บทที่ 2
การแก้ปัญหาและขั้นตอนวิธีการ

สาระสำคัญ

การคิดเชิงคำนวณเป็นพื้นฐานของการแก้ปัญหาคอมพิวเตอร์ สำหรับการออกแบบขั้นตอนวิธีในการแก้ปัญหาด้วยโปรแกรมคอมพิวเตอร์ จำเป็นต้องระบุขั้นตอนการทำงาน รวมถึงเงื่อนไขต่างๆ ที่ชัดเจน เช่น ข้อมูลเข้า ข้อมูลออก ขอบเขตของข้อมูลที่ต้องการ มนุษย์จึงจะสามารถเขียนโปรแกรมให้คอมพิวเตอร์ทำงานตามคำสั่งได้

ตัวชี้วัด

1. ใช้เหตุผลเชิงตรรกะในการอธิบายและออกแบบวิธีการแก้ปัญหาที่พบในชีวิตประจำวัน

ขอบข่ายเนื้อหา

เรื่องที่ 1 การแก้ปัญหาด้วยคอมพิวเตอร์

เรื่องที่ 2 การระบุข้อมูลเข้า ข้อมูลออก และเงื่อนไขของปัญหา

เรื่องที่ 3 การออกแบบขั้นตอนวิธี

เรื่องที่ 4 การทำซ้ำ

เรื่องที่ 5 การจัดเรียงและค้นคว้าข้อมูล

เวลาที่ใช้ในการศึกษา 15 ชั่วโมง

สื่อการเรียนรู้

1. หนังสือแบบเรียน วิทยาการคำนวณ รหัสรายวิชา พว3300102

เรื่องที่ 1 การแก้ปัญหาด้วยคอมพิวเตอร์

คอมพิวเตอร์มีบทบาทในการปฏิวัติการทำงานในทุกภาคส่วนของสังคม การประยุกต์ใช้คอมพิวเตอร์มีผลให้ประสิทธิภาพการทำงานเพิ่มขึ้น ลดภาระงานที่ทำซ้ำๆ รวมถึงเพิ่มความแม่นยำของผลลัพธ์ที่ได้ ขั้นตอนวิธีที่จะสั่งงานให้คอมพิวเตอร์ทำงานได้ตรงตามความต้องการ ต้องผ่านการคิดวิเคราะห์และการออกแบบที่สมบูรณ์ครบถ้วน

ปัญหาที่สามารถแก้ไขด้วยคอมพิวเตอร์ไม่จำเป็นต้องเป็นปัญหาทางคณิตศาสตร์เสมอไป ซึ่งอาจเป็นปัญหาต่างๆ ในชีวิตประจำวันก็ได้ อย่างไรก็ตาม โปรแกรมคอมพิวเตอร์ต้องระบุขั้นตอนการทำงาน รวมถึงเงื่อนไขต่าง ๆ ที่ชัดเจน ดังนั้นก่อนจะแก้ปัญหาด้วยคอมพิวเตอร์ จึงควรทำความเข้าใจกับปัญหาและความต้องการให้ชัดเจน แล้วจึงพัฒนาขั้นตอนวิธีที่สามารถใช้งานได้

การแก้ปัญหาเป็นกิจกรรมพื้นฐานในการดำรงชีวิตของมนุษย์ ปัญหาบางปัญหาสามารถหาคำตอบได้ในทันที ขณะที่บางปัญหาใช้เวลานานในการค้นหาคำตอบ อย่างไรก็ตามทุกคนต่างต้องการหาวิธีการในการแก้ปัญหาที่ทำให้ได้คำตอบที่ถูกต้องในเวลารวดเร็ว

ขั้นตอนการแก้ปัญหา มี 4 ขั้นตอน (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562) ดังนี้

1. วิเคราะห์และกำหนดรายละเอียดของปัญหา เป็นการทำความเข้าใจเกี่ยวกับรายละเอียดเงื่อนไข ข้อกำหนด รวมถึงข้อจำกัดต่าง ๆ ของปัญหา ข้อมูลที่จำเป็นในการแก้ปัญหา ตรวจสอบว่ามีข้อมูลเพียงพอหรือไม่ จะหาข้อมูลเพิ่มเติมให้ครบถ้วนต่อการใช้แก้ปัญหาได้อย่างไร ข้อมูลผลลัพธ์ที่ได้คืออะไร และจะตรวจสอบความถูกต้องของผลลัพธ์ที่ได้อย่างไร

2. การวางแผนการแก้ปัญหา เครื่องมือที่ใช้ในการวางแผนการแก้ปัญหาสำหรับการพัฒนาโปรแกรม อาจเลือกใช้รหัสลำลอง หรือผังงาน โดยวิธีการแก้ปัญหาที่ได้เรียกว่า **ขั้นตอนวิธีหรืออัลกอริทึม (Algorithm)** ซึ่งเป็นลำดับขั้นตอนในการแก้ปัญหาหรือการทำงานที่ชัดเจน

3. การดำเนินการปัญหา เป็นกระบวนการที่ได้วางแผนไว้มาปฏิบัติ หรือพัฒนาโปรแกรมเพื่อแก้ปัญหา โดยอาจใช้ภาษาโปรแกรมช่วยในการดำเนินการ

4. การตรวจสอบและประเมินผล ขั้นตอนนี้จะทำควบคู่ไปกับขั้นตอนการดำเนินการแก้ปัญหา โดยการตรวจสอบผลลัพธ์ที่ได้ไม่ถูกต้อง หรือยังมีส่วนที่ต้องแก้ไขปรับปรุงอยู่ต้องย้อนกลับไปทำซ้ำตั้งแต่ขั้นตอนแรกจนกว่าจะได้ผลลัพธ์ที่ถูกต้อง

ตัวอย่างที่ 2.1 การหาค่ามากที่สุดของจำนวนสามจำนวนที่กำหนดให้การแก้ปัญหามีขั้นตอนดังนี้

ขั้นตอนที่ 1 การวิเคราะห์และกำหนดรายละเอียดของปัญหา

ข้อมูลเข้า จำนวน 3 จำนวน ได้แก่ x, y และ z

ข้อมูลออกหรือสิ่งที่ต้องการ ตัวเลขที่มีค่ามากที่สุดของเลข 3 จำนวน

วิธีตรวจสอบความถูกต้อง ดำเนินการหาตัวเลขที่มากที่สุดด้วยตนเอง โดยกำหนดชุดตัวเลข 3 จำนวน เช่น 5, 4 และ 12 ในกรณีนี้ตัวเลขที่มีค่ามากที่สุด คือ 12

ขั้นตอนที่ 2 การวางแผนการแก้ปัญหา

2.1 เปรียบเทียบ x และ y เพื่อหาค่ามากกว่าระหว่างสองจำนวน

2.2 นำค่าที่ได้มาเปรียบเทียบกับ z เพื่อหาค่าที่มากกว่า

2.3 ค่าที่มากที่สุดคือ ผลลัพธ์ที่ได้จากขั้นตอนที่ 2.1

ขั้นตอนที่ 3 การดำเนินการแก้ปัญหา

ดำเนินการทดสอบตามขั้นตอนที่วางแผนไว้กับชุดตัวเลขที่กำหนด โดยสมมติ x, y และ z เป็น 5, 4 และ 8

3.1 เปรียบเทียบเพื่อหาค่าที่มากกว่าระหว่าง 5 และ 4 พบว่า 5 เป็นค่าที่มากกว่า

3.2 เปรียบเทียบเพื่อหาค่ามากกว่าระหว่าง 5 และ 8 พบว่า 8 เป็นค่าที่มากกว่า

3.3 ค่าที่มากที่สุดของ 5, 4 และ 8 คือ 8

ขั้นตอนที่ 4 การตรวจสอบและปรับปรุง

เมื่อพิจารณาคำตอบที่ได้คือ 8 กับค่าที่เหลือซึ่งได้แก่ 5 และ 4 พบว่า 8 มีค่ามากกว่าค่าที่เหลือทั้งคู่ คำตอบนี้จึงเป็นคำตอบที่ถูกต้องตามข้อกำหนดของสิ่งที่ต้องการ

