISPLAY a
        a , a + 1
    ENDWHILE
STOP
```

จากตัวอย่าง เป็นอัลกอริทึมของการวนซ้ำด้วยคำสั่ง WHILE ... ENDWHILE เพื่อแสดงค่าของตัวแปร a ได้ผลลัพธ์ออกมาเป็น 1, 2 และ 3 ตามลำดับ เหมือนกับการทำงานของผังงานตามภาพที่ 1.10

จากตัวอย่างเห็นได้ว่า การเขียนรหัสเทียมแสดงให้เห็นขั้นตอนการทำงานได้โดยง่าย คล้ายกับโปรแกรมคอมพิวเตอร์ ดังนั้นจึงสามารถนำอัลกอริทึมไปพัฒนาโปรแกรมต่อไปได้

กิจกรรม 1-3 การเขียนรหัสเทียมของการหาพื้นที่สามเหลี่ยม

ให้นักเรียนเขียนรหัสเทียมของการหาพื้นที่ของสามเหลี่ยม (Triangle)

1.4 การออกแบบอัลกอริทึมเพื่อแก้ปัญหาทางคณิตศาสตร์ วิทยาศาสตร์อย่างง่าย

ในการออกแบบอัลกอริทึมเพื่อแก้ปัญหาทางคณิตศาสตร์ วิทยาศาสตร์ ส่วนใหญ่จะเป็นการคำนวณค่าต่างๆ ตามสูตร สิ่งแรกที่จะต้องทำก่อนคือ การวิเคราะห์ปัญหา หรือวิเคราะห์งานที่จะทำ

1.4.1 หลักเกณฑ์ในการวิเคราะห์งาน

การวิเคราะห์งานสำหรับการออกแบบอัลกอริทึมที่ใช้ทำโปรแกรมเพื่อให้ได้ผลดี ควรพิจารณาดังต่อไปนี้

- 1) **สิ่งที่ต้องการ** เช่น ต้องการให้คำนวณหรือประมวลผลหาอะไร หรือต้องการให้แสดงข้อมูลอะไรบ้าง
- 2) **ข้อมูลที่ต้องนำเข้า** หรือข้อมูลที่ใช้ในการประมวลผล มีอะไรบ้าง และควรจะมีลักษณะ หรือรูปแบบอย่างไร เช่น เป็นเลขจำนวนเต็ม
- 3) **ตัวแปรที่ใช้ในโปรแกรม** ใช้ชื่อตัวแปรและแบบข้อมูลที่เหมาะสม สำหรับเก็บข้อมูลในการประมวลผล โดยเป็นไปตามหลักของภาษาที่ใช้เขียนโปรแกรม
- 4) **รูปแบบผลลัพธ์ที่ต้องการ** ควรจะมีลักษณะอย่างไร มีรายละเอียดมากน้อยเพียงใด งานบางอย่างอาจมีการกำหนดลักษณะผลลัพธ์เป็นรายงานมาให้อย่างชัดเจน แต่ถ้าไม่ได้กำหนดให้ ผู้เขียนโปรแกรมก็ต้องพิจารณากำหนดรูปแบบผลลัพธ์อย่างละเอียดรอบคอบ และเหมาะสมกับงานที่ต้องการ
- 5) **ขั้นตอนหรือวิธีการประมวลผล** ต้องการประมวลผลอย่างไรบ้าง เพื่อให้การทำงานตามลำดับ หรือตามเงื่อนไขอย่างถูกต้องตามความต้องการ ซึ่งก็คือ การออกแบบอัลกอริทึม

