

COMPUTER PROGRAMMING

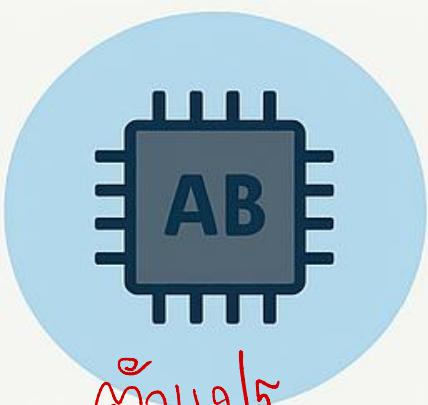
C++

ໄດ້ກຳສົດມາເຕີມທາງນະໝູນອົກກົດ



INPUT

Receive data
from the
user



ຕົວແປກ

VARIABLES

Store and
manage data



ກຽບມານາກາຣ

PROCESS

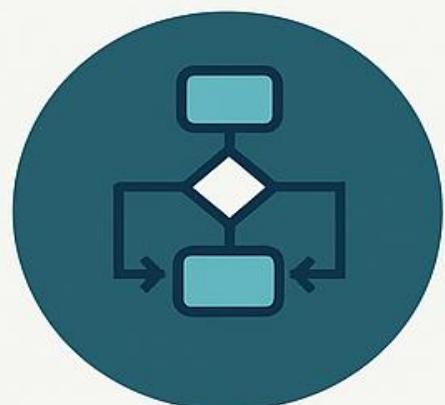
Perform
operations
on data



ລັກຄ່າໃຫຍ່ຕົວແປກແລະລົບ
ມາເຮັດວຽກນີ້ຈະຄົດ

OUTPUT

Produce
results for
the user



ALGORITHM

Define steps
to solve a
problem

โปรแกรมที่ 1 โปรแกรมคำนวณ BMI (Body Mass Index) ด้วยภาษา C++ โดยใช้ cin / cout และ if-else ตรวจสอบผลลัพธ์ตามเกณฑ์มาตรฐาน

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

```
int main() {
    // ประกาศตัวแปร
    float weight, height, bmi; // กองบังคับตัวแปรทางชื่อไม่ได้ทางค่า
    // รับข้อมูลจากผู้ใช้
    cout << "===== โปรแกรมคำนวณ BMI =====" << endl;
    cout << "กรุณากรอกน้ำหนัก (กิโลกรัม): ";
    cin >> weight; // ตัวแปรอินพุต
    cout << "กรุณากรอกส่วนสูง (เมตร): ";
    cin >> height; // ตัวแปรอินพุต
    // คำนวนค่า BMI Process
    bmi = weight / (height * height); // คำนวน BMI
    // แสดงผลค่า BMI
    cout << "\nค่า BMI ของคุณคือ: " << bmi << endl;
    // display
}
```

* ออกซอร์บ *

การคำนวณ

รับ input
พิมพ์ค่าผ่าน
หน้าจอคอมพิวเตอร์
weight
height
height
Unknown Variable

กระบวนการคำนวณ BMI
มี 3 ขั้นตอน คือ
-Input
-Process
-Output

ตัวแปรที่เก็บผลลัพธ์
การคำนวณ

รูปแบบที่ใช้ในการคำนวณ BMI นี้เป็นแบบการประมวลผล
ขั้นตอนๆ ๆ

// แสดงผลการประเมินสุขภาพ

```
if (bmi < 18.5) { // กรณี
    cout << "ผลการประเมิน: น้ำหนักน้อย / ผอม" << endl;
} else if (bmi >= 18.5 && bmi < 23) {
    cout << "ผลการประเมิน: สมส่วน (ปกติ)" << endl;
} else if (bmi >= 23 && bmi < 25) {
    cout << "ผลการประเมิน: น้ำหนักเกิน" << endl;
} else if (bmi >= 25 && bmi < 30) {
    cout << "ผลการประเมิน: โรคอ้วน ระดับ 1" << endl;
} else {
    cout << "ผลการประเมิน: โรคอ้วน ระดับ 2" << endl;
}
return 0;
```

