

MIIA0106#3-1



Python and C Programming Language

การโปรแกรมภาษาโปรแกรมและภาษาซี



Sutit Ongart



ตารางสอนรายวิชา MIIA0106

Section A

วัน	คบ	เวลาเรียน	ห้อง
อังคาร (TUE)	คบ 2	12.30 – 15.00 น.	D604
อังคาร (TUE)	คบ 3	15.30 – 18.00 น.	MII203

Section B

วัน	คบ	เวลาเรียน	ห้อง
เสาร์ (SAT)	คบ 4	16.00 – 18.30 น.	D503
อาทิตย์ (SUN)	คบ 4	15.30 – 17.30 น.	MII201

สอบประจำภาค วันอาทิตย์ ที่ 15 มีนาคม พ.ศ. 2569

เช้า 09:00-12:00 น.

ขาดสอบ จะได้เป็น FE

แหล่งค้นคว้าข้อมูลเพิ่มเติม



1. พื้นฐาน การแสดงผลใน C++
2. สัญลักษณ์ **Flowchart**: ภาษาภาพสำหรับอธิบายขั้นตอนการทำงาน
3. คำสั่งตัดสินใจ

และบางเนื้อหา สรุปมาจากการสอน CPEN1110 Introduction to Computer Programming

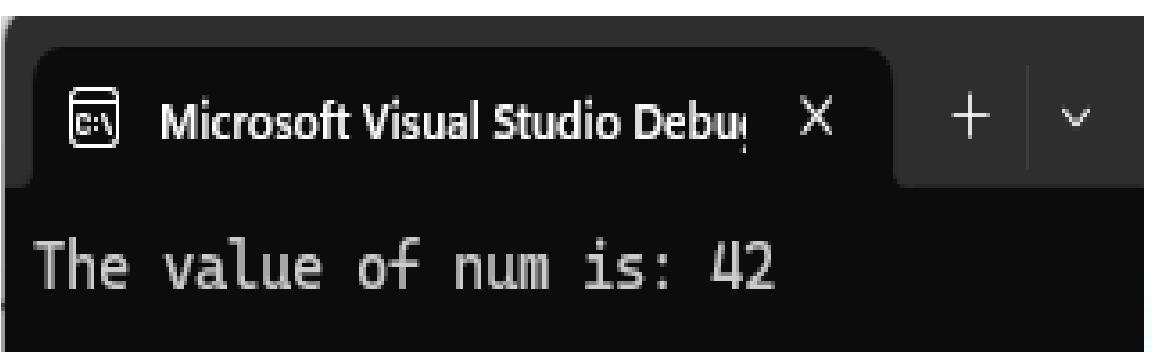
คุณสมบัติ	C++
รับข้อมูล (Input)	<code>cin >></code>
แสดงผล (Output)	<code>cout <<</code>
การจัดฟอร์แมตข้อความ	ต้องใช้ <code><<</code> และระวังช่องว่าง
การแปลงข้อมูล (Casting)	ทำงานโดยตรง (<code>int</code> , <code>float</code> , <code>ฯลฯ</code>)

Code

```
#include <iostream>

using namespace std;

int main() {
    int num = 42;
    cout << "The value of num is: " << num << endl;
    return 0;
}
```

**Code แสดงข้อมูลออกหน้าจอ**

- 1) #include <iostream>
เป็นคำสั่งสำหรับการนำเข้าไฟล์ส่วนหัว (header file) iostream ซึ่งให้โปรแกรมสามารถใช้งานคำสั่งสำหรับการอ่าน (input) และส่งออก (output) ข้อมูล เช่น cin และ cout
- 1) using namespace std;
คำสั่งนี้ทำให้โปรแกรมสามารถใช้ฟังก์ชันใน namespace std ได้โดยไม่ต้องเขียน std:: นำหน้า เช่น cout และ endl
- 1) int main()
เป็นฟังก์ชันหลักของโปรแกรมที่ทำหน้าที่เป็นจุดเริ่มต้นเมื่อโปรแกรมเริ่มทำงาน
int ระบุว่าโปรแกรมจะส่งค่าคืนให้ระบบปฏิบัติการในรูปแบบตัวเลขจำนวนเต็ม
- 1) int num = 42;
ประกาศตัวแปรชื่อ num ชนิด int (จำนวนเต็ม) และกำหนดค่าเริ่มต้นเป็น 42
- 1) cout << "The value of num is: " << num << endl;
ใช้ cout (console output) เพื่อพิมพ์ข้อความ "The value of num is: " และค่าของตัวแปร num ซึ่งคือ 42
ใช้ << เพื่อเชื่อมข้อความและค่าที่ต้องการพิมพ์
endl ใช้สำหรับขึ้นบรรทัดใหม่ (new line)
- 1) return 0;
ฟังก์ชัน main ส่งค่ากลับ 0 เพื่อบอกระบบปฏิบัติการว่าโปรแกรมทำงานสำเร็จเรียบร้อย

Code

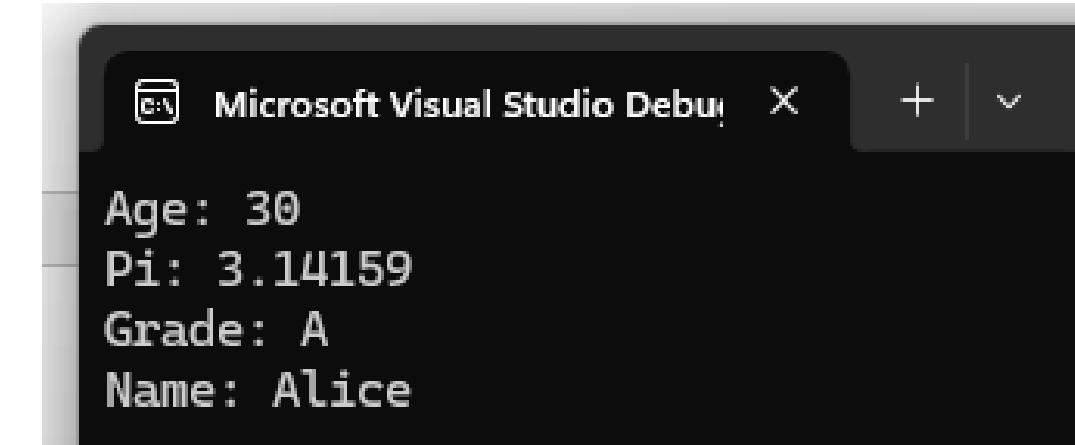
```
#include <iostream>

using namespace std;

int main() {
    int age = 30;
    double pi = 3.14159;
    char grade = 'A';
    string name = "Alice";

    cout << "Age: " << age << endl;
    cout << "Pi: " << pi << endl;
    cout << "Grade: " << grade << endl;
    cout << "Name: " << name << endl;
    return 0;
}
```

Code แสดงข้อมูลออกหน้าจอ





สัญลักษณ์ Flowchart: ภาษาภาพสำหรับอธิบายขั้นตอนการทำงาน

และบางเนื้อหา สรุปมาจากการสอนใน CPEN1110 Introduction to Computer Programming

ทำไมต้องใช้ Flowchart?

