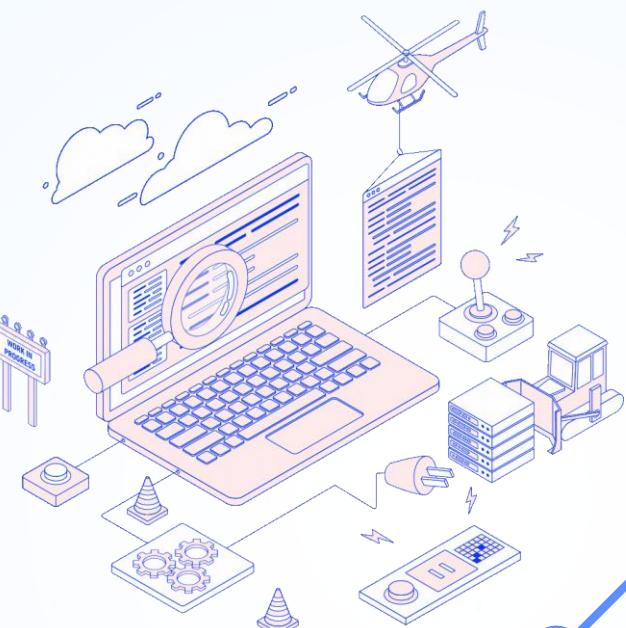


MIIA0106 #4

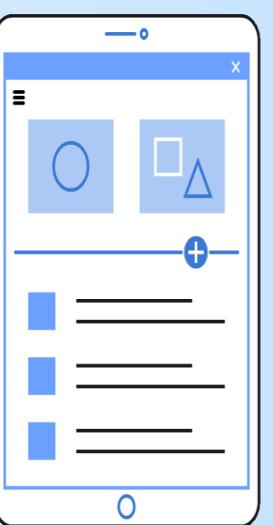


Python and C Programming Language

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

วนลูป for, while, do-while

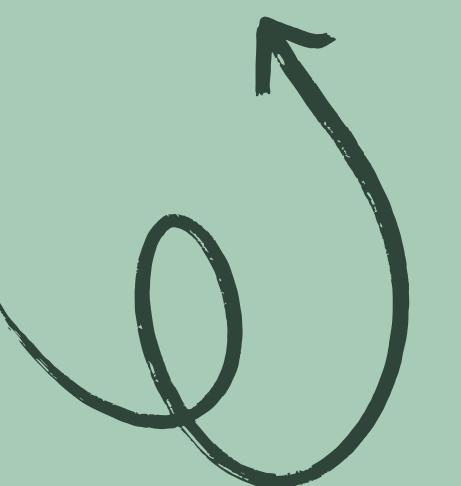


ลูป (Loop) คือโครงสร้างควบคุมการทำงานของโปรแกรมที่ช่วยให้คำสั่งหรือกลุ่มคำสั่งทำงานซ้ำได้หลายครั้งโดยไม่ต้องเขียนคำสั่งเดิมซ้ำ ๆ หลายบรรทัด

ความสำคัญของลูป

- 1.ลดการเขียนโค้ดซ้ำ: ใช้คำสั่งที่เหมือนกันโดยไม่ต้องเขียนโค้ดซ้ำหลายครั้ง
- 2.ปรับเปลี่ยนตามเงื่อนไขได้: การทำงานจะหยุดเมื่อเงื่อนไขที่กำหนดสิ้นสุด
- 3.ลดข้อผิดพลาด: การใช้ลูปช่วยลดโอกาสเกิดข้อผิดพลาดจากการคัดลอกคำสั่งหลายครั้ง
- 4.ประยุกต์เวลา: เขียนโปรแกรมที่ซับซ้อนและทำงานเร็วขึ้น
- 5.ยืดหยุ่น: ใช้กับข้อมูลจำนวนมาก เช่น การอ่านข้อมูลจากไฟล์หรือฐานข้อมูล

សែនមីយន ប្រពេករម សាច់រំបង់ រិមព័ន



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>)

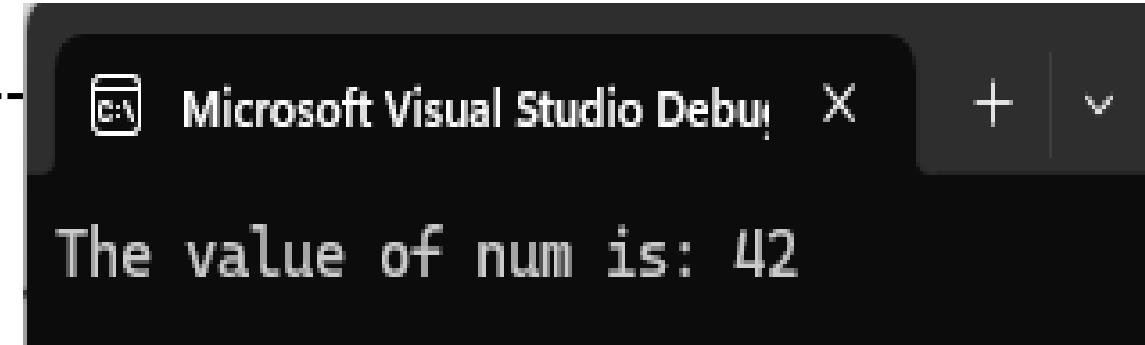
Code

```
#include <iostream>

using namespace std;

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

Code แสดงข้อมูลออกหน้าจอ



Code

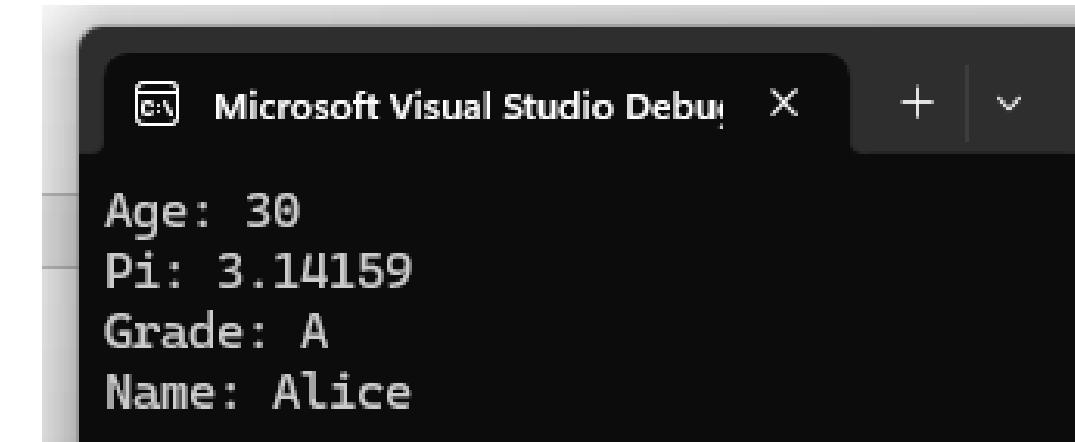
```
#include <iostream>

using namespace std;

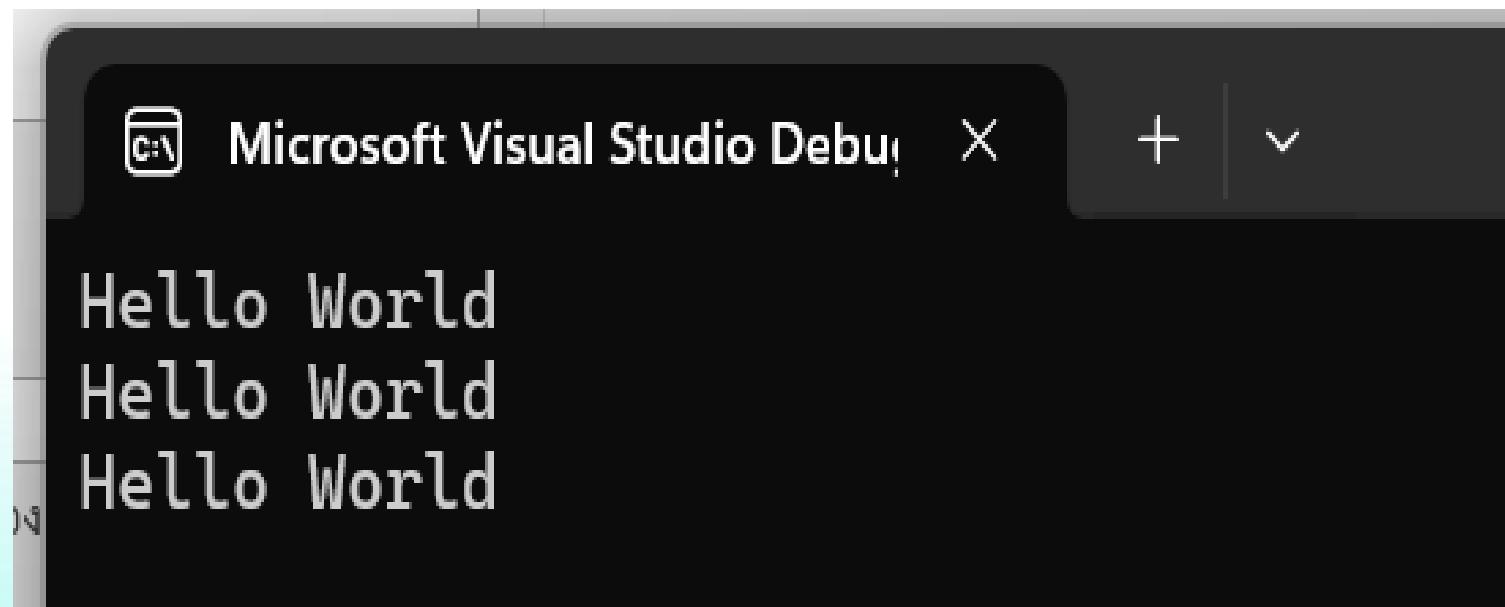
int main() {
    int age = 30;
    double pi = 3.14159;
    char grade = 'A';
    string name = "Alice";