ตัวอย่าง สนามหน้าบ้านหลังหนึ่ง กว้าง 4 เมตร ยาว 8 เมตร ต้องการปูหญ้าในสนาม โดยหญ้าหนึ่งผืนมีความกว้าง 50 เซนติเมตร ยาว 100 เซนติเมตร ดังภาพที่ 1.11 จะใช้หญ้าทั้งหมดกี่ผืน



ภาพที่ 1.11 แผ่นหญ้าที่ใช้ในการปูสนาม

การวิเคราะห์งาน

- 1) **สิ่งที่ต้องการ**
โปรแกรมคำนวณหาจำนวนแผ่นหญ้า
- 2) **ข้อมูลที่ต้องนำเข้า**
 - 2.1 ข้อมูลเกี่ยวกับขนาดของสนามที่จะปูหญ้า ความกว้าง ความยาว
 - 2.2 ข้อมูลเกี่ยวกับแผ่นหญ้า ความกว้าง ความยาว
- 3) **ตัวแปรที่ใช้** กำหนดชื่อตัวแปรที่ใช้เก็บค่าต่างๆ และแบบของข้อมูลได้ดังตารางที่ 1.5 โดยใช้ตัวย่อนำหน้าดังนี้
 - F หมายถึง สนาม (Field)
 - G หมายถึง หญ้า (Grass)
 - N หมายถึง จำนวน (Number)

ตารางที่ 1.5 ชื่อตัวแปรที่ใช้เก็บค่าต่างๆ และแบบของข้อมูล

ชื่อตัวแปร	ใช้เก็บค่า	แบบข้อมูล
F_area	พื้นที่ของสนาม	ตัวเลข
F_width	ความกว้างของสนาม	ตัวเลข
F_length	ความยาวของสนาม	ตัวเลข
G_area	พื้นที่ของแผ่นหญ้า	ตัวเลข
G_width	ความกว้างของหญ้า	ตัวเลข
G_length	ความยาวของหญ้า	ตัวเลข
N_grass	จำนวนแผ่นหญ้า	ตัวเลข

4) รูปแบบผลลัพธ์ที่ต้องการ

ต้องการให้โปรแกรมแสดงผลอะไรออกมาบ้าง เช่น จำนวนแผ่นหญ้า หรือข้อมูลทั้งหมด

5) ขั้นตอน หรือวิธีการประมวลผล

จากปัญหา สามารถแยกแยะและเขียนเป็นตาราง ตามลักษณะการประมวลผลของคอมพิวเตอร์ คือ ข้อมูลเข้าหรือข้อมูลที่ใช้ การประมวลผล และข้อมูลออกหรือผลลัพธ์ ตามลำดับการทำงานได้ดังตารางที่ 1.6

ตารางที่ 1.6 ขั้นตอนการประมวลผล

ข้อมูลเข้า/ข้อมูลที่ใช้	การประมวลผล	ข้อมูลออก
ความกว้างของสนาม ความยาวของสนาม	คำนวณหาพื้นที่ของสนาม	พื้นที่ของสนาม
ความกว้างของแผ่นหญ้า ความยาวของแผ่นหญ้า	คำนวณหาพื้นที่ของแผ่นหญ้า	พื้นที่ของแผ่นหญ้า
พื้นที่ของสนาม พื้นที่ของแผ่นหญ้า	คำนวณหาจำนวนแผ่นหญ้า	จำนวนแผ่นหญ้า

