# บทที่ 2

## ขั้นตอนการทำงาน

## Algorithm

ตามที่ได้กล่าวมาแล้วว่าก่อนที่จะลงมือเขียนโปรแกรมเราจะต้องออกแบบขั้นตอนการทำงานหรืออัลกอริทึม (Algorithm) ก่อน โดยจะเป็นเครื่องมือในการแสดงขั้นตอนการทำงานของระบบงานใด ๆ เพื่อให้การเขียนโปรแกรมเป็นไปได้รวมเร็วและง่ายขึ้น โดยเราอาจเขียนอัลกอริทึมในลักษณะผังงาน (Flowchart) หรือรหัสจำลองที่เรียกว่าซูโดโค้ด (Psudocodes) ก็ได้

### 2.1 ซูโดโค้ด (Psudocodes)

ซูโดโค้ดเป็นคำอธิบายขั้นตอนการทำงานของโปรแกรมโดยใช้ถ้อยคำผสมระหว่างภาษาอังกฤษและภาษาการเขียนโปรแกรมแบบโครงสร้าง หรืออาจใช้ภาษาไทยก็ได้แต่ควรเขียนเป็นภาษาอังกฤษ โดยให้ผู้เขียนโปรแกรมสามารถพัฒนาขั้นตอนต่าง ๆ ให้เป็นโปรแกรมได้ง่ายขึ้น แต่ส่วนใหญ่แล้วคำที่ใช้มักเป็นคำเฉพาะ (Reserve Word) ที่มีอยู่ในภาษาการเขียนโปรแกรมและมักจะเขียนด้วยตัวอักษรตัวใหญ่ ซูโดโค้ดที่ดีจะต้องมีความชัดเจน สั้น และได้ใจความ ข้อมูลต่าง ๆ ที่ใช้จะถูกเขียนอยู่ในรูปของตัวแปร ซูโดโค้ดนี้บางครั้งจะเรียกว่าอัลกอริทึม รูปแบบทั่วไปจะเป็นดังนี้

```
Algorithm < ชื่อของอัลกอริทึม >
1. ……..
2. ……..                    ◄──────── ขบวนการทำงานต่าง ๆ
……..
……..
END
```

ตัวอย่างเช่นในการเขียนซูโดโค้ดสำหรับให้คอมพิวเตอร์หาค่าเฉลี่ยจากข้อมูลที่รับเข้าทางแป้นพิมพ์อาจเขียนได้ดังนี้

Algorithm การหาค่าเฉลี่ย

1. ตัวนับ = 0
2. ผลรวม = 0
3. รับค่าทางแป้นพิมพ์เก็บไว้ในตัวแปร (ข้อมูล)
4. ถ้า ข้อมูล มากกว่า 0
   เพิ่มค่าตัวนับขึ้นหนึ่งค่า
   ผลรวม = ผลรวม + ค่าข้อมูล
   ย้อนกลับไปทำขั้นตอนที่ 3
   ถ้าไม่มากกว่าไปทำขั้นตอนที่ 5
5. ค่าเฉลี่ย = ผลรวมหารด้วยตัวนับ
6. แสดงค่าเฉลี่ยทางจอภาพ โดยมีทศนิยมสองตำแหน่ง
7. จบ

จะเห็นว่าขั้นตอนการหาค่าเฉลี่ยได้เขียนไว้อย่างเข้าใจ เราสามารถทราบได้ว่าในการทำงานต่าง ๆ จะต้องใช้ตัวแปรใดบ้าง แต่ละขั้นตอนมีการประมวลผลอย่างไร แต่โดยทั่วไปแล้วซูโดโค้ดจะถูกเขียนด้วยภาษาอังกฤษ ดังต่อไปนี้

```
Algorithm Avarage_Sum
1. count = 0
2. sum = 0
3. INPUT (value)
4. IF value > 0 THEN
                count = count +1
                sum = sum + value
                GOTO 3
   ELSE GOTO 5
5. avarage = sum / count
6. OUTPUT (avarage)
7. END
```

ในการคำนวณหาพื้นที่สามเหลี่ยมเราอาจเขียนซูโดโค้ดได้ดังต่อไปนี้

```
ซูโดโค้ดหาพื้นที่สามเหลี่ยม
เริ่มต้น
        1. รับค่าความยาวของด้านที่เป็นฐานมาเก็บในตัวแปร X
        2. รับค่าความยาวของส่วนสูงมาเก็บในตัวแปร Y
        3. คำนวณพื้นที่โดย  ARRAY  = (X * Y)/2
        4. แสดงผลพื้นที่
จบ
```

หรืออาจเขียนเป็นภาษาอังกฤษได้เป็น

```
START
        1. READ  X
        2. READ  Y
        3. Compute  ARRAY  =  (X * Y)/2
        4. Print ARRAY
END
```

แม้ว่าการเขียนซูโดโค้ดจะไม่มีรูปแบบที่แน่นอน แต่โดยทั่วไปแล้วมักจะทำกันดังลักษณะต่อไปนี้

**การรับข้อมูลเข้าและการแสดงผลข้อมูล**

ในการรับข้อมูลจะนิยมใช้คำว่า READ หรือ INPUT ตามด้วยตัวแปรที่ต้องการใช้เก็บข้อมูล ถ้าหากมีตัวแปรหลายตัวจะใช้เครื่องหมายคอมมา (“,”) คั่น ส่วนการแสดงผลมักใช้คำว่า PRINT