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

Code แสดงข้อมูลออกหน้าจอ



ตัวอย่าง เขียน code ต่อไปนี้



```
Microsoft Visual Studio Debug X + | ▾  
Hello World  
Hello World  
Hello World
```



ทำไมต้องใช้ loop

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

int main() {
    cout << "Hello World" << endl;
    cout << "Hello World" << endl;
    cout << "Hello World" << endl;
    return 0;
}
```

ลูป (Loop) คืออะไร?

Loop คือโครงสร้างควบคุมการทำงานของโปรแกรมที่ช่วยให้คำสั่งหรือกลุ่มคำสั่งทำงานซ้ำได้หลายครั้งโดยไม่ต้องเปลี่ยนคำสั่งเดิมซ้ำ ๆ ความสำคัญ:

ลดการเขียนโค้ดซ้ำ (Reduce redundancy)

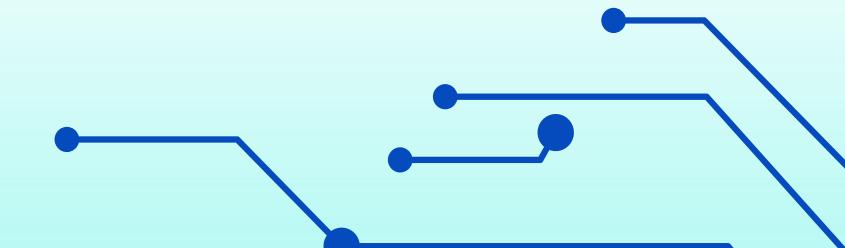
ปรับเปลี่ยนตามเงื่อนไข (Flexible)

ลดข้อผิดพลาด (Minimize errors)

ประหยัดเวลา (Save time)



"ทำซ้ำจนกว่าจะครบเงื่อนไข"



ความหมายและ ความสำคัญของ ลูป

ลูปคือโครงสร้างควบคุมในโปรแกรมที่ช่วยให้เขียนโค้ด
ง่ายได้อย่างมีประสิทธิภาพ ลดการทําซ้ำ คำสั่ง ช่วย
ประหยัดเวลาและเพิ่มความยืดหยุ่นในการพัฒนา
โปรแกรม



ปัญหาของการไม่ใช้ Loop

ตัวอย่าง: พิมพ์เลข 1-10

ถ้าไม่ใช้ Loop เราต้องเขียนคำสั่ง cout ซ้ำกันถึง 10 บรรทัด

หากต้องพิมพ์ถึง 100 หรือ 1000 บรรทัด จะเสียเวลามากและ

แก้ไขยาก

```
#include
using namespace std;

int main() {
    cout << "1\n";
    cout << "2\n";
    cout << "3\n";
    // ... เขียนต่อจนถึง 10
    cout << "10\n";
    return 0;
}
```

การแก้ปัญหาด้วย for Loop

```
#include  
using namespace std;  
  
int main() {  
    for (int i = 1; i <= 10; i++) {  
        cout << i << " ";  
    }  
    return 0;  
}
```

ผลลัพธ์ที่ได้:

0 1 2 3 4 5 6 7 8 9 10

แก้ไขจำนวนรอบได้ง่าย (แค่เปลี่ยนเลข 10)

อ่านเข้าใจง่ายและเป็นระบบ

ทำไมต้องใช้ loop

ต้องการ Hello World 10 บรรทัด

ทำไมต้องใช้ loop

Python and C Programming Language

ต้องการ Hello World 1,000 บรรทัด

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

int main() {
    cout << "Hello World" << endl;
    cout << "Hello World" << endl;
    cout << "Hello World" << endl;
    ...
    ...
    cout << "Hello World" << endl;
    cout << "Hello World" << endl;
    cout << "Hello World" << endl;
    return 0;
}
```

ประเภทของลูป

ลูปแบ่งออกเป็น 3 ประเภทหลัก:

1. For Loop

- ใช้เมื่อต้องการทำงานจำนวนรอบที่รู้ล่วงหน้า
- เช่น พิมพ์ตัวเลข 1 ถึง 10

2. While Loop

- ใช้เมื่อต้องการทำงานซ้ำจนกว่าจะถึงเงื่อนไขที่กำหนด
- เช่น รอนกว่าผู้ใช้จะป้อนค่า "exit"

3. Do-While Loop (เฉพาะใน C++)

- คล้ายกับ While แต่คำสั่งภายในจะถูกทำงานอย่างน้อยหนึ่งครั้งก่อนตรวจสอบเงื่อนไข

ประเภทของ Loop ในภาษา C



For Loop

ใช้เมื่อต้องการทำงานจำนวนรอบที่รู้ล่วงหน้า

เช่น พิมพ์เลข 1-10



While Loop

ใช้เมื่อต้องการทำงานซ้ำจนกว่าจะถึงเงื่อนไข

ที่กำหนด

เช่น วนลูปจนกว่า user พิมพ์ exit



Do-While Loop

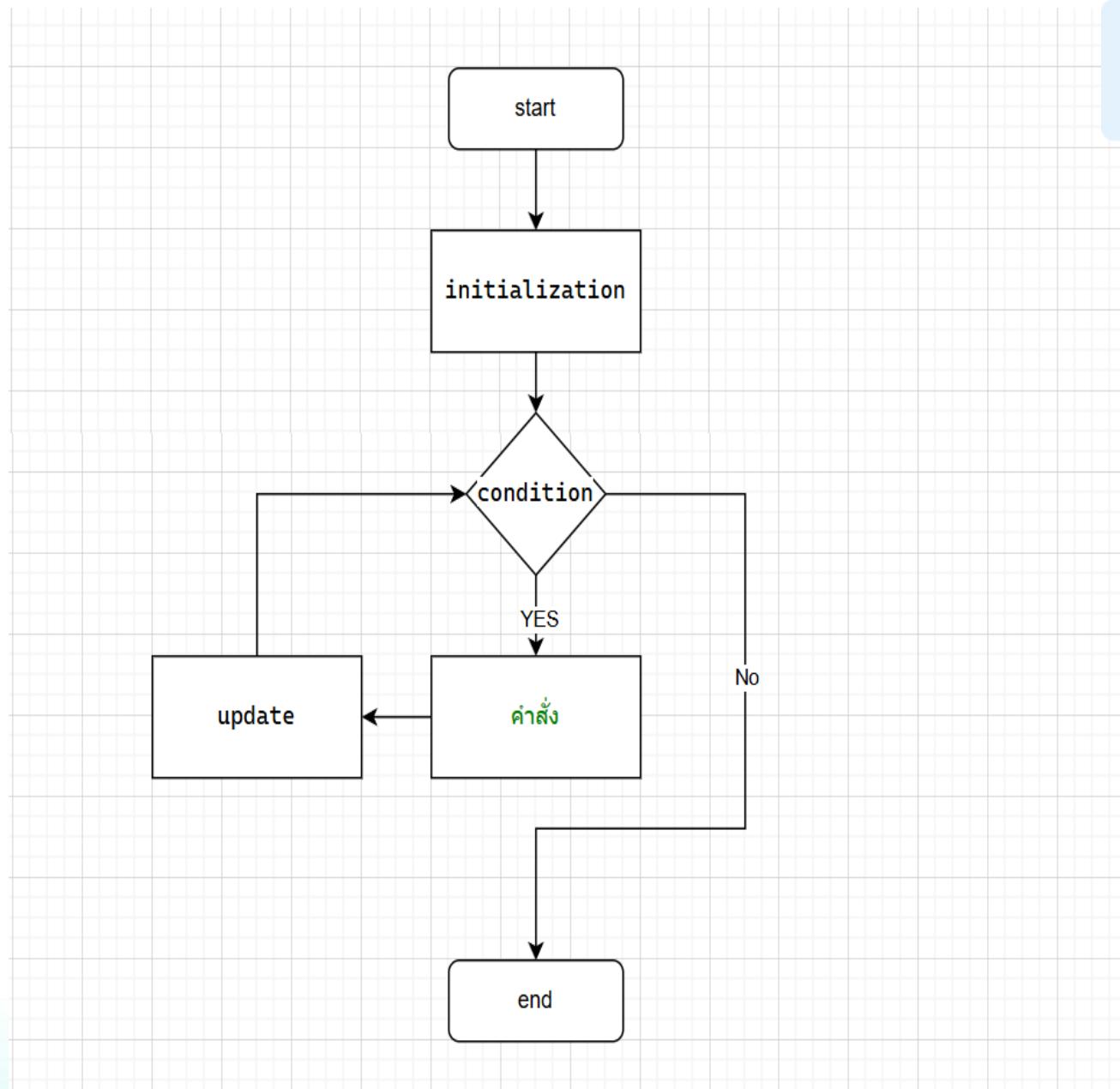
คล้าย While แต่ทำการ kontrol 1 ครั้ง

เสมอ

เช่น รับค่า input จนกว่าจะถูกต้อง

โครงสร้างของ For Loop

for (initialization; condition; update)

