

บทที่ 8

เงื่อนไขตัดสินใจ

จากหลายบทเรียนที่ผ่านมาได้มีการกล่าวถึงการออกแบบขั้นตอนวิธีสำหรับการแก้ปัญหา หนึ่งในสิ่งสำคัญของการออกแบบขั้นตอนวิธีสำหรับการแก้ปัญหา คือ เงื่อนไขสำหรับการแก้ปัญหา เพราะบางครั้งการแก้ปัญหาก็ต้องใช้การตัดสินใจ เมื่อตัดสินใจแล้วจะต้องมีการดำเนินการต่อจากการตัดสินใจทุก ๆ ทางเลือก สำหรับทางเลือกที่มีเพียงสองทางเลือกก็เช่นกัน โดยสองทางเลือกในที่นี้จะอ้างอิงถึงทางเลือกที่เป็นเท็จ และทางเลือกที่เป็นจริง การดำเนินการหลังจากเลือกทางเลือกที่เป็นเท็จ และทางเลือกที่เป็นจริงจะแตกต่างกัน เงื่อนไขที่ทำให้เกิดทางเลือกที่เป็นเท็จและเป็นจริงมาจากเครื่องหมายทางคณิตศาสตร์ และการดำเนินการทางตรรกศาสตร์ โดยมีเครื่องหมายต่าง ๆ ดังนี้

เครื่องหมาย	ความหมาย
>	มากกว่า
<	น้อยกว่า
=	เท่ากับ
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ
!=	ไม่เท่ากับ

นอกจากเครื่องหมายดำเนินการทางคณิตศาสตร์แล้ว ยังสามารถใช้เครื่องหมายดำเนินการทางตรรกศาสตร์อย่าง and or not ร่วมด้วยได้อีก

สมมุติตัวอย่างเป็นการเล่นเกมเศรษฐี

ถ้าในเหตุการณ์ที่ตัวละครติดคุก

ถ้าเป็นจริง คือ ขณะนี้ติดคุกอยู่ จะต้องดำเนินการให้ตนเองออกมาจากคุกให้ได้

ถ้าเป็นเท็จ คือ ขณะนี้ไม่ได้ติดคุกอยู่ สามารถเดินต่อไปบนแผนที่ได้เลย

หรือตัวอย่างที่แสดงถึงทางเลือกสำหรับการตัดสินใจอื่น ๆ ดังตัวอย่างต่อไปนี้

ตัวอย่าง : น้อยกว่า มากกว่าหรือเท่ากับ

สมมติสถานการณ์ตัวอย่างในห้องเรียน ให้นายข้าวโพดออกมายืนหน้าชั้นเรียน จากนั้นให้นายข้าวโพดล้วงมือลงไปในการ์ด ซึ่งภายในการ์ดจะเป็นสลากตัวเลขหลากหลายจำนวน และถ้านายข้าวโพดหยิบสลากเลขที่น้อยกว่า 50 จะต้องเขียนข้อความว่า “น้อยกว่า 50” ลงบนกระดาษและชูขึ้น ถ้านายข้าวโพดหยิบได้สลากเลขที่มากกว่าหรือเท่ากับ 50 จะต้องเขียนข้อความว่า “มากกว่าหรือเท่ากับ 50” ลงบนกระดาษและชูขึ้น สมมติให้นายข้าวโพดหยิบสลากทั้งหมด 4 ครั้ง โดยแต่ละครั้งมีผลการหยิบสลากดังนี้

ครั้งที่ 1 หยิบได้สลากหมายเลข 20

ดังนั้น นายข้าวโพดจะต้องเขียนข้อความว่า “น้อยกว่า 50”
ลงบนกระดาษและชูขึ้น



ครั้งที่ 2 หยิบได้สลากหมายเลข 49.5

ดังนั้น นายข้าวโพดจะต้องเขียนข้อความว่า “น้อยกว่า 50” ลง
บนกระดาษและชูขึ้น

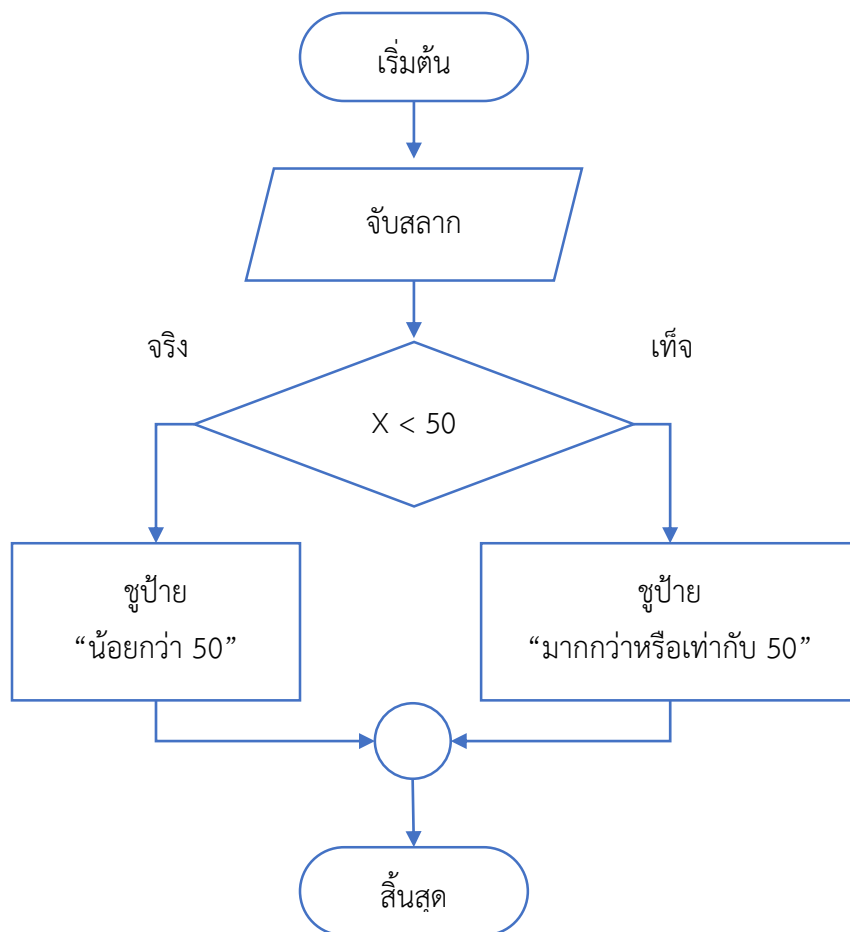


ครั้งที่ 3 หยิบได้สลากหมายเลข 50
ดังนั้น นายข้าวโพดจะต้องเขียนข้อความว่า “มากกว่าหรือ
เท่ากับ 50” ลงบนกระดาดและชูขึ้น



ครั้งที่ 4 หยิบได้สลากหมายเลข 50.0590
ดังนั้น นายข้าวโพดจะต้องเขียนข้อความว่า “มากกว่าหรือ
เท่ากับ 50” ลงบนกระดาดและชูขึ้น

จากตัวอย่างการหยิบสลากของนายข้าวโพด เงื่อนไขคือ “ <50 ” ถ้าเป็นจริง คือเลขในสลากน้อยกว่า 50 การดำเนินการคือ นายข้าวโพดต้องเขียนข้อความ “น้อยกว่า 50” ลงบนกระดาดและชูขึ้น แต่ถ้าเงื่อนไขเป็นเท็จ คือเลขในสลากมากกว่าหรือเท่ากับ 50 การดำเนินการคือ นายข้าวโพดจะต้องเขียนข้อความ “มากกว่าหรือเท่ากับ 50” ลงบนกระดาดและชูขึ้น จากตัวอย่างนี้สามารถเขียนผังงานได้โดยใช้สัญลักษณ์ Decision ดังนี้

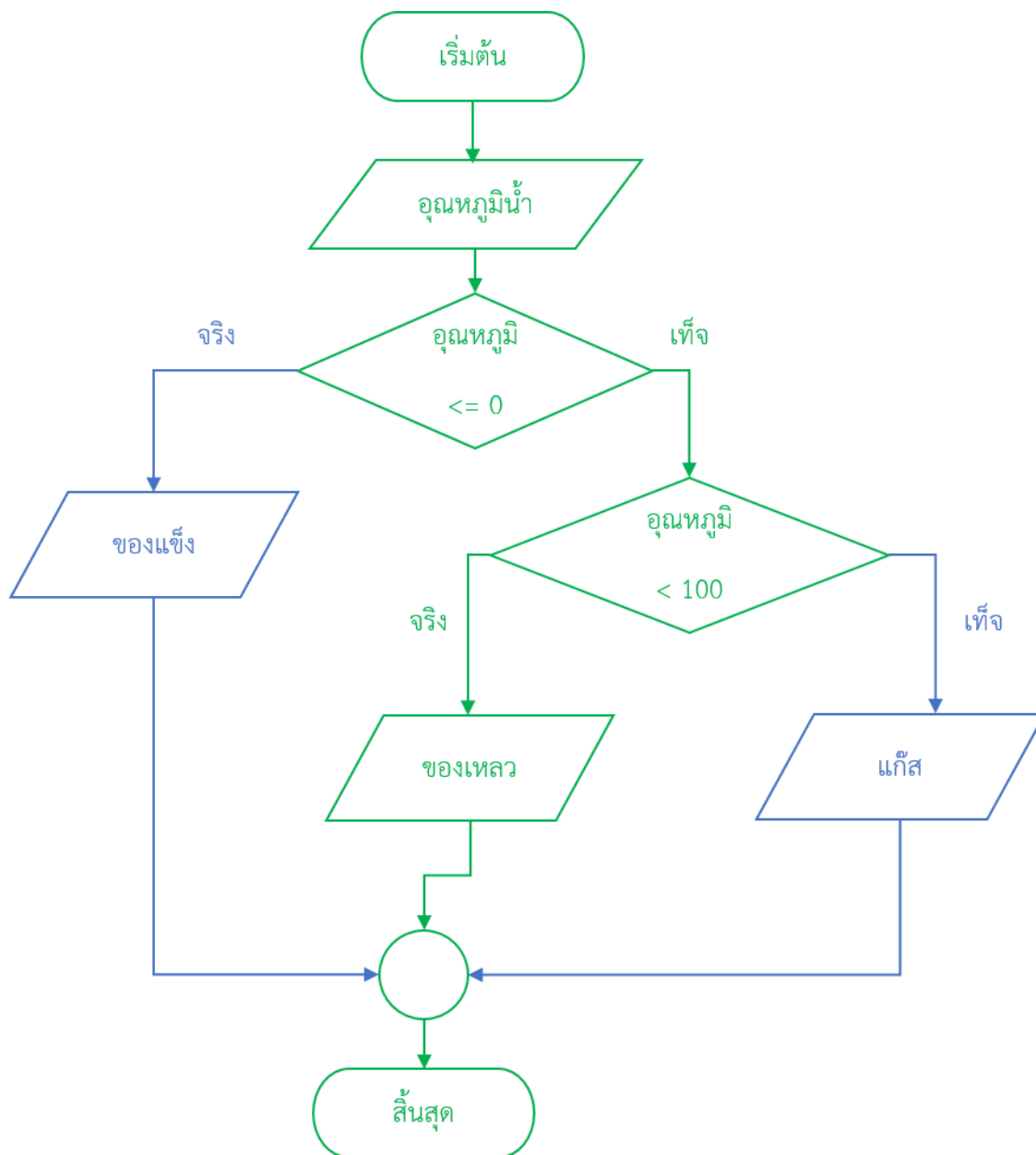


จากผังงาน สังเกตขั้นตอนการจับสลาก จะใช้สัญลักษณ์ Input/Output เพราะถือว่าการที่นายข้าวโพดหยิบสลากขึ้นมาจากกระป๋อง และอ่านด้วยสายตาของตนเองเป็นการรับข้อมูลเข้า บริเวณขั้นตอน $x < 50$ ใช้สัญลักษณ์ Decision เพราะเป็นการเปรียบเทียบจำนวนตัวเลขบนสลากกับตัวเลข 50 ว่าตัวเลขบนสลาคน้อยกว่าตัวเลข 50 หรือไม่ จากนั้นจะเป็นขั้นตอนการดำเนินการเขียนข้อความลงกระดาษ และทำการชูกระดาษนั้น จึงใช้สัญลักษณ์ Process เพื่อเขียนและแสดงข้อความ “น้อยกว่า 50” หรือ “มากกว่าหรือเท่ากับ 50”