**การคำนวณ**

ในการประมวลผลแบบคำนวณจะขึ้นต้นด้วยคำว่า Compute ตามด้วยตัวแปรที่ต้องการเก็บค่าจากการคำนวณ เครื่องหมายเท่ากับและนิพจน์การคำนวณ ตัวอย่างเช่น

```
Compute  ARRAT  =  (X * Y)/2
```

**การตัดสินใจและทดสอบทางเลือก**

การตัดสินใจเพื่อเลือกทำระหว่างทางสองทางจะใช้คำว่า IF หรือ IF-THEN-ELSE และ ENDIF โดยจะเปรียบเทียบเงื่อนไข ถ้าเงื่อนไขเป็นจริงจะทำกลุ่มคำสั่ง (Statement) กลุ่มหนึ่ง ถ้าเป็นเท็จจะทำกลุ่มคำสั่งอีกกลุ่มหนึ่ง ตัวอย่างเช่น

```
IF  number > 0 THEN
        PRINT  POSITIVE NUMBER
ELSE
        PRINT  NEGATIVE NUMBER
ENDIF
```

จากตัวอย่างหมายความว่าถ้าค่า number มีค่ามากกว่า 0 ให้คอมพิวเตอร์พิมพ์คำว่า POSITIVE NUMBER ถ้าไม่มากกว่าจะพิมพ์คำว่า NEGATIVE NUMBER

สำหรับกรณีที่มีทางเลือกมากกว่าสองทางจะใช้คำว่า CASE และ ENDCASE โดยจะทำกลุ่มคำสั่งที่มีค่านำหน้าเท่ากับตัวแปรที่อยู่หลัง CASE ตัวอย่างเช่น

```
CASE  num OF
        1 :     PRINT 11111
        2 :     PRINT 22222
        3 :     PRINT 33333
ENDCASE
```

จากตัวอย่างถ้าค่าในตัวแปร num เป็น 1 จะให้พิมพ์คำว่า 11111 ถ้าตัวแปร num มีค่าเป็น 2 จะให้พิมพ์คำว่า 22222

**การทำแบบวนซ้ำ**

ในการทำซ้ำหมายความว่าให้ระบบทำงานซ้ำ ๆ ตามเงื่อนไขที่กำหนด โดยจะมีการเปรียบเทียบเงื่อนไขในการทำซ้ำ แบ่งออกได้สามรูปแบบดังนี้

1. การทำซ้ำที่มีการเพิ่มค่าในแต่ละรอบ จะใช้คำว่า FOR และ ENDFOR โดยมีคำว่า IN STEPS OF เป็นการบอกค่าที่เพิ่มในแต่ละรอบ ถ้าไม่มีคำว่า IN STEPS OF หมายความว่าเพิ่มค่ารอบละหนึ่ง
2. การทำซ้ำจนระบบมีเงื่อนไขอย่างหนึ่งจึงหยุดทำ จะใช้คำว่า REPRAT – UNTIL ดังรูปแบบต่อไปนี้

```
REPEAT
        Statement_1
        ……
UNTIL (Condition)
```

3. ถ้าเงื่อนไขเป็นจริงจะทำคำสั่งภายใน จะใช้คำว่า DO – WHILE โดยจะตรวจสอบเงื่อนไขก่อนที่ทำชุดคำสั่งภายใน ดังรูปแบบต่อไปนี้

```
DO (Condition)  WHILE
        Statement_1
        ……
ENDDO
```

**การกระโดดข้าม**

การกระโดดข้ามไปทำชุดคำสั่งใด ๆ จะใช้คำว่า LABEL กำหนดตำแหน่งที่จะกระโดดมาและใช้คำว่า GOTO ในตำแหน่งที่จะกระโดด ตัวอย่างเช่น

```
START :
                Statement_1
                ……..
        AB1:    ……
                ……
                GOTO AB1
END
```

**ตัวอย่าง 2.1** ถ้าหากต้องการเขียนซูโดโค้ดในการบวกเลข 1 + 2 + 3 +.....+100 และพิมพ์ผลลัพธ์ออกมาอาจเขียนได้ดังนี้

```
START
        I = 0
        SUM  =  0
        DO ( I <= 100 ) WHILE
                Compute  SUM  =  SUM + I
                Conpute  I      =  I + 1
        ENDDO
        PRINT  SUM
END
```

## 2.2 การเขียนผังงาน (Flowchart)

ผังงานหรือเรียกกันว่าโฟลวชาร์ต เป็นแผนภาพที่ใช้ออกแบบและอธิบายการทำงานของโปรแกรมโดยอาศัยรูปทรงต่าง ๆ ควบคู่ไปกับลูกศร แต่ละรูปในแผนภาพจะหมายถึงการทำงานหนึ่งขั้นตอน ส่วนลูกศรจะแทนลำดับการทำงานขั้นตอนต่าง ๆ รวมทั้งทิศทางการไหลของข้อมูลตั้งแต่เริ่มต้นจนได้ผลลัพธ์ตามต้องการ ระบบงานทุกชนิดที่ผ่านการวิเคราะห์เป็นลำดับขั้นตอนแล้วจะสามารถเขียนเป็นผังงานได้