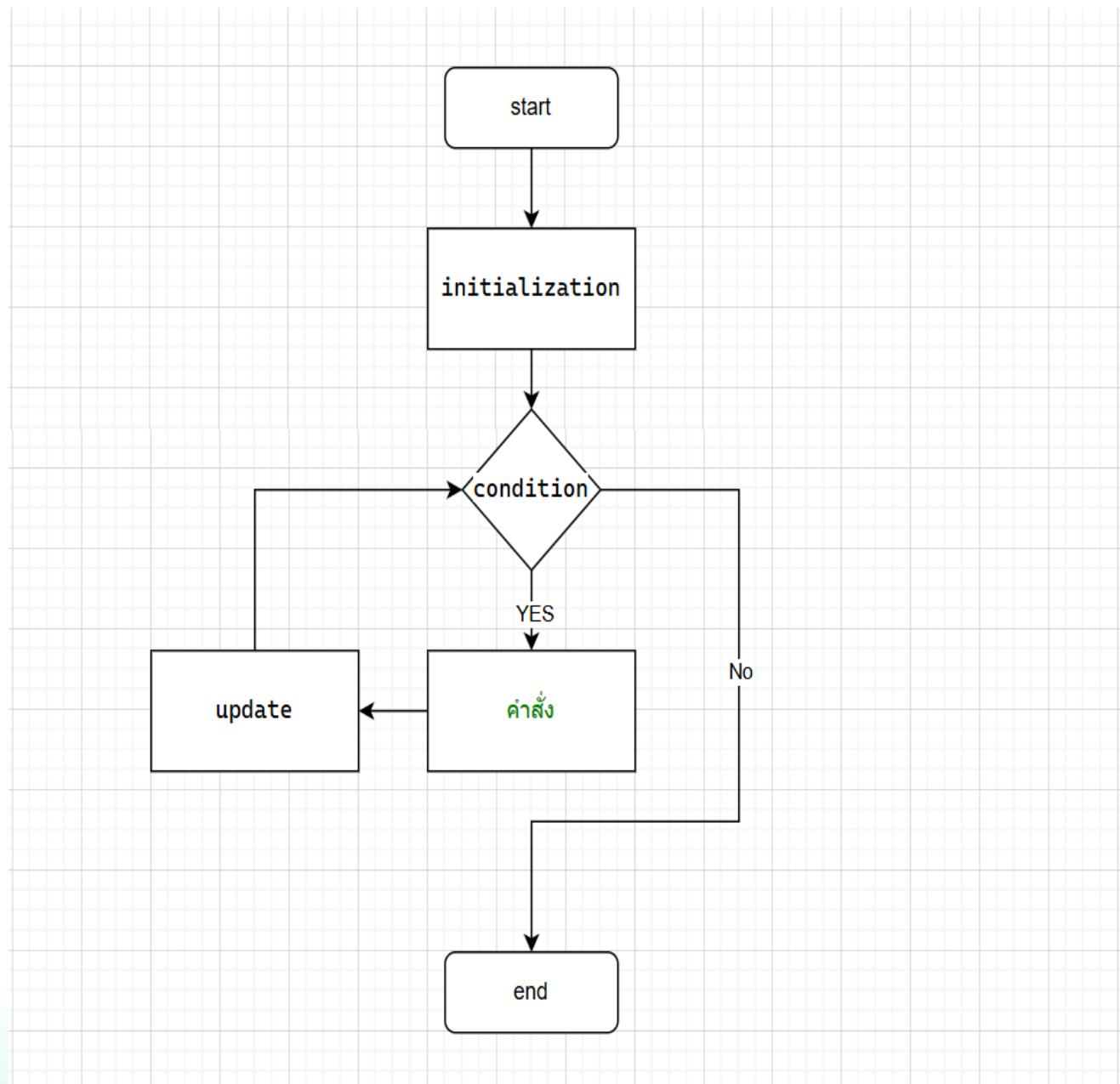


- Initialization (C++): กำหนดค่าเริ่มต้น เช่น int i = 0;
- Condition: เงื่อนไขที่ตรวจสอบก่อนทำซ้ำ
- Update: การปรับเปลี่ยนค่าตัวแปร เช่น i++

```
for (initialization; condition; update) {  
    // คำสั่งที่ต้องการทำซ้ำ  
}
```

โครงสร้างของ For Loop

Python and C Programming Language



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

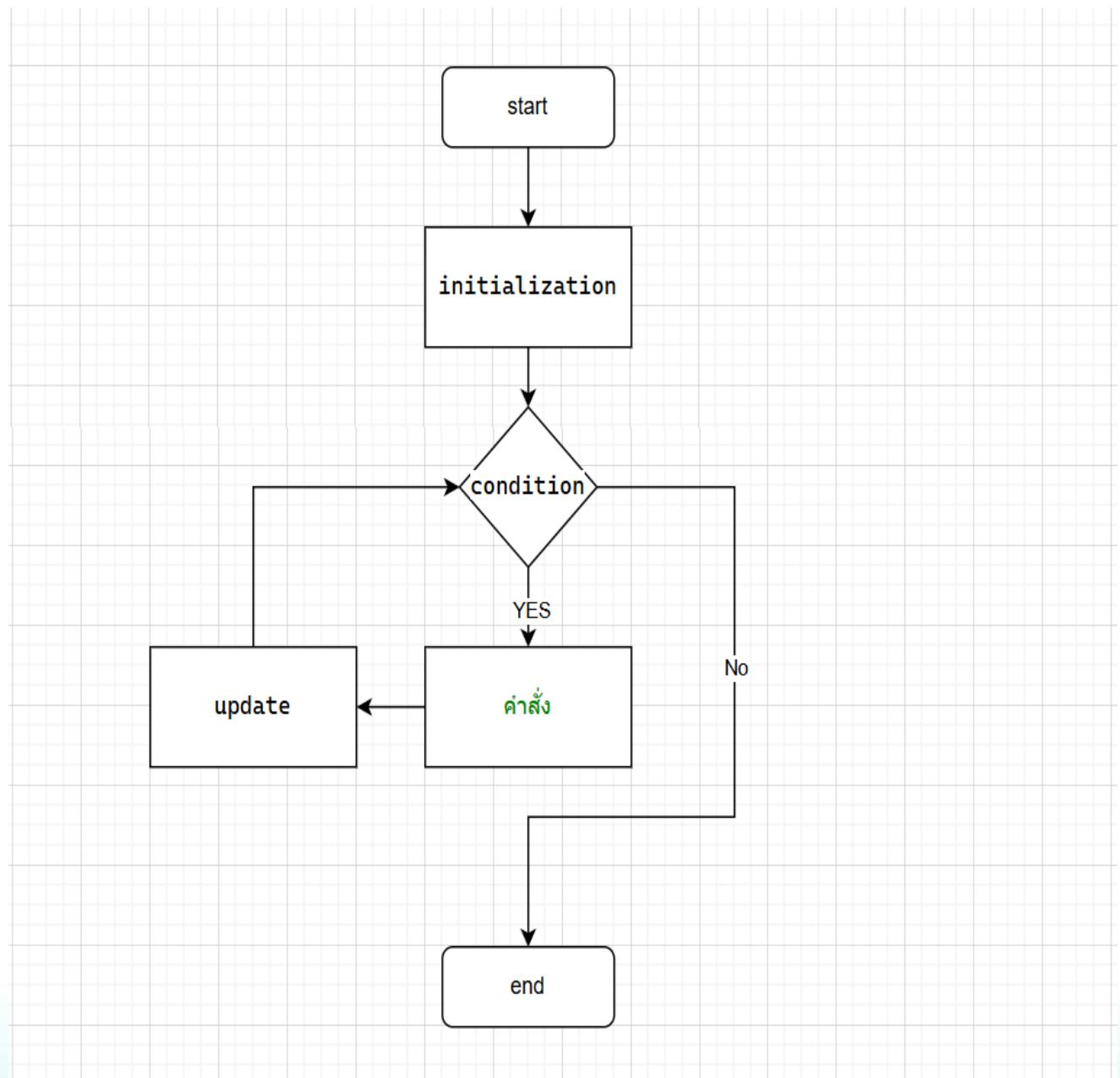
int main() {
    for (int i = 1; i <= 10; i++) {
        cout << i << " ";
    }
    return 0;
}
```

A screenshot of the Microsoft Visual Studio Debug window. The window title is 'Microsoft Visual Studio Debug'. The output pane displays the numbers 1 through 10, each on a new line, representing the execution of the for loop code.