ตัวอย่าง : สถานะของน้ำ

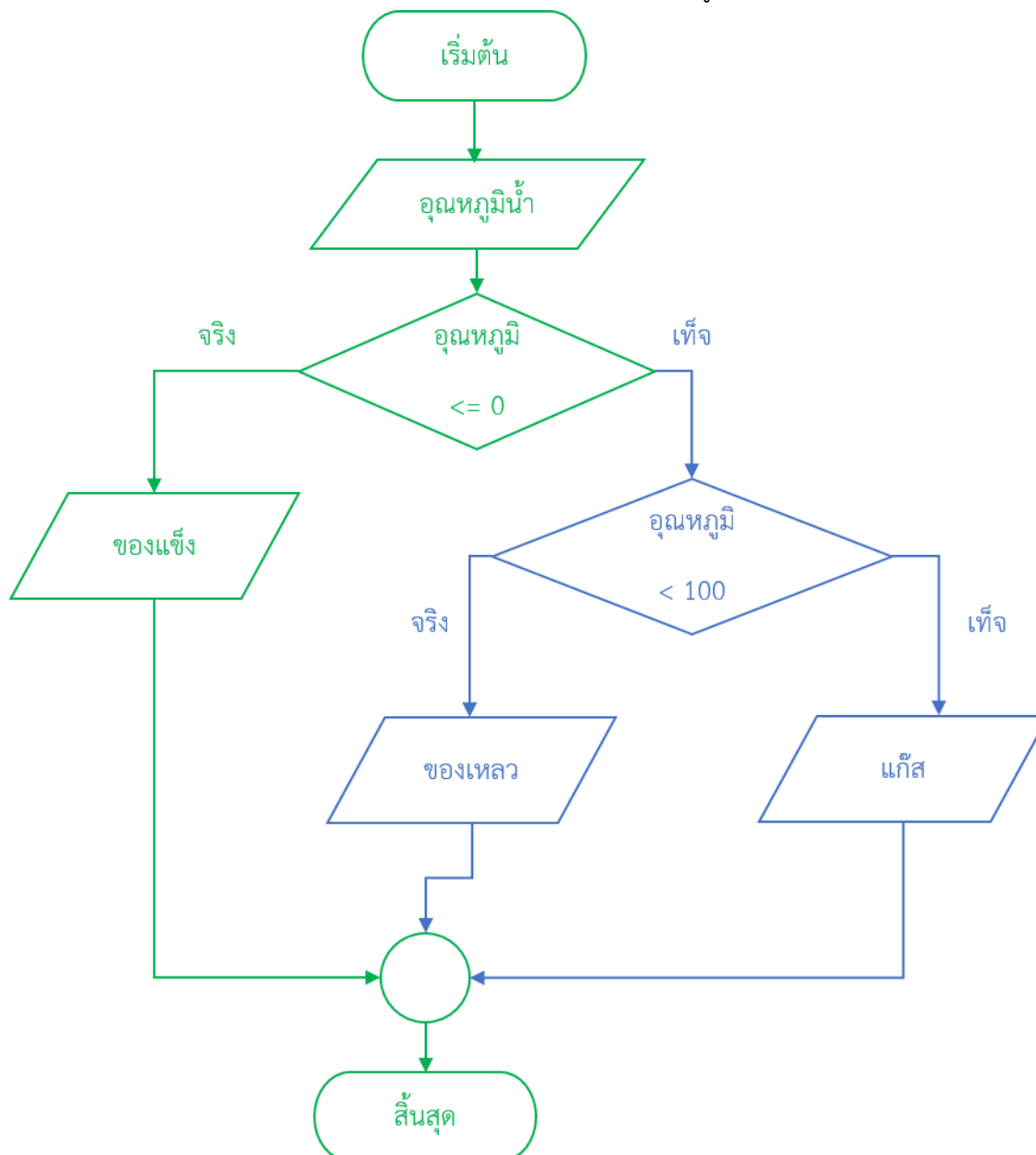
น้ำสามารถมีสถานะได้ 3 สถานะคือของแข็ง ของเหลว และแก๊ส โดยเงื่อนไขสำหรับการเปลี่ยนสถานะของน้ำคือ จะเป็นของแข็งเมื่ออุณหภูมิน้อยกว่าหรือเท่ากับ 0 องศา จะเป็นแก๊สเมื่ออุณหภูมิมากกว่าหรือเท่ากับ 100 องศา ส่วนน้ำที่มีอุณหภูมิมากกว่า 0 องศาแต่น้อยกว่า 100 องศา จะมีสถานะเป็นของเหลวหรือน้ำ เช่นเดิม เราจะทำการระบุสถานะให้กับน้ำที่อุณหภูมิต่าง ๆ ตามที่โจทย์กำหนด โดยมีอุณหภูมิต่าง ๆ ดังนี้

น้ำที่อุณหภูมิห้อง โดยอุณหภูมิห้องมาตรฐานอยู่ที่ 20-25 องศา ดังนั้นน้ำที่อุณหภูมิห้องจะมีอุณหภูมิอยู่ในช่วง 20-25 องศาเช่นกัน ดังนั้นน้ำอุณหภูมิห้องจึงมีสถานะเป็นของเหลวหรือน้ำ ตามเดิม โดยสามารถแสดงในรูปแบบผังงานได้ดังนี้



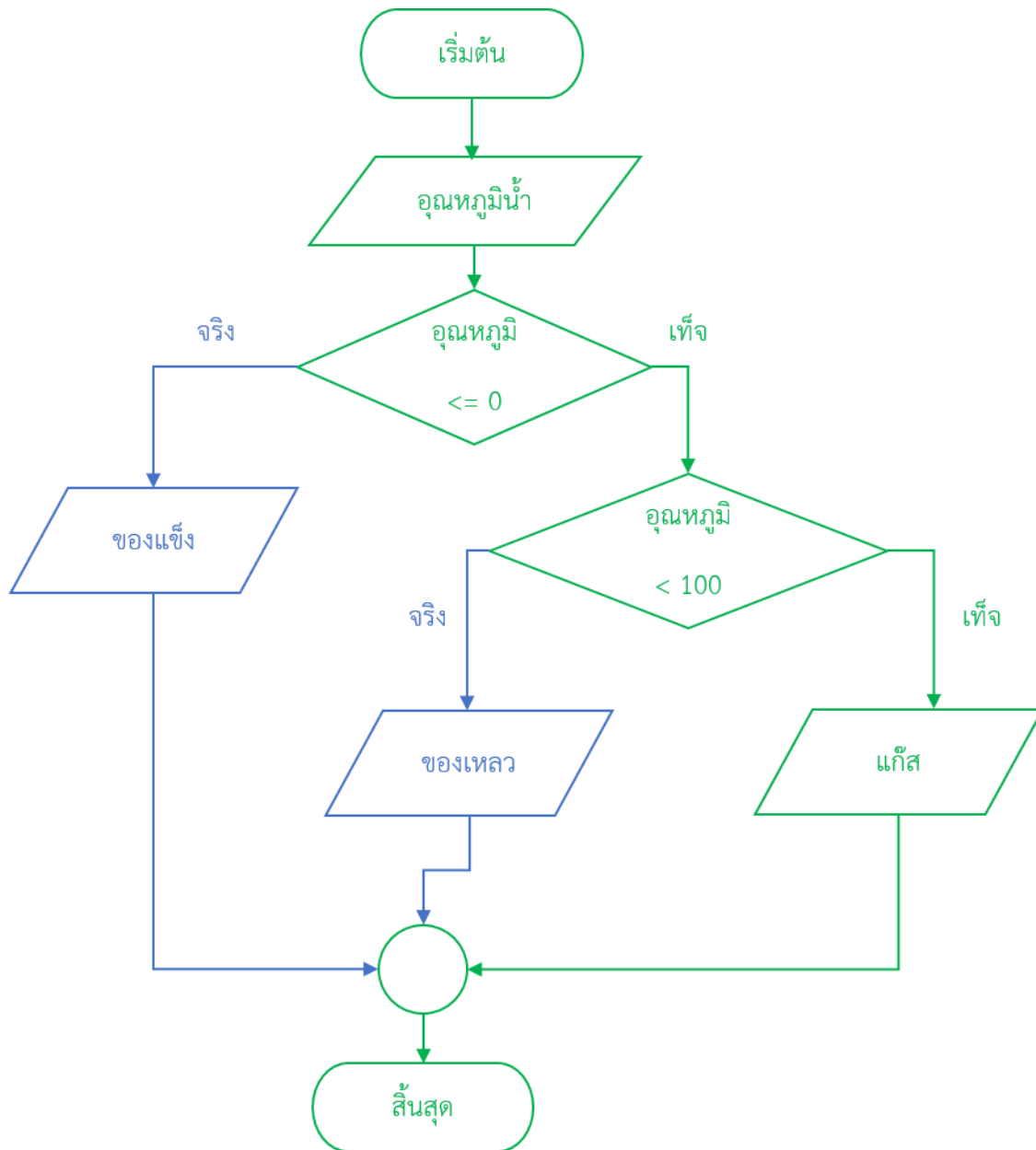
จากผังงาน ผังงานทั้งหมดแสดงให้เห็นถึงเงื่อนไขทั้งหมดในการระบุสถานะของน้ำ (กรอบและเส้นสีฟ้า) และส่วนที่ถูกดำเนินการเมื่อสถานะของน้ำเป็นอุณหภูมิห้อง คือ 20-25 องศา (กรอบและเส้นสีเขียว) เมื่อน้ำมีอุณหภูมิ 20-25 องศาเมื่อเข้าสู่สัญลักษณ์ Decision “อุณหภูมิ <= 0” ส่งผลให้เป็นเท็จ จึงดำเนินการต่อและเข้าเงื่อนไข “อุณหภูมิ <= 100” ส่งผลให้เป็นจริง ดังนั้น น้ำอุณหภูมิห้องจึงมีสถานะเป็น “ของเหลว”