- ช่วยให้เข้าใจกระบวนการได้ง่ายขึ้น: โดยเฉพาะอย่างยิ่งสำหรับกระบวนการที่ซับซ้อน
- ใช้ในการสื่อสาร: สามารถนำเสนอให้ผู้อื่นเข้าใจได้ง่าย
- ใช้ในการวางแผน: ช่วยให้เห็นภาพรวมของกระบวนการ และระบุจุดที่อาจเกิดปัญหาได้
- ใช้ในการแก้ปัญหา: ช่วยในการวิเคราะห์ปัญหา และหาทางแก้ไขได้อย่างมีระบบ

สัญลักษณ์ Flowchart พื้นฐาน

Python and C Programming Language



Start/Stop



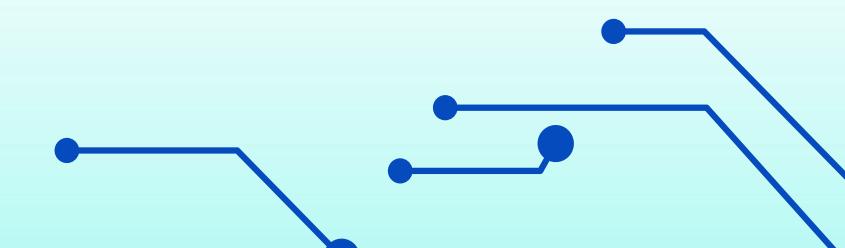
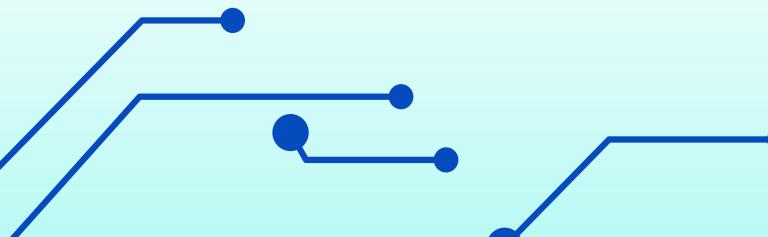
Process

1. Terminator (รูปวงรี)

- ความหมาย: ใช้สำหรับจุดเริ่มต้น (Start) หรือจุดสิ้นสุด (End) ของกระบวนการ
- ตัวอย่างข้อความในวงรี: "Start" หรือ "End"

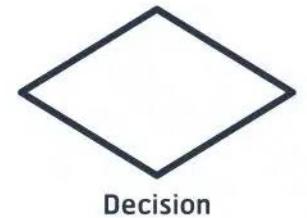
2. Process (รูปสี่เหลี่ยมผืนผ้า)

- ความหมาย: ใช้แทนกระบวนการหรือขั้นตอนการทำงานที่ต้องปฏิบัติ
- ตัวอย่างข้อความในสี่เหลี่ยม: "คำนวณยอดรวม" หรือ "ส่งอีเมล"



สัญลักษณ์ Flowchart พื้นฐาน

Python and C Programming Language



Decision



Flowline

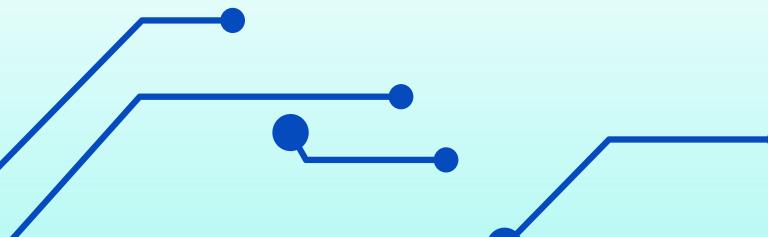
3. Decision (รูปข้าวหวานตัด)

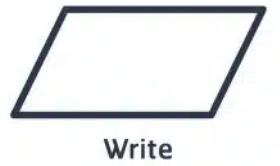
- ความหมาย: ใช้แทนจุดที่ต้องตัดสินใจ โดยมีคำถามที่สามารถตอบได้ด้วย "ใช่ (Yes)" หรือ "ไม่ใช่ (No)"
- ตัวอย่างข้อความในข้าวหวานตัด: "ข้อมูลครบถ้วนหรือไม่?"

4. Arrow (ลูกศร)

- ความหมาย: ใช้เพื่อแสดงทิศทางของการทำงานหรือการเชื่อมโยงระหว่างสัญลักษณ์ต่างๆ

การใช้งาน: เชื่อมต่อระหว่างวงรี, สี่เหลี่ยม, และข้าวหวานตัด เพื่อแสดงลำดับขั้นตอน





IN-Page Connector

5. Input/Output (รูปสี่เหลี่ยมคงที่)

- ความหมาย: ใช้แทนการรับข้อมูล (Input) หรือแสดงผลข้อมูล (Output)
ตัวอย่างข้อความในสี่เหลี่ยมคงที่: "ป้อนรหัสผ่าน" หรือ "แสดงผลรวม"

6. Between-Page สัญลักษณ์เพื่อชาร์ตแสดงการกำหนดจุดอ้างอิงในการเชื่อมต่อระหว่างหน้ากระดาษของการเขียนผังงาน

สัญลักษณ์ Flowchart พื้นฐาน

Python and C Programming Language



Between-Page

7.Annotation สัญลักษณ์ Flowchart ที่แสดงการระบุหมายเหตุเพื่อใช้อธิบายรายละเอียดเพิ่มเติมของการดำเนินงาน

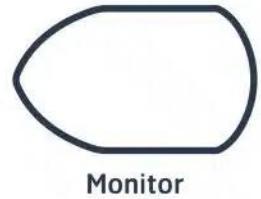


Read

8.Keyboard สัญลักษณ์ Flowchart แสดงถึงการรับค่าข้อมูลหรืออ่านข้อมูลเข้ามาจากคีย์บอร์ด