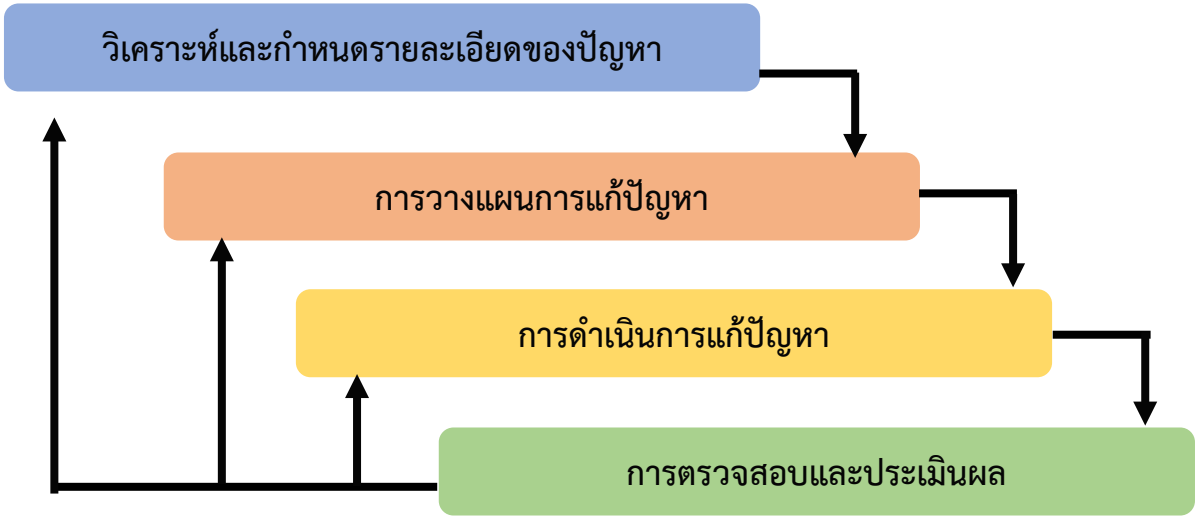
แนวคิดข้างต้นใช้งานได้เนื่องจากว่าหากพิจารณาจำนวนสามจำนวนใด ๆ เมื่อ x > y และ y > z แล้ว x > z ด้วย

จากตัวอย่างแม้ว่าเราจะไม่ได้นำค่า 8 มาเปรียบเทียบกับ 4 โดยตรง แต่เราได้นำมาเปรียบเทียบกับ 5 ซึ่ง 5 ถูกตรวจสอบมาก่อนหน้านี้แล้วมากกว่า 4 เพราะฉะนั้น 8 จึงมากกว่า 4 ด้วย

ตัวอย่างโปรแกรมภาษา Python การหาค่ามากที่สุดของจำนวนสามจำนวน (FindMaxNumber.py)

```
1  #การหาค่ามากที่สุดของจำนวนสามจำนวนที่กำหนด
2  #รับค่าจำนวนสามจำนวน
3  a=int(input("ป้อนจำนวนที่ 1 :"))
4  b=int(input("ป้อนจำนวนที่ 2 :"))
5  c=int(input("ป้อนจำนวนที่ 3 :"))
6
7  #นำจำนวนทั้งสามจำนวนมาเก็บไว้ใน List
8  num = list((a,b,c))      # ข้อมูลเข้าจำนวน 3 จำนวน เก็บอยู่ในรูปแบบ List
9
10 #เปรียบเทียบจำนวนทั้งสามจำนวน
11 nummax = num[0]          # ตั้งค่าสูงสุดเริ่มต้น
12 for n in num:            # ทำการวนซ้ำเพื่อเปรียบเทียบจำนวนแต่ละตัวใน List
13     if(n>nummax):        # ถ้าพบมากกว่าค่าสูงสุดเดิม
14         nummax = n       # ให้เปลี่ยนเป็นค่านั้น
15
16 print('จำนวนที่มากที่สุดคือ  %.1d '%(nummax))
```

การแก้ปัญหา ประกอบด้วย 4 ขั้นตอนดังรูป คือ



เกร็ดความรู้

ข้อมูล

ข้อมูล หมายถึง ข่าวสาร เอกสาร ข้อเท็จจริงเกี่ยวกับบุคคล สิ่งของหรือเหตุการณ์ที่มีอยู่ในรูปของตัวเลข ภาษา ภาพ สัญลักษณ์ต่างๆ ที่มีความหมายเฉพาะตัว ซึ่งยังไม่มีการประมวลเกี่ยวกับการนำไปใช้ได้อย่างมีประสิทธิภาพ (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562)

เงื่อนไขที่ชัดเจน

เงื่อนไข หมายถึง ข้อแม้, ข้อจำกัด หรือข้อตกลง เงื่อนไขที่ชัดเจน คือการระบุสิ่งต่างๆ ที่จำเป็นสำหรับการตัดสินใจอย่างใดอย่างหนึ่ง เพื่อให้คอมพิวเตอร์สามารถทำงานตามที่ต้องการ (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562)

ตัวแปร

ตัวแปร คือ ชื่อที่ใช้แทนข้อมูลขณะใดขณะหนึ่งในขั้นตอนวิธี เช่น การเลือกอาหารในรายการอาหาร พบว่า มีการใช้งานตัวแปร เช่น ตัวแปร Q, R และ S โดยทั่วไปแล้วในทางคอมพิวเตอร์ ตัวแปรจะถูกใช้เพื่อเก็บข้อมูล และอาจจะมีการเปลี่ยนแปลงค่าได้ตามการทำงาน (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562)

ในการเขียนขั้นตอนวิธีอาจจะระบุว่า ให้กำหนดค่าให้กับตัวแปร หรืออาจใช้สัญลักษณ์ ⟵ ในการเขียน เช่น

x ⟵ 10 จะเป็นการระบุให้ตัวแปร x มีค่าเท่ากับ 10

z ⟵ z + 1 จะเป็นการกำหนดให้เพิ่มค่าตัวแปร z ขึ้นอีก 1 กล่าวคือ ถ้าก่อนการทำงานตัวแปร z มีค่าเท่ากับ 10 หลังการทำงานตัวแปร z จะมีค่าเท่ากับ 11

สัญลักษณ์ที่ใช้ในการเขียนขั้นตอนวิธีนอกเหนือจากการใช้สัญลักษณ์ทางคณิตศาสตร์ทั่วไป

⟵ , := , <-- , =	หมายถึง การกำหนดค่าให้ตัวแปร
! , ~ , -	หมายถึง นิเสธ
!= , <>	หมายถึง การไม่เท่ากัน
>=	หมายถึง การเปรียบเทียบมากกว่าหรือเท่ากับ
<=	หมายถึง การเปรียบเทียบน้อยกว่าหรือเท่ากับ
** , ^	หมายถึง การยกกำลัง

การแก้ปัญหาด้วยคอมพิวเตอร์ นอกจากจะช่วยลดระยะเวลาในการตัดสินใจ ยังเพิ่มขีดความสามารถในการแก้ปัญหา เมื่อข้อมูลมีจำนวนมากเกินกว่าที่มนุษย์จะคำนวณได้ ผู้เรียนจะได้ศึกษาวิธีการและแนวคิดอื่นเพิ่มเติม เพื่อให้สามารถนำไปใช้การแก้ปัญหาที่ซับซ้อนได้