ถ้าน้ำมีอุณหภูมิ 0 องศาเซลเซียส น้ำจะมีสถานะเป็นของแข็ง หรือน้ำแข็งนั่นเอง เพราะเมื่อน้ำมีอุณหภูมิน้อยกว่าหรือเท่ากับ 0 องศาเซลเซียส น้ำจะมีสถานะเป็นของแข็ง สามารถแสดงในรูปแบบผังงานได้ ดังนี้



จากผังงาน แสดงถึงเงื่อนไขทั้งหมดในการระบุสถานะของน้ำ (เส้นและข้อความสีฟ้า) และแสดงเส้นทางการดำเนินงานโดยที่อุณหภูมิของน้ำเป็น 0 องศาเซลเซียส (เส้นและข้อความสีเขียว) ด้วย เมื่อน้ำมีอุณหภูมิ 0 องศาเซลเซียสดำเนินการมาถึงสัญลักษณ์ Decision “อุณหภูมิ <= 0” แล้วส่งผลให้เป็นจริง ดังนั้นจึงดำเนินการต่อที่สัญลักษณ์ Input/Output เพื่อแสดงผลรหัสสถานะของน้ำว่าเป็น “ของแข็ง”

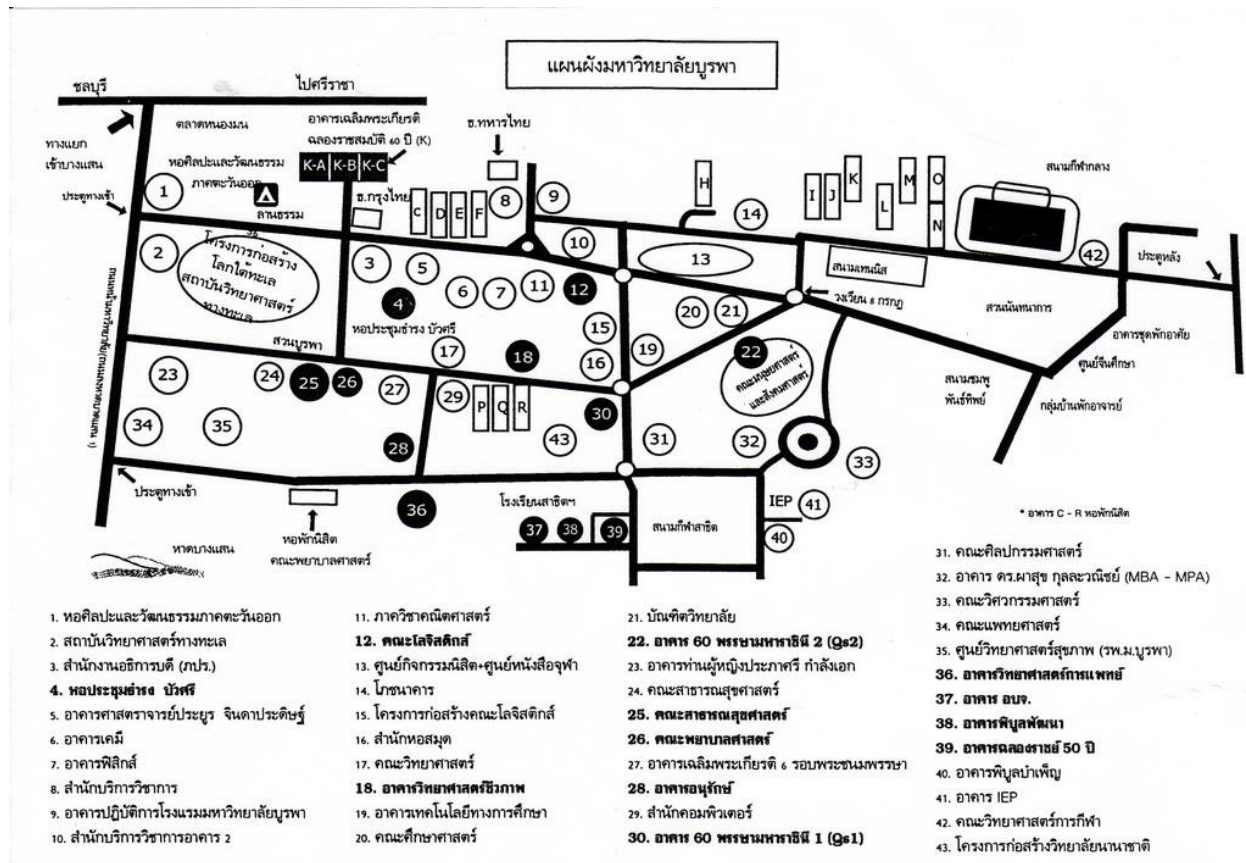
ถ้าน้ำมีอุณหภูมิ 100 องศาเซลเซียส ตามเงื่อนไขแล้ว จะทำให้สามารถระบุสถานะของน้ำอุณหภูมิ 100 องศาเซลเซียสได้ว่าเป็น แก๊ส ซึ่งสามารถแสดงแทนได้ด้วยผังงาน ดังนี้



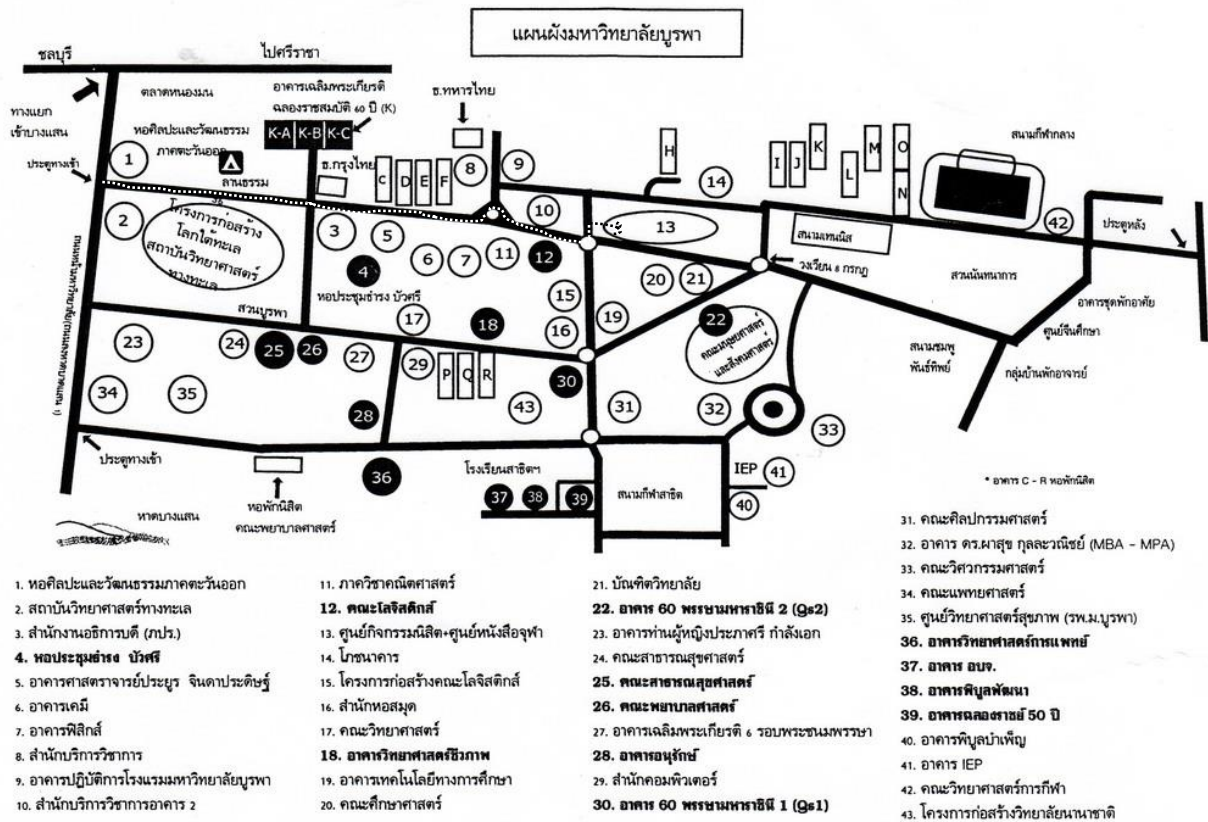
จากผังงาน แสดงถึงการดำเนินการเพื่อระบุสถานะของน้ำที่อุณหภูมิ 100 องศาเซลเซียส (เส้นและข้อความสีเขียว) โตรเริ่มจากสัญลักษณ์ Input/Output “อุณหภูมิน้ำ” ลงมาที่สัญลักษณ์ Decision “อุณหภูมิ <= 0” ส่งผลให้เป็น เท็จ จึงดำเนินการต่อที่สัญลักษณ์ Decision “อุณหภูมิ < 100” ส่งผลเป็นเท็จ จึงดำเนินการต่อที่สัญลักษณ์ Input/Output “แก๊ส” เป็นอันระบุสถานะของน้ำอุณหภูมิ 100 องศาเซลเซียสได้ว่าเป็นแก๊ส