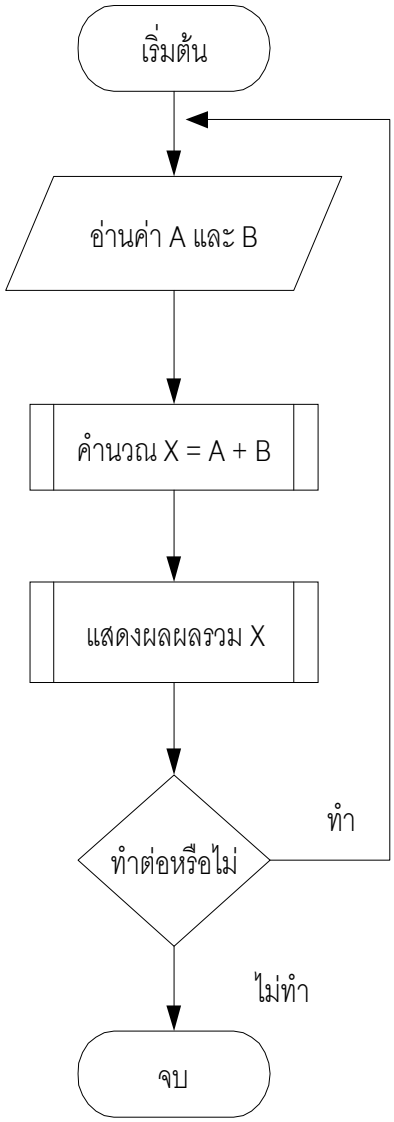
### ประโยชน์ของผังงาน

1. ช่วยอธิบายลำดับขั้นตอนการทำงานของโปรแกรม
2. ทำให้ตรวจสอบข้อผิดพลาดของโปรแกรมได้ง่าย
3. ทำให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมและแก้ไขโปรแกรมได้ง่าย

### การเขียนผังงานที่ดี

1. เขียนตามสัญลักษณ์ที่กำหนด
2. ใช้ลูกศรแสดงทิศทางการทำงานจากบนลงล่าง
3. อธิบายสั้น ๆ ให้เข้าใจง่าย
4. ทุกแผนภาพต้องมีทิศทางเข้าออก
5. ไม่ควรโยงลูกศรไปที่ไกลมาก ๆ ถ้าต้องทำให้ใช้สัญลักษณ์การเชื่อมต่อแทน

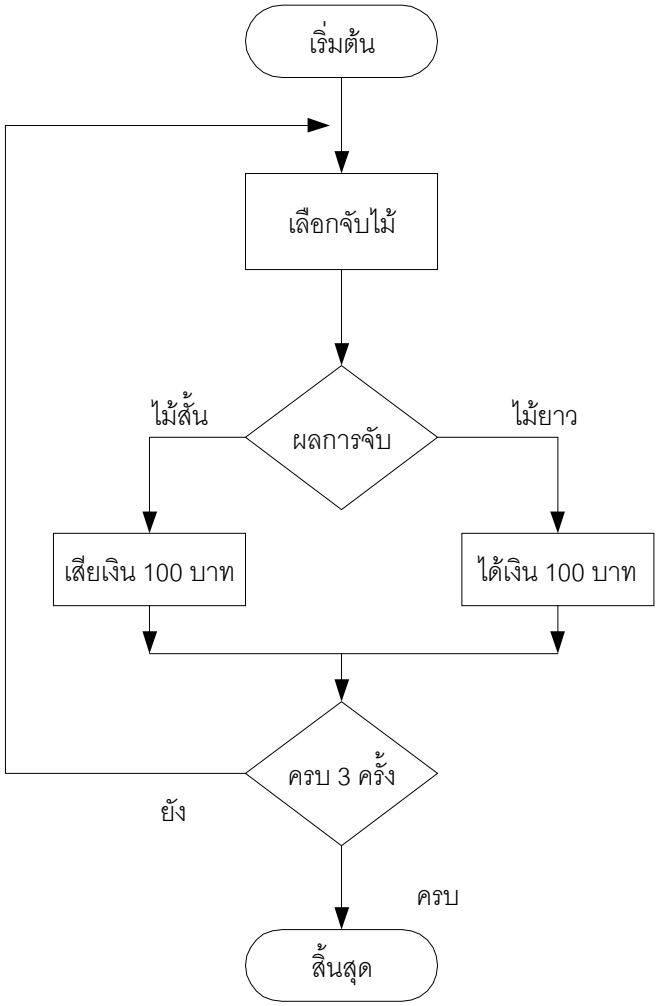
**ตัวอย่าง** 2.2 ถ้าหากเราต้องการให้คอมพิวเตอร์ทำงานเป็นเครื่องคิดเลข โดยรับข้อมูล A และ B เข้าไป จากนั้นแสดงผลบวกออกมาเราสามารถเขียนเป็นผังงานได้ดังนี้