```
1
2
3
4
5
6
7
8
9
10
```

โครงสร้างของ For Loop

Python and C Programming Language



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

int main() {
    for (int i = 1; i <= 10; i++) {
        cout << i << " ";
    }
    return 0;
}
```

A screenshot of the Microsoft Visual Studio Debug window. The window title is 'Microsoft Visual Studio Debug'. The output pane displays the numbers 1 through 10, each on a new line, representing the execution of the for loop code.

```
1
2
3
4
5
6
7
8
9
10
```

ตัวอย่าง for Loop: แสดงเลข 1-10

```
for (int i = 1; i <= 10; i++) {  
    cout << i << " ";  
}
```

Step-by-Step Explanation:

- 1 Start: กำหนด $i = 1$
- 2 Check: $1 \leq 10$? (True) -> พิมพ์ 1
- 3 Update: $i++$ (i กลายเป็น 2)
- 4 Check: $2 \leq 10$? (True) -> พิมพ์ 2
- 5 ...ทำซ้ำไปเรื่อยๆ...
- 6 Stop: เมื่อ i เป็น 11 (False) -> จบลูป
- .



โครงสร้าง while Loop

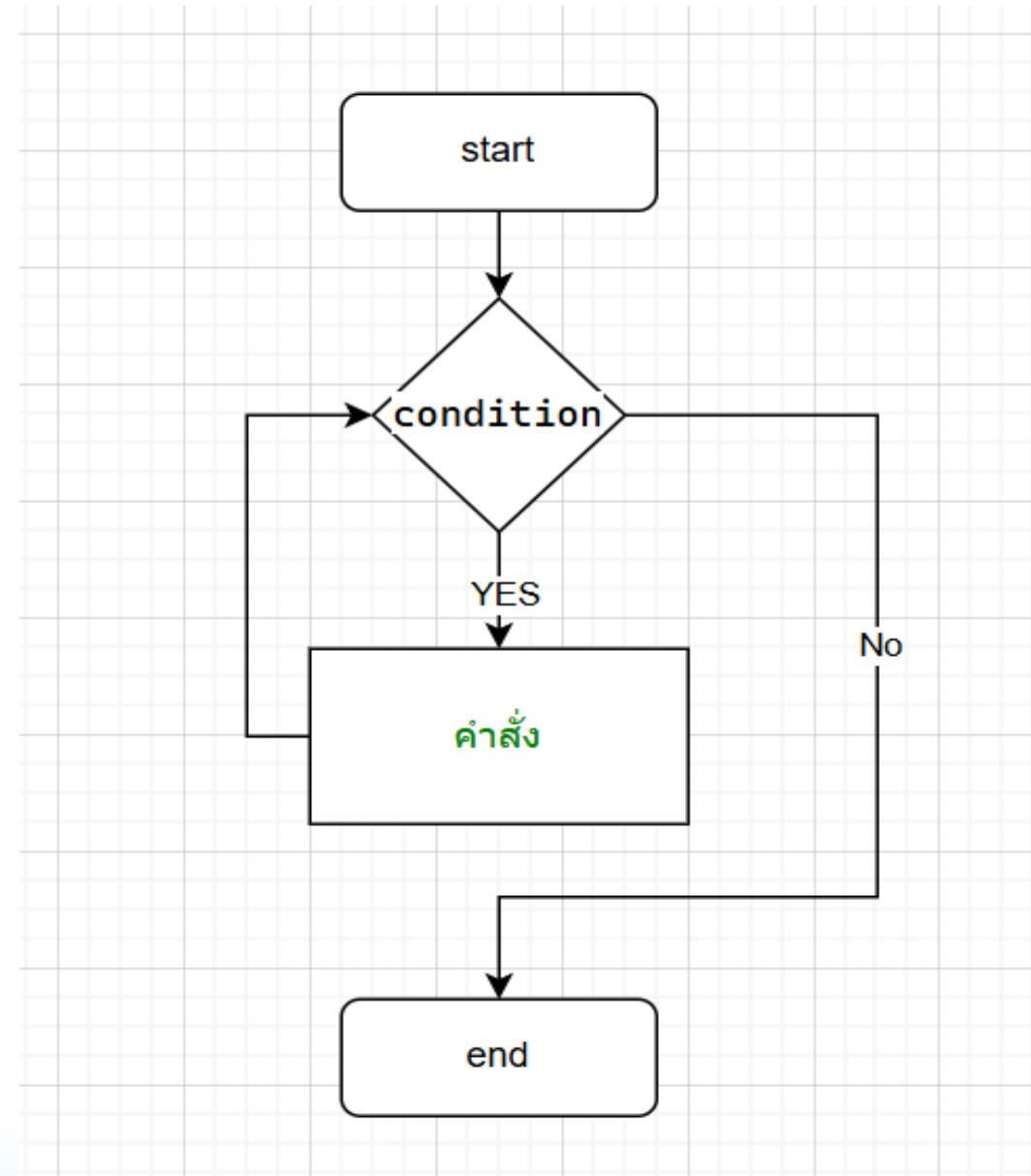
จุดสำคัญ:

- ตรวจสอบเงื่อนไขก่อนเริ่มทำงานเสมอ
- ถ้าเงื่อนไขเป็นเท็จตั้งแต่แรก จะไม่ทำงานเลย
- ต้องระวัง Infinite Loop (ลูปไม่รู้จบ) ถ้าลืม update ค่าตัวแปร

Syntax:

```
while (เงื่อนไข) {  
    // คำสั่งที่ต้องการทำซ้ำ  
    // ต้องมีการ Update ค่าที่นี่  
}
```

โครงสร้าง while Loop



- Condition: เงื่อนไขที่ตรวจสอบก่อนเริ่มทำงาน
- ทำงานซ้ำจนกว่าจะไม่เป็นจริงโครงสร้างพื้นฐานของ while loop

```
while (เงื่อนไข) {  
    // โค้ดที่ต้องการทำงาน  
}
```

ตัวอย่าง while Loop: แสดงเลข 1-10

```
int main() {  
    int i = 1; // Initialization  
    while (i <= 10) { // Condition  
        cout << i << "\n";  
        i++; // Update  
    }  
    return 0;  
}
```

การทำงาน:

- กำหนด $i = 1$ ไว้ในคลูป
- ตรวจสอบ $i \leq 10$ ก่อนเข้าคลูป
- พิมพ์ค่า i
- เพิ่มค่า $i++$ ภายในบล็อกของ while
- วนกลับไปตรวจสอบเงื่อนไขใหม่

Example:

โครงสร้าง do-while Loop

Syntax:

```
do {  
    // คำสั่งที่ต้องการทำซ้ำ  
} while (condition);
```

จุดเด่น (Key Feature):

"ทำก่อน เช็คทีหลัง"

- คำสั่งภายในจะถูกทำงานอย่างน้อย 1 ครั้งเสมอ
- เหมาะสมสำหรับเมนู หรือการรับค่า Input ที่ต้องถามผู้ใช้อย่างน้อยหนึ่งครั้ง

ตัวอย่าง do-while: รับค่าตัวเลขบวก

```
int num;  
do {  
    cout << "Enter a positive number: ";  
    cin >> num;  
} while (num <= 0);  
  
cout << "You entered: " << num;
```

ลำดับการทำงาน:

1. โปรแกรมจะถามให้ใส่ตัวเลขทันที 1 ครั้ง
2. รับค่า num
3. ตรวจสอบเงื่อนไข: ถ้า num <= 0 (ไม่ใช่เลขบวก) ให้วนกลับไปตามใหม่
4. ถ้าเป็นเลขบวกแล้ว ให้จบลูปและแสดงผล

คำสั่งควบคุม: break และ continue

Break

"หยุดทันที"

ออกจากลูปทันทีเมื่อเจอคำสั่งนี้ ไม่สนใจในลูป

```
if (i == 5) break;  
// จบลูปเมื่อ i เป็น 5
```

Continue

"ข้ามรอบนี้"

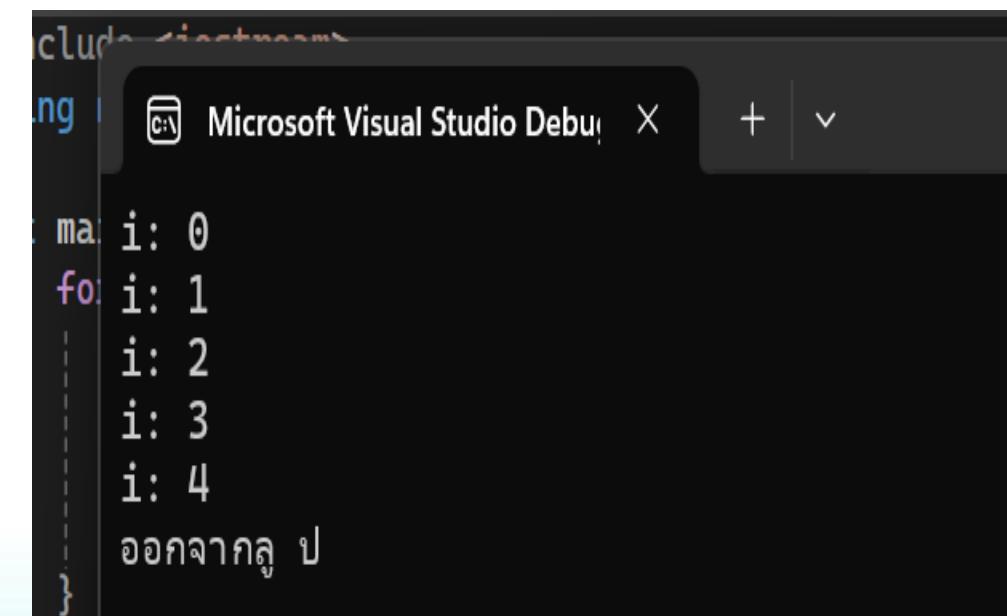
ข้ามคำสั่งที่เหลือในรอบนั้น แล้วเริ่มรอบถัดไปทันที

```
if (i % 2 == 0) continue;  
// ข้ามถ้าเป็นเลขคู่
```

