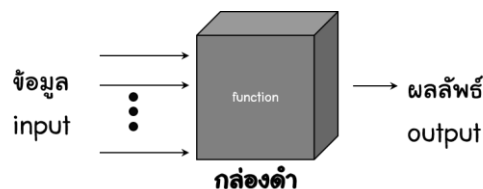


บทที่ 10

การแตกปัญหาออกเป็นปัญหาย่อยและการสร้างลำดับขั้นตอนย่อย

การแตกปัญหาออกเป็นปัญหาย่อย หรือเรียกว่า ฟังก์ชัน โดยฟังก์ชันคือการนำปัญหาใหญ่ๆ มาแบ่งออกเป็นส่วนๆ สำหรับส่วนไหนที่จะต้องใช้งานบ่อย ๆ ถ้าสร้างขึ้นในรูปแบบฟังก์ชันจะสามารถเรียกใช้งานได้เลย ไม่ต้องสร้างขึ้นใหม่ทุกครั้งที่ต้องการใช้งาน ช่วยให้สามารถประหยัดเวลา และเกิดความสะดวกมากยิ่งขึ้น ฟังก์ชันทางเทคโนโลยีสารสนเทศแต่ละฟังก์ชันจะมีชื่อ และลักษณะการทำงานที่แตกต่างกันออกไป หรือบางครั้งฟังก์ชันสำเร็จรูปที่สร้างมาแล้วก็มี



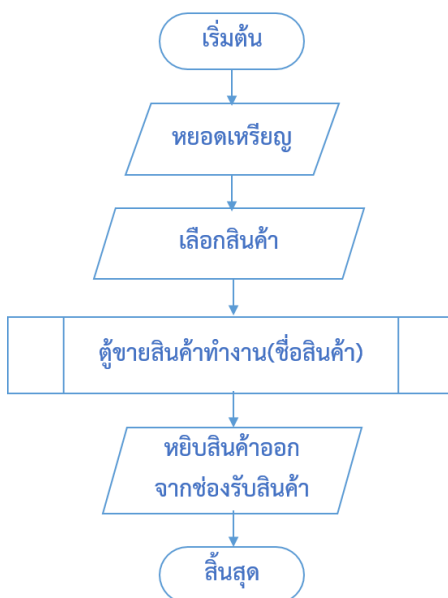
ฟังก์ชันเปรียบเสมือนกล่องดำกล่องหนึ่ง เมื่อใส่ค่าหรือตัวเลขเข้าไป 1 หรือหลายค่า ผลลัพธ์ที่ได้ออกมาจะเป็นตัวเลขที่ผ่านกระบวนการต่าง ๆ ที่อยู่ภายในกล่องดำเรียบร้อยแล้ว โดยข้อมูลเข้าและผลลัพธ์จะมีความสัมพันธ์บางอย่างต่อกันอยู่ ความสัมพันธ์นั้นคือกระบวนการที่ทำให้ได้ผลลัพธ์นั่นเอง

ฟังก์ชันสำหรับอุปกรณ์ฮาร์ดแวร์ คือการทำงานที่สมบูรณ์แบบ ที่มีผลมาจากการมองเห็นของเครื่องคอมพิวเตอร์ อย่างเช่น เครื่องคอมพิวเตอร์ที่ต่อพ่วงกับเครื่องพิมพ์ ถือได้ว่าเครื่องพิมพ์เป็นหนึ่งในฟังก์ชันของคอมพิวเตอร์ ผลลัพธ์ที่ได้จากฟังก์ชันอย่างเครื่องพิมพ์คือเอกสารที่ถูกพิมพ์ออกมาจากเครื่องพิมพ์ ผ่านการสั่งงานจากเครื่องคอมพิวเตอร์





ตู้กดเครื่องดื่มอัตโนมัติหยอดเหรียญนับเป็นตัวอย่างที่สามารถแสดงถึงฟังก์ชันได้ เพราะมีการทำงานที่แตกต่างกันอย่างชัดเจนตรงที่เมื่อกดปุ่มเครื่องดื่มยี่ห้อใด หลังจากการหยอดเหรียญแล้ว เครื่องดื่มยี่ห้อนั้นจะร่วงลงมาที่บริเวณช่องรับเครื่องดื่ม โดยผู้ใช้ไม่ทราบกระบวนการขั้นตอนการทำงานภายในของตู้กดเครื่องดื่มอัตโนมัติเลย ข้อมูลข่าวของตู้กดเครื่องดื่มนี้คือเหรียญหรือธนบัตรที่ผู้ใช้ใส่เข้าไปในตู้และยี่ห้อหรือเบอร์ของเครื่องดื่มที่ผู้ใช้กดหลังจากการหยอดเหรียญหรือธนบัตร และข้อมูลผลลัพธ์คือเครื่องดื่มที่ร่วงลงมาทางช่องสำหรับรับสินค้านั่นเอง ตัวอย่างฟังก์ชันจากตู้กดเครื่องดื่มอัตโนมัติสามารถเขียนผังงานได้ ดังนี้



จากผังงาน สังเกตได้ว่าการใช้สัญลักษณ์ใหม่เข้ามา คือสัญลักษณ์ Predefined Process เป็นสัญลักษณ์สำหรับการเรียกใช้ฟังก์ชันนั่นเอง ส่วนขยายหรือรายละเอียดของฟังก์ชันนั้นจะถูกเขียนแยกเป็นผังงานอีกหนึ่งผังงาน โดยผังงานที่เขียนแยกสำหรับฟังก์ชันจะใช้สัญลักษณ์ จะใช้สัญลักษณ์ Terminator แต่ภายในจะใช้ชื่อฟังก์ชันแทนการใช้ข้อความว่า "start" หรือ "เริ่มต้น" ต่อด้วยการทำงานต่าง ๆ ที่แทนด้วยสัญลักษณ์เช่นเดียวกับการเขียนผังงานปกติ แต่สัญลักษณ์สุดท้ายจะเปลี่ยนเป็นข้อความว่า “คืนค่า” แทนข้อความว่า “สิ้นสุด”

ตัวอย่าง : โรงงานผลิตตุ๊กตา

โรงงานส่วนใหญ่จะถูกแบ่งส่วนผลิตออกเป็นแผนกๆ โดยแต่ละแผนกจะผลิตชิ้นงานออกมาต่าง ๆ กัน ทุก ๆ ชิ้นส่วนจากทุก ๆ แผนก อาจนำมาใช้ หรือไม่สำหรับการประกอบเป็นผลผลิตที่สมบูรณ์หนึ่งชิ้น ตัวอย่างเช่น โรงงานผลิตตุ๊กตา โดยโรงงานผลิตตุ๊กตาจะแบ่งส่วนการผลิตออกเป็นแผนกละชิ้นส่วน

- แผนกสำหรับเย็บส่วนหัว
- แผนกสำหรับเย็บส่วนตัว
- แผนกสำหรับเย็บส่วนแขน
- แผนกสำหรับเย็บส่วนขา
- แผนกสำหรับเย็บส่วนหู
- แผนกสำหรับเย็บเสื้อผ้า

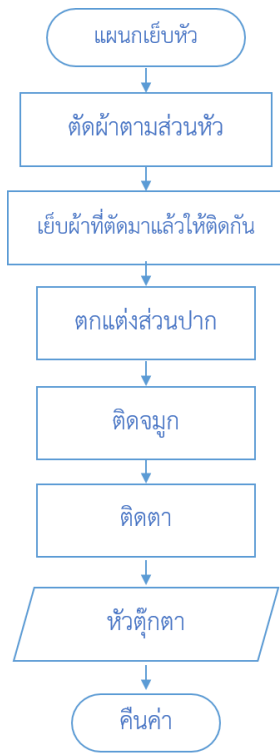


และมีหนึ่งแผนกที่จะรวมส่วนต่าง ๆ ของตุ๊กตาเป็นตุ๊กตาที่เสร็จสมบูรณ์ 1 ตัว สามารถมองให้แผนกเย็บรวมเพื่อให้เป็นตุ๊กตาที่สมบูรณ์นี้เป็นส่วนของโปรแกรมหลัก สามารถเขียนผังงานของแผนกรวม และแต่ละแผนกได้ ดังนี้



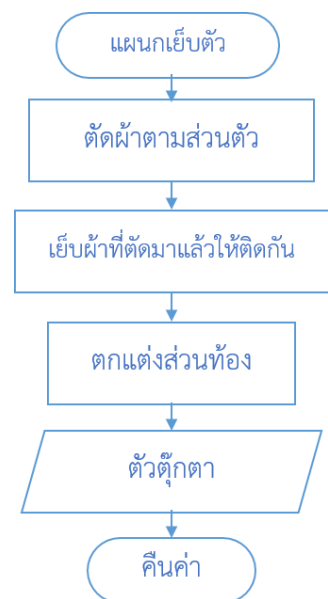
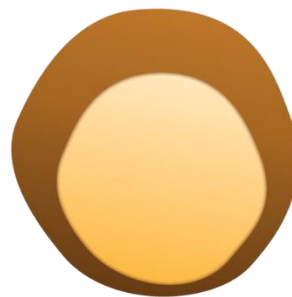
เริ่มต้นที่สัญลักษณ์ Input/Output จะรับของเข้าสู่โรงงาน โดยเป็นวัตถุดิบที่จะต้องใช้ในการเย็บตุ๊กตาชิ้นส่วนต่าง ๆ รวมทั้งใช้ประกอบตุ๊กตาด้วย ต่อมาที่สัญลักษณ์ Predefined Process "หัว = แผนกเย็บหัว" จะทำการเย็บส่วนหัวและส่งผลลัพธ์ออกมาเป็นหัวของตุ๊กตาที่เสร็จเรียบร้อยแล้ว ต่อมาที่สัญลักษณ์ Predefined Process "ตัว = แผนกเย็บตัว" แผนกนี้จะทำการเย็บส่วนตัวและคืนผลลัพธ์มาเป็นส่วนตัวของตุ๊กตาที่เย็บเสร็จเรียบร้อยแล้ว แล้ว สัญลักษณ์ Predefined Process "แขน = แผนกเย็บแขน" จะทำการตัดเย็บแขนของตุ๊กตาและคืนค่าเป็นแขนของตุ๊กตาที่ตัดเย็บเรียบร้อยแล้ว ดำเนินการต่อที่ "ขา = แผนกเย็บขา" จะทำการตัดเย็บส่วนขาและคืนค่าเป็นขาของตุ๊กตาที่ตัดเย็บเรียบร้อยแล้ว ถัดมาเป็นสัญลักษณ์ Predefined Process "หู = แผนกเย็บหู" จะทำการตัดเย็บหูและคืนค่าออกมาเป็นผลงานหูของตุ๊กตาที่เสร็จเรียบร้อยแล้ว เครื่องหมาย Predefined Process "หู = แผนกเย็บหู" แผนกเย็บหูจะทำการตัดเย็บส่วนหูของตุ๊กตาและคืนค่าออกมา