ตัวอย่าง : เดินทางไปกองกิจ

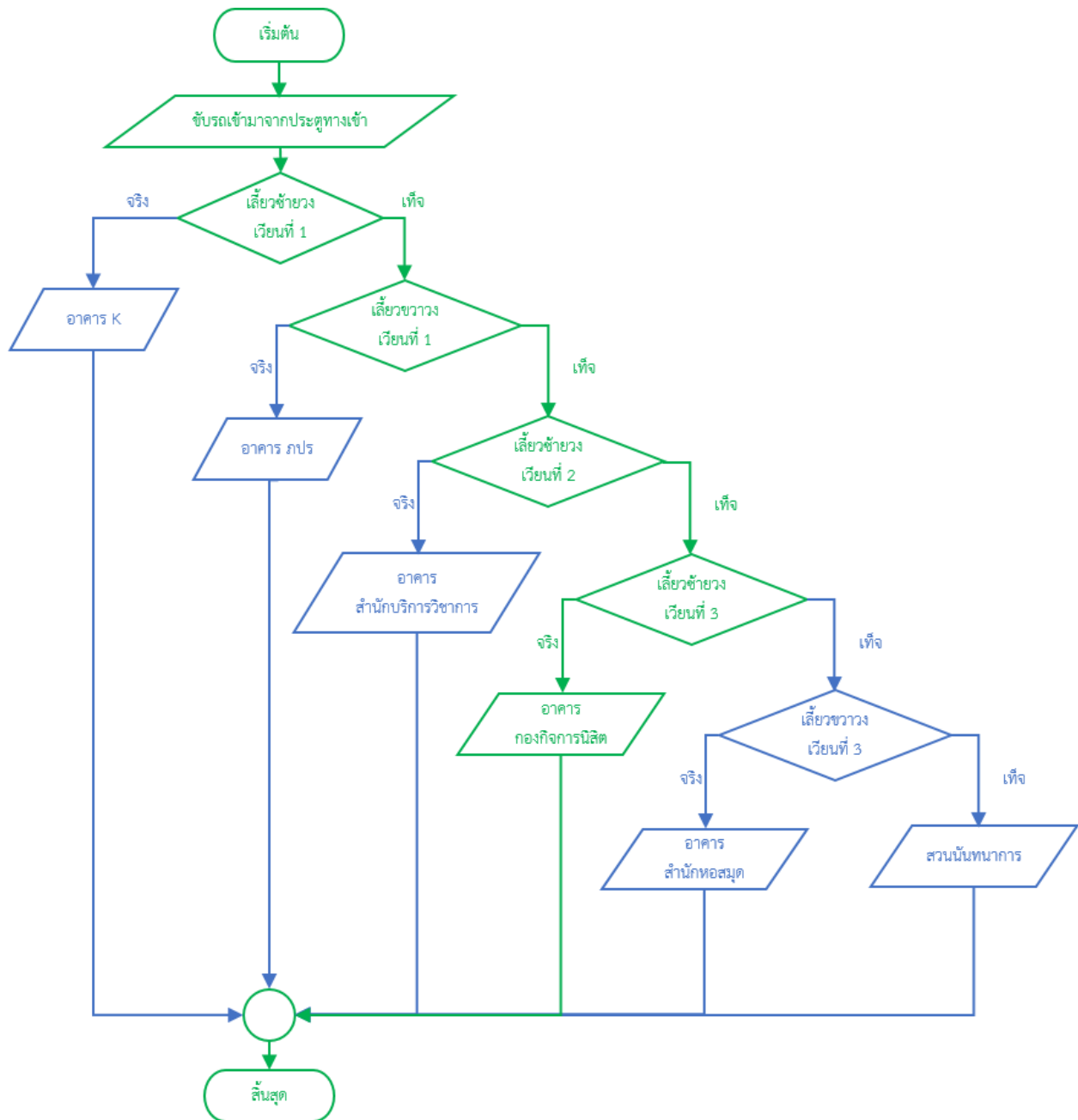
หากนิสิตเคยมาสอบสัมภาษณ์หรือเข้ามาซื้อเครื่องแบบนิสิตภายในมหาวิทยาลัย นิสิตอาจจะคุ้นชินกับแผนที่ด้านล่างนี้ ซึ่งเป็นแผนที่ภายในมหาวิทยาลัยบูรพาตนเอง



หากนิสิตจะต้องเดินทางจาก ประตูทางเข้า (หมายเลข 1) ไปยัง ศูนย์กิจกรรมนิสิต+ศูนย์หนังสือจุฬา (หมายเลข 13) บนแผนที่ โดยบังคับให้ใช้เส้นทางหลัก ตามเส้นประบนแผนที่ ทำให้ต้องผ่านวงเวียนหรือสี่แยก 3 วงเวียน ก่อให้เกิดเงื่อนไขการตัดสินใจเมื่อเจอวงเวียนว่าจะวนรถเพื่อไปยังเส้นทางไหนต่อไปให้ถึงกองกิจการนิสิต นั่นคือการใช้เงื่อนไขเพื่อตัดสินใจ



เมื่อถึงวงเวียนที่ 1 (บริเวณอาคารหมายเลข 3 และธนาคารกรุงไทย) มีทางเลือก ตรงไป เลี้ยวซ้าย เลี้ยวขวา สำหรับวงเวียนนี้จะต้องเลือก ตรงไป เมื่อตรงมาแล้ววงเวียนต่อไปที่จะพบคือวงเวียนที่ 2 (บริเวณอาคารหมายเลข 10 และ 11) มีทางเลือกตรงไป และเลี้ยวซ้าย จะต้องเลือก ตรงไป อีกครั้ง ต่อไปจะพบวงเวียนที่ 3 (บริเวณอาคารหมายเลข 13) มีเส้นทางที่สามารถตรงไป เลี้ยวซ้าย และเลี้ยวขวาได้ แต่เนื่องจากจุดประบั้งคับให้เลี้ยว จึงต้องเลี้ยวซ้าย เพื่อจอดรถบริเวณหน้าอาคาร เป็นอันถึงเป้าหมายของการเดินทางในที่สุด จากเงื่อนไขของทั้ง 3 วงเวียน และทางแยกต่าง ๆ ของแต่ละวงเวียน สามารถแสดงในรูปของผังงานได้ ดังนี้



จากผังงาน กรอบและข้อความสีฟ้าคือเส้นทางทั้งหมดที่เป็นไปได้เมื่อเลี้ยวแยกต่าง ๆ ในวงเวียน ส่วนกรอบและข้อความสีเขียวคือส่วนที่เลือกและตัดสินใจเลี้ยวเมื่อถึงแต่ละวงเวียนเพื่อไปยังอาคารกองกิจการนิสิต เมื่อเดินทางเข้ามาถึงวงเวียนที่ 1 ใช้สัญลักษณ์ Decision เพื่อเลือกว่า “เลี้ยวซ้ายวงเวียนที่ 1” จริงหรือไม่ เนื่องจากเป็นเท็จ เพราะไม่ได้เลี้ยวซ้ายที่วงเวียนที่ 1 จึงใช้สัญลักษณ์ Decision เพื่อเลือกต่อว่า “เลี้ยวขวาวงเวียนที่ 1” จริงหรือไม่ ปรากฏว่าไม่จริง เพราะไม่ได้เลี้ยวขวาที่วงเวียนที่ 1 จึงเดินทางต่อไปยังวงเวียนที่ 2 และใช้สัญลักษณ์ Decision ถามว่า “เลี้ยวซ้ายวงเวียนที่ 2” ปรากฏว่าไม่ได้เลี้ยวซ้ายที่วงเวียนที่ 2 ส่งผลให้เป็นเท็จ จึงเหลือ 1 ทางเลือก (วงเวียนที่ 2 มีทางตรง กับเลี้ยวซ้าย) คือตรงไป และเดินทางต่อไปยังวงเวียนที่ 3 ใช้สัญลักษณ์ Decision แสดง

ทางเลือกกว่า “เลี้ยวซ้ายวงเวียนที่ 3” ส่งผลให้เป็นจริง เพราะโจทย์กำหนดให้เลี้ยวซ้ายที่วงเวียนนี้ ทำให้เดินทางถึงอาคารกองกิจการนิสิตในที่สุด เป็นอันจบการเดินทาง พบว่าผังงานมีความซับซ้อนมากขึ้น เพราะมีทางเลือกหลักหลายทาง และแต่ละทางเลือกหลักก็มีทางเลือกรองอยู่อีกหนึ่งชั้น ทำให้ผังงานมีสัญลักษณ์ Decision ซ้อนกันจำนวนมาก

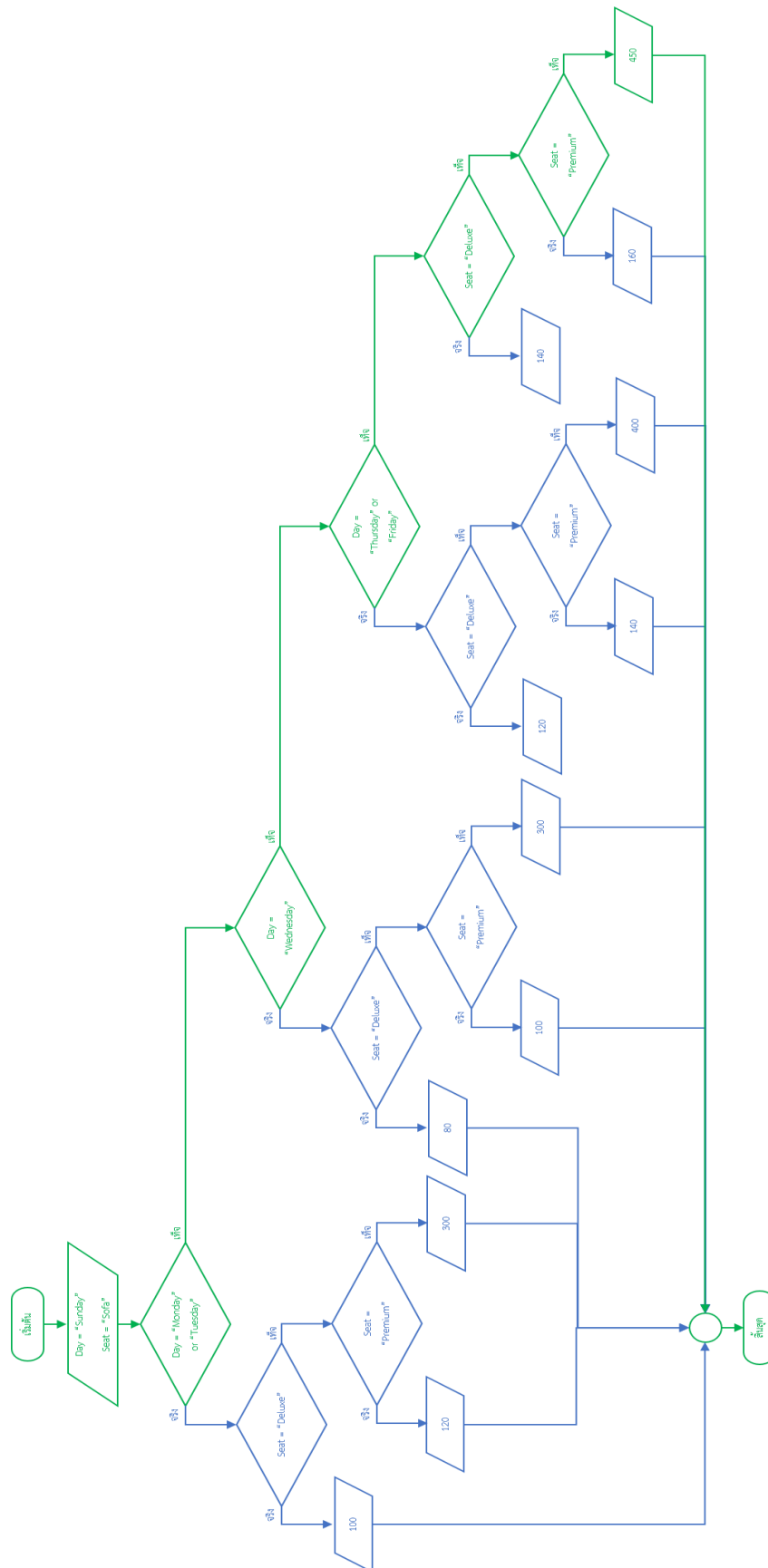
ตัวอย่างต่อไป อาจจะเหมาะสำหรับผู้ที่ชื่นชอบในการชมภาพยนตร์ และผู้ที่ชื่นชอบเป็นอย่างมาก หากการชมภาพยนตร์นั้น เสียค่าใช้จ่ายที่เหมาะสม ไม่แพงมากเกินไป