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

int main() {
    for (int i = 0; i < 10; i++) {
        if (i == 5) {
            break; // ออกจากลูปทันทีเมื่อ i เท่ากับ 5
        }
        cout << "i: " << i << endl;
    }
    cout << "ออกจากลูป" << endl;
    return 0;
}
```

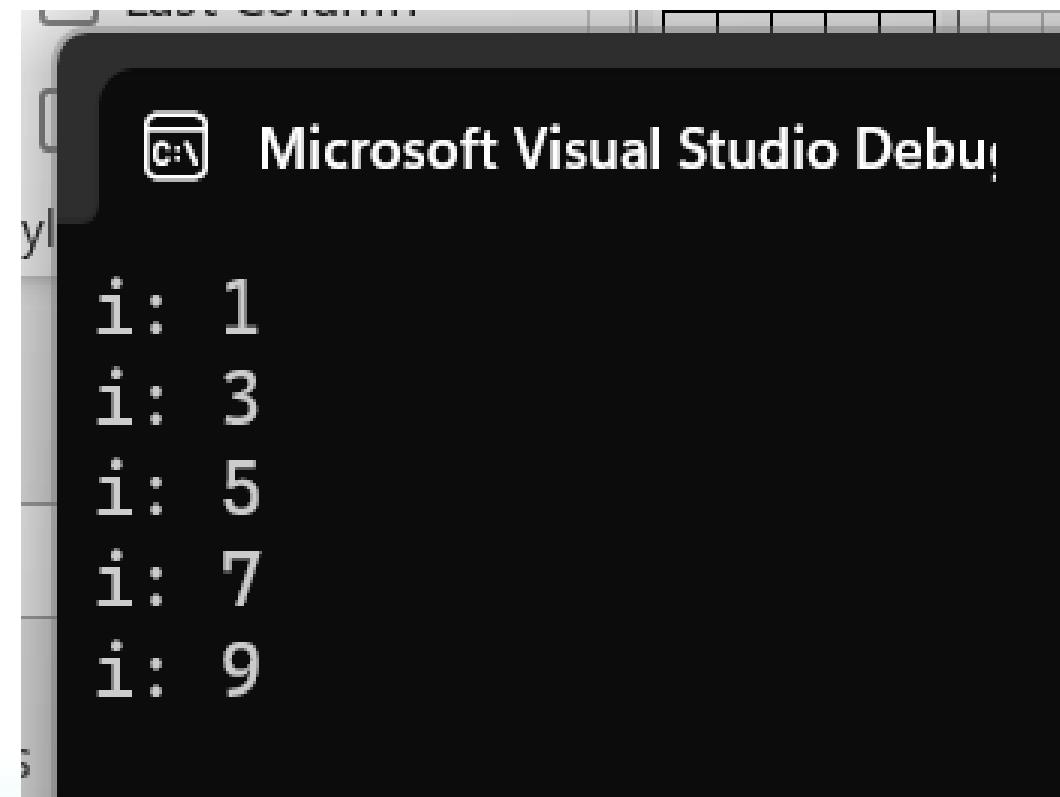
- ใช้เพื่้ออกจากลูปก่อนที่จะครบเงื่อนไขของการวนซ้ำ
- เมื่อโปรแกรมเจอคำสั่ง break ในลูป จะหยุดการวนซ้ำทันทีและออกไปยังคำสั่งถัดไปหลังลูป



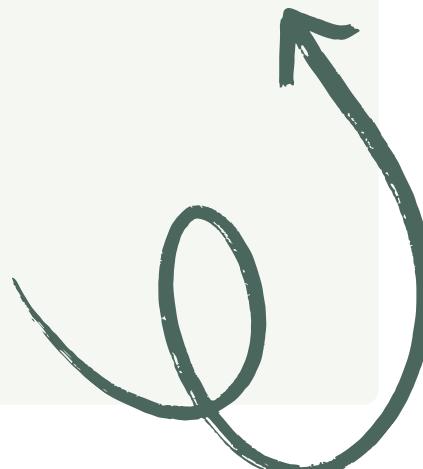
```
#include <iostream>
using namespace std;