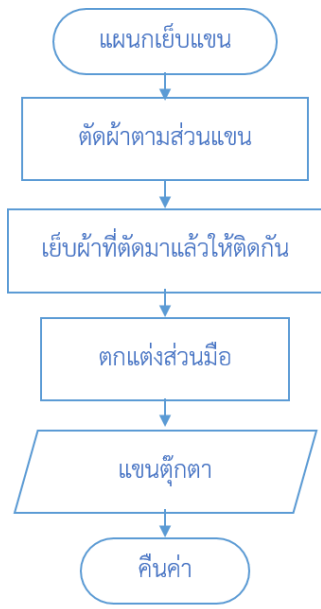
เป็นหูของตุ๊กตาที่ตัดเย็บเสร็จเรียบร้อยแล้ว และต่อที่สัญลักษณ์ Predefined Process "เสื้อผ้า = แผนกเย็บเสื้อผ้า" แผนกเย็บเสื้อผ้าก็จะทำการตัดเย็บเสื้อผ้า และคืนค่ามาเป็นเสื้อผ้าสำหรับสวมใส่ให้ตุ๊กตา และสุดท้ายเมื่อทุก ๆ แผนกส่งชิ้นส่วนที่เรียบร้อยแล้วคืนมาให้ส่วนรวมชิ้นส่วน ส่วนที่รวมชิ้นส่วนจะรวมทุกส่วนจากทุกแผนกเข้าด้วยกันเป็นตุ๊กตาที่เสร็จสมบูรณ์แล้ว 1 ตัว และไม่ว่าโรงงานจะผลิตตุ๊กตา 1 ตัว 10 ตัว 100 ตัวหรือมากกว่า ทุก ๆ แผนกจะทำเพียงหน้าที่ของตนเองเท่านั้น โดยไม่ต้องเรียนรู้ใหม่ แผนกหูเพียงตัดเย็บส่วนหูเหมือนเดิมเพียงแต่จำนวนเพิ่มมากขึ้น ไม่ต้องเรียนรู้การตัดเย็บชุดซึ่งต้องใช้เวลาและอาจเปลืองทรัพยากรหากเกิดความผิดพลาดขึ้น



ผังงานส่วนของแผนกหัว หรือฟังก์ชันสำหรับทำหัว เขียนโดยเริ่มต้นด้วยสัญลักษณ์ Terminator และภายในเป็นชื่อฟังก์ชัน สำหรับแผนกหัวจะระบุว่า “แผนกเย็บหัว” ต่อด้วยสัญลักษณ์ Process สำหรับขั้นตอนภายในแผนกอย่าง “ตัดผ้าตามส่วนหัว” “เย็บผ้าที่ตัดมาแล้วให้ติดกัน” “ตกแต่งส่วนปาก” “ติดจมูก” “ติดตา” และได้ผลลัพธ์ออกมาเป็นหัวตุ๊กตาที่เสร็จสมบูรณ์ที่สัญลักษณ์ Input/Output “หัวตุ๊กตา” จากนั้นคืนค่าที่สัญลักษณ์ Terminator เป็นการส่งชิ้นส่วนหัวของตุ๊กตาที่เรียบร้อยแล้วให้กับส่วนงานที่ทำการประกอบตุ๊กตา

ส่วนต่อมาของตุ๊กตาคือส่วนตัว ส่วนนี้จะถูกผลิตโดยแผนกเย็บส่วนตัว ผังงานจะเริ่มที่สัญลักษณ์ Terminator “แผนกเย็บส่วนตัว” เป็นการเริ่มเข้าสู่ฟังก์ชันหรือแผนกเย็บส่วนตัว ภายในแผนกนี้จะประกอบไปด้วยสัญลักษณ์ Process อย่าง “ตัดผ้าตามส่วนตัว” “เย็บผ้าที่ตัดมาแล้วให้ติดกัน” “ตกแต่งส่วนท้อง” และสุดท้ายจะได้ชิ้นงานออกมาเป็นส่วนตัวของตุ๊กตาที่เสร็จสมบูรณ์ และส่งต่อไปให้กับส่วนงานรวมชิ้นส่วนตุ๊กตาต่อไปที่สัญลักษณ์ Terminator “คืนค่า”

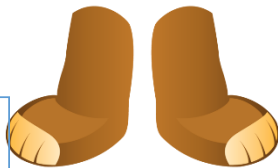
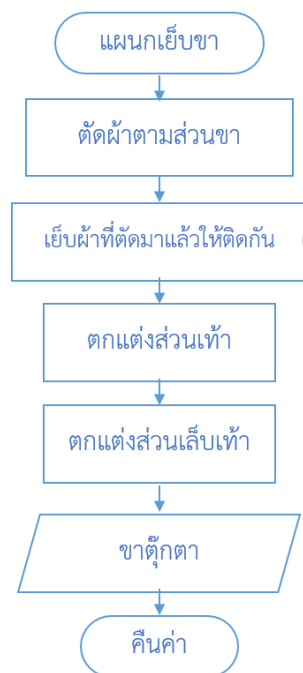




ถัดมาเป็นส่วนของแขนตุ๊กตา ฝั่งงานส่วนของแผนกเย็บแขนตุ๊กตา เริ่มต้นด้วยสัญลักษณ์ Terminator “แผนกเย็บแขน” และภายในฟังก์ชันหรือแผนกนี้จะเริ่มด้วยกระบวนการทำงานด้วยสัญลักษณ์ Process “ตัดผ้าตามส่วนแขน” “เย็บผ้าที่ตัดมาแล้วให้ติดกัน” “ตกแต่งส่วนมือ” เรียบร้อยแล้วจะได้ชิ้นส่วนของแขนตุ๊กตาออกมา 1

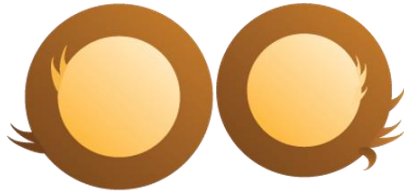
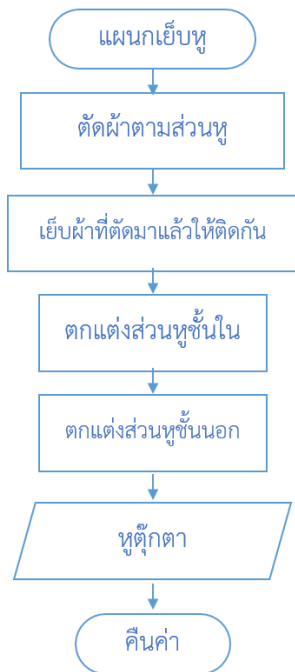
คู่ ประกอบด้วยแขนข้างซ้าย 1 ข้างและแขนข้างขวา 1 ข้าง จากนั้นทำการส่งแขนทั้งสองข้างไปยังส่วนงานสำหรับรวมตุ๊กตาให้เป็นตุ๊กตาที่สมบูรณ์ที่สัญลักษณ์

Terminator “สินค้า” เป็นอันจบการทำงานของแผนกเย็บแขน หรือฟังก์ชันแผนกเย็บแขน



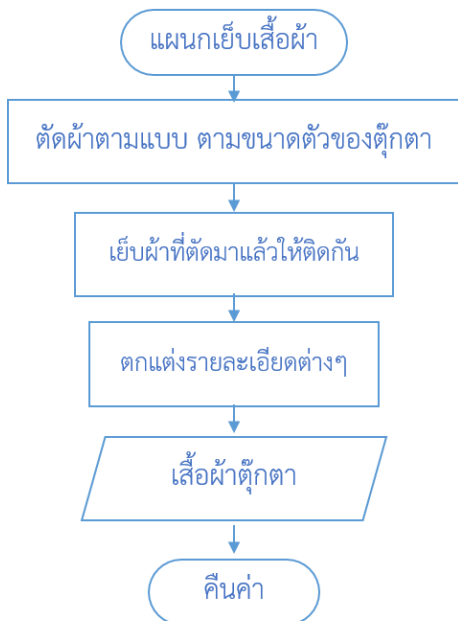
แผนกเย็บขาตุ๊กตาหรือฟังก์ชันเย็บขาตุ๊กตา ส่วนของฟังก์ชันเริ่มต้นด้วยสัญลักษณ์ Terminator “แผนกเย็บขา” และกระบวนการภายในประกอบด้วยสัญลักษณ์ Process “ตัดผ้าตามส่วนขา” “เย็บผ้าที่ตัดมาแล้วให้ติดกัน”

“ตกแต่งส่วนเท้า” “ตกแต่งส่วนเล็บเท้า” ทำให้ได้ขาตุ๊กตามา 1 คู่ ประกอบด้วยขาข้างซ้าย 1 ข้างและขาข้างขวา 1 ข้าง ที่สัญลักษณ์ Input/Output “ขาตุ๊กตา” และส่งขาตุ๊กตาที่เสร็จสมบูรณ์จำนวน 1 คู่ไปยังส่วนที่รวมชิ้นส่วนของตุ๊กตาเข้าด้วยกันเป็นตุ๊กตา 1 ตัวที่เสร็จเรียบร้อยแล้ว ที่สัญลักษณ์ “สินค้า”



ส่วนแผนกเย็บหูก็เริ่มต้นปฏิบัติงานด้วยสัญลักษณ์ Terminator เช่นกัน และภายในระบุข้อความว่า “แผนกเย็บหู” จากนั้นภายในฟังก์ชันหรือภายในแผนกจะใช้สัญลักษณ์ Process ที่

ประกอบด้วย “ตัดผ้าตามส่วนหู” “เย็บผ้าที่ตัดมาแล้วให้ติดกัน” “ตกแต่งส่วนหูชั้นใน” “ตกแต่งส่วนหูชั้นนอก” เรียบร้อยแล้วจะได้หูของตุ๊กตามา 1 คู่ ประกอบด้วยหูข้างซ้าย 1 ข้างและหูข้างขวา 1 ข้างนั่นเอง จากนั้นทำการส่งชิ้นส่วนของหู 1 คู่นี้ไปให้ส่วนงานรวมชิ้นส่วนตุ๊กตาที่สัญลักษณ์ Terminator “คินค่า” เป็นอันจบการทำงานของแผนกเย็บส่วนหู