เรื่องที่ 2 การระบุข้อมูลเข้า ข้อมูลออก และเงื่อนไขของปัญหา

การแก้ปัญหาด้วยคอมพิวเตอร์นั้น ก่อนที่จะระบุขั้นตอนวิธีที่ชัดเจนได้ จะต้องวิเคราะห์และทำความเข้าใจกับปัญหา เพื่อให้ทราบว่ามีข้อมูลอะไรบ้างที่สามารถใช้การประมวลผลได้มีเงื่อนไขต่างๆ อย่างไร ผลลัพธ์ที่ต้องการคืออะไร โดยจะแบ่งข้อที่เกี่ยวกับการทำงานออกเป็น 2 ส่วน (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562) คือ

1. ข้อมูลเข้า (Input) เป็นข้อมูลที่ใช้เพื่อประมวลผล

2. ข้อมูลออก (Output) เป็นข้อมูลผลลัพธ์ที่ต้องการ

จากการวิเคราะห์ข้อมูลทั้งสองส่วนนี้ นอกจากระบุว่าคืออะไรแล้ว ยังอาจจะระบุเงื่อนไขเพิ่ม เช่น ข้อมูลเข้าอาจมีการระบุขอบเขตหรือเงื่อนไข หรือข้อมูลออกอาจมีการระบุคุณสมบัติที่ต้องวิเคราะห์นี้ เป็นการระบุข้อกำหนดต่าง ๆ ที่เกี่ยวข้องกับปัญหาให้ชัดเจน ซึ่งจำเป็นต่อการออกขั้นตอนวิธีที่ถูกต้อง

ตัวอย่างที่ 1 ปัญหาการหา ห.ร.ม.

พิจารณาตัวอย่างปัญหาการหา ห.ร.ม. ผู้เรียนสามารถระบุข้อมูลเข้า ข้อมูลออก รวมทั้งเงื่อนไขได้ดังนี้

ข้อมูลเข้า : จำนวนเต็มบวกสองจำนวน a และ b

ข้อมูลออก : จำนวนเต็มบวกหนึ่งจำนวน c ที่มีคุณสมบัติดังนี้

- c เป็นจำนวนเต็มบวก
- c หาร a และ b ลงตัว
- c มีค่ามากที่สุดเท่าที่เป็นไปได้

ตัวอย่างโปรแกรมภาษา Python ปัญหาการหา ห.ร.ม. (FindGDC.py)

```python
#การหา ห.ร.ม. ของตัวเลข 2 จำนวน

#ข้อมูลเข้า (Input)
#รับค่าตัวเลขจำนวนเต็มตัวที่ 1
a=int(input("ป้อนจำนวนเต็มตัวที่ 1 : "))
#รับค่าตัวเลขจำนวนเต็มตัวที่ 2
b=int(input("ป้อนจำนวนเต็มตัวที่ 2 : "))

#ชุดคำสั่งการหา ห.ร.ม. ด้วยวิธีหารสั้น
n=min(a,b)
result=1
st = 2
num1=a
num2=b
while st<n:
  if((num1%st == 0) and (num2%st == 0)):
   result=result*st
   num1=num1/st
   num2=num2/st
   st=1
  st += 1

#ข้อมูลออก (Output)
#ตัวแปร c รับผลการคำนวณค่า ห.ร.ม. มาเก็บไว้
c=result
#แสดงผลลัพธ์
print("ห.ร.ม. ของ %d และ %d คือ %d"%(a,b,c))
```

ตัวอย่างที่ 2 คะแนนการตรวจแบบฝึกทักษะ

พิจารณาสถานการณ์สมมติต่อไปนี้

ครูได้ตรวจแบบฝึกทักษะของผู้เรียน 50 คน ประกาศคะแนนไว้หน้าห้อง หากต้องการหาคะแนนสูงสุด คะแนนต่ำสุด และคำนวณคะแนนเฉลี่ยของผู้เรียนทุกคน ในกรณีนี้ ระบุข้อมูลเข้า ข้อมูลออกได้ดังนี้

ข้อมูลเข้า : รายการคะแนนแบบฝึกทักษะของผู้เรียน 50 คน

ข้อมูลออก : คะแนนสูงสุด คะแนนต่ำสุด และคะแนนเฉลี่ย

แม้ว่าในหลาย ๆ กรณี การระบุข้อมูลเข้าและข้อมูลออกนั้น อาจไม่สามารถทำได้อย่างชัดเจน แต่ความพยายามในการระบุข้อมูลทั้งสอง มักเป็นเงื่อนไขให้ต้องทำความเข้าใจกับปัญหามากขึ้น ลองพิจารณาตัวอย่างต่อไปนี้

ตัวอย่างโปรแกรมภาษา Python คะแนนสอบ (FindMinMaxAvgScore.py)

```
#ข้อมูลเข้า
#จัดการข้อมูลคะแนนจำนวน 40 ชุด ให้อยู่ในรูปแบบของ List
score=[18,42,36,25,37,23,42,37,41,16,28,47,33,35,44,48,49,25,28,29,
26,24,45,21,38,38,31,44,35,37,31,39,26,28,29,24,25,34,26,43]

#หาค่าคะแนนต่ำสุดโดยการเปรียบค่าคะแนนแต่ละตัว
min = score[0]          # ตั้งค่าต่ำสุดเริ่มต้น
for x in score:         # ทำการวนซ้ำเพื่อเปรียบเทียบจำนวนแต่ละตัว
    if(x<min):          # ถ้าพบมากกว่าค่าต่ำสุดเดิม
        min = x         # ให้เปลี่ยนเป็นค่านั้น

#หาค่าคะแนนสูงสุดโดยการเปรียบค่าคะแนนแต่ละตัว
max = score[0]          # ตั้งค่าสูงสุดเริ่มต้น
for x in score:         # ทำการวนซ้ำเพื่อเปรียบเทียบจำนวนแต่ละตัว
    if(x>max):          # ถ้าพบมากกว่าค่าสูงสุดเดิม
        max = x         # ให้เปลี่ยนเป็นค่านั้น

#หาคะแนนเฉลี่ย
total=0
count=0
for x in score:
  total=total+x         # หาผลรวมของคะแนนทั้งหมด
  count=count+1         # หาจำนวนชุดข้อมูล
avgscore=total/count    # หาค่าเฉลี่ยคะแนน=คะแนนรวม/จำนวนชุดข้อมูล

#ข้อมูลออก (Output)
print("คะแนนต่ำสุด เท่ากับ %f" %(min))
print("คะแนนสูงสุด เท่ากับ %f" %(max))
print("คะแนนเฉลี่ย เท่ากับ %f" %(avgscore))
```

ตัวอย่างที่ 3 แบ่งกลุ่มทำงาน

ผู้เรียนในห้องต้องการจัดกิจกรรมวันรักการอ่าน จากการประชุมมีงานที่ต้องทำดังนี้

1. จัดบอร์ดหน้าห้องเกี่ยวกับวันรักการอ่าน
2. จัดเตรียมงานการประกวดวาดภาพ
3. เป็นกลุ่มผู้ประกวดวาดภาพ โดยมีสามกลุ่ม กลุ่มละ 2 คน
4. อ่านกลอนวันรักการอ่าน
5. ร้องเพลงลูกทุ่ง

เพื่อให้ทุกคนได้ทำงานที่ต้องการทำหรืออย่างน้อยเป็นงานที่ยินดีทำ จึงได้ให้ผู้เรียนทุกคนกรอกข้อมูลว่าสามารถทำงานใดได้บ้าง และมีงานใดบ้างที่ต้องการทำเป็นพิเศษ โดยมีเงื่อนไขว่า ผู้เรียนหนึ่งคนไม่ควรทำงานเกิน 2 อย่าง และผู้ประกวดวาดภาพไม่ควรเป็นคนจัดเตรียมงานประกวดวาดภาพ จากข้อมูลดังกล่าว ต้องการจัดกลุ่มว่าผู้เรียนคนใด จะทำงานใดบ้าง สามารถระบุข้อมูลเข้าและข้อมูลออกได้ดังนี้