int main() {
    for (int i = 0; i < 10; i++) {
        if (i % 2 == 0) {
            continue; // ข้ามการทำงานในรอบที่ i เป็นเลขคู่
        }
        cout << "i: " << i << endl;
    }
    return 0;
}
```

- . ใช้เพื่อข้ามการทำงานในรอบนั้น และวนลูปต่อไปยังรอบถัดไปทันที
- . ไม่ออกจากลูป แต่ข้ามคำสั่งที่อยู่หลัง continue



ໂລຍໍຕ້າວອຢ່າງແລກປະປະບຸກຕໍ່ໃຈໜັນລຸບ



ระบบควบคุมอุณหภูมิในห้องปฏิบัติการ Temperature Control System

คำอธิบาย:

ห้องปฏิบัติการต้องการรักษาอุณหภูมิให้อยู่ในช่วง 20-25 องศาเซลเซียสเพื่อให้แน่ใจว่าอุปกรณ์และวัสดุดิบไม่เสียหาย ระบบควบคุมอุณหภูมิจะตรวจสอบอุณหภูมิทุกๆ 5 วินาทีและทำการปรับอุณหภูมิตามเงื่อนไขต่อไปนี้:

- ถ้าอุณหภูมิต่ำกว่า 20 องศาเซลเซียส ให้เปิดเครื่องทำความร้อน
- ถ้าอุณหภูมิสูงกว่า 25 องศาเซลเซียส ให้เปิดเครื่องทำความเย็น
- ถ้าอุณหภูมิอยู่ในช่วง 20-25 องศาเซลเซียส ให้ปิดทั้งเครื่องทำความร้อนและเครื่องทำความเย็น



กรณีศึกษา 1: ระบบควบคุมอุณหภูมิ

โจทย์:

รักษาอุณหภูมิห้องให้อยู่ในช่วง $20-25^{\circ}\text{C}$ โดยตรวจสอบทุก ๆ 5 วินาที

เงื่อนไข:

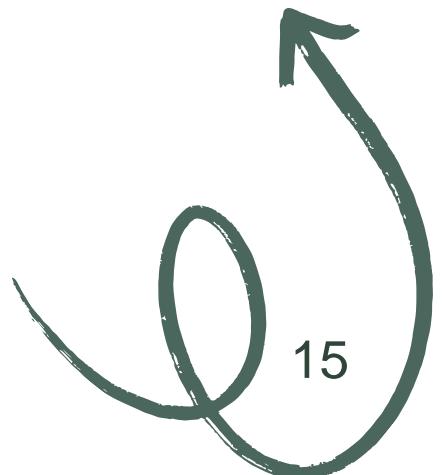
- $< 20^{\circ}\text{C}$: เปิดเครื่องทำความร้อน (Heater)
- $> 25^{\circ}\text{C}$: เปิดเครื่องทำความเย็น (Cooler)
- $20-25^{\circ}\text{C}$: ปิดทั้งสองอย่าง

1) วัตถุประสงค์ของการเขียนโปรแกรม

พัฒนาโปรแกรมที่สามารถตรวจสอบและควบคุมอุณหภูมิของห้องปฏิบัติการให้อยู่ในช่วง 20-25 องศาเซลเซียส

ป้องกันไม่ให้อุปกรณ์และวัตถุดิบในห้องปฏิบัติการเสียหายเนื่องจากอุณหภูมิที่ไม่เหมาะสม

แสดงสถานะการทำงานของระบบ (เปิดเครื่องทำความร้อน, เปิดเครื่องทำความเย็น, หรือปิดทิ้งสอง)



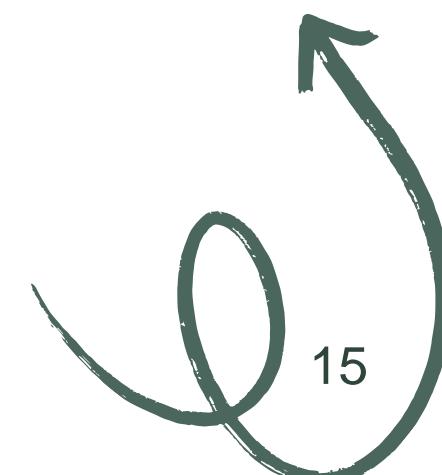
2) รูปแบบผลลัพธ์ที่ต้องการ

โปรแกรมจะแสดงข้อความแสดงสถานะของระบบทุก 5 วินาที เช่น:

"อุณหภูมิ: 18°C -> เปิดเครื่องทำความร้อน"

"อุณหภูมิ: 26°C -> เปิดเครื่องทำความเย็น"

"อุณหภูมิ: 22°C -> ปิดเครื่องทำความร้อนและเครื่องทำความเย็น"



3) ข้อมูลนำเข้า

- . อุณหภูมิปัจจุบัน (Current Temperature): อ่านค่าจากเซ็นเซอร์หรือป้อนค่าจำลอง (ในกรณีทดสอบโปรแกรม)

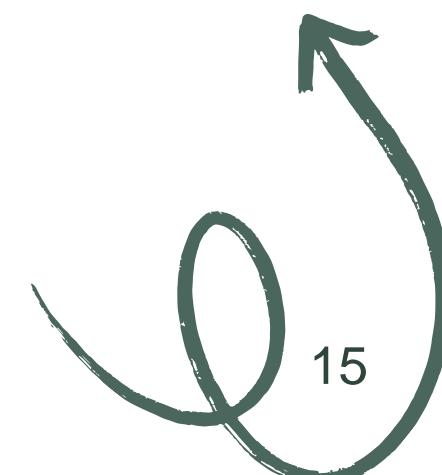
4) ตัวแปรที่ใช้

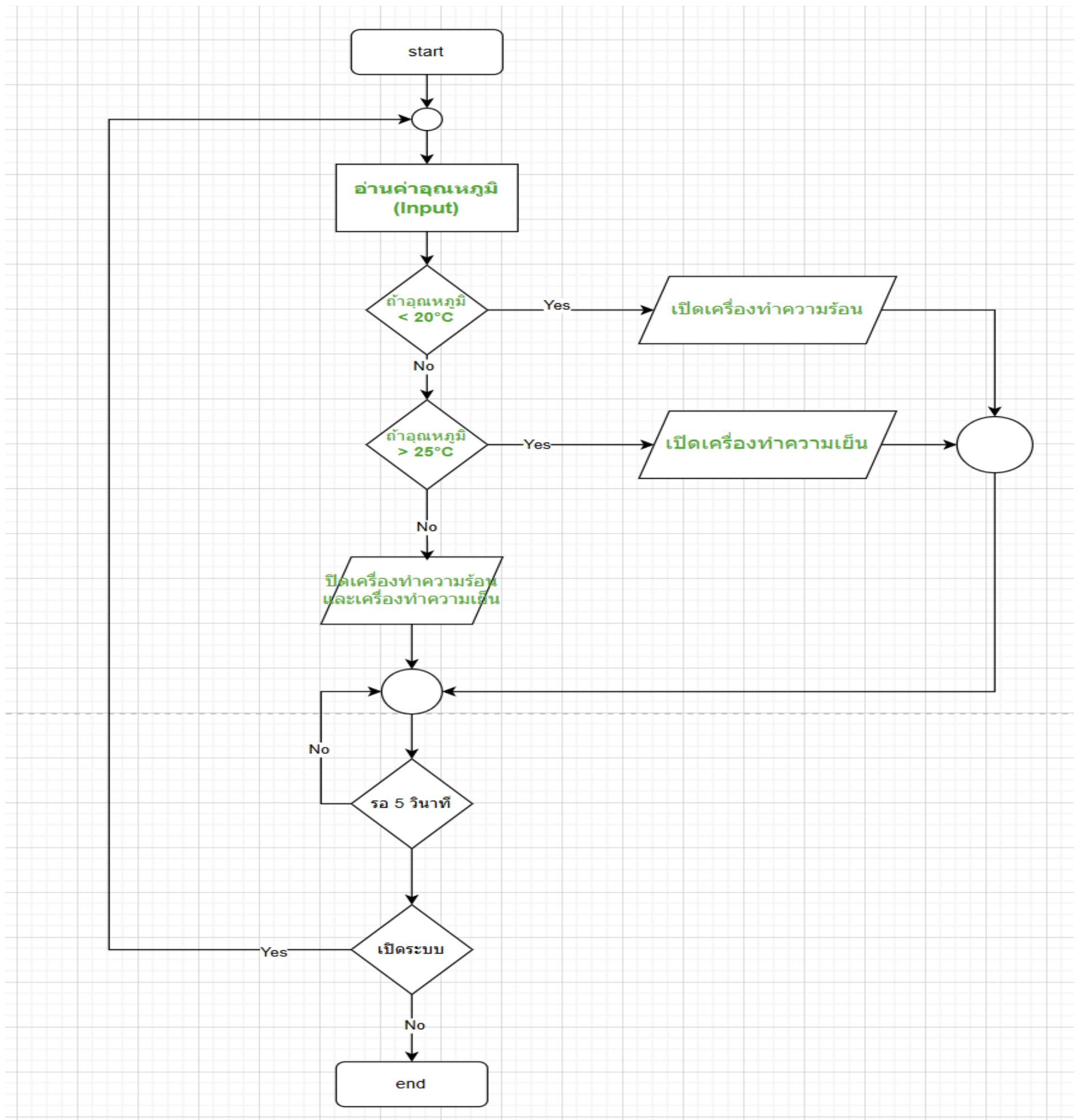
. C++:

- float currentTemperature: เก็บค่าอุณหภูมิที่ตรวจสอบ
- string action: เก็บข้อความแสดงสถานะของระบบ

5) วิธีการประมวลผล

- 1. อ่านค่าอุณหภูมิ (Input):** อ่านค่าจากเซ็นเซอร์ (หรือจำลองค่า)
- 2. ตรวจสอบเงื่อนไข (Processing):**
 - ถ้าอุณหภูมิ $< 20^{\circ}\text{C}$: แสดงข้อความ "ปิดเครื่องทำความร้อน"
 - ถ้าอุณหภูมิ $> 25^{\circ}\text{C}$: แสดงข้อความ "ปิดเครื่องทำความเย็น"
 - ถ้าอุณหภูมิ อยู่ในช่วง $20-25^{\circ}\text{C}$: แสดงข้อความ "ปิดเครื่องทำความร้อนและเครื่องทำความเย็น"
- 3. แสดงสถานะ (Output):** แสดงข้อความสถานะทุก 5 วินาที
- 4. วนซ้ำ:** ดำเนินการซ้ำทุก 5 วินาที





```
#include <iostream>
#include <thread> // สำหรับการหน่วงเวลา
#include <chrono> // สำหรับการหน่วงเวลา
using namespace std;

int main() {
    float currentTemperature;
    char choice; // ตัวแปรสำหรับเก็บตัวเลือกของผู้ใช้
    // จะเขียน while