สัญลักษณ์ Flowchart พื้นฐาน

Python and C Programming Language



Monitor



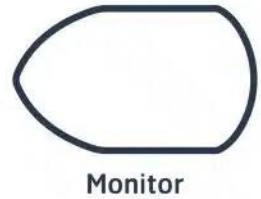
Printer

9. Monitor แสดงรายละเอียดข้อมูลหรือผลลัพธ์ของการประมวลผลไปที่จอภาพ(Monitor)

10. Printer แสดงรายละเอียดข้อมูลหรือผลลัพธ์ของการประมวลผลไปที่เครื่องพิมพ์(Printer)

สัญลักษณ์ Flowchart พื้นฐาน

Python and C Programming Language



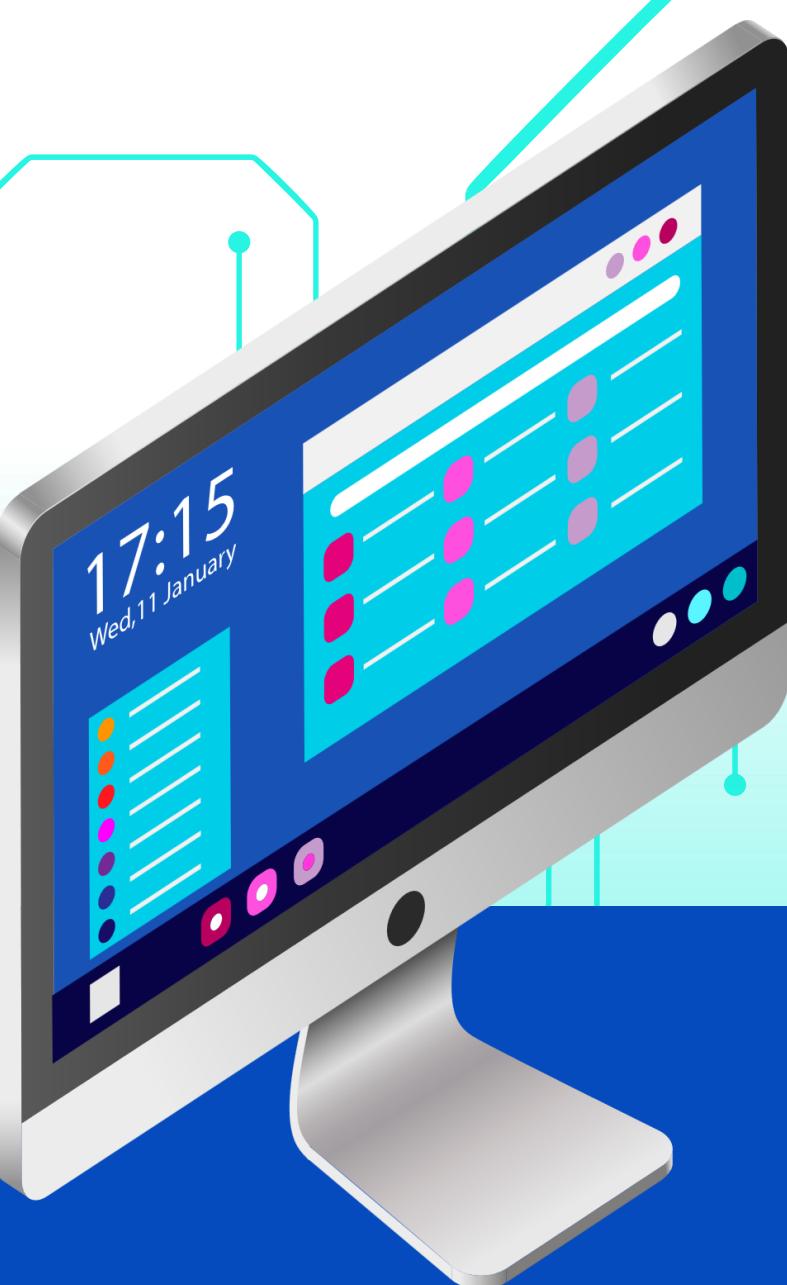
Monitor



Printer

9. Monitor แสดงรายละเอียดข้อมูลหรือผลลัพธ์ของการประมวลผลไปที่จอภาพ(Monitor)

10. Printer แสดงรายละเอียดข้อมูลหรือผลลัพธ์ของการประมวลผลไปที่เครื่องพิมพ์(Printer)



คำสั่ง if-else, switch-case

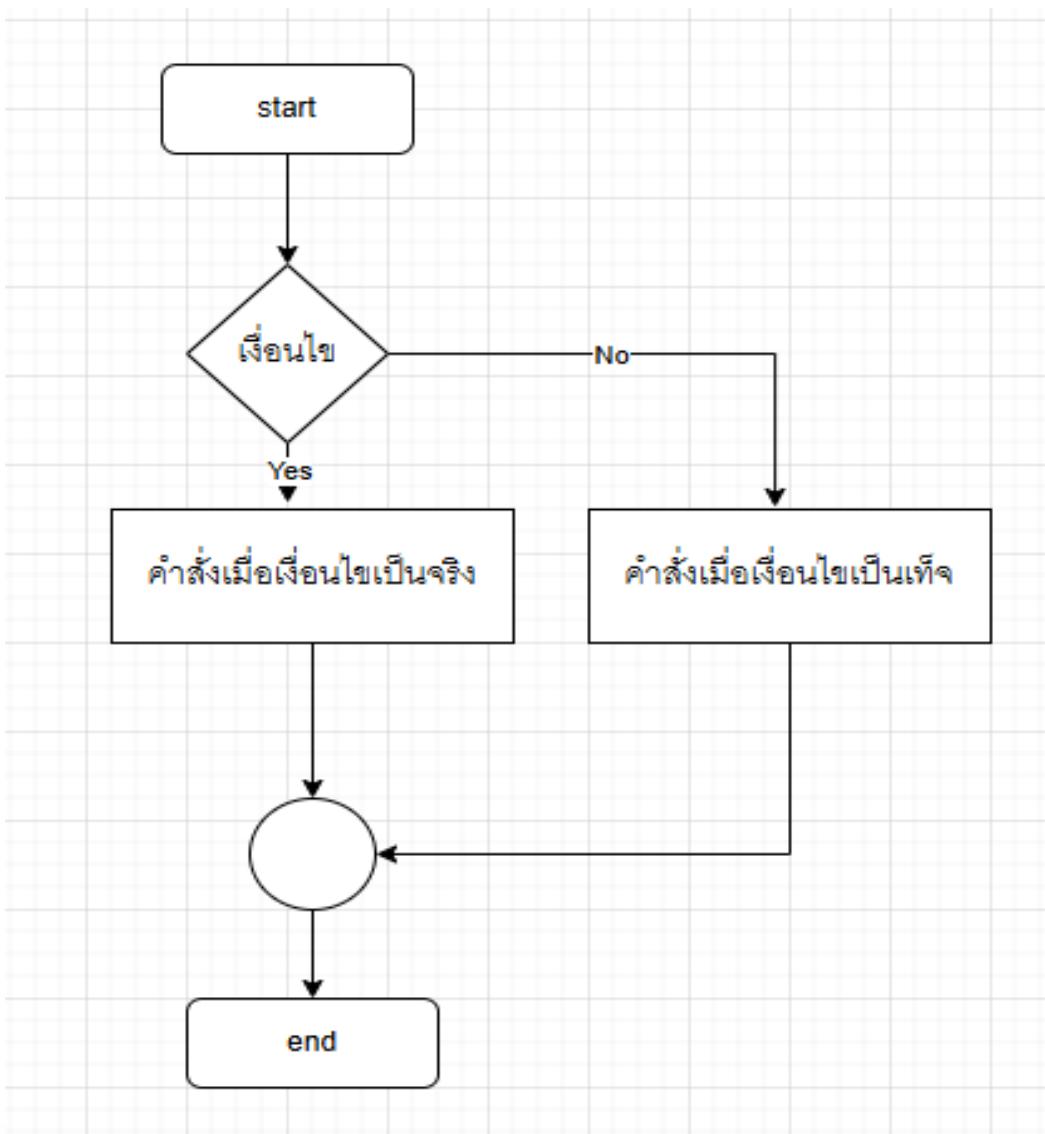
คำสั่ง if-else

คำสั่ง if-else ใช้ในการตรวจสอบเงื่อนไขที่ต้องการดำเนินการในกรณีที่เงื่อนไขนั้นเป็นจริง (True) หรือไม่จริง (False) ซึ่งหมายความว่าการตรวจสอบเงื่อนไขเพียง 1 ถึงหลายเงื่อนไขที่มีลำดับการตรวจสอบแบบต่อเนื่อง

โครงสร้างพื้นฐานของ if-else

C++

```
if (เงื่อนไข) {
    // คำสั่งเมื่อเงื่อนไขเป็นจริง
} else {
    // คำสั่งเมื่อเงื่อนไขเป็นเท็จ
}
```