ข้อมูลเข้า : รายการของงานทั้งหมด ข้อมูลของผู้เรียนแต่ละคนที่ระบุว่าสามารถทำงานใดได้บ้างและต้องการทำงานใดเป็นพิเศษบ้าง

ข้อมูลออก : ข้อมูลที่ระบุว่าผู้เรียนคนใดทำงานอะไร โดยมีเงื่อนไขดังนี้

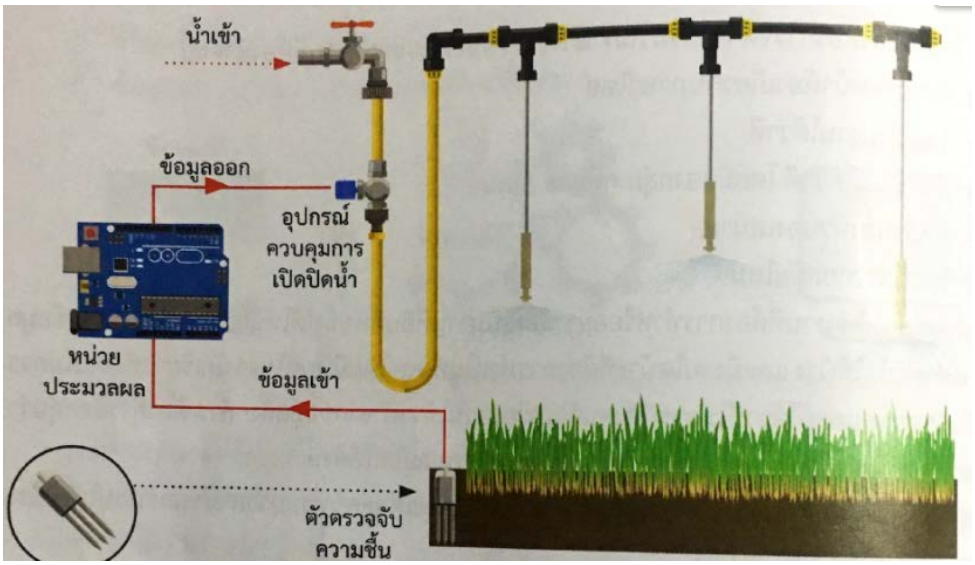
- ผู้เรียนควรได้ทำงานที่ตนเองเลือกว่าสามารถทำงานนั้นได้
- ถ้าเป็นไปได้ผู้เรียนควรได้ทำงานที่ตนเองเลือกว่าอยากทำเป็นพิเศษ
- ผู้เรียนแต่ละคน ควรมีงานที่ต้องทำอย่างน้อย 1 อย่าง แต่ไม่ควรเกิน 2 อย่าง
- ผู้เรียนที่อยู่ในกลุ่มผู้ประกวดวาดภาพ ไม่ควรเป็นผู้เตรียมงานประกวดวาดภาพ

จากการวิเคราะห์ข้อมูลข้างต้น ผู้เรียนอาจจะพบปัญหา เมื่อเริ่มดำเนินการ เช่น ถ้าผู้เรียนบางคนไม่ระบุงานที่สามารถทำได้ ก็จะทำให้ไม่สามารถจัดกลุ่มได้ สังเกตว่า

การระบุข้อมูลที่ชัดเจน ทำให้สามารถวิเคราะห์สถานการณ์ได้ชัดเจนยิ่งขึ้น และจะช่วยปรับปรุงกระบวนการต่าง ๆ ได้ดีกว่าเดิม ในกรณีนี้ เพื่อให้สามารถจัดกลุ่มได้ อาจเพิ่มเงื่อนไขให้ผู้เรียนทุกคนต้องเลือกงานที่สามารถทำได้อย่างน้อยหนึ่งอย่าง

ตัวอย่างที่ 4 อุปกรณ์รดน้ำต้นไม้อัตโนมัติ

ตัวอย่างนี้จะพิจารณาการสร้างอุปกรณ์ เพื่อตรวจสอบระดับความชื้นของดิน ถ้าดินแห้งจะสั่งให้ระบบรดน้ำต้นไม้อัตโนมัติ แสดงดังรูป (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562)



ระบบการรดน้ำต้นไม้อัตโนมัติ

ที่มาภาพ : หนังสือเรียนรายวิชาพื้นฐานวิทยาศาสตร์เทคโนโลยี (วิทยาการคำนวณ) ชั้นมัธยมศึกษาปีที่ 4 สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562

สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการได้สรุปไว้ว่า ระบบการรดน้ำต้นไม้อัตโนมัตินี้ มีการรับและสั่งงานระหว่างคอมพิวเตอร์ และอุปกรณ์อื่น เช่น ตัวตรวจจับ (Sensor) เพื่อใช้อ่านข้อมูลจากสภาพแวดล้อมหรือจากสิ่งที่สนใจ โดยข้อมูลเข้า คือ ระดับความชื้นของที่อ่านจากตัวตรวจจับ และเครื่องคอมพิวเตอร์จะประมวลผลเพื่อสั่งงาน ไปยังอุปกรณ์ควบคุมมการเปิด-ปิด ดังนั้นข้อมูลออกในกรณีนี้ คือ สัญญาณควบคุมอุปกรณ์ เปิด-ปิดน้ำ โดยสรุปสามารถระบุข้อมูลเข้าและข้อมูลออกได้ดังนี้

ข้อมูลเข้า : ระดับความชื้นของดิน (ผ่านทางตัวตรวจจับ)

ข้อมูลออก : สัญญาณควบคุมการเปิด-ปิดน้ำ

เรื่องที่ 3 การออกแบบขั้นตอนวิธี

ทักษะการคิดเชิงคำนวณ เช่น การแยกส่วนประกอบและการย่อยปัญหา การหารูปแบบ และการคิดเชิงนามธรรม สามารถนำมาใช้ในการออกแบบขั้นตอนวิธีเพื่อแก้ปัญหาต่างๆ การออกแบบไม่มีขั้นตอนที่ตายตัว จำเป็นต้องอาศัยประสบการณ์และการฝึกฝน จึงเป็นสิ่งที่ท้าทาย ซึ่งจะเป็นประโยชน์กับผู้เรียนในอนาคต (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562)

ตัวอย่างที่ 1 การตัดสินใจรดน้ำในโรงเรือนของระบบรดน้ำในโรงเรือนอัตโนมัติ

การตัดสินใจรดน้ำในโรงเรือน ในขั้นตอนวิธีของระบบรดน้ำในโรงเรือนอัตโนมัติ ระบบจะต้องอ่านข้อมูลความชื้นในดิน แล้วเปรียบเทียบกับค่าที่กำหนดไว้ (สมมติค่าความชื้นที่กำหนดเป็น 0.5 หน่วย) หากค่าความชื้นต่ำกว่าค่าที่กำหนดระบบส่งสัญญาณเปิดน้ำ และหากมีค่าความชื้นเกินกว่าหรือเท่ากับค่าที่กำหนดไว้ระบบส่งสัญญาณปิดน้ำ

ในส่วนการทำงานหลักของขั้นตอนวิธี คือ การตัดสินใจรดน้ำในโรงเรือน มีการทำงานตามลำดับดังนี้

1. อ่านค่าความชื้นในดิน
2. ให้ X แทนค่าความชื้นดังกล่าว
3. ถ้า X < 0.5 แล้วระบบส่งสัญญาณเปิดน้ำ
4. ถ้าเงื่อนไขไม่เป็นจริงระบบส่งสัญญาณปิดน้ำ