ตัวอย่าง : ดูหนังกันเถอะ

ถ้าสมมติให้ตารางแสดงราคาที่นั่งในโรงภาพยนตร์แห่งหนึ่งเป็นดังนี้

Day	Deluxe	Premium	Sofa
Monday	100.-	120.-	300.-
Tuesday	100.-	120.-	300.-
Wednesday	80.-	100.-	300.-
Thursday	120.-	140.-	400.-
Friday	120.-	140.-	400.-
Saturday	140.-	160.-	450.-
Sunday	140.-	160.-	450.-

ถ้าน้องแคะต้องการชมภาพยนตร์ในวันอาทิตย์ ที่นั่งแบบโซฟา น้องแคะจะต้องซื้อตั๋วในราคาเท่าไร สังเกตจากตารางแสดงราคาแล้วสามารถตอบได้ทันทีเลยว่าน้องแคะจะต้องซื้อตั๋วในราคา 450 บาท แต่ถ้าใช้ผังงานแสดงการตัดสินใจซื้อตั๋วของน้องแคะในวันอาทิตย์และที่นั่งแบบโซฟา จะสามารถเขียนผังงานได้ ดังต่อไปนี้

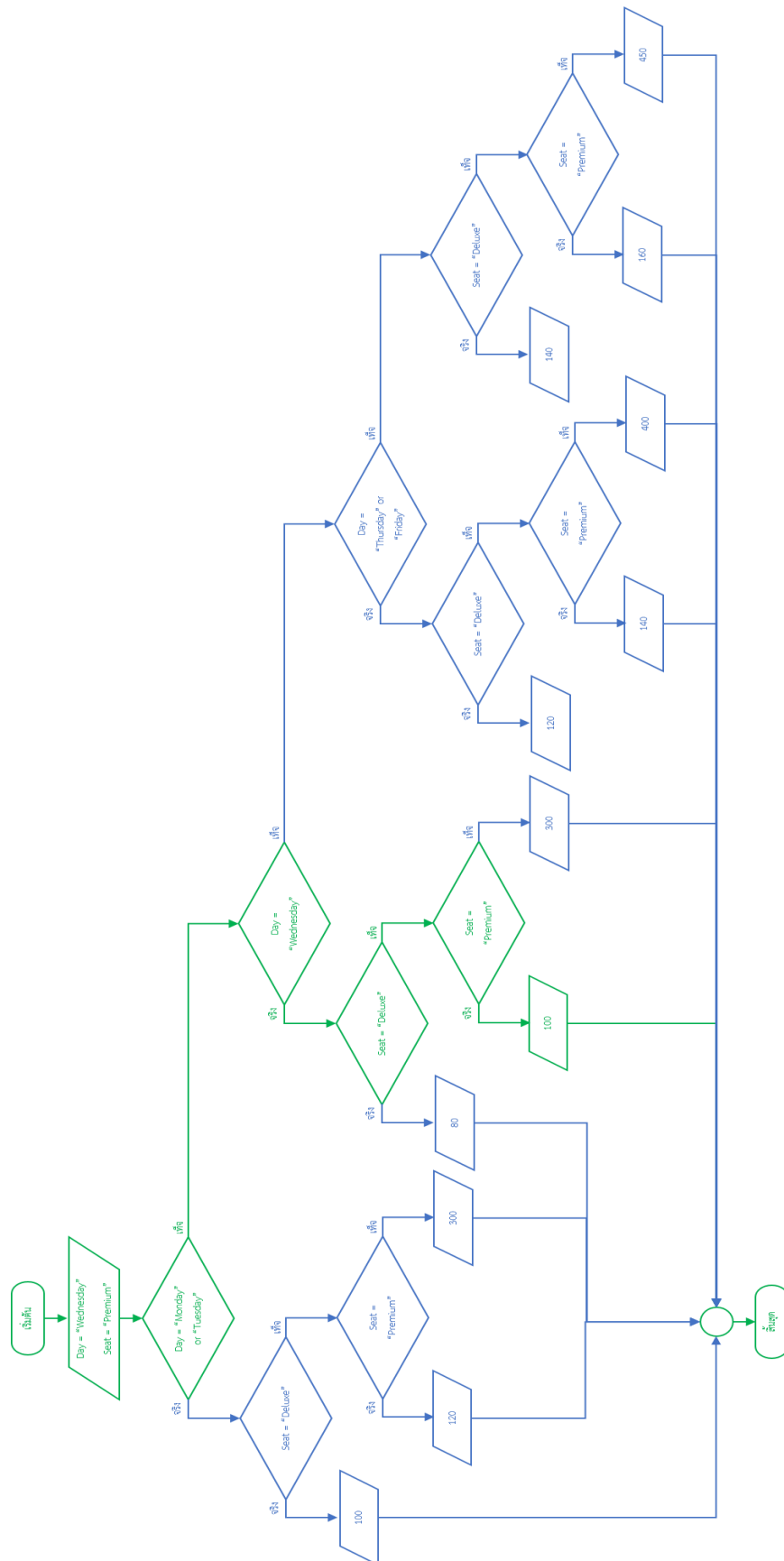


จากผังงานแสดงการซื้อตั๋วชมภาพยนตร์ในวันอาทิตย์ ที่นั่งแบบโซฟา (กรอบและข้อความสีเขียว) เริ่มต้นจะรับค่าวัน เป็นวันพุธ และที่นั่งเป็นโซฟาที่สัญลักษณ์ Input/Output “Day = Sunday” Seat = “Sofa” ต่อด้วยสัญลักษณ์ decision “Day = “Monday” or “Tuesday”” เพื่อเป็นทางเลือกกว่าวันที่รับเข้ามาเป็นวันจันทร์หรือวันอังคารหรือไม่ ทำให้ผลลัพธ์เป็นเท็จ เพราะไม่ใช่ทั้งวันจันทร์และวันอังคาร จึงดำเนินการต่อไปที่สัญลักษณ์ Decision “Day = “Wednesday”” แต่ก็ส่งผลให้เป็นเท็จอีก เพราะไม่ใช่วันพุธ จึงดำเนินการต่อไปที่สัญลักษณ์ Decision “Day = “Thursday” or “Friday”” เพื่อตรวจสอบว่าเป็นวันพฤหัสบดีหรือศุกร์หรือไม่ ส่งผลให้เป็นเท็จเมื่อถึงสัญลักษณ์นี้วันที่เหลือและมีโอกาสที่จะเป็นไปได้มีเพียง 2 วันคือวันเสาร์ และวันอาทิตย์ จึงดำเนินการต่อไปที่สัญลักษณ์ Decision “Seat = “Deluxe”” เพื่อทำการเลือกที่นั่งได้เลย และเมื่อถึงสัญลักษณ์นี้แล้วก็ส่งผลให้เป็นเท็จ เพราะไม่ต้องการที่นั่งแบบ Deluxe จึงทำงานต่อไปที่สัญลักษณ์ Decision “Seat = “Premium”” และส่งผลเป็นเท็จอีกครั้งเพราะไม่ต้องการที่นั่งแบบ Premium เมื่อดำเนินการถึงจุดนี้ที่นั่งจะเหลือเพียงรูปแบบเดียวคือ Sofa จึงทำงานต่อไปที่สัญลักษณ์ Input/Output “450” เพื่อแสดงราคาของตั๋วที่นั่งและต้องจ่าย

แล้วถ้าน้องแกะต้องการเปลี่ยนวันสำหรับชมภาพยนตร์ และเปลี่ยนรูปแบบที่นั่งด้วย เพราะน้องแกะเห็นว่าการชมภาพยนตร์ในวันอาทิตย์ ที่นั่งรูปแบบโซฟานั้นราคาสูงเกินกว่าที่น้องแกะจะรับไหว น้องแกะจึงปรึกษาน้องกลอง น้องกลองจึงแนะนำว่าให้น้องแกะชมภาพยนตร์ในวันพุธ เพราะจะมีราคาตั๋วถูกกว่าวันอื่นๆ ถ้าน้องแกะอยากจะประหยัดมาก ๆ น้องกลองก็บอกให้น้องแกะเลือกที่นั่งแบบ Deluxe แต่ถ้าน้องแกะจะสามารถจ่ายค่าตั๋วในราคาที่สูงกว่าเล็กน้อยเพื่อนั่งชมภาพยนตร์ได้อย่างสบาย ไม่เมื่อยคอ ก็แนะนำให้น้องแกะเลือกที่นั่งเป็นแบบ Premium น้องแกะจึงเชื่อคำแนะนำของน้องกลองและไม่อยากเมื่อยคอ น้องแกะจึงจะซื้อตั๋วชมภาพยนตร์ในวันพุธและที่นั่งรูปแบบ Premium

ถ้าเขียนผังงานการเลือกซื้อตั๋วชมภาพยนตร์ในวันพุธ ที่นั่งรูปแบบ Premium จะสามารถเขียนได้ดังต่อไปนี้

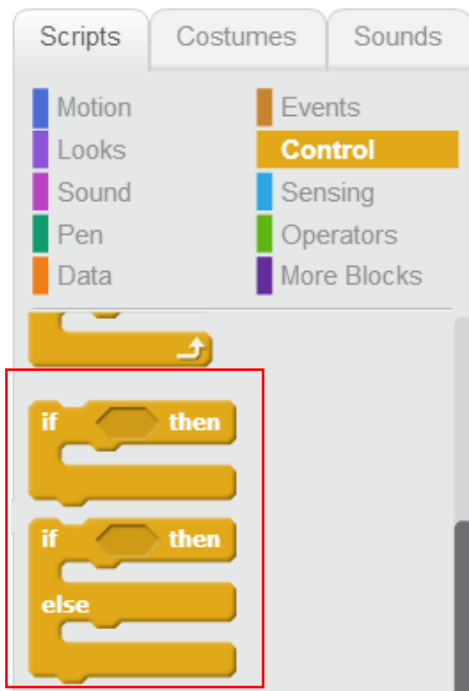




จากผังงานเมื่อเลือกชมภาพยนตร์วันพุธ ที่นั่งแบบ Premium หลังจากทราบวัน และรูปแบบที่นั่งแล้ว จึงดำเนินการต่อไปที่สัญลักษณ์ Decision “Day = “Monday” or “Tuesday”” ปรากฏว่าเป็นเท็จเพราะต้องการชมภาพยนตร์วันพุธ จึงดำเนินการต่อไปที่สัญลักษณ์ Decision “Day = “Wednesday”” และได้ผลลัพธ์เป็นจริง เพราะต้องการชมภาพยนตร์วันพุธ จากนั้นทำงานต่อไปที่สัญลักษณ์ Decision Seat = ““Deluxe”” แต่ส่งผลเป็นเท็จ เพราะต้องการนั่งที่นั่งแบบ Premium จึงทำงานต่อไปที่สัญลักษณ์ Decision “Seat = “Premium”” ทำให้ผลลัพธ์เป็นจริง จึงทำงานต่อสัญลักษณ์ Input/Output “100” เป็นการแสดงค่าตัวชมภาพยนตร์ที่นั่งเกาะ ต้องจ่ายเมื่อต้องการชมภาพยนตร์วันพุธ ที่นั่งแบบ Premium