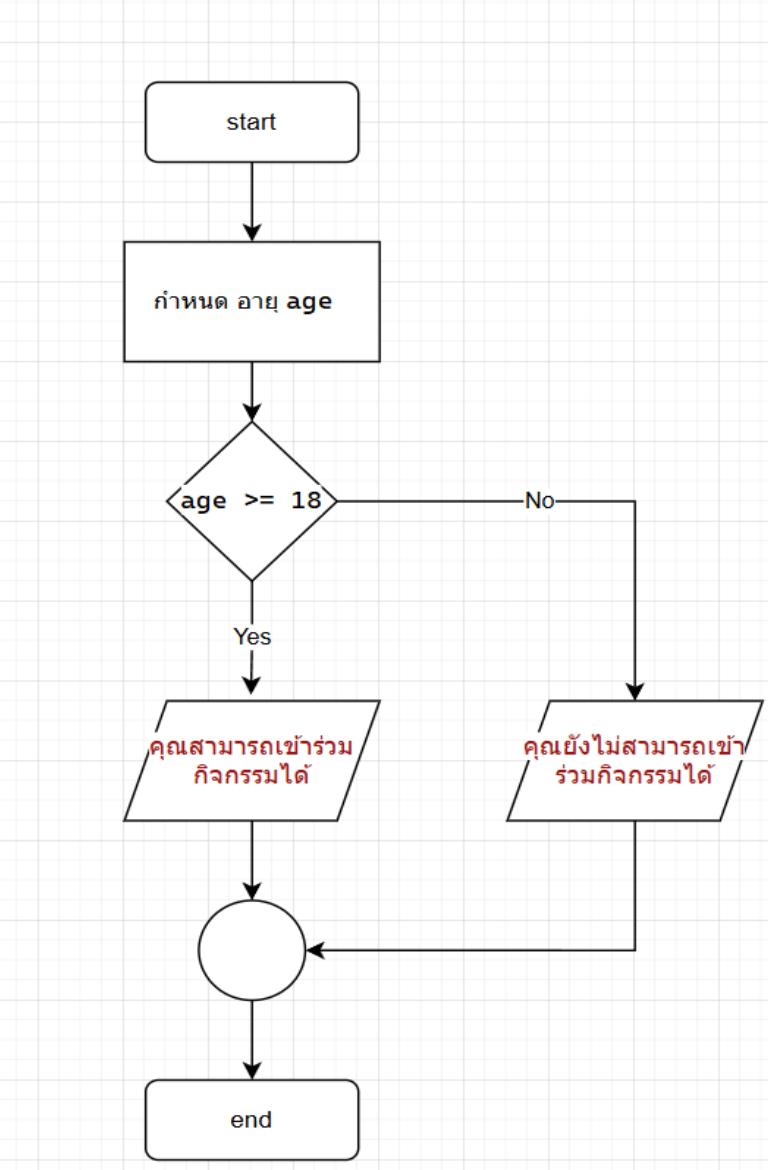
คำสั่ง if-else, switch-case

ตัวอย่าง if-else

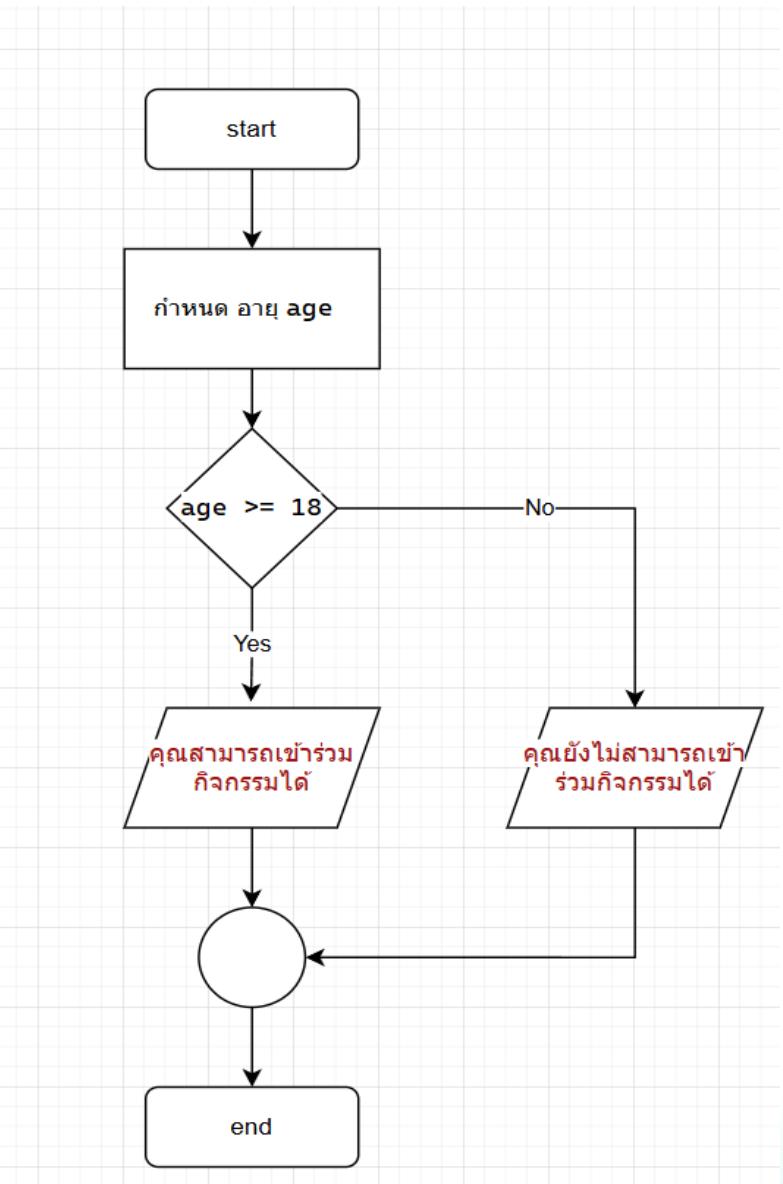
ตัวอย่างนี้จะทำการเปรียบเทียบค่าของตัวแปร age กับค่า 18 หากค่า age มากกว่าหรือเท่ากับ 18 จะแสดงข้อความว่าสามารถเข้าร่วมกิจกรรมได้ แต่ถ้าค่า age น้อยกว่า 18 จะแสดงข้อความว่ายังไม่สามารถเข้าร่วมกิจกรรมได้

ตัวอย่างการทำงาน:

- ถ้าเราเปลี่ยนค่าของ age เป็น 15 โปรแกรมจะแสดงผลลัพธ์ว่า "คุณยังไม่สามารถเข้าร่วมกิจกรรมได้"
- ถ้าเราเปลี่ยนค่าของ age เป็น 25 โปรแกรมจะแสดงผลลัพธ์ว่า "คุณสามารถเข้าร่วมกิจกรรมได้"