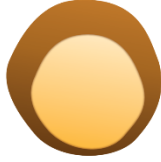
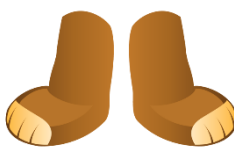


แผนกเย็บเสื่อผ้าของตุ๊กตาจะมีกระบวนการทำงานภายในประกอบด้วย “ตัดผ้าตามแบบตามขนาดตัวของตุ๊กตา” “เย็บผ้าที่ตัดมาแล้วให้ติดกัน” “ตกแต่งรายละเอียดต่างๆ” จะได้ชุดของตุ๊กตาที่เสร็จเรียบร้อยแล้ว และส่งให้ส่วนรวมชิ้นส่วน

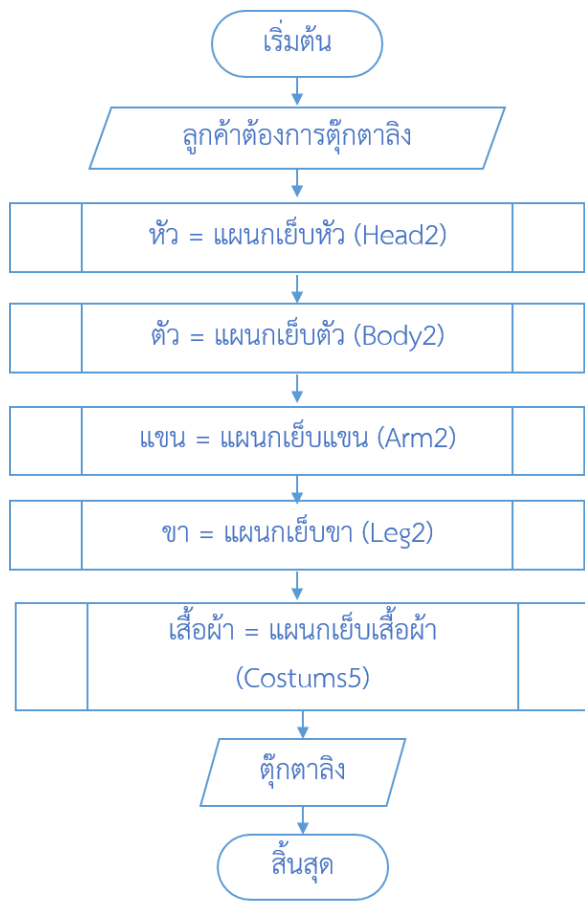
จากนั้นเมื่อทุก ๆ แผนกคินค่ามาเป็นชิ้นส่วนต่าง ๆ ของตุ๊กตาแล้ว ส่วนหลักของโรงงานที่ทำหน้าที่รวมชิ้นส่วนของตุ๊กตาเป็นตัวตุ๊กตาที่เรียบร้อยแล้ว จะทำการเย็บชิ้นส่วนตุ๊กตาประกอบเข้าด้วยกัน หุ้มห่อตุ๊กตา และส่งขายในตลาดต่อไป

ถ้าสมมติให้โรงงานตุ๊กตาไม่ได้ผลิตเพียงตุ๊กตาหมีเท่านั้น แต่ผลิตตุ๊กตาอื่น ๆ ด้วย แล้วยังรับคำสั่งซื้อจากลูกค้าถ้าลูกค้าต้องการตุ๊กตาแบบที่โรงงานไม่เคยผลิตมาก่อน การทำงานของแผนกต่าง ๆ จะเปลี่ยนไป โดนทุก ๆ แผนกจะต้องรองรับการตัดเย็บชิ้นส่วนของตุ๊กตาที่มีความหลากหลายมากขึ้น ส่วนงานที่รับคำสั่งจากลูกค้าจะต้องระบุว่าแต่ละแผนกจะต้องตัดเย็บตุ๊กตาอะไร คล้ายกับการกดยี่ห้อของสินค้าที่ตุ๊กตาคู่มืออัตโนมัตินั่นเอง โดยจะแสดงตัวอย่างการขยายกิจการของโรงงานผลิตตุ๊กตาต่อไป

โดยรูปแบบมาตรฐานของตุ๊กตาทั้งโรงงานมีข้อมูลดังนี้

ตัวอย่า งส่วน หัว					
รหัส	Head1	Head2	Head3	Head4	Head5
ตัวอย่า ง ส่วนตัว					
รหัส	Body1	Body2			
ตัวอย่า งแขน- ขา					
รหัส	Arm1	Arm2	Leg1	Leg2	
ตัวอย่า งเครื่อง แต่ง กาย					
รหัส	Costums1	Costums2	Costums3	Costums4	Costums5

ถ้าลูกค้าสั่งตุ๊กตาลิง ลูกค้าจะได้ลิงที่สมบูรณ์ เพราะจากตุ๊กตามาตรฐานทั้งหมดของโรงงานแล้ว ลิงและหมี เป็นตุ๊กตาที่มีรูปแบบที่ครบสมบูรณ์ที่สุด โดยแผนกรับคำสั่งจากลูกค้าจะต้องแจ้งรายละเอียดต่อแผนกผลิตว่าต้องผลิตตุ๊กตาอะไร เช่น สั่งแผนกเย็บหัวว่าให้เย็บหัวลิงจะเขียนข้อความในผังงานได้ว่า “แผนกเย็บหัว (Head2)” “แผนกเย็บตัว (Body2)” “แผนกเย็บแขน (Arm2)” “แผนกเย็บขา (Leg2)” ส่วนเครื่องแต่งกายนั้นจะมีหรือไม่ก็ได้ แต่กรณีนี้สมมุติให้ลูกค้าสั่งเครื่องแต่งกายเป็นโบว์สีชมพู จะสั่งแผนกเย็บเสื้อผ้าว่า “แผนกเย็บเสื้อผ้า (Costums5)” การสั่งทำตุ๊กตาลิง สามารถเขียนผังงานได้ ดังนี้



จากผังงาน เริ่มต้นด้วยการรับคำสั่งผลิตจากลูก้าคือ “ลูก้าต้องการตุ๊กตาลิง” ฝ่ายรับออเดอร์จึงต้องแจ้งฝ่ายผลิตให้ผลิตตุ๊กตาลิง และแผนกต่าง ๆ จะต้องรับทราบว่าการทำชิ้นส่วนของตุ๊กตาลิงนั้นจำเป็นต้องทำอะไรบ้าง แผนกผลิตจะกระจายคำสั่งไปที่แผนกเย็บหัวว่าต้องเย็บหัวลิง ใช้สัญลักษณ์ Predefined Process “หัว = แผนกเย็บหัว (Head2)” Head2 เป็นการระบุว่าต้องเย็บหัวตุ๊กตาตามแบบ Head2 นั่นคือหัวลิง สั่งแผนกเย็บตัวว่าต้องเย็บตัวลิง ใช้สัญลักษณ์ Predefined Process “ตัว = แผนกเย็บตัว (Body2)” Body2 คือการระบุว่าต้องเย็บตัวมีลักษณะเป็นตัวของลิงตามแบบมาตรฐานตุ๊กตาของบริษัท สั่งแผนกเย็บแขนว่าต้องเย็บแขนลิง ใช้สัญลักษณ์ Predefined Process “แขน = แผนกเย็บแขน (Arm2)” Arm2 เป็นการระบุว่าต้องเย็บแขนตามรูปแบบของ Arm2 สั่งแผนกเย็บขาว่าต้องเย็บขาแบบ Leg2 และ

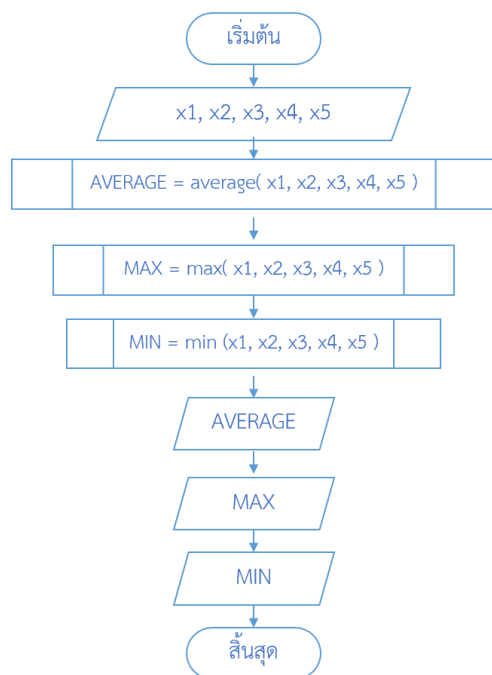
สั่งแผนกเย็บเสื้อผ้าให้เย็บโบว์สีชมพูตามที่ลูก้าต้องการ คือ Costums5 นั่นเอง จากนั้นทุก ๆ ส่วนจะถูกส่งมาที่แผนกรวมชิ้นส่วน และถูกเย็บชิ้นส่วนเข้าด้วยกันกลายเป็นตุ๊กตาลิงติดโบว์สีชมพู ที่สัญลักษณ์ Input/Output “ตุ๊กตาลิง” และนำไปส่งให้ลูก้าต่อไป



จากผังงานสังเกตว่าไม่มีการส่งงานไปให้แผนกเย็บหูเช่นตอนแรกที่โรงงานผลิตตุ๊กตาแบบเดียว นั่นเพราะถูกเย็บภายในแผนกเย็บหัวเรียบร้อยแล้วนั่นเอง สังเกตได้จากรูปแบบตุ๊กตามาตรฐานทั้งหมดที่โรงงานแห่งนี้มี

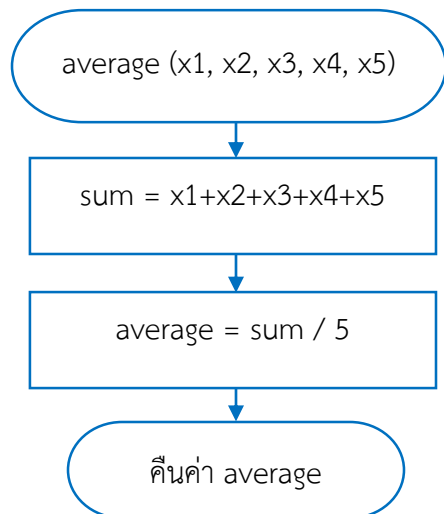
ตัวอย่าง : หาค่า Max, Min, Average

ตัวอย่างนี้จะแสดงการหาค่ามากที่สุด น้อยที่สุด และค่าเฉลี่ย จากจำนวนตัวเลข 5 จำนวนที่ผู้ใช้รับเข้ามา มีการส่งข้อมูลทั้ง 5 จำนวนที่ผู้ใช้รับเข้าไปยังฟังก์ชันการหาค่าเฉลี่ย ค่ามากที่สุด และค่าสูงที่สุด และให้ฟังก์ชันคืนค่าผลลัพธ์กลับมาให้ส่วนโปรแกรมหลัก ตัวอย่างการส่งค่าทั้ง 5 จำนวนไปยังฟังก์ชันต่าง ๆ สามารถเขียนได้ ดังนี้