รูปที่ 2.1 ผังงานแสดงการทำงานของโปรแกรมบวกเลข

นอกจากนี้ผังงานยังสามารถนำมาใช้กับงานที่เกี่ยวข้องกับชีวิตประจำวันได้

**ตัวอย่าง** 2.3 ผังงานในการจับไม้สั้นไม้ยาวสามครั้ง ถ้าได้ไม้ยาวให้ได้เงิน 100 บาท ถ้าได้ไม้สั้นให้เสียเงิน 100 บาท สามารถเขียนได้ดังนี้




รูปที่ 2.2 แสดงผังงานการจับไม้สั้นไม้ยาว

การเขียนผังงานสามารถแบ่งออกเป็นสองประเภทใหญ่ ๆ คือ

1. **ผังงานระบบ** (System Flowchart)

ใช้แสดงขั้นตอนการทำงานภายในระบบงานหนึ่ง ๆ โดยกล่าวถึงข้อมูลต่าง ๆ ที่เกี่ยวข้องทั้งหมด เช่น เอกสารเบื้องต้นคืออะไร วัสดุที่ใช้คืออะไร ใช้หน่วยความจำประเภทใด จะต้องส่งผ่านไปยังหน่วยงานใด วิธีการประมวลผลและการแสดงผลลัพธ์ โดยจะกล่าวอย่างกว้าง ๆ ไม่สามารถนำมาเขียนเป็นโปรแกรมได้

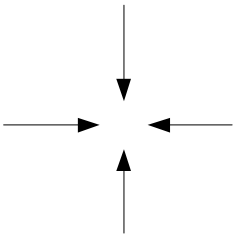
2. **ผังงานโปรแกรม** (Program Flowchart)

ผังงานประเภทนี้จะแสดงถึงขั้นตอนของคำสั่งที่ใช้ในโปรแกรม การรับข้อมูล การประมวลผล การแสดงข้อมูล บางครั้งจะเรียกว่าผังการเขียนโปรแกรม

## สัญลักษณ์ที่ใช้ในการเขียนผังงาน

การเขียนผังงานจะต้องใช้ภาพสัญลักษณ์ต่าง ๆ นำมาเรียงต่อกันเพื่อแสดงลำดับการทำงาน สัญลักษณ์มาตรฐานที่เรียกว่าสัญลักษณ์ ANSI (American National Standards Institute) ที่ควรทราบมีดังนี้

การเริ่มต้นและการสิ้นสุดการทำงานของโปรแกรม

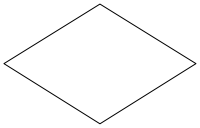


ลูกศรแสดงทิศทางการทำงานของโปรแกรมและการไหลของข้อมูล

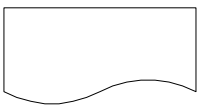
การประมวลผลการคำนวณต่าง ๆ

การทำงานย่อย

การรับหรือแสดงข้อมูลโดยไม่ระบุชนิดของอุปกรณ์



การตรวจสอบเงื่อนไข เพื่อเลือกทำอย่างใดอย่างหนึ่ง โดยจะมีเส้นลูกศรแสดงทิศทางการทำงานต่อไป



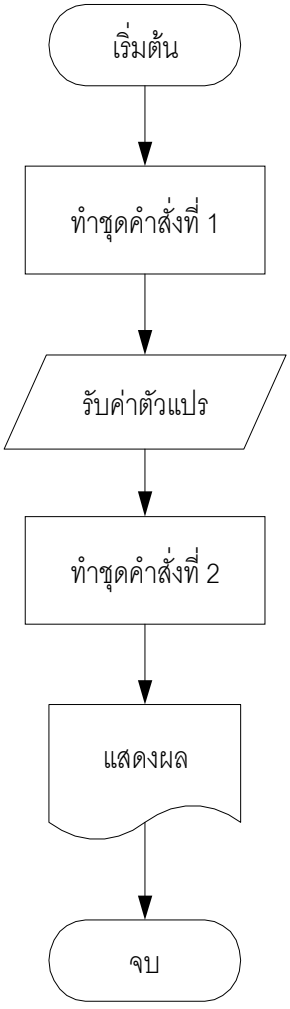
แสดงผลลัพธ์หรือรายงานทางเครื่องพิมพ์

จุดเชื่อมต่อของผังงาน

## 2.3 รูปแบบการจัดภาพของผังงาน

ต่อไปจะกล่าวถึงรูปแบบของผังงานซึ่งนิยมใช้เป็นมาตรฐานในการเขียนโปรแกรมแบบโครงสร้าง(Structure Programming) โดยโปรแกรมทุกโปรแกรมจะมีโครงสร้างการควบคุมเพื่อกำหนดทิศทางการทำงานของโปรแกรม โครงสร้างโดยทั่วไปจะมีอยู่ 5 รูปแบบดังต่อไปนี้