คำสั่ง if-else, switch-case



```
#include <iostream>
using namespace std;

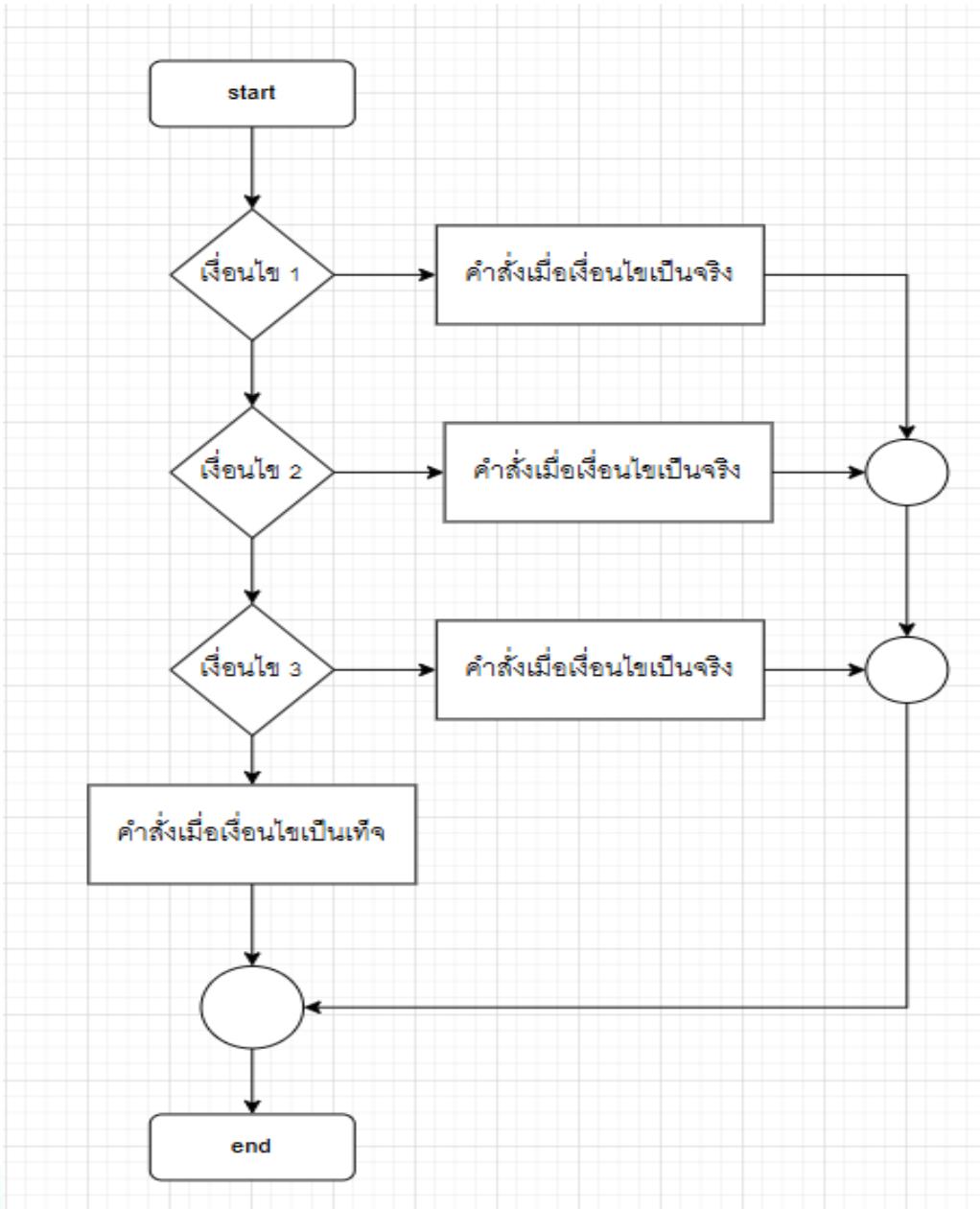
int main() {
    int age = 20;

    if (age >= 18) {
        cout << "คุณสามารถเข้าร่วมกิจกรรมได้" << endl;
    }
    else {
        cout << "คุณยังไม่สามารถเข้าร่วมกิจกรรมได้" << endl;
    }

    return 0;
}
```

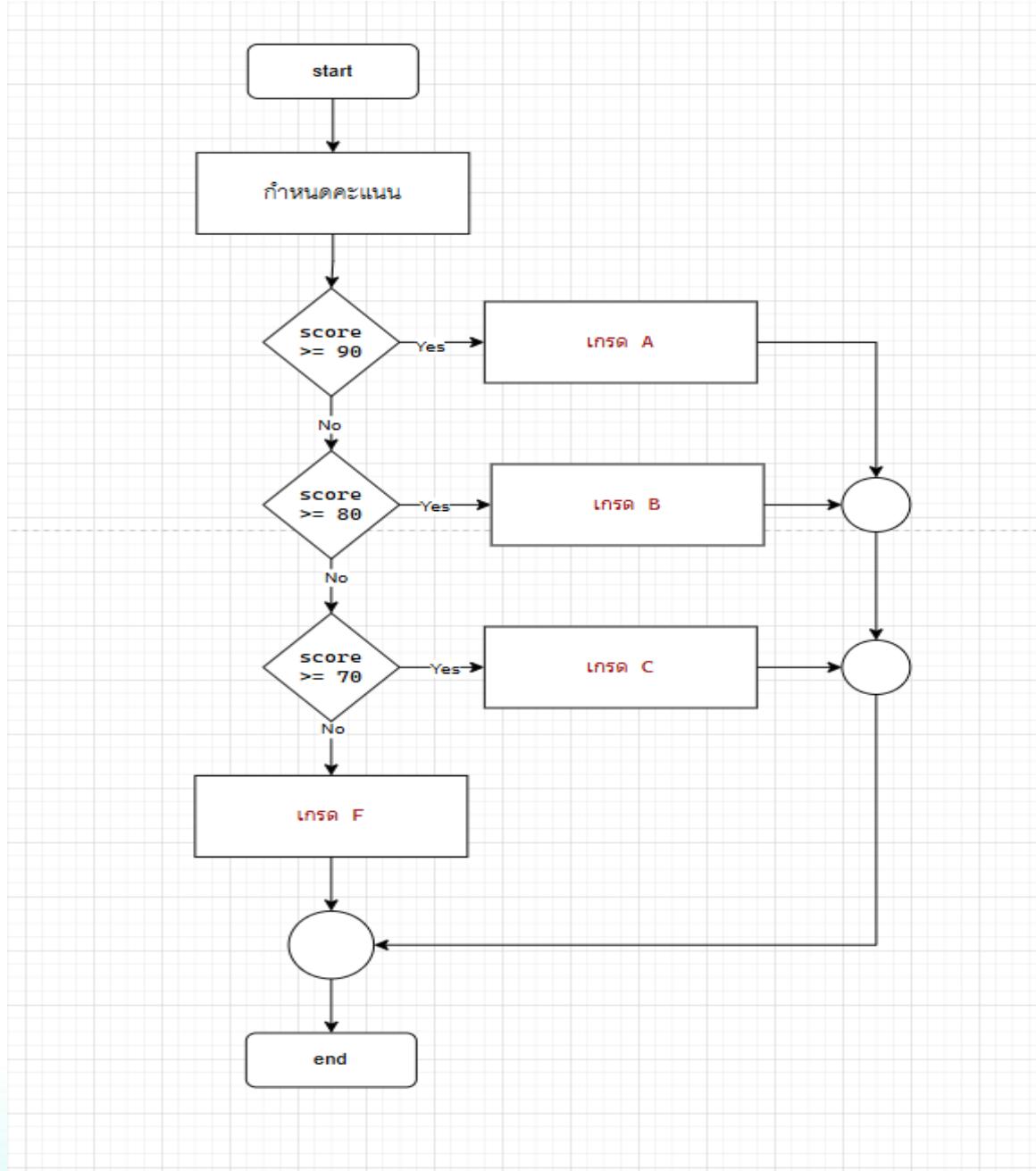
กรณีตรวจสอบหลายเงื่อนไข (if-else if-else)