จากตารางที่ 1.6 อาจทำการออกแบบอัลกอริทึมรวมทั้งหมด หรือแยกเป็น 3 อัลกอริทึม คือ อัลกอริทึมของการหาพื้นที่ของสนาม อัลกอริทึมของการหาพื้นที่ของแผ่นหญ้า และอัลกอริทึมของการหาจำนวนแผ่นหญ้า

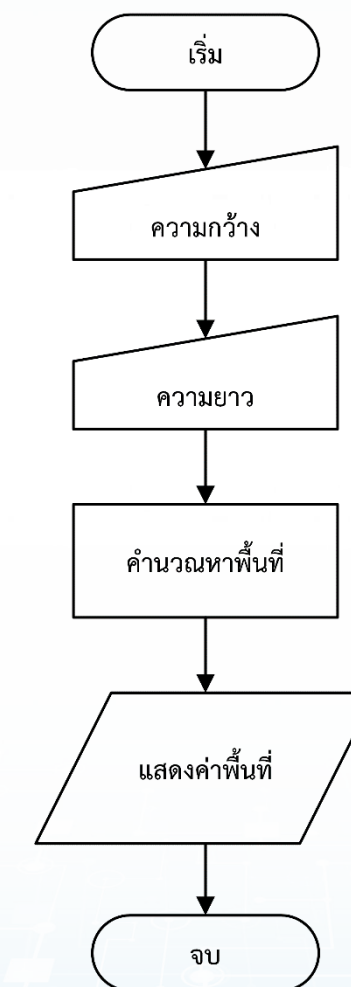
5.1) อัลกอริทึมของการหาพื้นที่ของสนาม

การคำนวณหาพื้นที่ ถ้าเป็นพื้นที่รูปสี่เหลี่ยมผืนผ้า ใช้สูตร $\text{พื้นที่} = \text{กว้าง} \times \text{ยาว}$ เมื่อใช้ตัวแปรที่กำหนดไว้ สามารถเขียนอัลกอริทึมส่วนการคำนวณได้ดังนี้

$F_area, F_width * F_length$

ถ้าพื้นที่เป็นรูปอื่นก็ต้องใช้สูตรต่างกันไป

สามารถกำหนดขั้นตอนออกมาเป็นอัลกอริทึมแบบผังงานได้ดังภาพที่ 1.12



ภาพที่ 1.12 อัลกอริทึมของการคำนวณหาพื้นที่

จากอัลกอริทึมแบบผังงานตามภาพที่ 1.12 สามารถเขียนเป็นอัลกอริทึมแบบรหัสเทียมได้ดังนี้

```

Algorithm_A1: Area_of_Field
BEGIN
  GET F_Width
  GET F_Length
  # Area Calculation
  F_Area , F_Width * F_Length
  DISPLAY "Field Area = ", F_Area
END

```

หมายเหตุ

ข้อความที่จะแสดงอยู่ในอัญประกาศ (เครื่องหมายคำพูด) และใช้จุลภาค (Comma) คั่นระหว่างข้อความกับตัวแปร

5.2) อัลกอริทึมของการหาพื้นที่แผ่นหญ้า

การคำนวณหาพื้นที่ เนื่องจากแผ่นหญ้าเป็นรูปสี่เหลี่ยมผืนผ้า

ใช้สูตร พื้นที่ = กว้าง x ยาว

สามารถกำหนดขั้นตอนออกมาเป็นอัลกอริทึมแบบผังงานได้เช่นเดียวกันกับภาพที่ 1.12

แต่ใช้ตัวแปรและค่าข้อมูลต่างกันเมื่อเขียนโปรแกรม คือ

เมื่อใช้ตัวแปรที่กำหนดไว้ จะได้

$G_area, G_width * G_length$

5.3) อัลกอริทึมของการหาจำนวนแผ่นหญ้า

การคำนวณหาจำนวนแผ่นหญ้า หาได้จาก

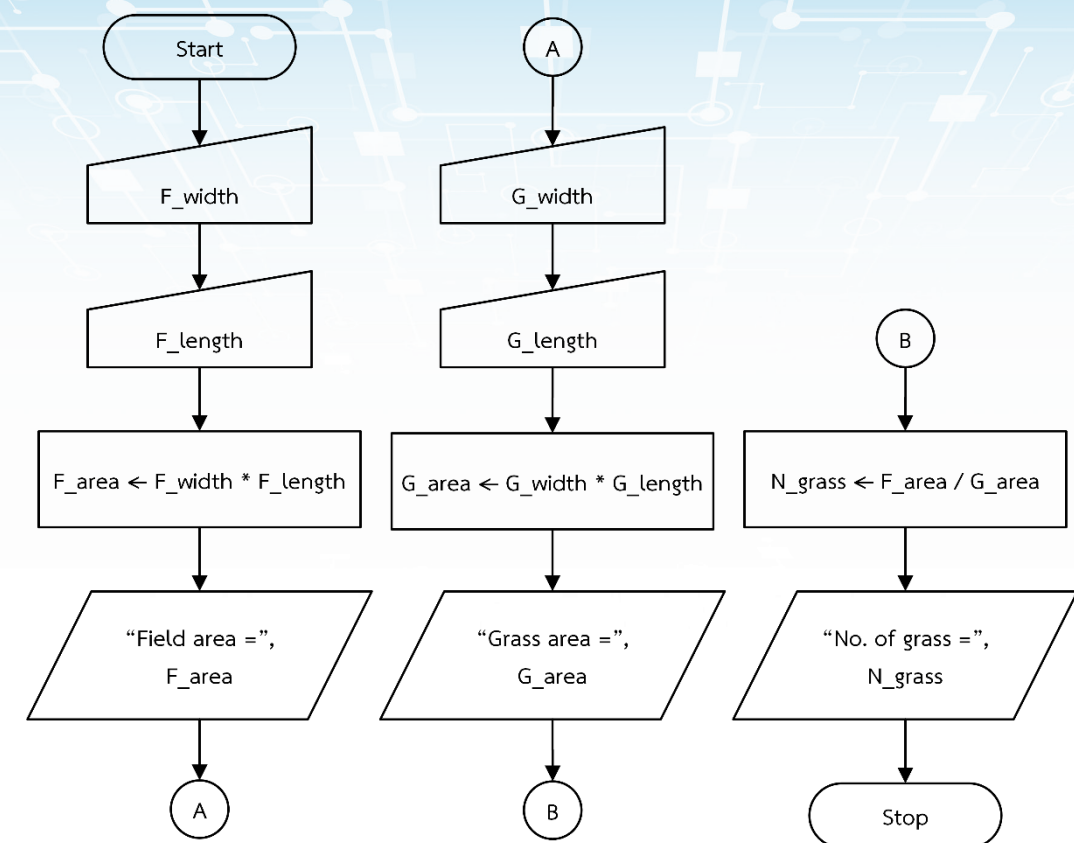
จำนวนแผ่นหญ้า = พื้นที่สนาม / พื้นที่แผ่นหญ้า

เมื่อใช้ตัวแปรที่กำหนดไว้ จะได้

$N_grass, F_area / G_area$

เมื่อได้อัลกอริทึมย่อยครบแล้ว สามารถนำมาจัดรวมกันแล้วออกแบบให้เป็นอัลกอริทึม