    return 0;
}
```

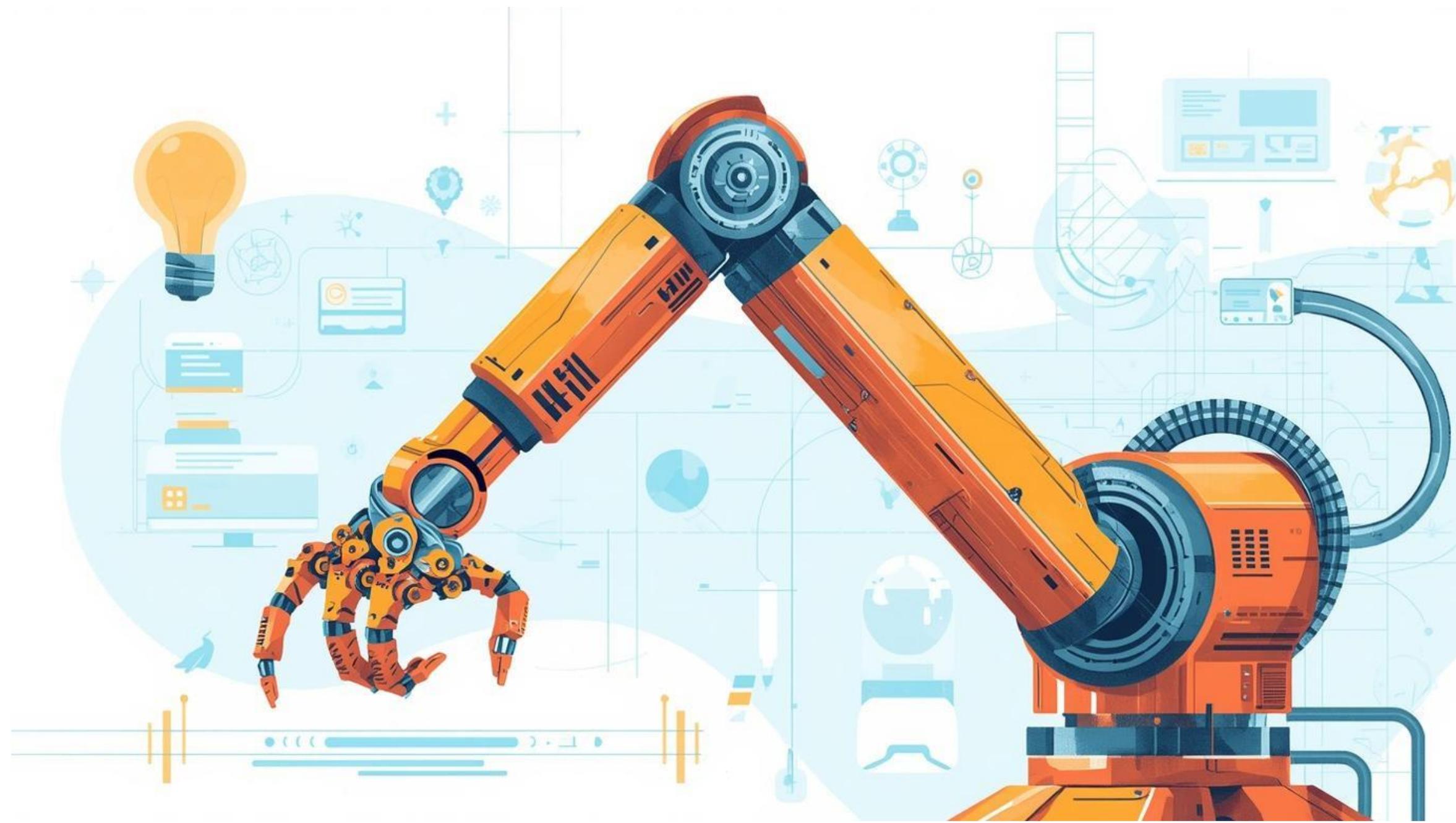
```

while (true) {
    // ถามผู้ใช้ว่าจะใส่อุณหภูมิหรือไม่
    cout << "Do you want to enter the temperature? (Y/N): ";
    cin >> choice;
    // ตรวจสอบคำตอบของผู้ใช้
    if (choice == 'Y' || choice == 'y') {
        // รับค่าอุณหภูมิจากผู้ใช้
        cout << "Enter the current temperature: ";
        cin >> currentTemperature;
        // ตรวจสอบเงื่อนไข
        if (currentTemperature < 20) {
            cout << "Temperature: " << currentTemperature << "°C -> Turning on the heater" << endl;
        }
        else if (currentTemperature > 25) {
            cout << "Temperature: " << currentTemperature << "°C -> Turning on the cooler" << endl;
        }
        else {
            cout << "Temperature: " << currentTemperature << "°C -> Turning off both heater and cooler" << endl;
        }
        // หน่วงเวลา 5 วินาที
        this_thread::sleep_for(chrono::seconds(5));
    }
    else {
        // ถ้าผู้ใช้ตอบ N หรือค่าที่ไม่ใช่ Y ให้หยุดโปรแกรม
        cout << "Exiting the program..." << endl;
        break;
    }
}

```

```
Microsoft Visual Studio Debug + ▾  
Do you want to enter the temperature? (Y/N): y  
Enter the current temperature: 21  
Temperature: 21°C -> Turning off both heater and cooler  
Do you want to enter the temperature? (Y/N): y  
Enter the current temperature: 50  
Temperature: 50°C -> Turning on the cooler  
Do you want to enter the temperature? (Y/N): y  
Enter the current temperature: 24  
Temperature: 24°C -> Turning off both heater and cooler  
Do you want to enter the temperature? (Y/N): N  
Exiting the program...
```

ระบบควบคุมแขนหุ่นยนต์



ระบบควบคุมขนส่งหุ้นยนต์

คำอธิบาย:

แขนหุ้นยนต์ในโรงงานผลิตต้องการจับวัตถุจากแท่นส่ง (conveyor belt) และวางลงในกล่อง
แขนหุ้นยนต์มีเซนเซอร์สำหรับตรวจจับวัตถุและตำแหน่งของวัตถุ ระบบควบคุมจะตรวจสอบ
ตำแหน่งของวัตถุและทำการควบคุมแขนหุ้นยนต์ให้เคลื่อนไหวตามเงื่อนไขดังไปนี้:

- ถ้าวัตถุอยู่ในตำแหน่ง A ให้แขนหุ้นยนต์เคลื่อนไหวไปทางซ้าย
- ถ้าวัตถุอยู่ในตำแหน่ง B ให้แขนหุ้นยนต์เคลื่อนไหวไปทางขวา
- ถ้าวัตถุอยู่ในตำแหน่ง C ให้แขนหุ้นยนต์เคลื่อนไหวลงล่างและจับวัตถุ
- ถ้าไม่มีวัตถุ ให้แขนหุ้นยนต์กลับไปยังตำแหน่งเริ่มต้น

กรณีศึกษา 2: ระบบควบคุมแขนหุ่นยนต์

โจทย์:

ควบคุมแขนหุ่นยนต์ตามตำแหน่งวัตถุบนสายพาน

- A: Move Left (ซ้าย)
- B: Move Right (ขวา)
- C: Pick Up (จับวัตถุ)
- N: Return Start (กลับจุดเริ่ม)

*ถามผู้ใช้ว่าจะทำงานต่อหรือไม่ (Y/N) โดยใช้ do-while

1. วัตถุประสงค์ของการเขียนโปรแกรม

- . พัฒนาโปรแกรมควบคุมการเคลื่อนไหวของแขนหุ่นยนต์เพื่อจับวัตถุจากสายพานลำเลียงและวางลงในกล่อง
- . กำหนดพฤติกรรมของแขนหุ่นยนต์ตามตำแหน่งของวัตถุ (A, B, C หรือไม่มีวัตถุ)
- . มีฟังก์ชันเพื่อถามผู้ใช้ว่าต้องการให้ระบบทำงานต่อหรือหยุด

2. รูปแบบผลลัพธ์ที่ต้องการ

- โปรแกรมแสดงข้อความระบุการเคลื่อนไหวของแขนหุ่นยนต์:
 - ตำแหน่ง A: "Moving arm to the left."
 - ตำแหน่ง B: "Moving arm to the right."
 - ตำแหน่ง C: "Moving arm down to pick up the object."
 - ไม่มีวัตถุ: "Returning arm to the starting position."
- มีการถามผู้ใช้ว่า "Do you want to continue? (Y/N):"

3. รูปแบบผลลัพธ์ที่ต้องการ

- ตำแหน่งของวัตถุ: A, B, C, หรือไม่มีวัตถุ
- ตัวเลือกจากผู้ใช้: Y (ทำงานต่อ) หรือ N (หยุดโปรแกรม)

4. ตัวแปรที่ใช้

- . **char position:** เก็บตำแหน่งของวัตถุ (A, B, C หรือ 'N' สำหรับไม่มีวัตถุ)
- . **char choice:** เก็บตัวเลือกของผู้ใช้ (Y หรือ N)

5. วิธีการประเมินผล

1. รับค่าตำแหน่งของวัตถุจากผู้ใช้

2. ตรวจสอบตำแหน่งของวัตถุ:

- ตำแหน่ง A: แนวเคลื่อนไปทางซ้าย

- ตำแหน่ง B: แนวเคลื่อนไปทางขวา

- ตำแหน่ง C: แนวเคลื่อนไปด้านล่างและจับวัตถุ

- ไม่มีวัตถุ: แขนกลับไปยังตำแหน่งเริ่มต้น

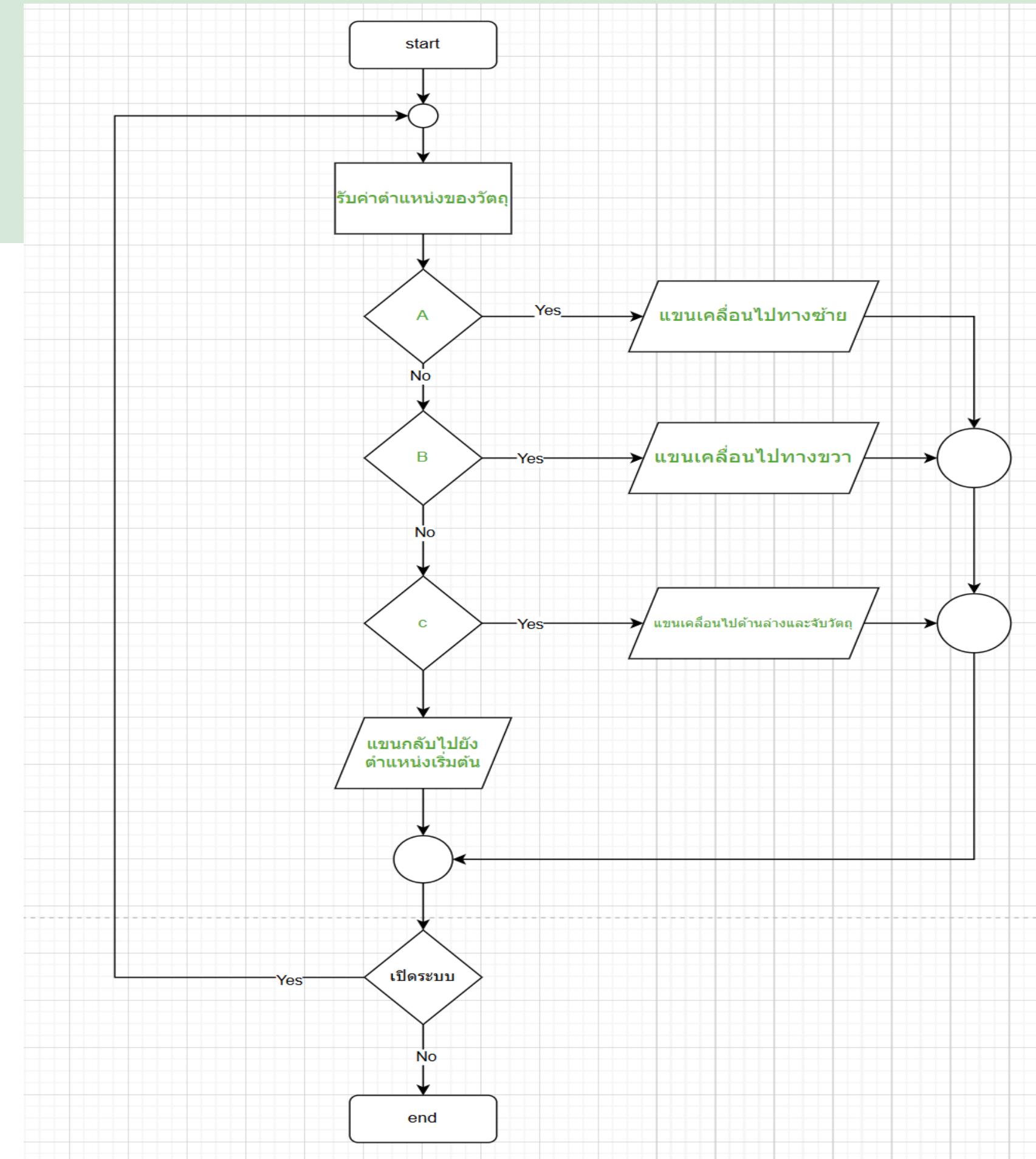
3. แสดงการกระทำของแขนหุ่นยนต์

4. ถามผู้ใช้ว่าจะทำงานต่อหรือไม่:

- ถ้าตอบ 丫: กลับไปเริ่มต้นใหม่

- ถ้าตอบ N: หยุดโปรแกรม

6. FlowChart



7. เขียนโปรแกรม

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

int main() {
    char position;
    char choice;

    do {

        return 0;
    }
```