เมื่อเราต้องการให้โปรแกรมตัดสินใจเลือกทำอะไรสักอย่างหนึ่งจากหลายๆ ทางเลือกตามเงื่อนไขที่แตกต่างกัน เราจะใช้โครงสร้างควบคุม if-else if-else



```

if (เงื่อนไขที่ 1) {
    // ทำงานเมื่อเงื่อนไขที่ 1 เป็นจริง
} else if (เงื่อนไขที่ 2) {
    // ทำงานเมื่อเงื่อนไขที่ 1 เป็นเท็จ แต่เงื่อนไขที่ 2 เป็นจริง
} else if (เงื่อนไขที่ 3) {
    // ทำงานเมื่อเงื่อนไขที่ 1 และ 2 เป็นเท็จ แต่เงื่อนไขที่ 3 เป็นจริง
} else {
    // ทำงานเมื่อทุกเงื่อนไขเป็นเท็จ
}
  
```



ตัวอย่าง if-else if-else

โปรแกรมนี้มีหน้าที่ตรวจสอบเกรดนักเรียนจากคะแนนที่กำหนดไว้

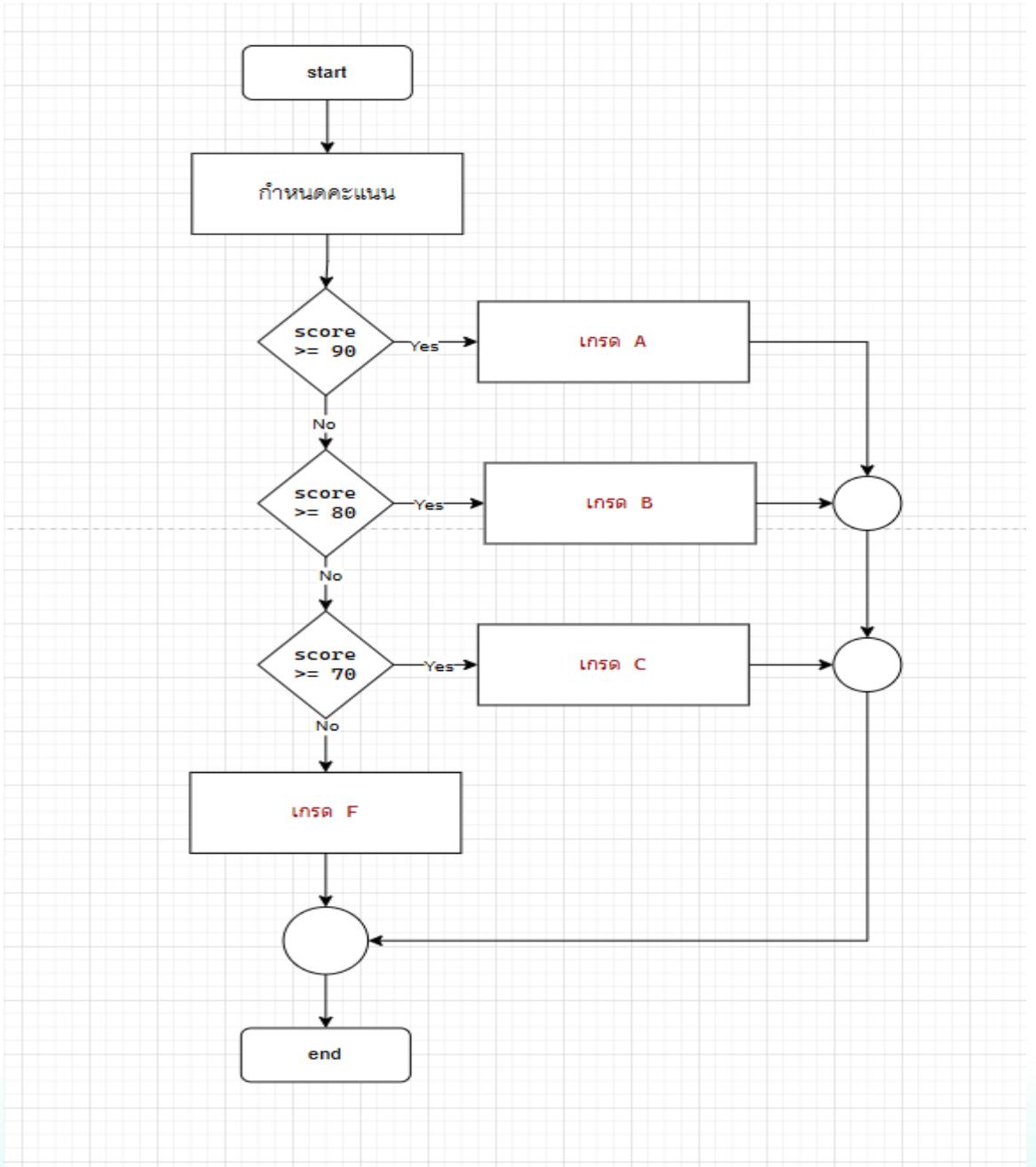
วิธีการทำงาน:

- 1) กำหนดคะแนน: โปรแกรมจะกำหนดคะแนนของนักเรียนไว้ในตัวแปร score โดยมีค่าเริ่มต้นเป็น 85 คะแนน
- 2) ตรวจสอบเงื่อนไข: โปรแกรมจะนำคะแนนไปเปรียบเทียบกับเกณฑ์ต่างๆ ที่กำหนดไว้ (90, 80, 70, 60) เพื่อหาว่าคะแนนอยู่ในช่วงของเกรดใด
- 3) แสดงผลลัพธ์: เมื่อพบเงื่อนไขที่ตรงกับคะแนน โปรแกรมจะแสดงเกรดที่สอดคล้องกันออกมาทางหน้าจอ

ตัวอย่างในกรณี:

เนื่องจากคะแนน score มีค่าเท่ากับ 85 ซึ่งอยู่ในช่วง 80-89 โปรแกรมจึงแสดงผลลัพธ์ออกมาว่า "เกรด B"

สรุปง่ายๆ: โปรแกรมนี้เหมือนกับเครื่องคิดเลขเกรด ที่เราป้อนคะแนนเข้าไป และโปรแกรมจะบอกให้เราทราบว่าได้เกรดอะไร



```
#include <iostream>
using namespace std;

int main() {
    int score = 85;

    if (score >= 90) {
        cout << "เกรด A" << endl;
    }
    else if (score >= 80) {
        cout << "เกรด B" << endl;
    }
    else if (score >= 70) {
        cout << "เกรด C" << endl;
    }
    else {
        cout << "เกรด F" << endl;
    }

    return 0;
}
```