ของโปรแกรมคำนวณหาจำนวนแผ่นหญ้าได้ ดังภาพที่ 1.13



ภาพที่ 1.13 อัลกอริทึมของโปรแกรมคำนวณหาจำนวนแผ่นหญ้า

เมื่อออกแบบอัลกอริทึมของโปรแกรมเสร็จแล้ว ทำการตรวจสอบว่าถูกต้อง ตรงตามลำดับขั้นตอน และมีขั้นตอนครบตามที่ต้องการหรือไม่ จากนั้นก็นำไปเขียนโปรแกรมต่อไป

กิจกรรม 1-4 การเขียนรหัสเทียมคำนวณหาพื้นที่แผ่นหญ้า

ให้นักเรียนเขียนรหัสเทียมของการคำนวณหาจำนวนแผ่นหญ้าจากผังงานตามภาพที่ 1.1

สรุป

อัลกอริทึม หมายถึง ชุดขั้นตอนการทำงานตามลำดับที่คอมพิวเตอร์ และอุปกรณ์อัจฉริยะอื่นๆ สามารถทำได้ เพื่อทำการคำนวณ ประมวลผลข้อมูล และทำงานที่ต้องใช้เหตุผลอย่างอัตโนมัติ

ขั้นตอนการแก้ปัญหาด้วยคอมพิวเตอร์ มีดังนี้

- 1) วิเคราะห์ปัญหา
- 2) ออกแบบอัลกอริทึม
- 3) ตรวจสอบความถูกต้องของอัลกอริทึม
- 4) นำอัลกอริทึมไปออกแบบและเขียนโปรแกรม

การคิดเชิงคำนวณ เป็นกระบวนการคิดที่ต้องใช้ทักษะและเทคนิคเพื่อแก้ปัญหาในเชิงนามธรรม ประกอบด้วย การแบ่งย่อยปัญหา การรู้จำรูปแบบของปัญหา การมองภาพรวม หรือแนวคิดเชิงนามธรรม และการออกแบบอัลกอริทึม

คุณสมบัติของอัลกอริทึม

- 1) มีการทำงานที่ชัดเจน และทำได้จริง ไม่คลุมเครือ
- 2) มีลำดับขั้นตอนการทำงานที่ชัดเจน จนจบการทำงาน
- 3) ง่าย หรือตรวจสอบแก้ไขได้
- 4) มีความถูกต้อง ทั้งการทำงาน และผลลัพธ์
- 5) ทำงานได้ครบทุกเงื่อนไข
- 6) มีประสิทธิภาพ คือ ใช้เวลา และหน่วยความจำที่เหมาะสม

อัลกอริทึมเขียนได้หลายรูปแบบ ดังต่อไปนี้

- 1) แบบบรรยาย (Narrative description)
- 2) แบบขั้นตอน (Step by step)
- 3) แบบผังงาน (Flowchart)
- 4) แบบรหัสเทียม (Pseudo-code)
- 5) แบบโปรแกรม (Program)

ประโยชน์ของอัลกอริทึม มีดังนี้

- 1) ทำให้การทำงาน หรือการแก้ปัญหาชัดเจนขึ้น
- 2) ช่วยในการแก้ปัญหา
- 3) ทำให้การทำงานสำเร็จอย่างรวดเร็ว
- 4) อัลกอริทึมที่ออกแบบไว้ดีแล้ว สามารถนำไปใช้ได้ อีก หรืออาจนำไปใช้กับงานอื่นได้