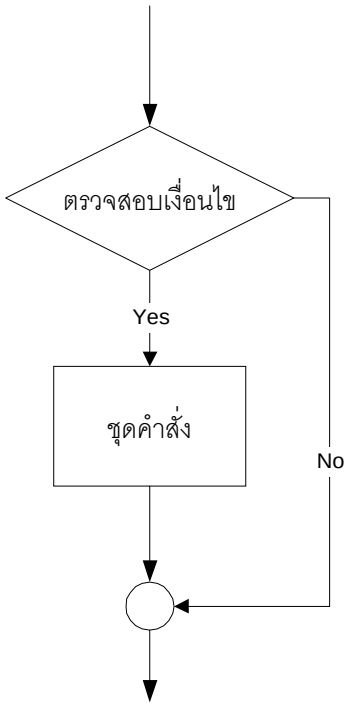
2.3.1 **โครงสร้างการทำงานแบบลำดับ** (Sequence) จะแสดงขั้นตอนการทำงานที่เรียงลำดับกับไป ไม่มีการข้ามขั้น หรือย้อนกลับไปทำคำสั่งที่ได้ทำไปแล้ว ดังตัวอย่างในรูปที่ 2.3




รูปที่ 2.3 ผังโปรแกรมแบบมีลำดับ

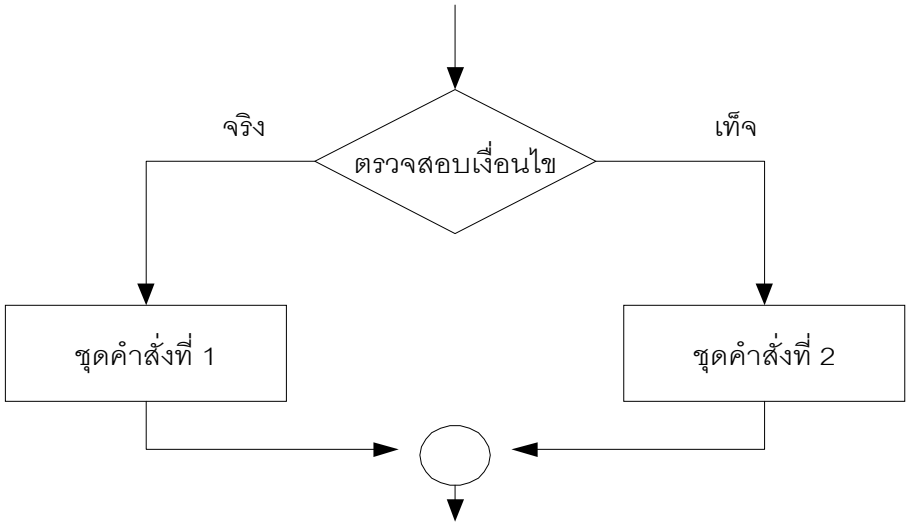
2.3.2 **ผังโปรแกรมแบบมีการเลือก** (Selection) เป็นโครงสร้างที่ตรวจสอบเงื่อนไข ให้โปรแกรมเลือกทำอย่างใดอย่างหนึ่ง ซึ่งมีอยู่ 3 กรณีดังต่อไปนี้

**การเลือกแบบหนึ่งเส้นทาง** จะทำงานเฉพาะเมื่อเงื่อนไขเป็นจริงเท่านั้น ผังงานแสดงได้ดังรูปที่ 2.4



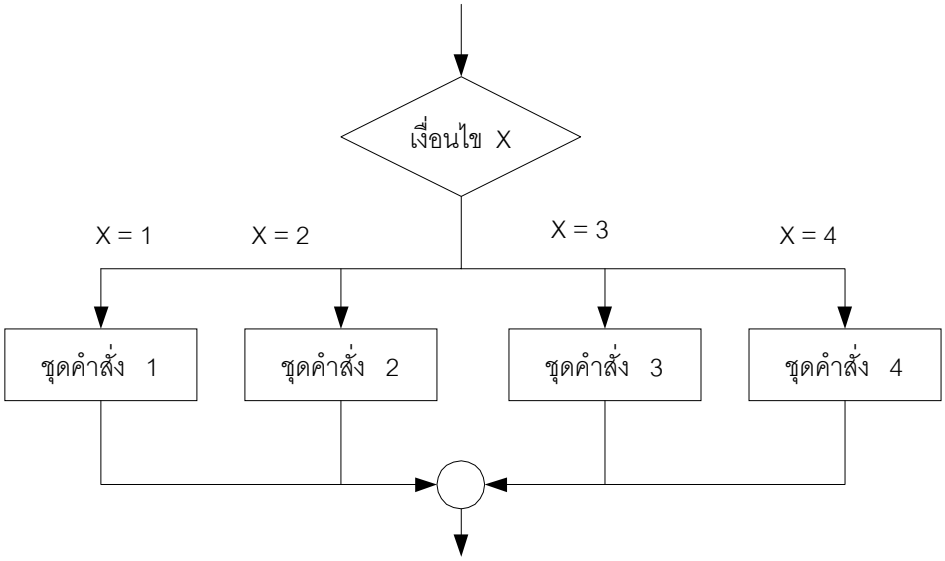

รูปที่ 2.4 ผังงานแบบเลือกทำหนึ่งเส้นทาง

**การเลือกทำแบบสองเส้นทาง** จะพิจารณาเงื่อนไขที่เป็นจริงและเป็นเท็จ โดยถ้าเป็นจริงจะทำอย่างหนึ่ง ถ้าเป็นเท็จจะทำอีกอย่างหนึ่ง ผังงานแสดงได้ดังรูปที่ 2.5