เมื่อสามารถเข้าใจการเขียนผังงานและการทำงานแบบมีเงื่อนไขแล้ว ต่อไปจะเป็นการนำผังงานที่เขียนร่างเรียบร้อยแล้วมาสร้างเป็นผลงานที่สามารถเข้าใจได้มากยิ่งขึ้นด้วยโปรแกรม Scratch โดยจะใช้ตัวอย่างแรกคือ รับคะแนนและตัดสินว่าสอบผ่านหรือสอบตก ตัวอย่างที่สองคือ รับคะแนนและแสดงเกรดที่ผู้เรียนได้รับ

สร้างชิ้นงานด้วยโปรแกรม Scratch



ก่อนการสร้างชิ้นงาน ขอแนะนำบล็อกคำสั่งที่เกี่ยวข้องกับการสร้างชิ้นงานโดยมีเงื่อนไขก่อน ดังนี้

บล็อกคำสั่งที่เกี่ยวข้องกับการทำงานแบบมีเงื่อนไขจะอยู่ในบล็อกคำสั่งหมวด Control คือบล็อกคำสั่ง if.....then และบล็อกคำสั่ง if.....then.....else..... ความแตกต่างของบล็อกคำสั่งทั้งสองคือ



if.....then..... เหมาะกับใช้งานกับเงื่อนไขที่มีเพียงเงื่อนไขเดียวเท่านั้น และสามารถนำบล็อกคำสั่งนี้ต่อกันได้

หลายบล็อก เมื่อมีเงื่อนไขเดียว ๆ หลายเงื่อนไข โดยจะเข้าทำงานเพื่อตรวจสอบค่าความจริงทุก ๆ บล็อกคำสั่งที่ต่อกันอยู่

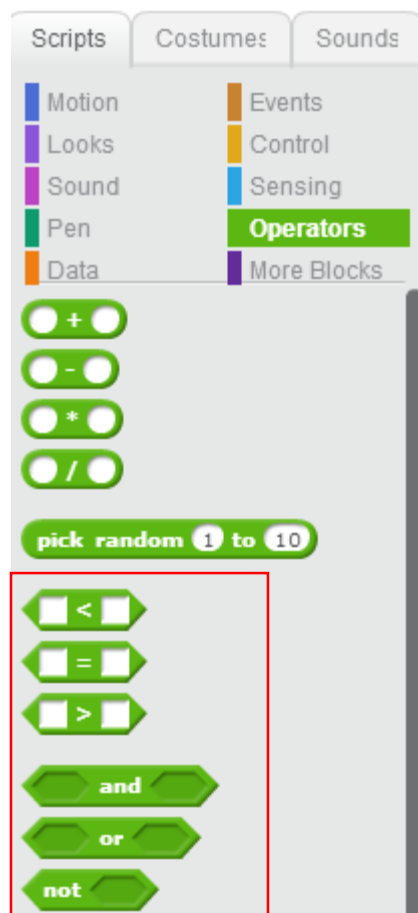


if.....then.....else..... เหมาะกับการทำงานที่มีเงื่อนไข 2

เงื่อนไขหรือมากกว่าขึ้นไป โดยจะทำงานในส่วนของ

if.....then..... ก่อนและถ้าการตรวจสอบผลลัพธ์ในส่วนนี้เป็นเท็จ
 จึงจะทำงานต่อที่บล็อกคำสั่งภายในส่วน else แต่ถ้าผลการตรวจสอบ
 ในส่วน if.....then..... เป็นจริงจะดำเนินการต่อที่บล็อกคำสั่งอื่น ๆ
 ในส่วน if.....then..... และจะไม่ดำเนินการภายในส่วนของ else
 เลย

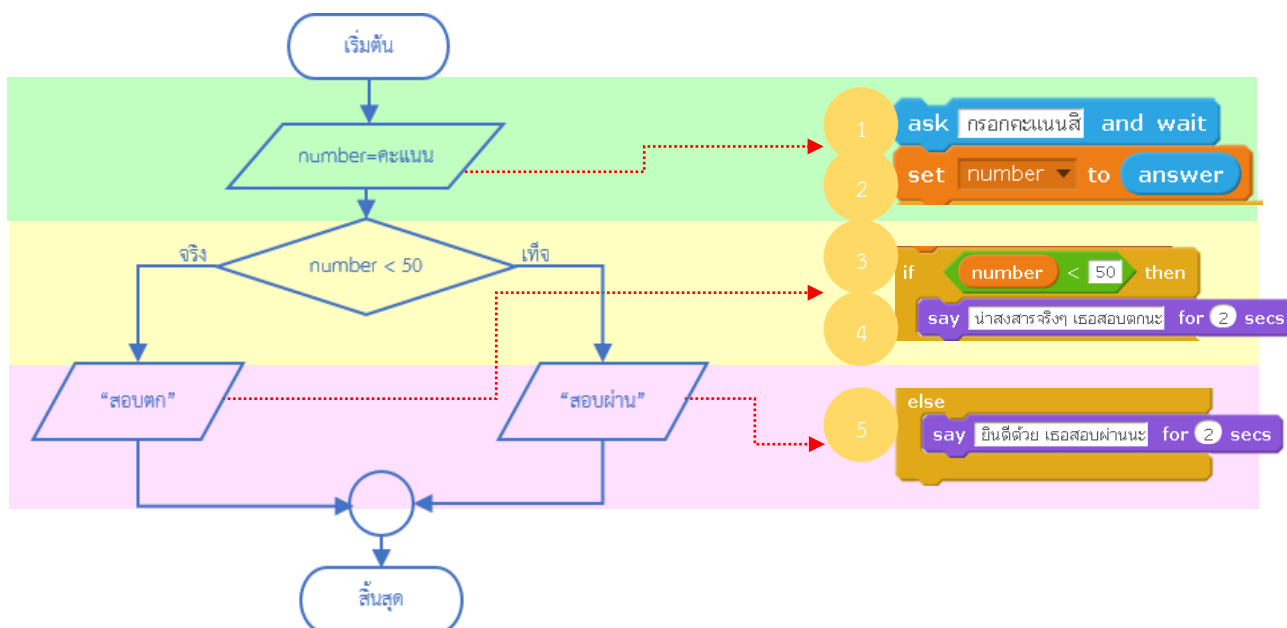
เมื่อสังเกตบล็อกคำสั่งทั้งสองแล้วจะพบว่าระหว่าง if.... และthen จะมีช่องว่างอยู่ ช่องว่างนี้ไว้สำหรับวาง
 บล็อกคำสั่งทางตรรกศาสตร์อย่าง and, or และ not และคืนค่าผลลัพธ์เป็นจริงหรือเป็นเท็จ บล็อกคำสั่งที่
 สามารถวางในช่วงว่างนี้ได้จะอยู่ในหมวดบล็อกคำสั่ง Operators ดังนี้



บล็อกคำสั่ง and จะให้ผลลัพธ์เป็นจริงเมื่อเงื่อนไขด้านหน้าและด้านหลัง
 เป็นความจริงทั้งคู่เท่านั้น ส่วนกรณีอื่น ๆ จะคืนค่าผลลัพธ์เป็นเท็จทั้งหมด
 บล็อกคำสั่ง or จะให้ผลลัพธ์เป็นจริงเมื่อเงื่อนไขด้านหน้าและเงื่อนไข
 ด้านหลังมีอย่างน้อย 1 เงื่อนไขเป็นจริงหรือเป็นจริงทั้งสองเงื่อนไข ส่วน
 กรณีที่ทั้งเงื่อนไขด้านหน้าและเงื่อนไขด้านหลังเป็นเท็จจะให้ผลลัพธ์เป็น
 เท็จ และบล็อกคำสั่ง not จะคืนค่าผลลัพธ์เป็นค่าตรงข้ามกับเงื่อนไขที่อยู่
 ด้านหลัง ถ้าเงื่อนไขที่อยู่ด้านหลังเป็นเท็จจะคืนค่าผลลัพธ์เป็นจริง ถ้า
 เงื่อนไขที่อยู่ด้านหลังเป็นจริงจะคืนค่าผลลัพธ์เป็นเท็จ

สังเกตบริเวณด้านหน้าและด้านหลังของบล็อกคำสั่งand.....
 และor..... มีช่องว่างสำหรับวางบล็อกคำสั่ง and or not หรือ
 บล็อกคำสั่ง > = < ก็ได้ โดยบล็อกคำสั่ง < จะเป็นจริงเมื่อค่าด้านหน้า
 น้อยกว่าค่าด้านหลัง บล็อกคำสั่ง = จะเป็นจริงเมื่อค่าด้านหน้าเท่ากับค่า
 ด้านหลัง และบล็อกคำสั่ง > จะเป็นจริงเมื่อค่าด้านหน้ามากกว่าค่า
 ด้านหลัง และบล็อกคำสั่งทั้ง 6 นี้สามารถนำมาวางซ้อนกันได้หลายชั้นอีก
 ด้วย

ตัวอย่างการสร้างชิ้นงานด้วยโปรแกรม Scratch ตัวอย่างแรกนี้จะแสดงถึงการรับค่าจากผู้ใช้เป็นตัวเลข ตัวเลขนี้คือคะแนนของผู้ใช้นั่นเอง ถ้าตัวเลขน้อยกว่า 50 จะแจ้งผู้ใช้งานว่าผู้ใช้สอบตก แต่ถ้าตัวเลขมากกว่าหรือเท่ากับ 50 จะแสดงข้อความแจ้งผู้ใช้งานว่าผู้ใช้สอบผ่าน ตัวอย่างนี้สามารถเขียนผังงานได้ ดังนี้



จากแผนภาพ เริ่มจากรับคะแนนของผู้ใช้เก็บลงในตัวแปร number ที่สัญลักษณ์ Input/Output และดำเนินการต่อที่สัญลักษณ์ Decision ถ้า number น้อยกว่า 50 จริงจะแสดงข้อความว่า “สอบตก” แต่ถ้า number ไม่น้อยกว่า 50 ส่งผลให้เป็นเท็จจะแสดงข้อความ “สอบผ่าน” เป็นอันจบการทำงาน และการทำงานของบล็อกคำสั่ง Scratch ทำงานดังนี้