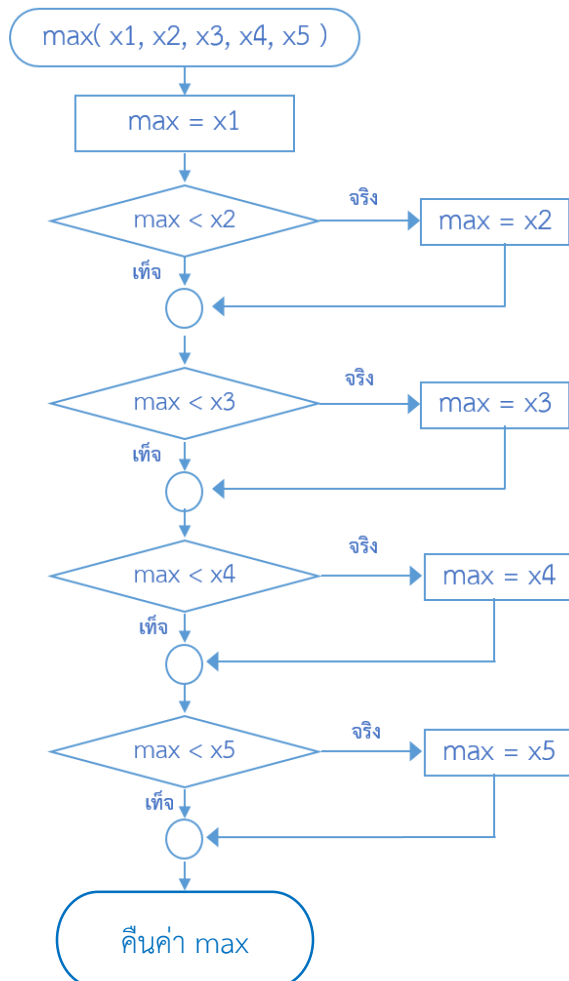


เริ่มจากสัญลักษณ์ Input/Output “x1, x2, x3, x4, x5” เป็นการรับตัวเลขทั้ง 5 จำนวนจากผู้ใช้ เก็บลงตัวแปรทั้ง 5 ตัวตามลำดับการป้อนของผู้ใช้ จากนั้นที่สัญลักษณ์ Predefined Process “AVERAGE = average (x1, x2, x3, x4, x5)” เป็นการส่งค่าทั้ง 5 ค่าไปยังฟังก์ชันเพื่อหาผลลัพธ์ค่าเฉลี่ยทั้ง 5 ค่าและเมื่อฟังก์ชันคืนผลลัพธ์กลับมาให้โปรแกรมหลัก ผลลัพธ์นั้นจะถูกเก็บลงในตัวแปร AVERAGE โปรแกรมหลักส่งค่าทั้ง 5 จำนวนไปยังฟังก์ชัน max และ min อีกด้วย โดยผลลัพธ์จากฟังก์ชัน max ที่คืนจากการทำงานของฟังก์ชันจะถูกเก็บลงในตัวแปร MAX และผลลัพธ์ที่ได้คืนจากการทำงานของฟังก์ชัน min จะถูกเก็บลงในตัวแปร MIN จากนั้นที่สัญลักษณ์ Input/Output “AVERAGE” “MAX” “MIN” คือการแสดงผลของแต่การหาค่าเฉลี่ย ค่ามากที่สุด และค่าสูง

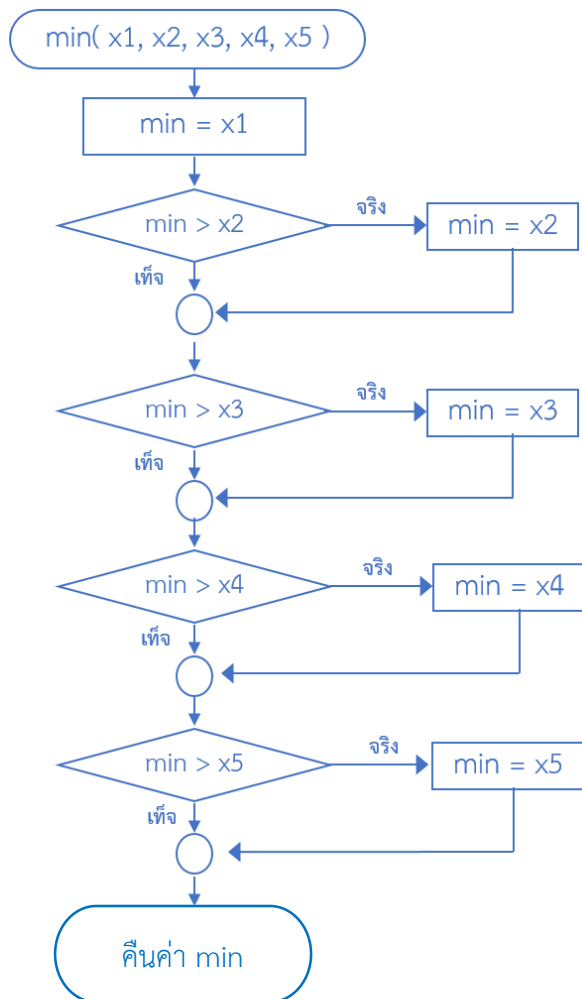
ที่สุด ที่ถูกเก็บอยู่ในตัวแปรเหล่านั้น ตามลำดับ เป็นอันจบการทำงาน รายละเอียดการทำงานของฟังก์ชันต่างๆ เขียนด้วยผังงานได้ ดังนี้



ผังงานส่วนของฟังก์ชันการหาค่าเฉลี่ย ภายในจะทำงานโดยการนำค่าในตัวแปร x1, x2, x3, x4, x5 จากที่โปรแกรมหลักส่งมาบวกกันและเก็บผลบวกลงในตัวแปร sum จากนั้นจะหารด้วย 5 และเก็บผลลัพธ์หลังการหารลงในตัวแปร average และคืนค่าในตัวแปร average กลับสู่โปรแกรมหลัก ซึ่งค่านี้จะถูกเก็บลงในตัวแปร AVERAGE ของฟังก์ชันหลักเพื่อแสดงผลต่อไป



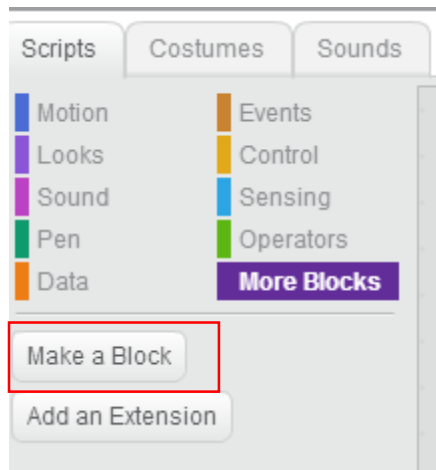
ผังงานแสดงฟังก์ชัน max การทำงานภายในฟังก์ชันเริ่มด้วยการนำค่าในตัวแปร x1 เก็บลงในตัวแปร max จากนั้นนำค่าในตัวแปร max เปรียบเทียบกับค่าในตัวแปร x2 ถ้าค่าในตัวแปร max น้อยกว่าค่าในตัวแปร x2 ค่าในตัวแปร x2 จะถูกเก็บลงในตัวแปร max แทนค่าเดิมซึ่งเป็น x1 แต่ถ้าค่าในตัวแปร max ไม่ได้น้อยกว่าค่าในตัวแปร x2 ค่าในตัวแปร max จะยังคงเป็นค่า x1 เหมือนเดิม และเปรียบเทียบค่าในตัวแปร max ละตัวแปร x3 ต่อ ถ้าค่าในตัวแปร max น้อยกว่าค่าในตัวแปร x3 ค่าในตัวแปร x3 จะถูกเก็บลงในตัวแปร max แทนค่าเดิม จากนั้นนำค่าในตัวแปร max เปรียบเทียบกับค่าในตัวแปร x4 ต่อ ถ้าค่าในตัวแปร max น้อยกว่าค่าในตัวแปร x4 ค่าในตัวแปร x4 จะถูกเก็บลงในตัวแปร max แทนค่าเดิม และสุดท้ายเปรียบเทียบค่าในตัวแปร max กับค่าในตัวแปร x5 ถ้าค่าในตัวแปร max น้อยกว่าค่าในตัวแปร x5 ค่าในตัวแปร x5 จะถูกเก็บลงในตัวแปร max แทนค่าเดิม จากนั้นฟังก์ชันจะคืนค่าในตัวแปร max ให้กับโปรแกรมหลัก และโปรแกรมหลักมีตัวแปร MAX รองรับผลลัพธ์จากฟังก์ชัน MAX และนำไปแสดงผล