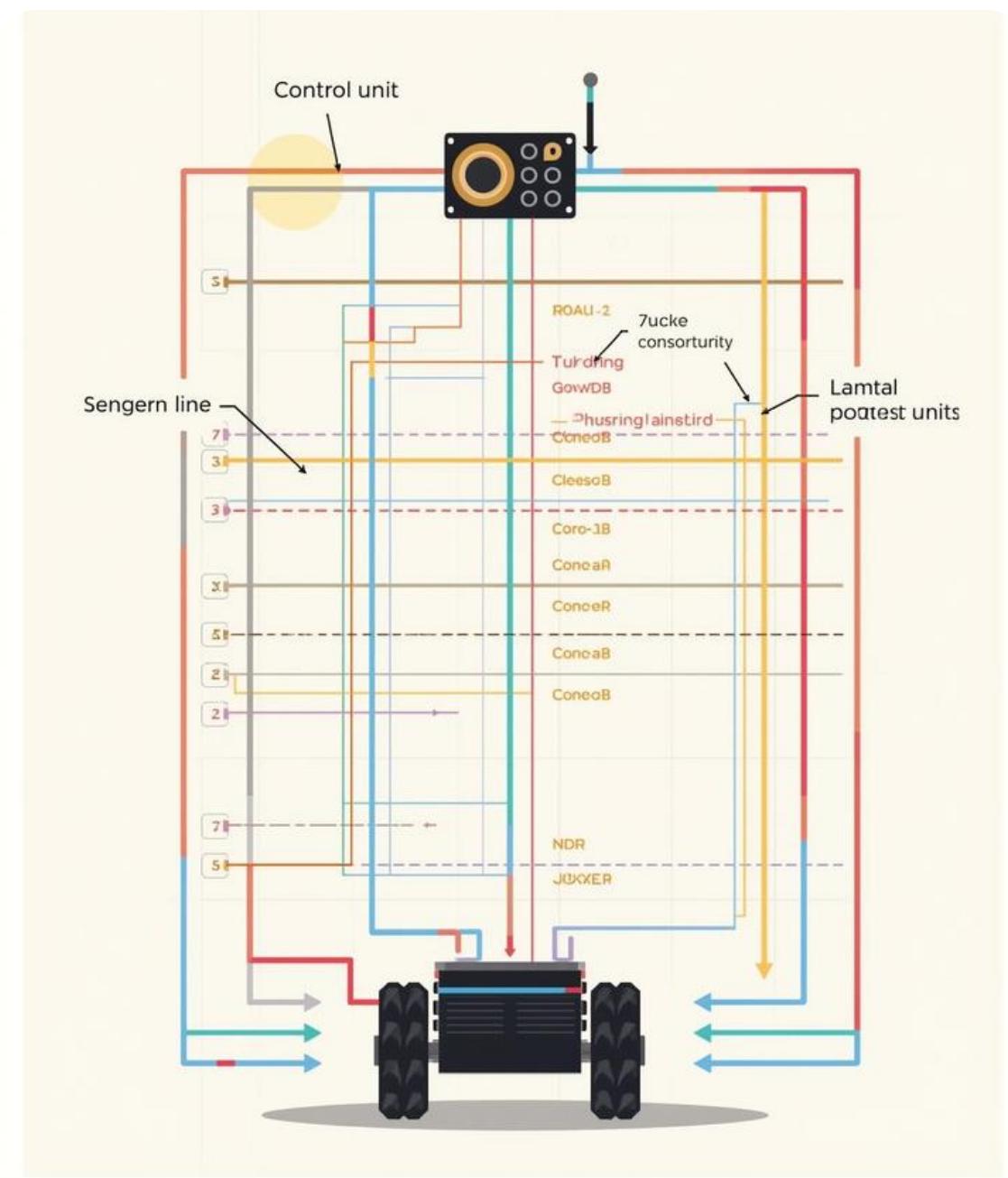
7. เขียนโปรแกรม

```
do {  
    // รับค่าตำแหน่งของวัตถุ  
    cout << "Enter the position of the object (A, B, C, or N for no object): ";  
    cin >> position;  
    // ตรวจสอบตำแหน่งและแสดงการกระทำของแขนหุ่นยนต์  
    switch (position) {  
        case 'A':  
        case 'a':  
            cout << "Moving arm to the left." << endl;  
            break;  
        default:  
            cout << "Invalid position entered!" << endl;  
    }  
    // ถามผู้ใช้ว่าจะทำงานต่อหรือไม่  
    cout << "Do you want to continue? (Y/N): ";  
    cin >> choice;  
  
} while (choice == 'Y' || choice == 'y'); // ทำงานต่อเมื่อผู้ใช้ตอบ Y  
cout << "Exiting the program..." << endl;
```

7. เขียนโปรแกรม

```
// ตรวจสอบตำแหน่งและแสดงการกระทำของแขนหุ่นยนต์
switch (position) {
    case 'A':
    case 'a':
        cout << "Moving arm to the left." << endl;
        break;
    case 'B':
    case 'b':
        cout << "Moving arm to the right." << endl;
        break;
    case 'C':
    case 'c':
        cout << "Moving arm down to pick up the object." << endl;
        break;
    case 'N':
    case 'n':
        cout << "Returning arm to the starting position." << endl;
        break;
    default:
        cout << "Invalid position entered!" << endl;
}
```

ระบบควบคุมรถวิ่งตามเส้น



ระบบควบคุมรถวิ่งตามเส้น

คำอธิบาย:

รถวิ่งตามเส้นต้องการวิ่งตามเส้นสีดำบนพื้นสีขาว รถมีเซนเซอร์สี (color sensor) ที่ติดตั้งที่ด้านหน้าของรถ เช่นเซนเซอร์จะตรวจจับสีและส่งข้อมูลกลับมาว่าเส้นสีดำอยู่ทางซ้าย, ขวา, หรือกลาง ระบบควบคุมจะตรวจสอบข้อมูลจากเซนเซอร์และทำการควบคุมรถให้เคลื่อนไหวตามเงื่อนไขต่อไปนี้:

- 1) ถ้าเส้นสีดำอยู่ทางซ้าย ให้รถเลี้ยวไปทางซ้าย
- 2) ถ้าเส้นสีดำอยู่ทางขวา ให้รถเลี้ยวไปทางขวา
- 3) ถ้าเส้นสีดำอยู่กลาง ให้รถวิ่งตรงไปข้างหน้า

สรุปและแบบฝึกหัด

For Loop

รูจํานวนรอบแน่นอน

While Loop

ทำเมื่อเงื่อนไขจริง

Do-While

ทำก่อน 1 ครั้งเสมอ

งานมอบหมาย (Assignment)

- ให้นักศึกษาเขียนโปรแกรมตัดเกรด โดยรับคะแนนซี่ๆ จนกว่าจะใส่ -1
- วาด Flowchart และเขียนโค้ดภาษา C ส่งในค้าบถัดไป



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

END



www.mut.ac.th