- (1) รับคะแนนจากผู้ใช้
- (2) เก็บคะแนนลงตัวแปร number
- (3) เปรียบเทียบค่าในตัวแปร number ว่าน้อยกว่า 50 หรือไม่
- (4) แสดงข้อความ “น่าสงสารจริง ๆ เธอสอบตกนะ”
- (5) ถ้าคะแนนมากกว่าหรือเท่ากับ 50 แสดงข้อความ “ยินดีด้วย เธอสอบผ่านนะ”

สามารถชมตัวอย่างและบล็อกคำสั่งจากตัวอย่างนี้ได้ที่ <https://scratch.mit.edu/projects/111056343/>

อีกหนึ่งตัวอย่างของชิ้นงานจากโปรแกรม Scratch คือการตัดเกรด โดยจะต้องรับคะแนนของผู้ใช้ 3 ค่าคือคะแนน
เก็บ คะแนนสอบกลางภาค และคะแนนสอบปลายภาค นำมารวมกันก่อนจึงเข้าสู่การตัดเกรด ซึ่งมีช่วงของ
คะแนนของเกรดที่ได้รับ ดังนี้

คะแนนรวม <50 : เกรด F

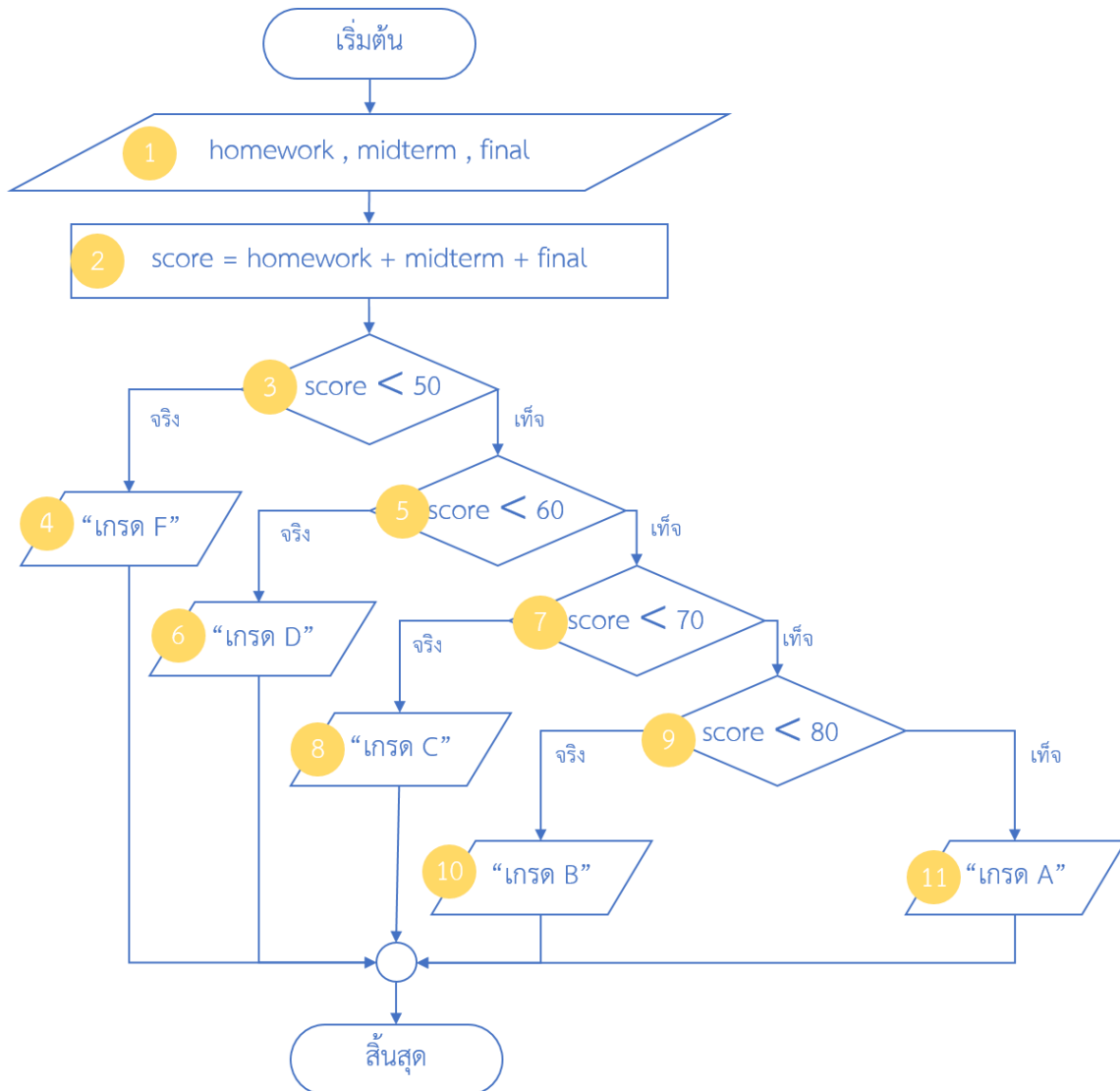
คะแนนรวม <60 : เกรด D

คะแนนรวม <70 : เกรด C

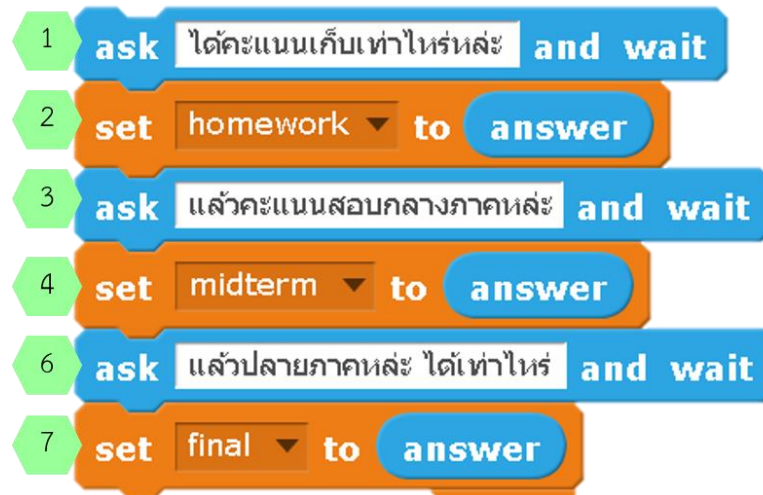
คะแนนรวม <80 : เกรด B

คะแนนรวม <100 : เกรด A

สามารถเขียนผังงานได้บล็อกคำสั่งโปรแกรม Scratch (โดยจะแสดงผังงานก่อน และตามด้วยบล็อกคำสั่ง ระหว่าง
ผังงานจะจับคู่ด้วยหมายเลข)



- (1) สัญลักษณ์ Input/Output รับค่า 3 จำนวนคือคะแนนเก็บ เก็บลงในตัวแปร homework คะแนนสอบกลางภาค เก็บลงในตัวแปร midterm คะแนนสอบปลายภาค เก็บลงในตัวแปร final วางบล็อกรหัสคำสั่งได้ดังนี้



บล็อกหมายเลข 1 ถามผู้ใช้ว่า “ได้คะแนนเก็บเท่าไรหะ” และรอผู้ใช้กรอกคะแนน

บล็อกหมายเลข 2 นำคะแนนที่ผู้ใช้กรอกเข้ามา เก็บลงในตัวแปร homework

บล็อกหมายเลข 3 ถามผู้ใช้ว่า “แล้วคะแนนสอบกลางภาคหะ” และรอผู้ใช้กรอกคะแนน

บล็อกหมายเลข 4 นำคะแนนที่ผู้ใช้กรอกเข้ามา เก็บลงในตัวแปร midterm

บล็อกหมายเลข 5 ถามผู้ใช้ว่า “แล้วปลายภาคหะ ได้เท่าไร” และรอผู้ใช้กรอกคะแนน

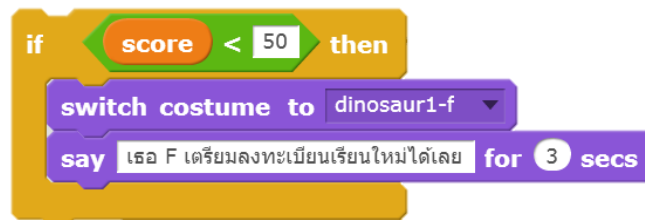
บล็อกหมายเลข 6 นำคะแนนที่ผู้ใช้กรอกเข้ามา เก็บลงในตัวแปร final

- (2) สัญลักษณ์ Process เป็นการนำคะแนนเก็บ คะแนนสอบกลางภาค และคะแนนสอบปลายภาคมารวมกัน และเก็บลงในตัวแปร score วางบล็อกคำสั่งได้ ดังนี้



บล็อกคำสั่งนี้จะนำค่าในตัวแปร homework + midterm + final และเก็บผลลัพธ์ลงในตัวแปร score

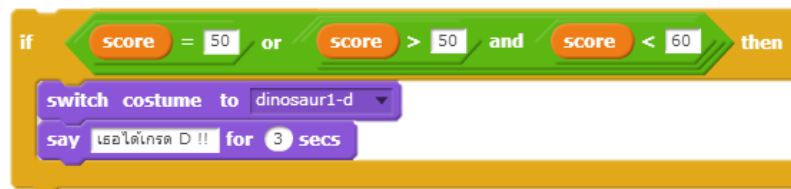
- (3) สัญลักษณ์ Decision ตรวจสอบว่าคะแนนน้อยกว่า 50 จริงหรือไม่ ถ้าจริงจะดำเนินการต่อที่ (4) ส่วนนี้สามารถวางบล็อกคำสั่งได้ ดังนี้



บล็อกคำสั่งชุดนี้จะตรวจสอบว่าค่าในตัวแปร score น้อยกว่า 50 หรือไม่ ถ้าน้อยกว่าจริงจะดำเนินการต่อภายในชุดบล็อกคำสั่งถัดเข้ามาคือ สลับรูปร่างตัวละคร และแสดงข้อความ “เธอ F เตรียมลงทะเลเป็นเรียนใหม่ได้เลย”

- (4) แสดงข้อความ “เกรด F” โดยใช้บล็อกคำสั่งภายใน (3)

- (5) ตรวจสอบว่าค่าในตัวแปร score น้อยกว่า 60 จริงหรือไม่ ถ้าจริงจะดำเนินการตอนที่ (6) แสดงข้อความ “เกรต F” สามารถวางบล็อกคำสั่งได้ ดังนี้



บล็อกคำสั่ง if score=50 or score>50 and score<60 จะดำเนินการตรวจสอบ score>50 and score<60 ก่อน เมื่อได้ผลลัพธ์แล้วจึงนำไปเปรียบเทียบกับ score=50 ที่หลัง เมื่อผลลัพธ์เป็นจริงจึงดำเนินการต่อบล็อกคำสั่งภายใน คือ ตัวละครสลับรูปร่าง และแสดงข้อความ “เกรตได้เกรต D !!”