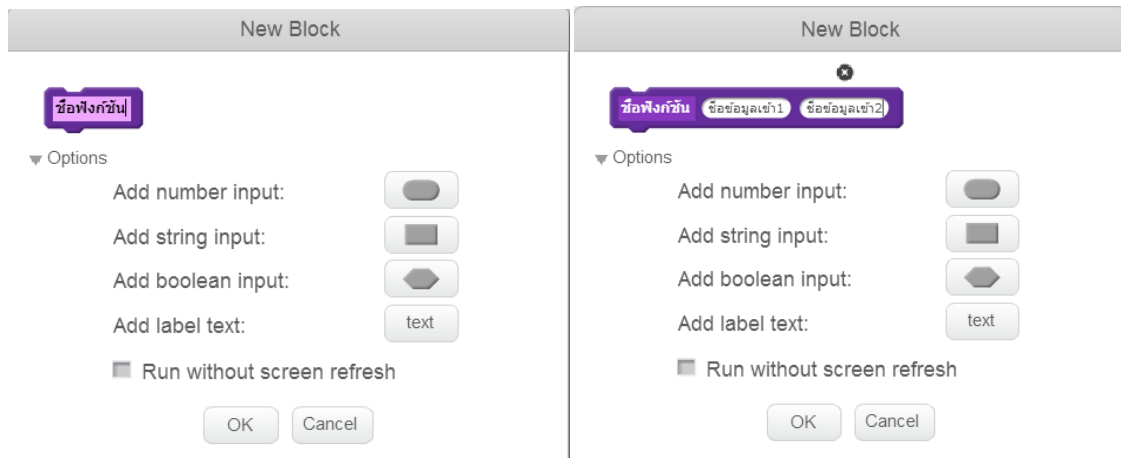
ผังงานแสดงฟังก์ชัน min การทำงานภายในของฟังก์ชัน min เริ่มต้นด้วยการนำค่าจากตัวแปร x1 ที่โปรแกรมหลักส่งมาเก็บลงในตัวแปร min และนำค่าในตัวแปร min เปรียบเทียบกับค่าในตัวแปร x2 ถ้าค่าในตัวแปร min มากกว่าค่าในตัวแปร x2 ค่าในตัวแปร x2 จะถูกเก็บลงในตัวแปร min แทนค่าเดิม นำค่าในตัวแปร min เปรียบเทียบกับค่าในตัวแปร x3 ถ้าค่าในตัวแปร min มากกว่าค่าในตัวแปร x3 ค่าในตัวแปร x3 จะถูกเก็บลงในตัวแปร min แทนค่าเดิม นำค่าในตัวแปร min เปรียบเทียบกับค่าในตัวแปร x4 ต่อไป ถ้าค่าในตัวแปร min มากกว่าค่าในตัวแปร x4 ค่าในตัวแปร x4 จะถูกเก็บลงในตัวแปร min แทนค่าเดิม และสุดท้ายนำค่าในตัวแปร min เปรียบเทียบกับค่าในตัวแปร x5 ถ้าค่าในตัวแปร min มากกว่าค่าในตัวแปร x5 ค่าในตัวแปร x5 จะถูกเก็บลงในตัวแปร min เมื่อเปรียบเทียบข้อมูลทั้ง 5 ตัวที่ถูกเก็บลงในทั้ง 5 ตัวแปรเรียบร้อยแล้ว ฟังก์ชันจะคืนค่าเป็นผลลัพธ์สุดท้ายที่ถูกเก็บลงในตัวแปร min ไปให้โปรแกรมหลัก และโปรแกรมหลักจะเก็บค่าที่ได้คืนจากฟังก์ชัน min ลงในตัวแปร MIN และแสดงผลต่อไป

สร้างชิ้นงานด้วยโปรแกรม Scratch

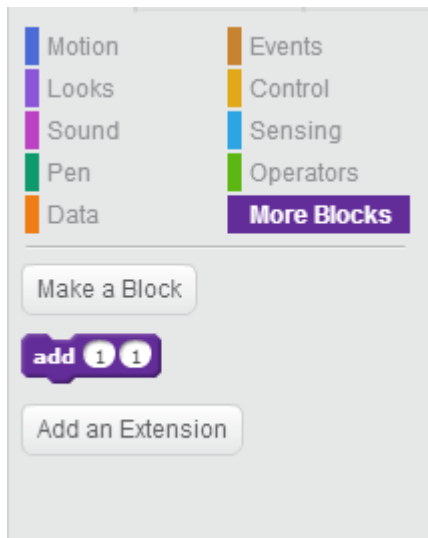
จากการอธิบายและฝึกงาน อาจเข้าใจการทำงานแบบแยกส่วนการทำงานย่อยออกจากการทำงานหลักไปบ้างแล้ว แต่เพื่อเพิ่มความเข้าใจ และสามารถเห็นผลลัพธ์จริงได้ จะแสดงการสร้างชิ้นงานด้วยโปรแกรม Scratch โดยการใช้บล็อกคำสั่งสำหรับการสร้างฟังก์ชัน ดังนี้



บล็อกคำสั่งสำหรับการสร้างฟังก์ชันจะอยู่ในบล็อกคำสั่งหมวด More Blocks ซึ่งเป็นหมวดสุดท้ายจากหมวดของบล็อกคำสั่งทั้งหมด ภายในบล็อกคำสั่งหมวดนี้จะประกอบไปด้วยปุ่ม 2 ปุ่ม กดปุ่มแรกคือ Make a Block จากนั้นจะปรากฏหน้าต่างขึ้นบริเวณกลางหน้าจอ เพื่อให้ผู้ใช้ตั้งชื่อของฟังก์ชัน และกำหนดรูปแบบของค่าที่ฟังก์ชันจะรับจากโปรแกรมหลัก ดังนี้



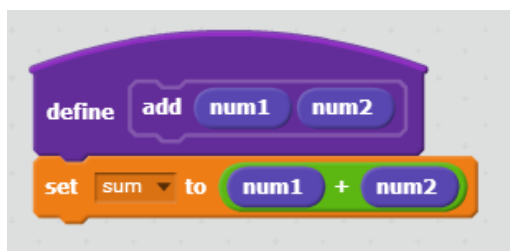
(รูปด้านซ้าย) บล็อกสีม่วงบริเวณด้านบนซ้าย คือส่วนที่ผู้ใช้จะต้องตั้งชื่อให้กับฟังก์ชัน และส่วนถัดมา ภายใต้ข้อความ “options” เป็นส่วนที่ผู้ใช้สามารถคลิกเลือกรูปแบบข้อมูลที่ฟังก์ชันรับจากโปรแกรมหลัก (รูปด้านขวา) เมื่อคลิก 1 ครั้งจะปรากฏบล็อกลักษณะนั้น ๆ 1 บล็อก เมื่อคลิกอีกครั้งจะปรากฏอีก 1 บล็อก โดยบล็อกจะเรียงลำดับกันจากการคลิกครั้งแรก ไปยังการคลิกครั้งสุดท้าย เมื่อปรากฏบล็อกภายในบล็อกสีม่วงด้านบนแล้ว ผู้ใช้สามารถตั้งชื่อให้ตัวแปรนั้น ๆ สำหรับนำไปใช้ภายในฟังก์ชันต่อได้เลย เมื่อสร้างส่วนหัวและข้อมูลรับเข้าของฟังก์ชันได้แล้ว ต่อไปคือการใส่ลำดับการทำงานให้กับฟังก์ชัน



หลังจากผู้ใช้กดปุ่ม OK บนหน้าต่างการสร้างฟังก์ชันแล้ว จะปรากฏบล็อกคำสั่งสีม่วง บริเวณหมวดคำสั่ง โดยมีชื่อและตัวแปรที่รับเข้าตามที่ใช้กำหนด การกำหนดการทำงานภายในฟังก์ชันทำได้โดยการนำบล็อกคำสั่งที่สร้างเรียบร้อยแล้วนี้ วางลงบริเวณพื้นที่สำหรับวางบล็อกคำสั่ง เช่นเดียวกับการวางบล็อกคำสั่งทั่วไป



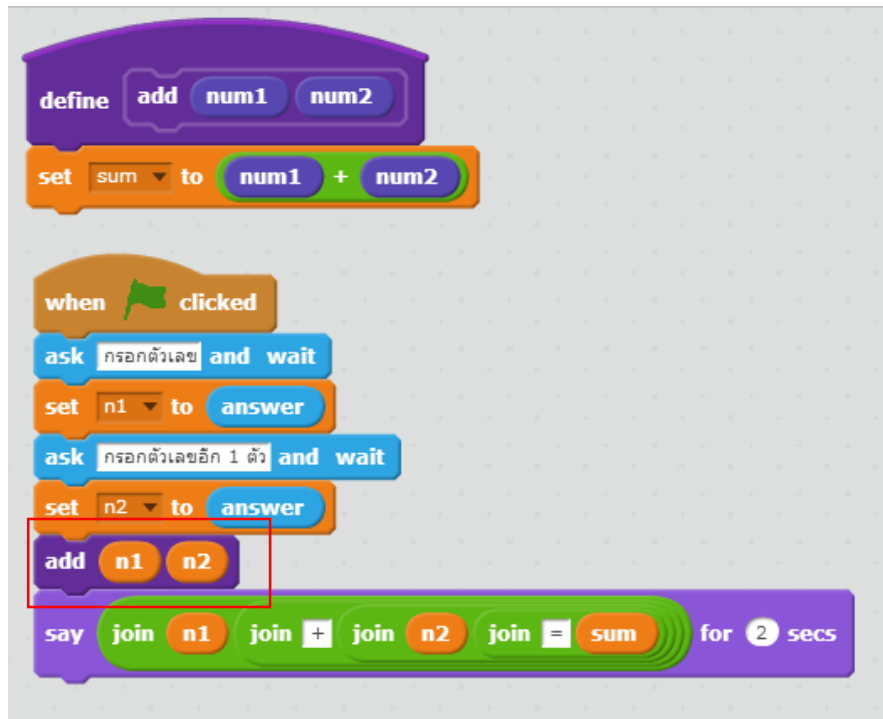
ตัวอย่างฟังก์ชันนี้คือฟังก์ชันสำหรับหาผลบวกจากจำนวนใด ๆ 2 จำนวน ชื่อฟังก์ชัน add ตัวแปรที่รับเข้ามี 2 ตัวแปรคือ num1 และ num2 การหาผลลัพธ์ของฟังก์ชันจะต้องสร้างตัวแปร 1 ตัวเพื่อรองรับค่าผลลัพธ์ สำหรับการสร้างตัวแปรนั้น สามารถทำได้เช่นเดียวกับการสร้างตัวแปรปกติ โดยกดที่บล็อกคำสั่งหมวด Data และดำเนินการตั้งชื่อให้เรียบร้อย แล้วนำมาใช้งาน โดยการลากมาวางต่อบล็อกคำสั่งสีม่วงนี้ได้เลย