ส่วนของขั้นตอนวิธีดังกล่าวเป็นการตัดสินใจเพียงครั้งเดียว ดังนั้นเพื่อความสมบูรณ์ของขั้นตอน วิธีที่จะทำให้ระบบรดน้ำในโรงเรือน มีการอ่านค่าและส่งสัญญาณควบคุมจะต้องทำสม่ำเสมอ จึงต้องสั่งให้ขั้นตอนวิธีด้านบนทำงานซ้ำๆ ต่อเนื่องกันไป

ขั้นตอนวิธี : ควบคุมการเปิด-ปิดน้ำของเครื่องรดน้ำในโรงเรือน
ข้อมูลเข้า : ความชื้นของดิน
ข้อมูลออก : สัญญาณเปิด-ปิดน้ำ

1. ทำซ้ำทุก ๆ 1 วินาที

1.1 อ่านค่าความชื้นของดิน

1.2 ให้ X แทนค่าความชื้นดังกล่าว

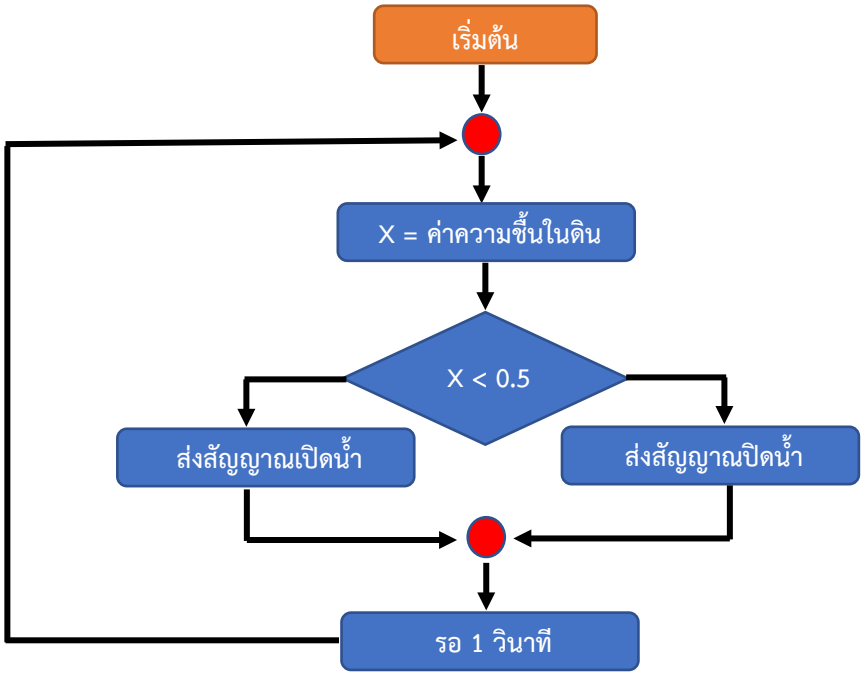
1.3 ถ้า X < 1 แล้ว

1.3.1 ส่งสัญญาณเปิดน้ำ

ถ้าเงื่อนไขไม่เป็นจริง

1.3.2 ส่งสัญญาณปิดน้ำ

ขั้นตอนวิธีดังกล่าวเขียนเป็นผังงานได้ดังนี้



ผังงานระบบการรดน้ำในโรงเรือนอัตโนมัติ

ตัวอย่างโปรแกรมภาษา Python
การตัดสินใจรดน้ำต้นไม้ของระบบรดน้ำต้นไม้อัตโนมัติ (WaterThePlants.py)

```python
import time

while True:
  #ค่าตัวแปร H รับสัญญาณข้อมูลจากเครื่องมือวัดความชื้น
  H = float(input("ทดลองป้อนค่าความชื้นของดิน (ค่าระหว่าง 0.01-2.00) :"))
  if H<0.1:              # เปรียบเทียบค่าความชื้นที่รับเข้ามากับค่าเงื่อนไขที่กำหนดไว้
    signal="เปิดน้ำ"     # ถ้าเงื่อนไขเป็นจริงส่งสัญญาณ "เปิดน้ำ"
  else:
    signal="ปิดน้ำ"      # ถ้าเงื่อนไขไม่เป็นจริงส่งสัญญาณ "ปิดน้ำ"

  print('ค่าความชื้นของดินเท่ากับ %f' %(H))
  print("ส่งสัญญาณ : "+ signal)

  #กำหนดเวลารอคอย 1 วินาที สำหรับรับค่าตัวแปร H ค่าใหม่
  time.sleep(1)
```

ตัวอย่างที่ 2 การคำนวณคะแนนสอบ

พิจารณาสถานการณ์ที่ต้องการคำนวณคะแนนสูงสุด คะแนนต่ำสุด และคะแนนเฉลี่ยของผู้เรียนในห้อง โดยมีข้อมูลเข้าและข้อมูลออก (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562) ดังนี้

ข้อมูลเข้า : รายการคะแนนสอบของผู้เรียน 40 คน

ข้อมูลออก : คะแนนสูงสุด คะแนนต่ำสุด คะแนนเฉลี่ย

ปัญหานี้สามารถแบ่งได้เป็นสามส่วน คือ การหาคะแนนสูงที่สุด การหาคะแนนต่ำที่สุด และการคำนวณคะแนนเฉลี่ย จะเริ่มพิจารณาจากปัญหาการหาคะแนนสูงที่สุดก่อนในการออกแบบนั้น จะเริ่มสังเกตวิธีการที่ผู้เรียนใช้ในการหาค่าสูงสุดของข้อมูล

โดยทั่วไปแล้ว ถ้ามีจำนวนข้อมูลน้อย ผู้เรียนสามารถหาค่าสูงสุดได้ทันที อย่างไรก็ตาม ถ้าข้อมูลอยู่คนละหน้า จะพบว่าระหว่างที่พลิกหน้าไปพิจารณาข้อมูล ผู้เรียนจะต้องจำค่าสูงสุดที่พบจากข้อมูลชุดแรก สังเกตว่าการตัดสินใจดังกล่าวจะเกิดขึ้นระหว่างที่พิจารณาข้อมูลไปทีละจำนวนด้วย ผู้เรียนสามารถใช้แนวคิดนี้ ในการเขียนขั้นตอนวิธีได้ โดยในการจดจำค่า จะใช้ตัวแปรชื่อ Max แทนค่าสูงสุดที่พบขั้นตอนวิธีดังกล่าวแสดงได้ดังนี้

ขั้นตอนวิธี : หาค่าสูงที่สุดของข้อมูลในรายการ

ข้อมูลเข้า : รายการข้อมูล

ข้อมูลออก : ค่าสูงสุดของข้อมูล

1. พิจารณาข้อมูลตัวแรก ให้ Max มีค่าเป็นข้อมูลดังกล่าว
2. พิจารณาข้อมูลตัวถัดไป ทีละจำนวน
    2.1 เรียกข้อมูลตัวที่กำลังพิจารณาว่า x
    2.2 ถ้า x > Max แล้ว
        ให้ Max ⟵ x
3. ตอบว่าค่าสูงที่สุดคือ Max

ตัวอย่างโปรแกรมภาษา Python (FindMinMaxAvgScore.py)

```
11  #หาค่าคะแนนสูงสุดโดยการเปรียบค่าคะแนนแต่ละตัว
12  max = score[0]          # ตั้งค่าสูงสุดเริ่มต้น
13  for x in score:         # ทำการวนซ้ำเพื่อเปรียบเทียบจำนวนแต่ละตัว
14      if(x>max):          # ถ้าพบมากกว่าค่าสูงสุดเดิม
15          max = x         # ให้เปลี่ยนเป็นค่านั้น
```