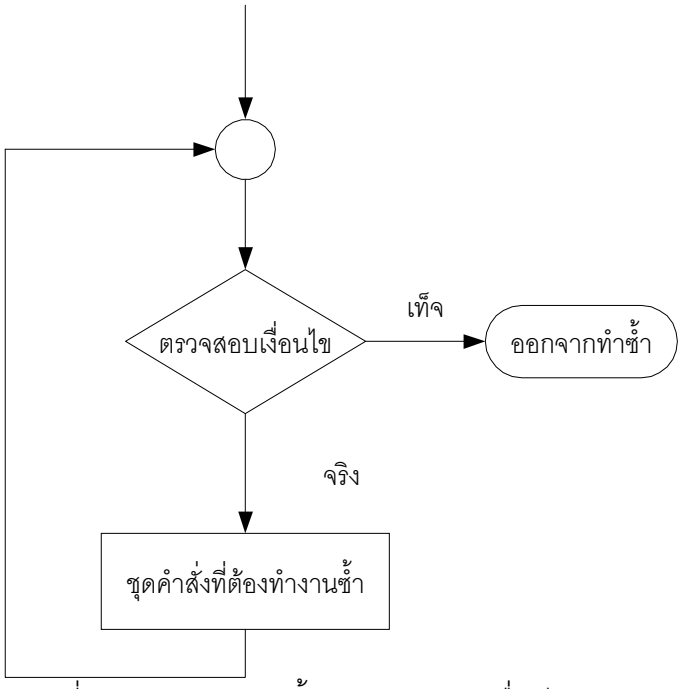

รูปที่ 2.5 แสดงผังงานสำหรับการเลือกทำอย่างใดอย่างหนึ่ง

**การเลือกทำแบบหลายเส้นทาง** จะพิจารณาเงื่อนไขต่าง ๆ ที่เกิดขึ้น ถ้าเท่ากับทางเลือกใดจะให้ไปทำงานตามทางเลือกนั้น ผังงานแบบเลือกหลายทางสามารถเขียนได้ดังรูปที่ 2.6



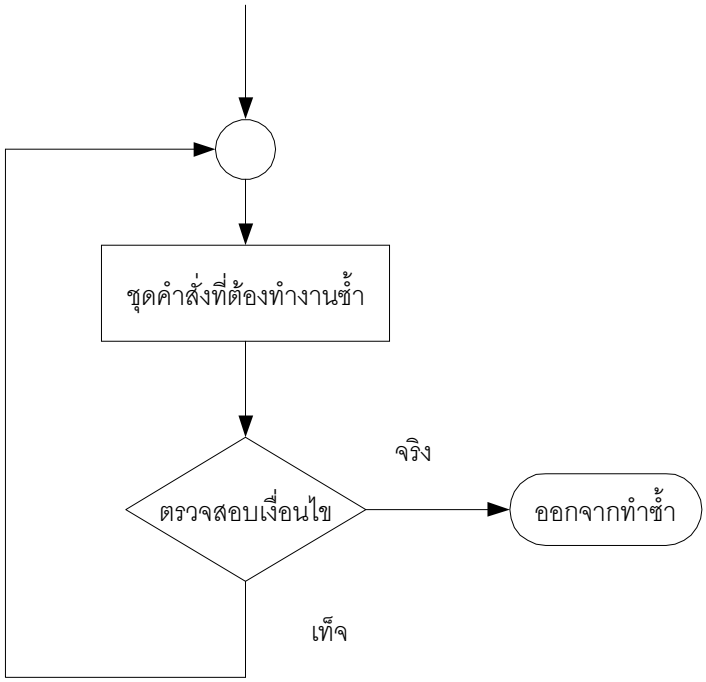

รูปที่ 2.6 ผังโปรแกรมแบบทางเลือกหลายทาง

2.3.3 **ผังโปรแกรมทำซ้ำแบบเงื่อนไขเป็นจริง** จะใช้ในงานที่มีการตรวจสอบเงื่อนไข ถ้าเป็นจริงจะทำงานซ้ำ โดยจะตรวจสอบเงื่อนไขก่อนการทำงานทุกครั้ง โดยเขียนได้ดังรูปที่ 2.7



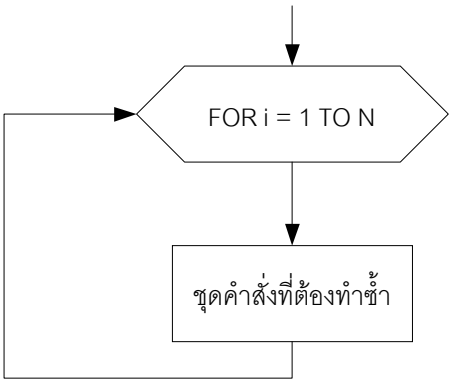

รูปที่ 2.7 ผังงานการทำซ้ำแบบตรวจสอบเงื่อนไขก่อน

2.3.4 **ผังโปรแกรมแบบทำซ้ำจนเงื่อนไขเป็นจริง** จะใช้ในระบบที่ต้องทำงานก่อนการตรวจสอบเงื่อนไข และทำงานซ้ำจนเงื่อนไขเป็นจริง สามารถเขียนได้ดังรูปที่ 2.8