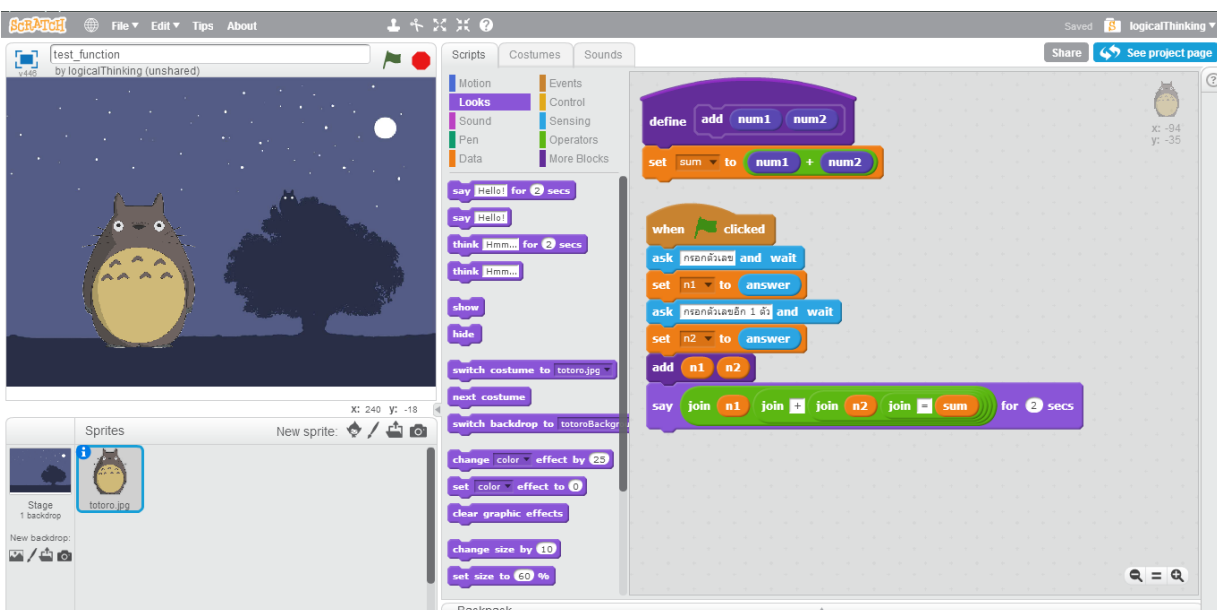
ตัวอย่างนี้จะสร้างตัวแปร sum เพื่อรองรับผลบวกของ num1 และ num2 การใช้ตัวแปร num1 และ num2 สามารถคลิกเมาส์บริเวณตัวแปรนั้น ที่บล็อกคำสั่งสีม่วง แล้วลากมาวางบริเวณที่ต้องการใช้ได้เลย จากรูป ทำการคลิกที่ตัวแปร num1 และ num2 แล้วลากมาวางซ้อนทับบล็อกคำสั่งการดำเนินการ

ทางคณิตศาสตร์อย่างบล็อกคำสั่งบวก (+) เลย การนำผลลัพธ์ที่ถูกเก็บในตัวแปร sum สามารถนำไปใช้ต่อได้เลย เช่นเดียวกับตัวแปรอื่น ๆ เช่น ให้ตัวละครแสดงข้อความว่า say...join...“ผลรวมเท่ากับ”...sum..3...secs สามารถนำบล็อกตัวแปร sum วางซ้อนทับบล็อกคำสั่ง say และบล็อกคำสั่ง join ได้เลย ดังนี้



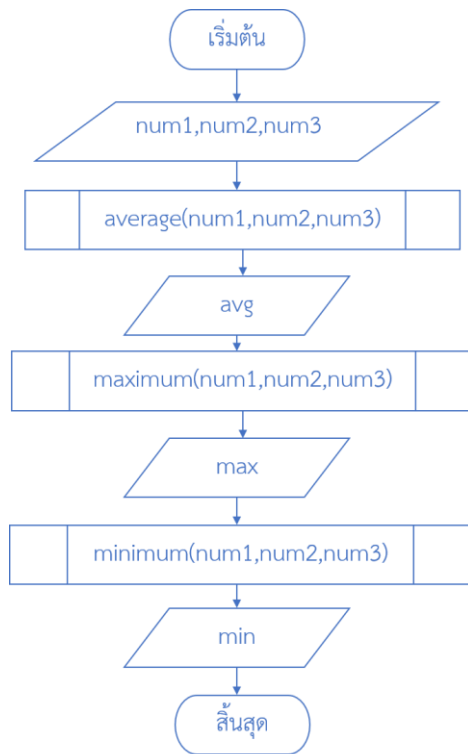


ภาพนี้ แสดงการสร้างบล็อกคำสั่งฟังก์ชัน การวางบล็อกคำสั่งการทำงานของฟังก์ชัน และการใช้งานบล็อกคำสั่งฟังก์ชัน จากภาพบริเวณที่มีกรอบสีแดง คือการเรียกใช้งานฟังก์ชัน โดยจะต้องส่งค่าไปให้ฟังก์ชัน 2 ค่าคือ n1 และ n2 และแสดงผลผ่านการพูดของตัวละคร ด้วยผลลัพธ์ที่ถูกเก็บอยู่ในตัวแปร sum สามารถชมชิ้นงานและบล็อกคำสั่งของตัวอย่างฟังก์ชัน add ได้ที่ <https://scratch.mit.edu/projects/111957257/>

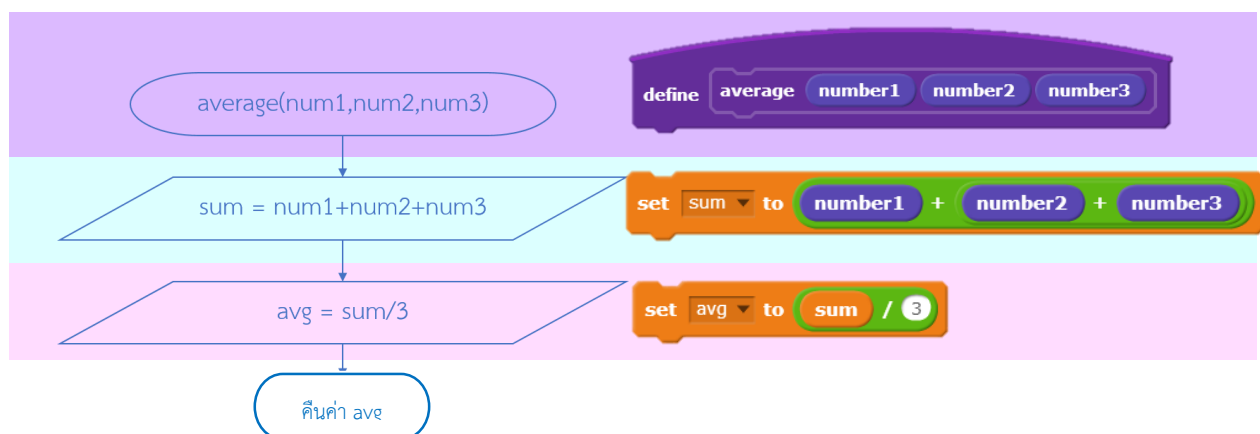


ตัวอย่าง : หาค่า Max, Min, Average

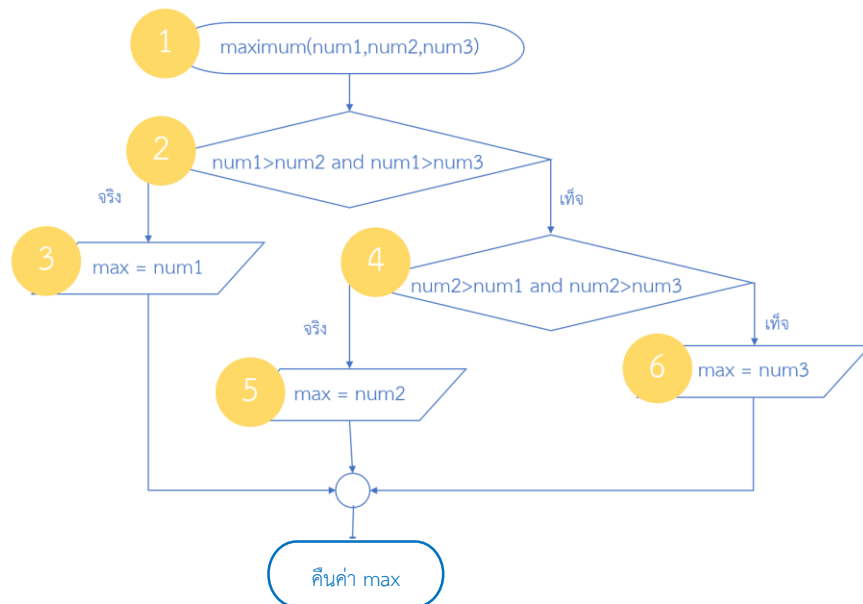
จากตัวอย่างก่อนหน้านี้ที่แสดงถึงการหาค่าที่มากที่สุด ค่าที่น้อยที่สุด และค่าเฉลี่ย จากข้อมูล 5 จำนวน แล้ว จะแสดงการสร้างฟังก์ชันการหาค่ามากที่สุด สูงที่สุด และค่าเฉลี่ย จากข้อมูล 3 จำนวน แต่ละฟังก์ชันจะต้องรับข้อมูลเข้า 3 จำนวน และแสดงผังงานของโปรแกรมหลัก ดังนี้



โปรแกรมหลักจะรับข้อมูลเข้า 3 จำนวนเก็บลงในตัวแปร num1 num2 num3 ตามลำดับ จากนั้นส่งค่าทั้ง 3 ไปยังฟังก์ชัน average(num1, num2, num3) ที่สัญลักษณ์ Predefined Process แสดงผลลัพธ์ของตัวแปร avg จากนั้นส่งค่าทั้ง 3 ค่าไปยังฟังก์ชัน maximum(num1, num2, num3) แสดงผลลัพธ์ของตัวแปร max ส่งค่าทั้ง 3 ค่าไปยังฟังก์ชัน minimum(num1, num2, num3) แสดงผลลัพธ์ของตัวแปร min แต่ละฟังก์ชันมีรายละเอียดการทำงานที่แตกต่างกัน ดังแสดงในรูปแบบของผังงานและการวางบล็อกคำสั่งได้ ดังนี้



จากบล็อกคำสั่ง ตัวแปร sum จะเก็บผลลัพธ์ของ number1 + number2 + number3 และหารด้วย 3 เพื่อหาค่าเฉลี่ย จากนั้นเก็บค่าเฉลี่ยลงในตัวแปร avg เพื่อให้โปรแกรมหลักนำไปใช้แสดงผล



- (1) ชื่อฟังก์ชัน maximum รับข้อมูลเข้า 3 ตัวชื่อเป็นตัวแปรชื่อ num1 num2 num3 วางบล็อกคำสั่งในโปรแกรม Scratch ดังนี้