- (6) แสดงข้อความ “เกรต D” โดยใช้บล็อกคำสั่งภายใน (5)
- (7) เปรียบเทียบค่าในตัวแปร score น้อยกว่า 70 หรือไม่ ถ้าน้อยกว่า 70 จริงจะดำเนินการตอนที่ (8) แสดงข้อความ “เกรต C” วางบล็อกคำสั่งได้ ดังนี้



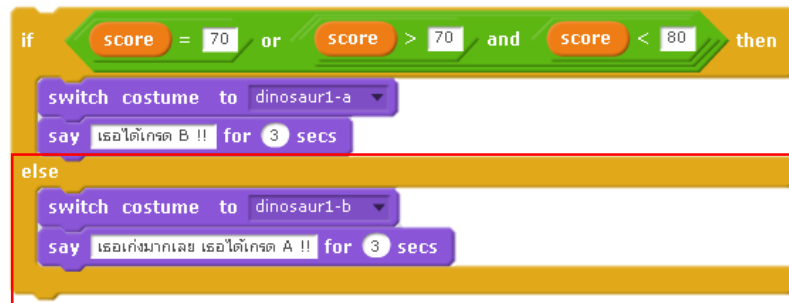
บล็อกคำสั่ง if score=60 or score>60 and score<70 จะดำเนินการตรวจสอบ score>60 and score<70 ก่อน เมื่อได้ผลลัพธ์แล้วจึงนำไปเปรียบเทียบกับ score=60 ที่หลัง เมื่อผลลัพธ์เป็นจริงจึงดำเนินการต่อบล็อกคำสั่งภายใน คือ ตัวละครสลับรูปร่าง และแสดงข้อความ “เกรตได้เกรต C !!”

- (8) แสดงข้อความ “เกรต C” โดยใช้บล็อกคำสั่งภายใน (7)
- (9) – (11) เปรียบเทียบค่าในตัวแปร score น้อยกว่า 80 หรือไม่ ถ้าน้อยกว่า 80 จริงจะดำเนินการตอนที่ (10) แสดงข้อความ “เกรต B” วางบล็อกคำสั่งได้ ดังนี้ (ภายในกรอบสีแดง)



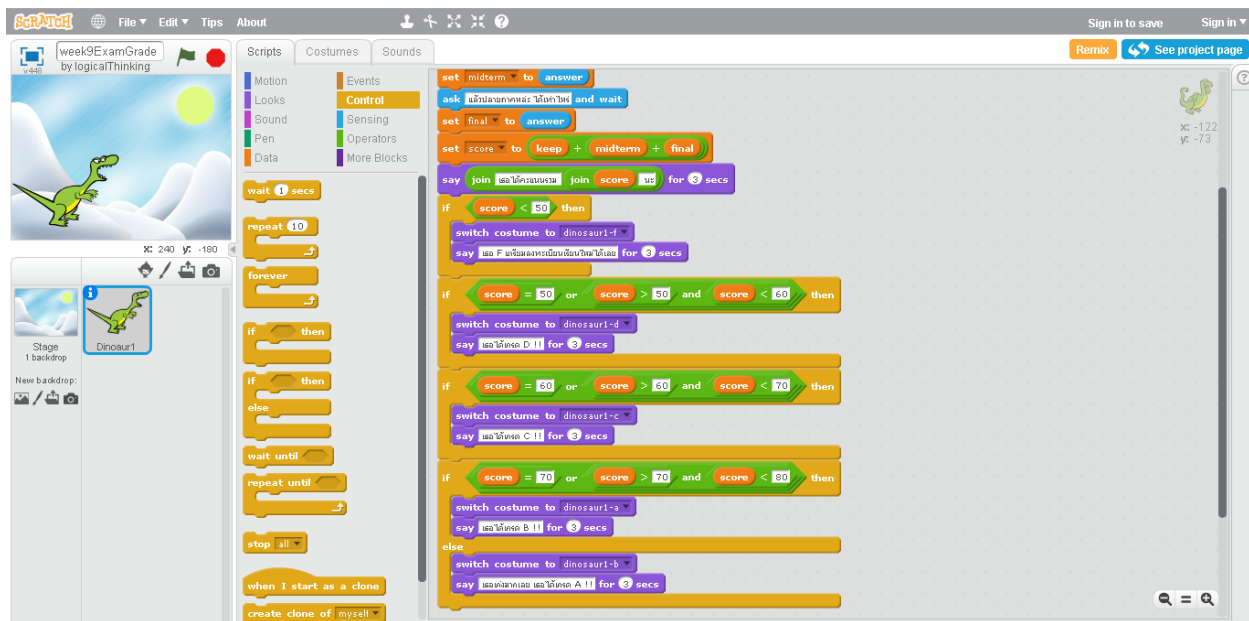
บล็อกคำสั่ง if score=70 or score>70 and score<80 จะดำเนินการตรวจสอบ score>70 and score<80 ก่อน เมื่อได้ผลลัพธ์แล้วจึงนำไปเปรียบเทียบกับ score=70 ที่หลัง เมื่อผลลัพธ์เป็นจริงจึงดำเนินการต่อบล็อกคำสั่งภายใน คือ ตัวละครสลับรูปร่าง และแสดงข้อความ “เกรตได้เกรต B !!”

จากนั้น หากตรวจสอบเงื่อนไขที่ (9) แล้วเป็นเท็จ จะแสดงข้อความ “เกรด A” และวางบล็อกคำสั่งได้ดังนี้ (ภายในกรอบสีแดง)



ภายในบล็อกคำสั่ง else เป็นผลลัพธ์ โดยดำเนินการที่บล็อกคำสั่งภายในคือ สลับรูปร่างตัวละคร และแสดงข้อความ “เธอเก่งมากเลย เธอได้เกรด A !!”

สามารถชมตัวอย่างผลงานและบล็อกคำสั่งทั้งหมดได้ที่ <https://scratch.mit.edu/projects/111057429/>



นอกจากเงื่อนไขที่ตรวจสอบจากตัวเลขอย่างในตัวอย่างการชูป้ายหมายเลขแล้ว สังเกตจากตัวอย่างการชมภาพยนตร์ จะพบว่าเงื่อนไขสามารถตรวจสอบได้ด้วยข้อความ หรือตัวอักษรเดี่ยว ๆ ได้ด้วย ตัวอย่างชิ้นงาน Scratch ที่เปรียบเทียบเงื่อนไขตัวตัวอักษรเดี่ยว ๆ จะแสดงดังต่อไปนี้

กำหนดข้อมูลสำหรับสร้างชิ้นงานนี้ คือช่วงอายุของคน (อ้างอิงจาก http://www.elearning.msu.ac.th/opencourse/0709451/page04_02_04.html) ดังนี้

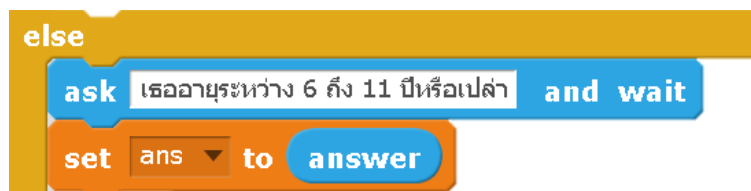
1. วัยทารกและเด็กเล็ก (แรกเกิดถึง 5 ขวบ)
2. วัยเด็กโต (อายุ 6-11 ปี)
3. วัยรุ่น (อายุ 12 – 20 ปี)
4. วัยผู้ใหญ่ (อายุ 20-60 ปี)
5. วัยสูงอายุ (อายุ 60 ปีขึ้นไป)

ชิ้นงานนี้จะรับค่าจากแป้นพิมพ์เพียง 2 ค่าเท่านั้นคือ y หรือ n โดย y แปลว่าผู้ใช้ยอมรับว่าสิ่งที่โปรแกรมถามนั้นเป็นเรื่องจริง และ n แปลว่าผู้ใช้ยอมรับว่าสิ่งที่โปรแกรมถามนั้นเป็นเรื่องไม่จริง

โดยเริ่มต้นโปรแกรมจะถามก่อนว่าผู้ใช้นั้นอายุอยู่ในช่วงแรกเกิดถึง 6 ปีหรือไม่ ดังบล็อกคำสั่งดังนี้



จากบล็อกคำสั่งด้านบน ถ้าผู้ใช้กรอก n หรือ N เข้ามานั้น โปรแกรมจะแสดงข้อความว่า “อู๋!! เธอเป็นเด็กทารกหรอ” แต่ถ้าผู้ใช้กรอก y หรือ Y เข้ามา โปรแกรมจะทำงานที่ส่วน else ดังบล็อกคำสั่งนี้



โดยการทำงานของบล็อกคำสั่งนี้จะแสดงข้อความถามผู้ใช้งานว่า “เธออายุระหว่าง 6 ถึง 11 ปีหรือเปล่า” และเก็บค่า y หรือ n ที่ผู้ใช้กรอกเข้ามาลงในตัวแปร ans และดำเนินการต่อดัง if-else ด้วยบล็อกคำสั่งดังนี้



ถ้าผู้ใช้กรอก y หรือ Y เข้ามา โปรแกรมจะแสดงข้อความ “เธอเป็นเด็กโตสินะ” แต่ถ้าผู้ใช้ป้อน n หรือ N เข้ามา โปรแกรมจะแสดงข้อความเพื่อถามผู้ใช้ต่อว่า “เธออายุระหว่าง 12 ถึง 20 ปีหรือไม่” และเก็บค่าที่ผู้ใช้กรอกเข้ามาลงในตัวแปร ans และทำการตรวจสอบที่บล็อกคำสั่ง if-else ต่อไป ดังนี้



จากบล็อกคำสั่ง ถ้าผู้ใช้กรอก y หรือ Y เข้ามา โปรแกรมจะแสดงข้อความ “อย่างเธอเนี่ย เรียกว่าวัยรุ่นนะ” แต่ถ้าผู้ใช้กรอก n หรือ N เข้ามา โปรแกรมจะถามต่อว่า “เธออายุระหว่าง 20 ถึง 60 ปีหรือไม่” และตรวจสอบต่อไป ดังนี้



ส่วนของบล็อกคำสั่งด้านบนนี้จะเป็นส่วนสุดท้ายของโปรแกรม เนื่องจากหมดเงื่อนไขที่จะต้องตรวจสอบแล้ว เพราะเมื่อถึงเงื่อนไขนี้ หากผู้ใช้กรอก y หรือ Y โปรแกรมจะแสดงข้อความ “อายุขนาดนี้ เป็นผู้ใหญ่แล้วนะ” แต่ถ้าผู้ใช้กรอก n หรือ N โปรแกรมจะลงมาทำงานที่บล็อกคำสั่งส่วนล่าง คือแสดงข้อความ “ฮู้ว!! เธอแก่แล้วนะ”

สามารถชมตัวอย่างผลงานและบล็อกคำสั่งทั้งหมดได้ที่ <https://scratch.mit.edu/projects/134165635>