รูปที่ 2.8 ผังโปรแกรมแบบทำซ้ำจนเงื่อนไขเป็นจริง

2.3.5 **ผังโปรแกรมแบบทำซ้ำตามจำนวนที่ระบุ** ใช้ในระบบที่ต้องทำงานตามจำนวนรอบที่กำหนด โดยเริ่มจากรอบเริ่มต้นไปยังรอบสุดท้าย ตามปกติแล้วค่าการนับรอบจะเพิ่มขึ้นครั้งละหนึ่งค่า โดยเขียนได้ดังรูปที่ 2.9




รูปที่ 2.9 ผังงานการทำซ้ำรอบตามจำนวนที่ระบุ

**ตัวอย่าง 2.4** ถ้าหากต้องการนำคะแนนของนักศึกษามาตัดเกรดตามเงื่อนไขต่อไปนี้

คะแนน 80 – 100 ได้เกรด A

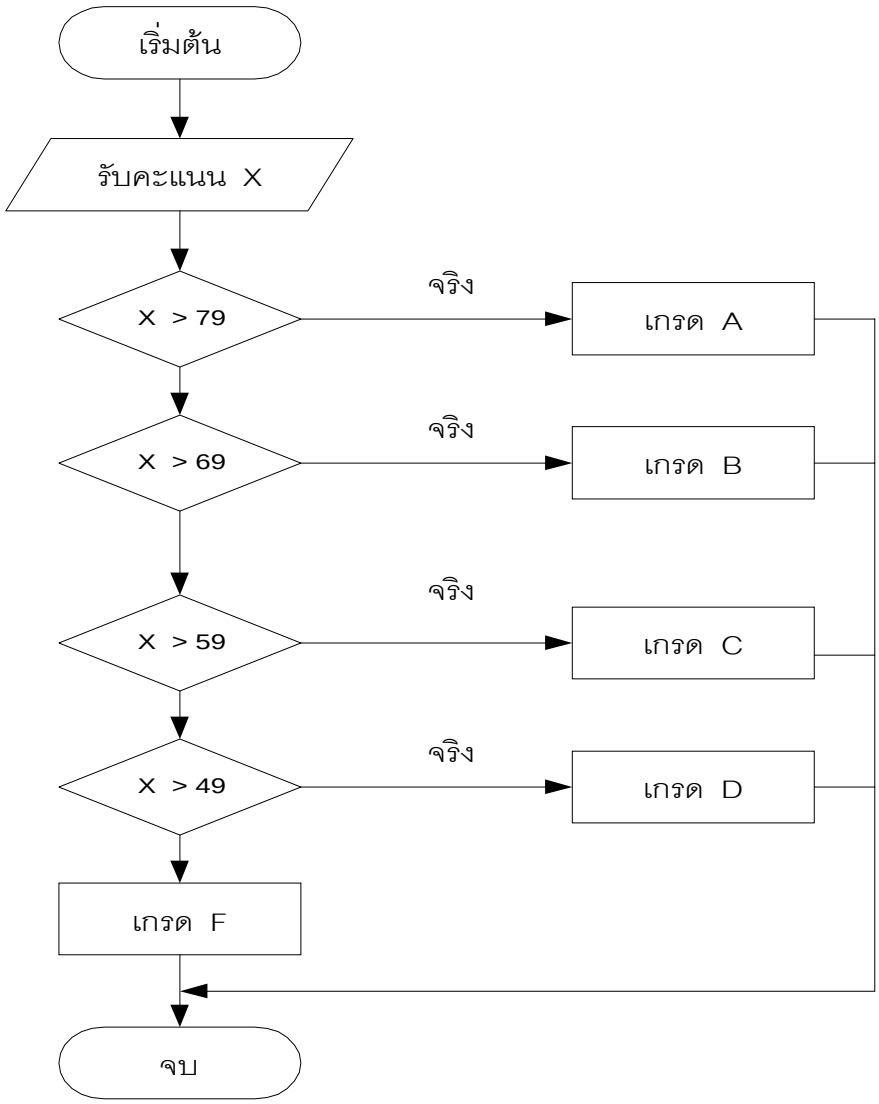
คะแนน 70 - 79 ได้เกรด B

คะแนน 60 – 69 ได้เกรด C

คะแนน 50 – 59 ได้เกรด D

คะแนนต่ำกว่า 50 ได้เกรด F

จากคะแนนดังกล่าวสามารถเขียนผังงานได้ดังรูปที่ 2.10



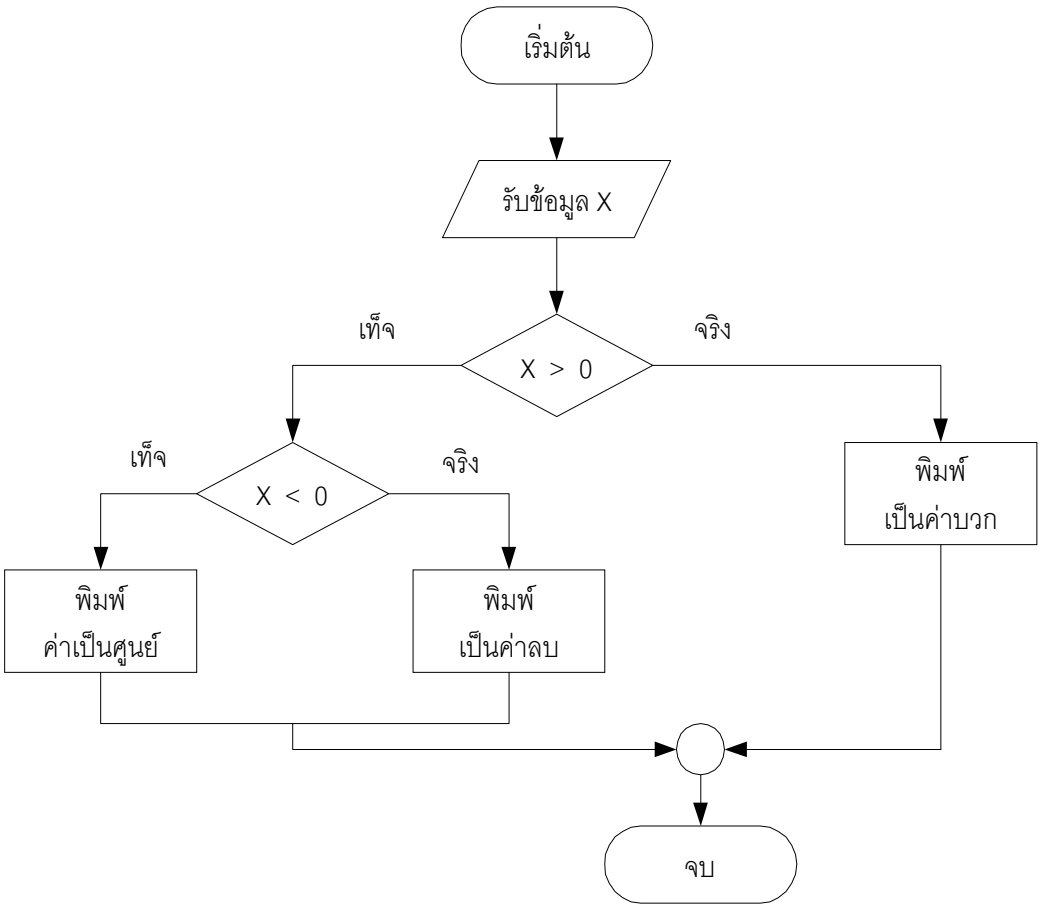

รูปที่ 2.10 แสดงผังงานของระบบงานตัดเกรดนักศึกษา

**ตัวอย่าง 2.5** จงเขียนผังงานให้รับข้อมูลตัวเลขเข้าไปหนึ่งตัว เก็บในตัวแปร X จากนั้นให้พิมพ์ตามเงื่อนไขต่อไปนี้

ถ้า X > 0 " เป็นเลขบวก "

ถ้า X < 0 " เป็นเลขลบ "

ถ้า X = 0 " ค่าเป็นศูนย์ "




รูปที่ 2.11 ผังงานของการรับค่าตัวเลขและแสดงผล