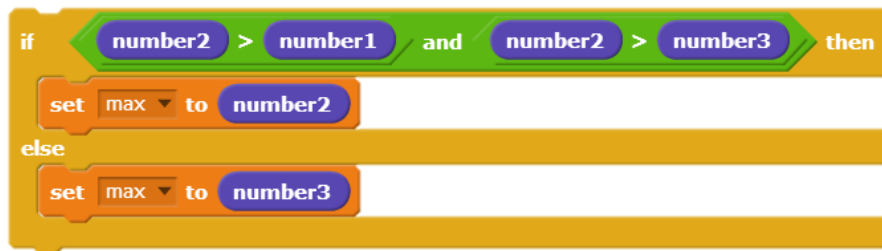
บล็อกคำสั่งสำหรับฟังก์ชันชื่อ maximum รับตัวแปรเข้าฟังก์ชัน 3 ตัวคือ number1 number2 และ number3

- (2) เปรียบเทียบหาค่าที่มากที่สุด โดยเงื่อนไขนี้ตรวจสอบว่า num1>num2 และ num1>num3 จริงหรือไม่ ถ้า num1 มากกว่าทั้ง num2 และ num3 จริงแปลว่า num1 คือค่าที่มากที่สุด จึงเก็บค่าในตัวแปร num1 ลงตัวแปร max วางบล็อกคำสั่งได้ ดังนี้

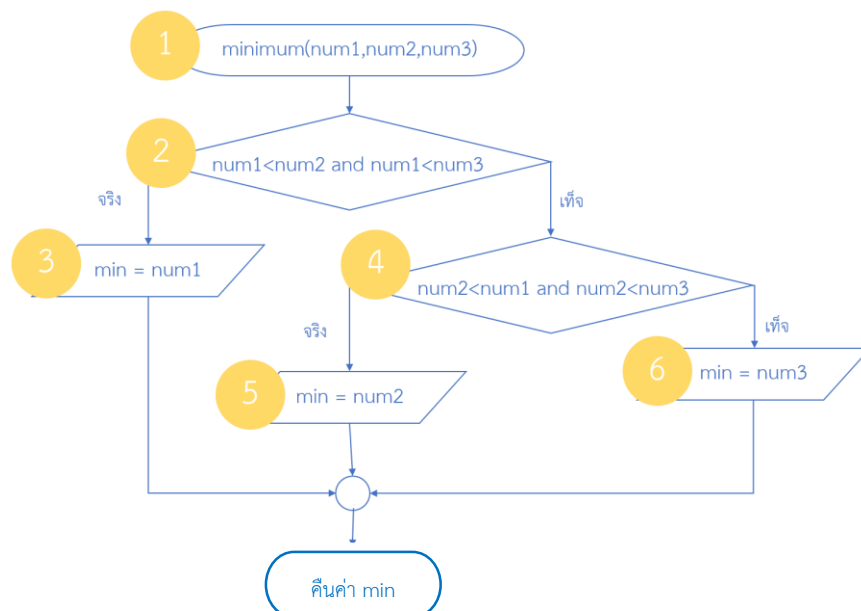


วางบล็อกคำสั่ง if.....then วางบล็อกคำสั่ง and ที่เงื่อนไขของบล็อกคำสั่ง if วางบล็อกคำสั่งมากกว่า (>) จากนั้นนำตัวแปรของฟังก์ชันมาวางเพื่อหาเปรียบเทียบ ถ้าเงื่อนไขทั้งหมดเป็นจริง จะเก็บค่าในตัวแปร number1 สู่ตัวแปร max

- (3) เก็บค่าจากตัวแปร max สู่วัสดุแปร max จะทำงานเมื่อ num1 มากกว่า num2 และ num3 เท่านั้น วางบล็อกคำสั่งเป็นบล็อกคำสั่งภายใน (2)
- (4) – (6) เมื่อเงื่อนไข $num1 > num2$ and $num1 > num3$ เป็นเท็จจึงดำเนินการมาส่วนนี้ โดยจะทำการตรวจสอบว่า $num2 > num1$ และ $num2 > num3$ จริงหรือไม่ ถ้าจริงจะเก็บค่าของตัวแปร num2 ลงในตัวแปร max วางบล็อกคำสั่งของเงื่อนไข ดังนี้



บล็อกคำสั่งนี้จะเปรียบเทียบค่าในตัวแปร number2 กับ number1 และ number3 ว่ามีค่ามากกว่าทั้งสองจำนวนหรือไม่ ถ้ามีค่ามากกว่าทั้งสองจำนวนจริง ค่าในตัวแปร number2 จะถูกเก็บลงในตัวแปร max แต่ถ้าค่าในตัวแปร number2 ไม่มากกว่า number1 หรือ number3 แล้วแสดงว่าจะเหลืออีกเพียง 1 ค่าที่มากที่สุด มากกว่า number1 และมากกว่า number2 นั่นคือ number3 จะลงมาทำงานที่บล็อกคำสั่งภายใน else คือเก็บค่าในตัวแปร number3 ลงสู่วัสดุแปร max นั่นเอง



- (1) ชื่อฟังก์ชัน minimum รับเข้าตัวแปร 3 ตัวคือ num1 num2 และ num3 วางบล็อกคำสั่งโดยสร้างบล็อกคำสั่งฟังก์ชันและตั้งชื่อให้เรียบร้อย ดังนี้



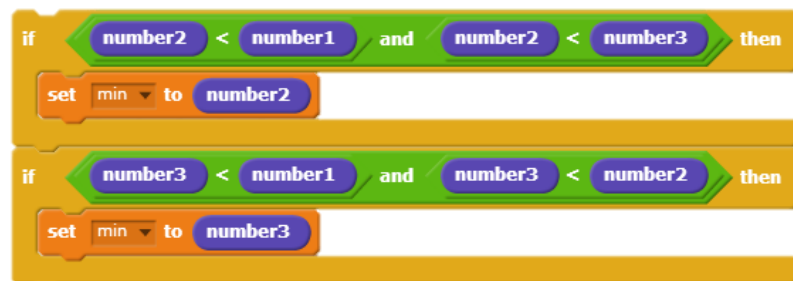
จากบล็อกคำสั่ง เป็นบล็อกคำสั่งฟังก์ชันชื่อ minimum รับข้อมูลเข้า 3 จำนวน ชื่อตัวแปร number1 number2 และ number3

- (2) เปรียบเทียบ num1 กับ num2 และ num3 ถ้าค่าในตัวแปร num1 มากกว่าค่าในตัวแปร num2 และ num3 จริง จะทำการเก็บค่าในตัวแปร num1 สู่ตัวแปร min วางบล็อกคำสั่งดังนี้



บล็อกคำสั่ง if.....then ถูกวางซ้อนด้วยบล็อกคำสั่ง and และบล็อกคำสั่งมากกว่า (>) ถ้า $number1 < number2$ และ $number1 < number3$ จริงจะทำการเก็บค่าจากตัวแปร number1 ลงสู่ตัวแปร min

- (3) ทำการเก็บค่าในตัวแปร num1 ลงสู่ตัวแปร min จะทำงานเมื่อค่าในตัวแปร min น้อยกว่าค่าในตัวแปร num2 และ num3 เท่านั้น
- (4) – (6) เปรียบเทียบค่าในตัวแปร $num2 < num1$ และ $num2 < num3$ เมื่อค่าในตัวแปร num2 น้อยกว่าค่าทั้งในตัวแปร num1 และ num3 แปลว่าค่าในตัวแปร num2 คือค่าที่น้อยที่สุด และเมื่อค่าในตัวแปร num2 คือค่าที่น้อยที่สุดจริงจึงเก็บค่าในตัวแปร num2 ลงในตัวแปร min แต่ถ้าค่าในตัวแปร num2 ไม่น้อยกว่าค่าในตัวแปร num1 และ num3 แล้ว จะเหลือเพียงค่าเดียวที่น้อยที่สุด เมื่อไม่ใช่ค่าในตัวแปร num1 และ num2 แล้ว ค่าที่น้อยที่สุดคือค่าในตัวแปร num3 วางบล็อกคำสั่งของเงื่อนไขได้ ดังนี้

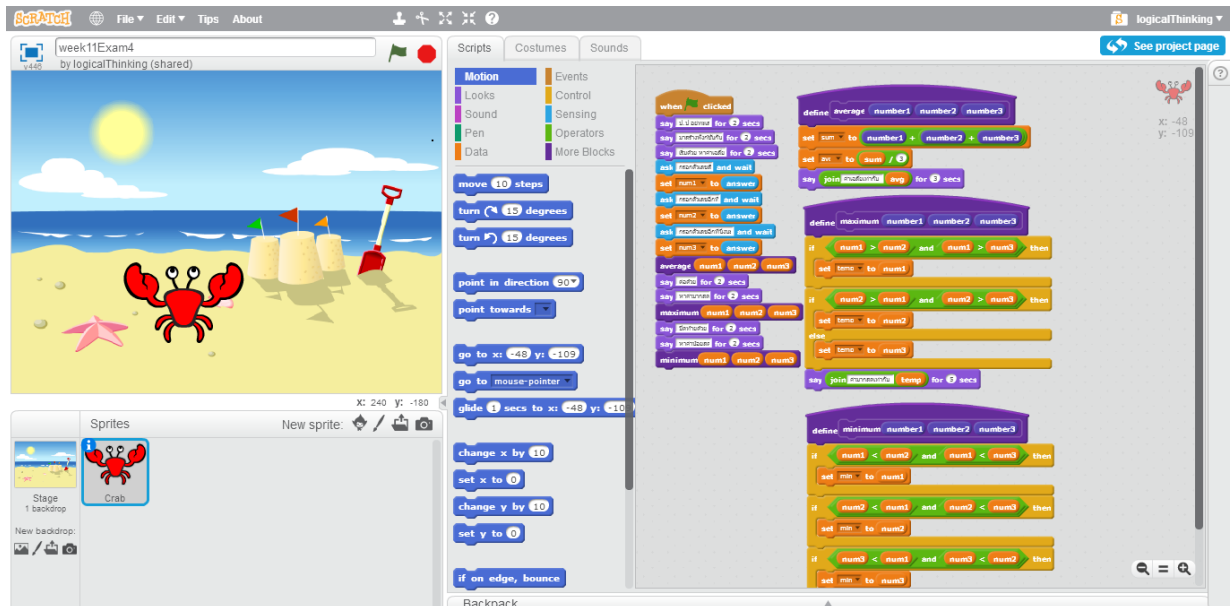


จากบล็อกคำสั่ง ถ้าค่าในตัวแปร $number2 < number1$ และ $number2 < number3$ แสดงว่าค่าในตัวแปร number2 คือค่าที่น้อยที่สุด จึงเก็บค่าในตัวแปร number2 ลงในตัวแปร min เพื่อเป็นผลลัพธ์ และ