เนื่องจากคะแนนเฉลี่ยคือคะแนนรวมหารด้วยจำนวนผู้เรียนในห้อง ซึ่งในที่นี้มีค่าเท่ากับ 40 คน ดังนั้นในการคำนวณคะแนนเฉลี่ยจะสนใจเฉพาะคะแนนรวม ผู้เรียนสามารถคำนวณคะแนนรวมด้วยมีการใกล้เคียงกับการหาคะแนนสูงสุด โดยจะใช้ตัวแปร Total เก็บค่าผลรวมของคะแนนทั้งหมด เมื่อเริ่มต้น จะให้ Total มีค่าเป็น 0 ขั้นตอนวิธีดังกล่าวแสดงได้ดังนี้

ขั้นตอนวิธี : หาผลรวมของข้อมูลในรายการ

ข้อมูลเข้า : รายการข้อมูล

ข้อมูลออก : ผลรวมของข้อมูล

1. ให้ Total มีค่าเป็น 0
2. พิจารณาข้อมูล ทีละจำนวนจนครบทุกจำนวน
    2.1 เรียกข้อมูลตัวที่กำลังพิจารณาว่า x
    2.2 ให้ Total ⟵ Total + x
3. ตอบว่าผลรวม คือ Total

เมื่อคำนวณผลรวมได้แล้ว ค่าเฉลี่ยจะมีค่าเท่ากับ Total ÷ 40

การหาค่าเฉลี่ยโดยหารผลรวมด้วย 40 นั้น ใช้ได้กับกรณีที่มีข้อมูลจำนวน 40 จำนวน เท่านั้น สามารถแก้ไขขั้นตอนวิธีให้ทำงานได้ครอบคลุมมากขึ้น โดยนับจำนวนข้อมูลไปพร้อมๆ กับการหาผลรวม ขั้นตอนวิธีที่แก้ไขแล้ว เป็นดังนี้

ขั้นตอนวิธี : หาค่าเฉลี่ยของข้อมูลในรายการ

ข้อมูลเข้า : รายการข้อมูล

ข้อมูลออก : ค่าเฉลี่ยของข้อมูล

1. ให้ Total มีค่าเป็น 0
2. ให้ Count มีค่าเป็น 0
3. พิจารณาข้อมูล ทีละจำนวน
    3.1 เรียกข้อมูลตัวที่กำลังพิจารณาว่า x
    3.2 ให้ Total ⟵ Total + x
    3.3 ให้ Count ⟵ Count + 1
4. ตอบว่าค่าเฉลี่ย คือ Total ÷ Count

ขั้นตอนในข้อ 3.2 และ 3.3 มีการกำหนดให้ Total มีค่าเป็น Total + x เพื่อเพิ่มค่าให้ตัวแปร Total และมีการกำหนดให้ Count มีค่าเป็น Count + 1 เพื่อเพิ่มค่าให้ตัวแปร Count การเขียนลักษณะนี้ จะพบเห็นได้บ่อยครั้งในการเขียนโปรแกรม ซึ่งแตกต่างจากการเขียนสมการทางคณิตศาสตร์

ตัวอย่างโปรแกรมภาษา Python (FindMinMaxAvgScore.py)

```
17  #หาคะแนนเฉลี่ย
18  total=0
19  count=0
20  for x in score:
21      total=total+x          # หาผลรวมของคะแนนทั้งหมด
22      count=count+1          # หาจำนวนชุดข้อมูล
23  avgscore=total/count       # หาค่าเฉลี่ยคะแนน=คะแนนรวม/จำนวนชุดข้อมูล
```

สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ (2562) ได้อธิบายการออกแบบและพิจารณาเงื่อนไขดังนี้

1. การสร้างเงื่อนไขอย่างง่าย

การออกแบบเงื่อนไขที่ถูกต้องและชัดเจน จะเป็นปัจจัยสำคัญของการออกแบบขั้นตอนวิธี ซึ่งเงื่อนไขที่กำหนด อาจเป็นเงื่อนไขอย่างง่ายหรือเงื่อนไขที่ซับซ้อน โดยเงื่อนไขอย่างง่าย มักจะเป็นการเปรียบเทียบ มากกว่า น้อยกว่า หรือไม่เท่ากัน เช่น การหาค่าสูงสุด มีการใช้เงื่อนไข

ถ้า x > Max แล้ว ให้ Max ⟵ x

ขั้นตอนวิธีดังกล่าว มีการกำหนดให้การทำงานขึ้นกับเงื่อนไข “x > Max” ซึ่งเป็นรูปแบบที่ง่ายที่สุด โดยเปรียบเทียบค่าของตัวแปร x และค่าของตัวแปร Max

ตัวอย่างที่ 3 ตรวจสอบพิกัด

แผนที่ฉบับหนึ่ง มีอัตราส่วน 1 เซนติเมตร ต่อ 1 กิโลเมตร และมีการตีตารางพิกัดทุกๆ หนึ่งเซนติเมตร ถ้าพิกัดของโรงเรียนอยู่ที่ตำแหน่ง (2,3) พิกัดของร้านอาหารอยู่ที่ตำแหน่ง (x,y) ในแผนที่ ถ้าต้องการตรวจสอบว่าร้านอาหารมีระยะห่างจากโรงเรียนไม่เกิน 1 กิโลเมตรหรือไม่ จะระบุเงื่อนไขดังนี้

เงื่อนไข : จุด (x,y) ห่างจากจุด (2,3) ในแผนที่ไม่เกิน 1 เซนติเมตร

ในการตรวจสอบเงื่อนไขดังกล่าว อาจต้องใช้ไม้บรรทัดวัดระยะห่างในแผนที่ หรือใช้ทฤษฎีบทพีทาโกรัส คำนวณระยะห่างของพิกัดร้านอาหาร (x,y) กับจุด (2,3) ได้เท่ากับ $\sqrt{(x-2)^2+(y-3)^2}$ ดังนั้นสามารถเขียนเงื่อนไขดังกล่าวในอีกรูปแบบหนึ่งเป็น

เงื่อนไข : $\sqrt{(x-2)^2+(y-3)^2} \leq 1$

ตัวอย่างโปรแกรมภาษา Python ตรวจสอบหาพิกัด (CheckDistance.py)

```
1  import math
2  #ตรวจสอบระยะห่างร้านอาหารกับโรงเรียน
3
4  #รับค่าพิกับตำแหน่งของร้านอาหาร
5  x=float(input("ป้อนพิกัด x บนแผนที่ของร้านอาหาร : "))
6  y=float(input("ป้อนพิกัด Y บนแผนที่ของร้านอาหาร : "))
7  print('พิกัดบนแผ่นที่ของร้านอาหาร คือ X = %f , Y = %f'%(x,y))
8
9  #พิกัดบนแผนที่ของโรงเรียน
10 XSchool=2
11 YSchool=3
12
13 #คำนวณระยะห่างระหว่างโรงเรียนกับร้านอาหาร
14 Distance=math.sqrt(pow(x-XSchool,2)+pow(y-YSchool,2))
15 print('ร้านอาหารห่างจากโรงเรียนเป็นระยะทาง %f กิโลเมตร'%(Distance))
16
17 #ตรวจสอบเงื่อนไขว่าระยะห่างเกินมีค่าเกิน 1 หรือไม่
18 if Distance<=1:
19     print("ร้านอาหารอยู่ห่างจากโรงเรียน ไม่เกิน 1 กิโลเมตร")    # เมื่อเงื่อนไขเป็นจริง
20 else:
21     print("ร้านอาหารอยู่ห่างจากโรงเรียน เกิน 1 กิโลเมตร")      # เมื่อเงื่อนไขไม่เป็นจริง
```