**ตัวอย่าง 2.6** จากการรับข้อมูลตัวเลขใด ๆ หนึ่งตัว จงหาว่าตัวเลขที่รับเข้ามาเมื่อพิจารณาแยกตามประเภทแล้วได้เลขศูนย์ เลขจำนวนบวก และเลขลบ ประเภทละกี่ตัว โดยเขียนซูโดโค้ดและผังงาน เพื่อแสดงรายละเอียดของการทำงาน

**วิธีทำ** การวิเคราะห์ปัญหา

| | |
|---|---|
| ข้อมูลที่รับเข้า | ตัวเลขใด ๆ หนึ่งตัว |
| ผลลัพธ์ที่ต้องการ | ตัวเลขแสดงการนับของจำนวนที่เป็นเลขศูนย์ |
| | ตัวเลขแสดงการนับของจำนวนที่เป็นเลขบวก |
| | ตัวเลขแสดงการนับของจำนวนที่เป็นเลขลบ |

**อัลกอริทึม**

1. กำหนดค่าเริ่มต้นให้กับตัวแปรต่าง ๆ
2. รับค่าข้อมูลเป็นตัวเลข
3. ตรวจสอบข้อมูล ถ้าเป็นศูนย์ ให้ตัวนับเลขศูนย์เพิ่มค่าขึ้นหนึ่ง ถ้าไม่ใช่ค่าศูนย์ ให้ตรวจสอบว่าเป็นเลขบวกหรือไม่ ถ้าเป็นให้ตัวนับเลขบวกเพิ่มค่าขึ้นหนึ่ง ถ้าไม่เป็นให้ตัวนับเลขลบเพิ่มค่าขึ้นหนึ่ง
4. แสดงผลลัพธ์ที่ต้องการทั้ง 3 ค่า
5. จบการทำงาน

**รายละเอียดของตัวแปรที่ใช้เป็นดังนี้**

| | |
|---|---|
| Num | แทนข้อมูลตัวเลขใด ๆ |
| Count_0 | แทนค่าการนับจำนวนที่เป็นศูนย์ |
| Count_Plus | แทนค่าการนับจำนวนที่เป็นเลขบวก |
| Count_Minus | แทนค่าการนับจำนวนที่เป็นเลขลบ |

สำหรับซูโดโค้ดสามารเขียนได้ดังต่อไปนี้

รูปที่ 2.12 แสดงผังงานของตัวอย่างที่ 2.6