บล็อกคำสั่ง if.....then ด้านล่าง เปรียบเทียบ $\text{number3} < \text{number1}$ และ $\text{number3} < \text{number2}$ เมื่อ number3 น้อยกว่าจำนวนอีกสองจำนวนจริง แสดงว่า number3 คือค่าที่น้อยที่สุด จึงเก็บค่าตัวแปร number3 ลงในตัวแปร min เป็นผลลัพธ์ให้โปรแกรมหลักแสดงผลต่อไป

สามารถชมตัวอย่างการหาค่ามากที่สุด สูงที่สุด ค่าเฉลี่ย และบล็อกคำสั่งได้ที่

<https://scratch.mit.edu/projects/111687899/>

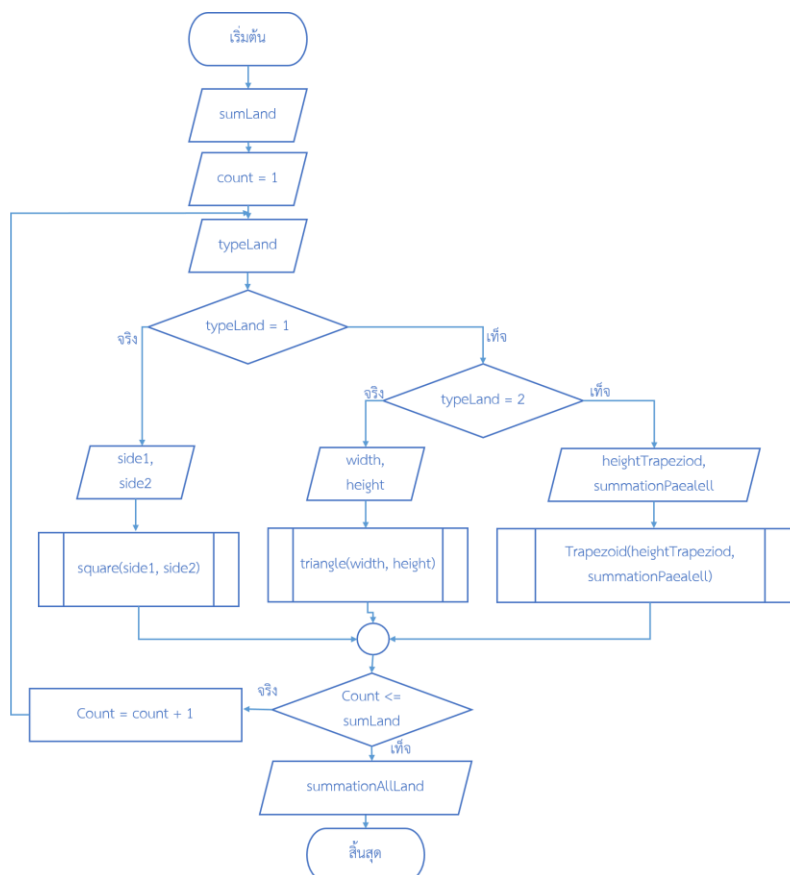


นอกจากตัดเครื่องอัตโนมัติแล้ว ยังมีเครื่องมืออุปกรณ์อื่น ๆ อีกที่สามารถเลือกรูปแบบการทำงาน และผลลัพธ์ได้ เช่น เครื่องซักผ้าที่บ้าน ไมโครเวฟ เครื่องพิมพ์เอกสาร แต่อุปกรณ์เหล่านั้น เราไม่ได้เป็นผู้เขียน ฟังก์ชัน และกำหนดการทำงานของฟังก์ชันเอง เราเป็นเพียงผู้ใช้นั้น ถ้ากลับกัน เราเป็นผู้สร้าง แล้วให้ผู้อื่นเป็นผู้ใช้บ้าง เราจะทราบถึงรายละเอียดการทำงานภายในฟังก์ชัน รวมถึงข้อมูลเข้า และข้อมูลผลลัพธ์ได้อย่างดีทีเดียว

ตัวอย่าง : ผมมีที่ดินเยอะมาก

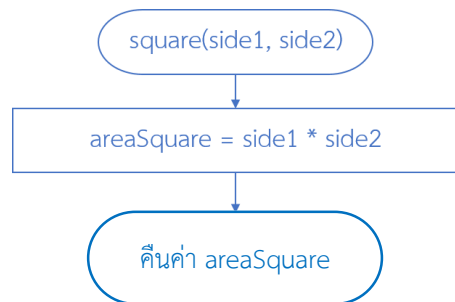
ลุงโชคเป็นชายวัยกลางคนที่มีฐานะร่ำรวยมาก ลุงโชคตระเวนซื้อที่ดินที่มีคนประกาศขายทั่วทุกแห่งหน จนลุงโชคจำได้ไม่หมดว่าลุงโชคซื้อที่ดินขนาดเท่าไร ไว้ที่ไหนบ้าง ลุงโชคจึงประกาศให้คนทั่วทุกแห่งหนช่วยลุงโชคสังเกตว่าพื้นที่ที่อยู่บริเวณใกล้เคียงกับพวกเขานั้น มีที่ดินของลุงโชคอยู่บ้างหรือเปล่า และลุงโชคเป็นคนที่เข้าถึงเทคโนโลยีใหม่ๆ มาก ลุงโชคจึงจะว่าจ้างให้นิสิตสร้างชิ้นงานที่สามารถคำนวณพื้นที่ที่ดินทั้งหมดที่คนอื่น ๆ พบเจอ และรวบรวมข้อมูลทั้งหมดมาให้ลุงโชค โดยลุงโชคใจดีให้ข้อมูลว่า ลุงโชคจำได้ว่าที่ดินของลุงโชคมี 3 รูปแบบคือที่ดินที่เป็นลักษณะสี่เหลี่ยมใด ๆ สามเหลี่ยม และสี่เหลี่ยมคางหมู

นิสิตตอบรับสร้างชิ้นงานให้กับลุงโชคโดยวางแผนการทำงานในรูปแบบของผังงานไว้ดังนี้



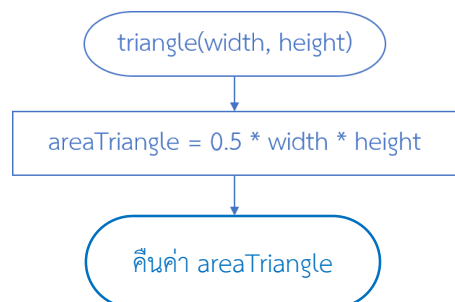
นิสิตวางแผนให้ผู้ใช้ที่เป็นผู้ค้นพบที่ดินของลุงโชค โดยที่อันดับแรกผู้ใช้จะต้องกรอกจำนวนที่ดินที่พบทั้งหมด จากนั้นผู้ใช้จะต้องกรอกรายละเอียดของที่ดินแต่ละแห่งที่พบ ถ้าผู้ใช้กดหมายเลข 1 แปลว่าผู้ใช้พบที่ดินรูปแบบสี่เหลี่ยมใด ๆ จะต้องกรอกความยาวทั้งสองของที่ดินรูปแบบสี่เหลี่ยมที่พบ ถ้าผู้ใช้กดหมายเลข 2 แปลว่าผู้ใช้พบที่ดินรูปแบบสามเหลี่ยม จะต้องกรอกความยาวฐานและความสูงของที่ดินที่พบ และถ้าผู้ใช้กดหมายเลข 3 แปลว่าผู้ใช้พบที่ดินรูปแบบสี่เหลี่ยมคางหมู จะต้องกรอกความสูงและผลบวกของด้านคู่ขนานของที่ดินที่พบ จากนั้นโปรแกรมจะทำการคำนวณพื้นที่แต่ละแห่งที่ผู้ใช้พบทั้งหมด และนำจำนวนผลลัพธ์ของที่ดินทุก ๆ แห่งมารวมกัน และแจ้งให้ลุงโชคทราบ โดยนิสิตวางแผนของการทำงานของที่ดินแต่ละรูปแบบเมื่อผู้ใช้กรอกหมายเลข 1 ถึงหมายเลข 3 ดังนี้

เมื่อผู้ใช้กรอกหมายเลข 1 แสดงว่าผู้ใช้พบที่ดินรูปแบบสี่เหลี่ยม



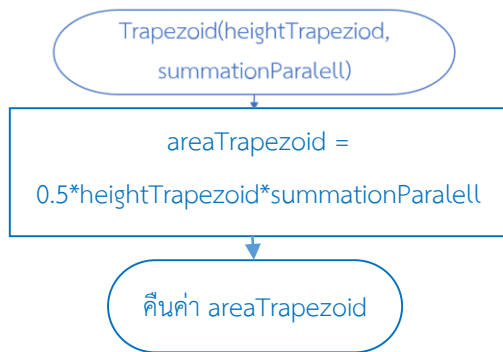
ฟังก์ชันการทำงานของการทำงานพื้นที่ที่ดินรูปแบบสี่เหลี่ยม จะนำค่าในตัวแปร `side1` และ `side2` มาคูณกัน จากนั้นจะนำผลลัพธ์ส่งให้โปรแกรมหลัก นำไปรวมกับที่ดินแห่งอื่น ๆ

เมื่อผู้ใช้กรอกหมายเลข 2 แสดงว่าผู้ใช้พบที่ดินรูปแบบสามเหลี่ยม



ฟังก์ชันการทำงานของการทำงานพื้นที่ที่ดินรูปแบบสามเหลี่ยม จะนำค่าในตัวแปร `width * height * 0.5` และเก็บผลลัพธ์ลงในตัวแปร `areaTriangle` เพื่อรอให้โปรแกรมหลักนำค่าไปรวมกับที่ดินแห่งอื่น ๆ

เมื่อผู้ใช้กรอกหมายเลข 3 แสดงว่าผู้ใช้พบที่ดินรูปแบบสี่เหลี่ยมคางหมู



ฟังก์ชันการทำงานของการทำงานหาพื้นที่ที่ดินรูปแบบสี่เหลี่ยมคางหมู จะนำค่าในตัวแปร heightTrapezoid * summationParalell * 0.5 และเก็บผลลัพธ์ลงในตัวแปร areaTrapezoid เพื่อรอให้โปรแกรมหลักนำค่าไปรวมกับที่ดินแห่งอื่น ๆ ที่ผู้ใช้พบ

จากตัวอย่างที่ดินของลุงโซนี่ สามารถทดสอบและศึกษาเพิ่มเติมได้ที่

<https://scratch.mit.edu/projects/113997467/>