คำสั่ง if-else, switch-case

คำสั่ง switch-case (Python ใช้ match-case ในเวอร์ชัน 3.10 ขึ้นไป)

switch-case (หรือใน Python คือ match-case) ใช้สำหรับการตรวจสอบค่าหลายกรณีที่ชัดเจน โดยแต่ละกรณีเป็นเงื่อนไขที่มีค่าเฉพาะเจาะจง มักใช้ในการที่ตัวแปรมีค่าแน่นอน จำนวนหนึ่ง เช่น ตัวเลขหรือสตริง

switch (ตัวแปร) {

case ค่า1:

// คำสั่งเมื่อค่าเท่ากับ ค่า1

break;

case ค่า2:

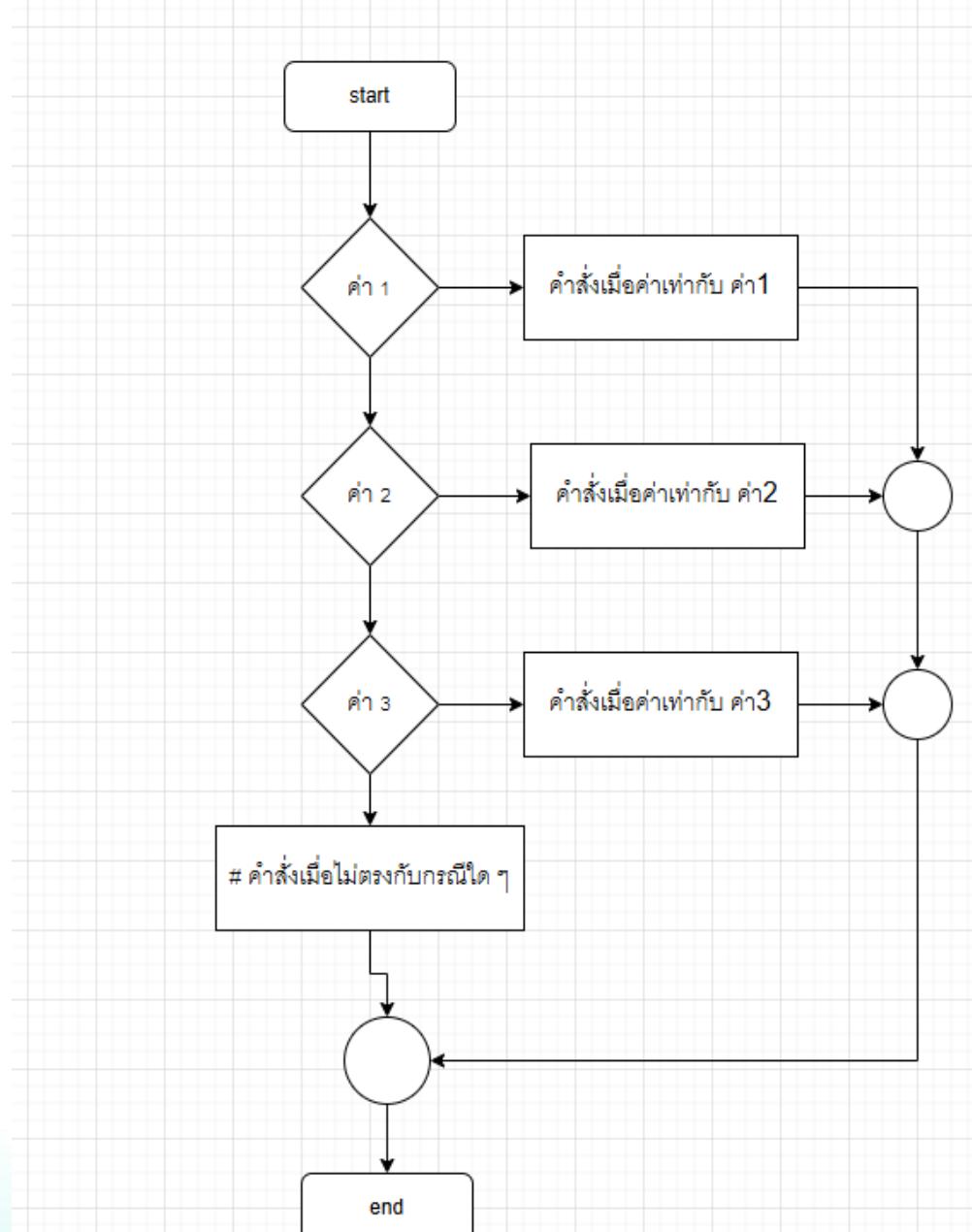
// คำสั่งเมื่อค่าเท่ากับ ค่า2

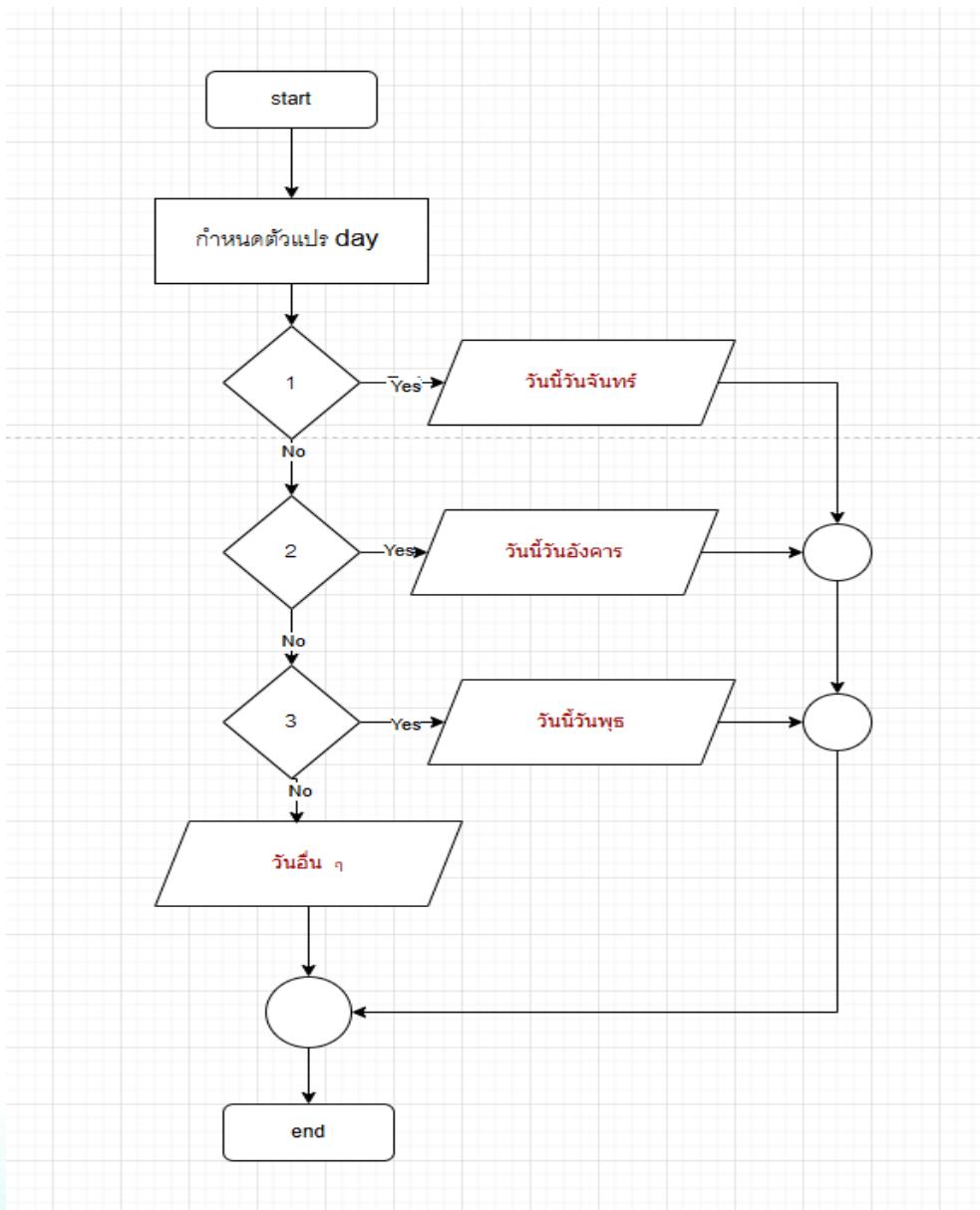
break;