2. การสร้างเงื่อนไขด้วยตัวดำเนินการตรรกะ

เงื่อนไขบางเงื่อนไข เช่น “รถประจำทางถึงโรงเรียนแล้ว” หรือ “รถยนต์มีความเร็วเหมาะสม” เป็นเงื่อนไขที่ระบุด้วยประโยคที่ชัดเจน ในการออกแบบขั้นตอนวิธี เราสามารถใช้เงื่อนไขเช่นนี้ได้ อย่างไรก็ตามระหว่างที่เราวิเคราะห์บางครั้ง จะพบว่าเงื่อนไขที่ระบุด้วยประโยคลักษณะ ถ้าพิจารณาด้วยแนวคิดการแยกส่วนประกอบและการย่อยปัญหาอาจจะประกอบด้วยเงื่อนไขย่อย ๆ อีกที เช่น เงื่อนไข “รถยนต์มีความเร็วที่เหมาะสม” อาจมีความหมายว่า รถยนต์มีความเร็วมากกว่า 40 กิโลเมตรต่อชั่วโมง และไม่เกิน 90 กิโลเมตรต่อชั่วโมง สังเกตว่าเงื่อนไขนี้ ประกอบด้วยเงื่อนไขย่อยสองเงื่อนไข และเชื่อมกันด้วยตัวดำเนินการตรรกะ “และ” นอกจากตัวดำเนินการ “และ” แล้ว ตัวดำเนินการที่พบบ่อยในการออกแบบขั้นตอนวิธี คือ “หรือ” และ “นิเสธ” (สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ, 2562)

เรื่องที่ 4 การทำซ้ำ

สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ (2562) ได้กล่าวถึงการทำงานรูปแบบเดียวกันซ้ำๆ หลายรอบ ซึ่งลักษณะการทำซ้ำ เช่น การทำซ้ำในรายการ การทำซ้ำด้วยเงื่อนไข

การทำซ้ำในรายการ

การทำซ้ำในรายการจะพิจารณาข้อมูลในรายการทีละตัวจนครบทุกรายการ โดยมีรูปแบบการพิจารณาคือ

1. ให้ตัวแปร x แทนข้อมูลที่พิจารณาอยู่
2. ประมวลผลตัวแปร x

ตัวอย่างการทำซ้ำในรายการ

ถ้าผู้เรียนมีเงิน M บาท และมีรายการราคาสินค้า A สามารถเขียนขั้นตอนวิธี นับจำนวนสินค้าที่มีราคาไม่เกิน M บาทได้ดังนี้

ขั้นตอนวิธี : หาจำนวนสินค้าที่มีราคาไม่เกิน M บาท

ข้อมูลเข้า : ราคาสินค้าในรายการ A

ข้อมูลออก : จำนวนสินค้าที่มีราคาไม่เกิน M บาท

ให้ตัวแปร count ⟵ 0

พิจารณาข้อมูลราคาสินค้าในรายการ A ทีละจำนวน จนครบ

2.1 ให้ x แทนข้อมูลราคาสินค้าที่พิจารณาอยู่

2.2 ถ้า x น้อยกว่าหรือเท่ากับ M แล้ว

ให้ count ⟵ count + 1

คืนค่าจำนวนสินค้าเท่ากับ count

ตัวอย่างโปรแกรมภาษา Python ตรวจสอบหาพิกัด (CheckPrice.py)

```
#เก็บข้อมูลราคาสินค้าจำนวน 50 ชิ้น ใน List
A=[18,42,36,25,37,23,42,37,41,16,28,47,33,35,44,48,49,25,28,29,26,
24,45,21,38,38,31,44,35,37,31,39,26,28,29,24,25,34,26,43,65,78,90,
85,100,75,89,63,71,94]

M=float(input("กรอกจำนวนเงินของคุณ (บาท) : "))

#กำหนดค่าตัวแปร count เป็น 0
count=0

# วนซ้ำเปรียบราคาสินค้ากับจำนวนเงิน
for x in A:
    if x<=M:
        count=count+1  #เพิ่มจำนวนรายการเมื่อเงื่อนไขเป็นจริง

print('มีสินค้าที่ราคาต่ำกว่า %d บาท จำนวน %d รายการ'%(M,count))
```

การทำซ้ำด้วยเงื่อนไข

การทำซ้ำแบบมีเงื่อนไขเป็นการทำซ้ำที่มีเงื่อนไขในการหยุดการทำซ้ำ

ตัวอย่าง ถ้าต้องการประมาณค่าของรากที่สองของ 10 ที่เป็นทศนิยม 3 ตำแหน่ง เขียนขั้นตอนวิธีได้ ดังนี้

ขั้นตอนวิธี : ประมาณค่ารากที่สองของ 10 ที่เป็นทศนิยม 3 ตำแหน่ง

ข้อมูลเข้า : -

ข้อมูลออก : ค่าประมาณของรากที่สองของ 10 ที่เป็นทศนิยม 3 ตำแหน่ง

1. ให้ s ⟵ 0
2. ให้ a ⟵ 0 (เก็บค่าประมาณที่ดีที่สุด)
3. ทำซ้ำในขณะที่ s ≤ 10
   - 3.1 ถ้า $|s^2 - 10| < |a^2 - 10|$ แล้ว a ⟵ s
   - 3.2 s ⟵ s + 0.001
4. คืนค่า a และจบการทำงาน

จากขั้นตอนวิธีข้างต้นเป็นการทำซ้ำด้วยเงื่อนไข s ≤ 10 ดังนั้นจะมีการทำซ้ำในข้อ 3.1 และ 3.2 จนกว่า s จะมีค่ามากกว่า 10

ตัวอย่างโปรแกรมภาษา Python ประมาณค่ารากที่สองของ 10 (AproximateSqrt10.py)

```
1  #ประมาณค่ารากที่สองของ 10 ที่เป็นทศนิยม 3 ตำแหน่ง
2  s=0
3  a=0
4  # ทำซ้ำในขณะเงื่อนไข
5  while s<=10:
6      if abs(pow(s,2)-10)<abs(pow(a,2)-10):  # ตรวจสอบเงื่อนไขเพื่อหาค่าที่ดีที่สุด
7          a=s                                # เก็บค่าประมาณที่ดีที่สุด
8      s=s+0.001                              # เพิ่มค่าให้กับ S
9
10 print('ค่าประมาณรากที่สองของ 10 ที่เป็นทศนิยม 3 ตำแหน่ง คือ %.3f'%(a))
```

ออกแบบขั้นตอนวิธีที่มีเงื่อนไขและการทำซ้ำ โดยใช้ code.org

หลักสูตร 1 (4-6 ปี) ที่ได้รับการออกแบบสำหรับผู้ที่เพิ่งเริ่มอ่านออกเขียนได้ ผู้เรียนจะสร้างโปรแกรมคอมพิวเตอร์ที่ช่วยให้พวกเขาเรียนรู้ที่จะทำงานร่วมกันกับผู้อื่น พัฒนาทักษะการแก้ปัญหา และมีความอดทนกับงานที่ยากลำบาก เมื่อจบหลักสูตรนี้ ผู้เรียนจะสร้างเกมหรือเรื่องราวที่เป็นของเขาเองขึ้นมา

Course 1

หลักสูตร 2 (6-18 ปี) ได้รับการออกแบบมาสำหรับผู้เรียนที่สามารถอ่านได้และไม่มีประสบการณ์ในการเขียนโปรแกรมมาก่อน ผู้เรียนจะได้สร้างโปรแกรมในการแก้ปัญหาและพัฒนาเกมแบบโต้ตอบหรือเรื่องราวที่พวกเขาสามารถแบ่งปันได้

Course 2

หลักสูตร 3 (8-18 ปี)

Course 3

หลักสูตร 4 (10-18 ปี)

Course 4

เรื่องที่ 5 การจัดเรียงและค้นหาข้อมูล

ในบทเรียนนี้จะได้เรียนรู้กับขั้นตอนวิธีพื้นฐานในการจัดเรียงข้อมูล (Sort) และการค้นหาข้อมูล (Search) ซึ่งเป็นกิจกรรมที่สัมพันธ์กันที่ใช้ในการแก้ปัญหาที่พบบ่อยในชีวิตประจำวัน