การเขียนผังงาน เป็นการออกแบบอัลกอริทึมที่ใช้รูปภาพสัญลักษณ์ แทนขั้นตอนการทำงานต่างๆ ในแต่ละขั้น และใช้เส้นหัวลูกศรเชื่อมต่อ เพื่อแสดงเส้นทางการทำงาน

หลักการเขียนผังงานที่ดี ควรทำดังต่อไปนี้

1. ใช้สัญลักษณ์ตรงตามที่กำหนดไว้ มีขนาด และระยะห่างที่เหมาะสม

2. มีจุดเริ่มต้น (Start) และจุดสิ้นสุด (End) เพียงชุดเดียว
3. ใช้เส้นหัวลูกศรแสดงทิศทางการไหลของข้อมูล
4. เส้นลูกศรต้องต่อเชื่อมกับสัญลักษณ์ โดยต่อเข้าทางเดียว และออกทางเดียวตรงกลางสัญลักษณ์ ยกเว้นสัญลักษณ์การตัดสินใจที่เข้าทางเดียว แต่ออกได้หลายทาง
5. ใช้คำอธิบายการทำงานในแต่ละขั้นตอนด้วยข้อความที่สั้น กระชับ ชัดเจน
6. ใช้สัญลักษณ์จุดต่อในหน้าเดียวกัน แทนการโยงเส้นยาว ๆ หรือเส้นที่ตัดกัน
7. ถ้าไม่สามารถเขียนในหน้าเดียวได้ ให้ใช้สัญลักษณ์จุดต่อระหว่างหน้า
8. ทดสอบความถูกต้องของการทำงานก่อน จึงนำไปใช้เขียนโปรแกรมต่อไป
9. เขียนให้ครบทุกฟังก์ชันที่ใช้ และกำหนดขั้นตอนและรายละเอียดให้ชัดเจน
10. กำหนดชื่อผังงานกำกับไว้

โครงสร้างของผังงาน มี 3 แบบดังต่อไปนี้

1. โครงสร้างแบบตามลำดับ (Sequence) เป็นการแสดงขั้นตอนการทำงานตามลำดับ
2. โครงสร้างแบบทางเลือก (Selection) เป็นการแสดงขั้นตอนการทำงานตามเงื่อนไขให้เลือกทำ ซึ่งจะมีการทดสอบเงื่อนไขเพื่อแยกแยะการเลือกทำงานให้เป็นไปตามกรณีที่ต้องการ ถ้าเงื่อนไขเป็นจริงจะให้ทำอะไร และถ้าเงื่อนไขเป็นเท็จจะให้ทำอะไร
3. โครงสร้างแบบวนซ้ำ (Iteration) เป็นการแสดงขั้นตอนการทำงานแบบวนซ้ำ ซึ่งจะมีการทดสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริงให้วนซ้ำการทำงานอีก ถ้าเป็นเท็จให้หยุดการวนซ้ำ

ประโยชน์ของผังงาน

1. ช่วยจัดลำดับขั้นตอนการทำงานของโปรแกรม และสามารถนำไปเขียนโปรแกรมได้อย่างถูกต้อง
2. ช่วยในการตรวจสอบ และแก้ไขโปรแกรมได้ง่าย เมื่อเกิดข้อผิดพลาด