default:

// คำสั่งเมื่อไม่ตรงกับกรณีใดเลย

}





ตัวอย่าง

โปรแกรมนี้ถูกออกแบบมาเพื่อแสดงชื่อวันในสัปดาห์ตามตัวเลขที่กำหนดให้

วิธีการทำงาน:

- 1) กำหนดตัวแปร: โปรแกรมจะกำหนดตัวแปร day ให้มีค่าเท่ากับ 2 ซึ่งหมายถึงวันอังคาร
- 2) ใช้คำสั่ง switch: คำสั่ง switch จะนำค่าในตัวแปร day ไปเปลี่ยนเทียบกับค่าในแต่ละ case
- 3) ตรวจสอบเงื่อนไข:

ถ้าค่าใน day ตรงกับค่าใน case ใด คำสั่งใน case นั้นจะถูกทำงาน

ถ้าไม่มี case ใดตรงกับค่าใน day คำสั่งใน default จะถูกทำงาน

- 1) แสดงผลลัพธ์: เมื่อพบ case ที่ตรงกัน โปรแกรมจะแสดงข้อความที่ระบุชื่อวันนั้นๆ ออกมาทางหน้าจอ

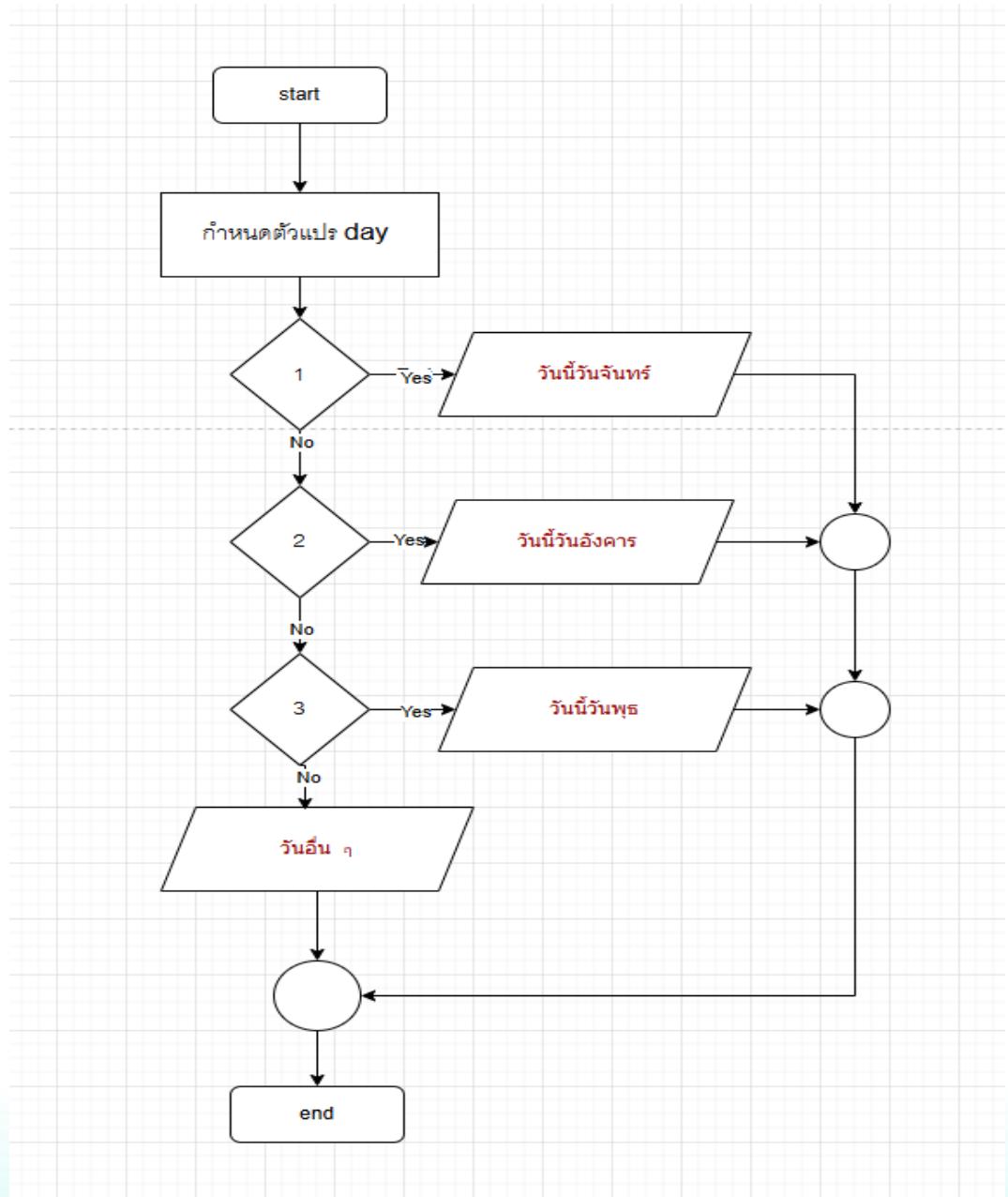
ในกรณีนี้:

เนื่องจากค่าใน day คือ 2 คำสั่งจะตรงกับ case 2

ดังนั้นโปรแกรมจะแสดงผลลัพธ์ออกมาว่า "วันนี้วันอังคาร"

สรุปง่ายๆ: โปรแกรมนี้เหมือนกับตารางเทียบเลขกับชื่อวันในสัปดาห์ เมื่อเราใส่เลขวันเข้าไป โปรแกรมจะบอกให้เราทราบว่าวันนั้นคือวันอะไร

คำสั่ง if-else, switch-case



```

#include <iostream>
using namespace std;

int main() {
    int day = 2;

    switch (day) {
        case 1:
            cout << "วันนี้วันจันทร์" << endl;
            break;
        case 2:
            cout << "วันนี้วันอังคาร" << endl;
            break;
        case 3:
            cout << "วันนี้วันพุธ" << endl;
            break;
        default:
            cout << "วันอื่น ๆ" << endl;
            break;
    }
    return 0;
}
  
```



Sutit Ongart
Sutit@mut.ac.th
sutit.ongart@gmail.com

END



www.mut.ac.th