การจัดเรียงข้อมูล

กิตติ์ชญาห์ เมธาวรารักษ์ (2563) กล่าวว่าการจัดเรียงข้อมูลเป็นสิ่งที่พบอยู่เสมอเมื่อต้องการประมวลผลข้อมูลเป็นจำนวนมาก เช่น ครูตรวจข้อสอบของผู้เรียนและต้องการบันทึกคะแนนลงสมุดบันทึกคะแนนผู้เรียนที่มีการเรียงเลขที่เอาไว้ การเรียงลำดับข้อมูลด้วยเงื่อนไขที่เหมาะสมจะทำให้การค้นหาข้อมูล ทำได้อย่างมีประสิทธิภาพ

โดยทั่วไปการเรียงลำดับจำนวนเต็ม อาจใช้**การจัดเรียงข้อมูลได้ 2 แบบ** คือ

1. การจัดเรียงแบบเลือก (Selection sort)

การเรียงลำดับแบบเลือก เป็นขั้นตอนวิธีการเรียงลำดับอย่างง่ายโดยใช้วิธีการเปรียบเทียบ ทำงานโดยการหาค่าเหมาะสมที่สุด (ค่ามากสุดหรือน้อยสุด) ที่อยู่ในรายการส่วนที่ยังไม่เรียงและนำค่าเหมาะที่สุดนั้นมาต่อท้ายของส่วนที่เรียงแล้ว

การจัดเรียงแบบเลือก (Selection sort)

ที่มาภาพ : https://krukitchaya.wordpress.com/ม-4-เทอม-1/เทคโนโลยีวิทยาการคำนวณ/8-การจัดเรียงและค้นหาข้อ/

2. การเรียงลำดับแบบแทรก (Insertion sort)

การเรียงลำดับแบบแทรก เป็นขั้นตอนวิธีการเรียงลำดับอย่างง่าย ทำงานโดยจะแบ่งข้อมูลในรายการเป็นสองส่วนคือส่วนที่เรียงแล้วและส่วนที่ยังไม่เรียง แน่นอนว่าในตอนเริ่มแรกส่วนที่เรียงแล้วก็จะมีอย่างน้อยหนึ่งตัว และจะเริ่มหยิบข้อมูลตัวหนึ่งของส่วนที่ยังไม่เรียงมาเปรียบเทียบเพื่อหาตำแหน่งที่เหมาะสมในการแทรกลงในข้อมูลส่วนที่เรียงแล้ว ลักษณะเดียวกับการเรียงไพ่ในมือ ดังนั้นการเรียงลำดับแบบแทรกจึงไม่เหมาะในการทำงาน ในรายการที่มีจำนวนสมาชิกมากๆ (กิตติ์ชญาห์ เมธาวรารักษ์, 2563)

การค้นหาข้อมูลแบบตามลำดับ (Sequential Search)

การค้นหาข้อมูลแบบลำดับ (Sequential Search) การหาข้อมูลแบบเป็นลำดับขั้นตอน โดยจะค้นหาตั้งแต่ตัวแรกเรียงลำดับไปทีละตัวจนกว่าจะพบข้อมูลที่ต้องการหรือเปรียบเทียบไปจนถึงตัวสุดท้าย การค้นหาวิธีนี้เป็นวิธีที่ง่ายที่สุด อัลกอริทึมในการค้นหาไม่ซับซ้อนสามารถใช้กับข้อมูลที่เรียงลำดับแล้วหรือข้อมูลที่ยังไม่ได้เรียงลำดับก็ได้ โดยผลลัพธ์จากการค้นหาข้อมูล จะมีความเป็นไปได้อยู่ 2 แบบ คือ

1. **พบตำแหน่งข้อมูล** ที่ต้องการภายในลิสต์ (Successful Search)
2. **ไม่พบตำแหน่งข้อมูล** ที่ต้องการภายในลิสต์ (Unsuccessful Search)

การค้นหาข้อมูลแบบทวิภาค (Binary Search)

การค้นหาข้อมูลแบบทวิภาค เหมาะสำหรับค้นหาข้อมูลที่มีการเรียงลำดับอยู่แล้ว โดยการค้นหาแต่ละรอบจะลดขอบเขตการค้นหาลงทีละครึ่ง การค้นหาข้อมูลแบบทวิภาคมีประสิทธิภาพดีมากและเป็นแนวคิด หลักในการพัฒนาระบบฐานข้อมูล หลังพิจารณาข้อมูลแต่ละครั้ง ขอบเขตของดัชนีที่เป็นไปได้จะลดลงประมาณครึ่งหนึ่ง ถ้าข้อมูลในรายการมีจำนวน n ตัว จำนวนรอบที่ต้องทำงานจะเท่ากับจำนวนครั้งในการลดค่าขอบเขตที่เป็นไปได้จาก n ทีละครึ่งจนเหลือค่าเท่ากับ 1 ซึ่งค่าดังกล่าวสอดคล้องกับฟังก์ชันลอการิทึม (logarithm) ฐาน 2 ของ n ดังนั้นความซับซ้อนของ ขั้นตอนวิธีการค้นหาแบบทวิภาคจะแปรผันตรงกับ $\log_2 n$ นั่นคือเรา สามารถเขียนว่าการค้นหาแบบทวิภาคมีความซับซ้อนเป็น 0 $\left(\log_2 n\right)$ (กิตติ์ชญาห์ เมธาวรารักษ์, 2563)

เทคนิคการค้นหาข้อมูลด้วยวิธีนี้

1. กำหนดข้อมูลที่ต้องการค้นหาและทำการเรียงข้อมูลตามความต้องการ เรียงจากมากไปน้อย หรือจากน้อยไปมากก็ได้

2. ทำการแบ่งข้อมูลออกเป็น 2 ส่วน แล้วทำการหาค่ากลาง

3. เมื่อทราบแล้วว่าค่าของคีย์ฟิลด์อยู่ครึ่งแรกหรือครึ่งหลังแล้ว ก็จะนำข้อมูลในครึ่งดังกล่าวทำการหาค่ากลางอีก ทำแบบนี้ไปเรื่อยๆ จนกระทั้งได้ข้อมูลที่ต้องการ หรือจนกระทั่งไม่สามารถแบ่งข้อมูลได้อีก

จะเห็นได้ว่า Binary Search สามารถหาข้อมูลที่ต้องการได้เร็วกว่า Sequential Search เมื่อมีจำนวนข้อมูลจำนวนมาก แต่ถ้าหากมีข้อมูลน้อยๆ และสิ่งที่ต้องการหานั้นอยู่เป็นต้นๆ ก็จำทำให้การหาแบบ Sequential Search เร็วกว่า (กิตติ์ชญาห์ เมธาวรารักษ์, 2563)

แบบฝึกหัดท้ายบทที่ 2

พิจารณารายการข้อมูลต่อไปนี้

8, 9, 13, 35, 42, 44, 50, 54, 58, 60, 61, 62, 77, 84, 86, 90, 92, 96

ให้เขียนข้อมูลจากรายการลงในตารางด้านล่างให้ตรงตามดัชนีในแถวแรก

ดัชนี	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ข้อมูล									58									

จากนั้น ให้แสดงการทำงานของขั้นตอนวิธีในการหา 62 ในรายการดังกล่าว โดยเติม ค่าตัวแปร เมื่อเริ่มต้นการทำงานในแต่ละรอบ

n = ...

รอบที่	**Left**	**Right**	**Mid**	**x**	**ผลการเปรียบเทียบ Target**
1	1	18	9	58	น้อยกว่า
2	10	18			
3					
4					
5					