3. ช่วยให้การดัดแปลง แก้ไข ทำได้อย่างสะดวกและรวดเร็ว
4. ช่วยให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมได้อย่างสะดวก และรวดเร็วมากขึ้น

การเขียนรหัสเทียม เป็นการเขียนขั้นตอนการทำงานโดยใช้รหัสเทียม หรือคำสั่งเทียมที่ไม่ใช่คำสั่งของภาษาคอมพิวเตอร์ แทนการทำงานในแต่ละขั้น (โดยไม่ต้องใช้สัญลักษณ์เหมือนผังงาน) โดยทั่วไปเขียนเป็นภาษาอังกฤษ

คำสั่งเทียมที่ใช้จะเป็นคำสั่งพื้นฐานในการทำงานของคอมพิวเตอร์ ดังต่อไปนี้

1. การนำข้อมูลเข้า เช่น GET, INPUT
2. การนำข้อมูลออก เช่น DISPLAY, PRINT
3. การคำนวณ เช่น COMPUTE

4. การกำหนดค่า เช่น SET
5. การเพิ่มค่า เช่น INCREMENT
6. การทดสอบเงื่อนไข เช่น IF ... THEN ... ELSE
7. การวนซ้ำ เช่น WHILE ... ENDWHILE

หลักการเขียนรหัสเทียมที่ดี ควรทำดังต่อไปนี้

1. กำหนดชื่ออัลกอริทึม
2. เริ่มจากคำ BEGIN, START หรือ เริ่มต้น
3. สิ้นสุดด้วยคำ END, STOP หรือ จบ หรือหยุด
4. คำสั่งต่าง ๆ อยู่ระหว่าง BEGIN และ END โดยจัดให้เยื้องกันไป
5. ใช้คำสั่งเดียวในแต่ละบรรทัด
6. คำสั่งหลักใช้ตัวพิมพ์ใหญ่
7. ใช้ # เพื่อเขียน Comment หรือหมายเหตุเพื่อให้เข้าใจง่ายขึ้น

โครงสร้างของรหัสเทียม มี 3 แบบ เช่นเดียวกับโครงสร้างของผังงาน คือ โครงสร้างแบบตามลำดับ แบบทางเลือก และแบบวนซ้ำ

การวัดและประเมินผลการเรียนรู้ หน่วยที่ 1

คำชี้แจง ตอบคำถามต่อไปนี้ให้ถูกต้อง

1. อัลกอริทึม หมายถึงอะไร
2. คุณสมบัติของอัลกอริทึม มีอะไรบ้าง
3. ประโยชน์ของอัลกอริทึม มีอะไรบ้าง
4. หลักการเขียนผังงานที่ดี ควรทำอย่างไรบ้าง
5. ผังงานมีประโยชน์อย่างไรบ้าง
6. เขียนผังงานของการหาอายุเฉลี่ยของนักเรียน 10 คน
7. หลักการเขียนรหัสเทียม ควรทำอย่างไรบ้าง
8. ยกตัวอย่างคำสั่งเทียมมา 5 คำสั่ง
9. เขียนอัลกอริทึมของการบวกเลข 1 ถึง 10 โดยใช้รหัสเทียม
10. ผังงาน และรหัสเทียม เหมือนกันหรือต่างกันอย่างไร อธิบายพอสังเขป