ค่าของ BMI
ผู้คนที่ 2 ผู้ดีกว่าผู้คนที่ 1

if - else
โครงสร้างเชิงลำดับ
(Selection)

ก้าวที่ 2 ให้คำแนะนำ
ให้ความช่วยเหลือและสนับสนุน



ตัวอย่างผลการรันโปรแกรม

===== โปรแกรมคำนวณ BMI =====

กรุณารอกน้ำหนัก (กิโลกรัม): 70

กรุณารอกส่วนสูง (เมตร): 1.75

} ล้วน Input ผ่านทางหน้าจอ 70 ผ่าน keyboard
หากต้องป้อนทางอื่นๆ ก็ต้องป้อน weight

ค่า BMI ของคุณคือ: 22.8571

ผลการประเมิน: สมส่วน (ปกติ)

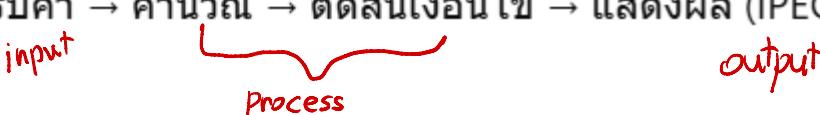
↓
ใช้ภาษา C++ หรือภาษา if-else

อธิบายโปรแกรม BMI แบบละเอียด

1) แนวคิดหลัก (Mapping จากโลจิริง → โค้ด)

- นิยาม: $BMI = \frac{\text{น้ำหนัก(กг.)}}{\text{ส่วนสูง(เมตร)}}^2$
- หน่วยสำคัญ: ส่วนสูงต้องเป็น “เมตร” ไม่ใช่เซนติเมตร
(เช่น 170 ซม. ต้องแปลงเป็น 1.70 ม. ก่อน)
- เกณฑ์ (ເອເຊີຍຕະວັນອອກເລື່ອງໃດ)
 - < 18.5 → น้ำหนักน้อย/ผอม
 - 18.5–22.9 → สมส่วน (ปกติ)
 - 23.0–24.9 → น้ำหนักเกิน
 - 25.0–29.9 → อ้วน ระดับ 1
 - ≥ 30.0 → อ้วน ระดับ 2

2) โครงสร้างโปรแกรม C++ ที่นักศึกษาควรเห็นชัด

- #include <iostream> เพื่อใช้ cin / cout
- #include <iomanip> เพื่อจัดรูปแบบทศนิยม
- main() เป็นจุดเริ่มต้นของโปรแกรม
- รับค่า → คำนวณ → ตัดสินใจ → แสดงผล (IPEO: Input–Process–Evaluate–Output)


The diagram illustrates the IPEO model. It consists of four boxes arranged horizontally. The first box is labeled 'Input' at the bottom left. An arrow points from it to the second box, which is labeled 'Process' at the bottom center. Another arrow points from the 'Process' box to the third box, which is labeled 'Output' at the bottom right. The fourth box, labeled 'Evaluate', is positioned above the 'Output' box.

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

int main() {
    cout << "===== BMI Calculator =====\n";
    double weightKg, heightM;

    // 1) รับอินพุตและตรวจสอบเบื้องต้น
    cout << "น้ำหนัก (kg): ";
    if (!(cin >> weightKg) || weightKg <= 0) {
        cout << "ข้อมูลน้ำหนักไม่ถูกต้อง\n";
        return 1;
    }

    cout << "ส่วนสูง (เมตร): ";
    if (!(cin >> heightM) || heightM <= 0) {
        cout << "ข้อมูลส่วนสูงไม่ถูกต้อง\n";
        return 1;
    }
}
```

```
// 2) คำนวณ BMI (ระวังวงเล็บ!)
double bmi = weightKg / (heightM * heightM);

// 3) จัดรูปแบบทศนิยม 2 ตำแหน่ง
cout << fixed << setprecision(2);
cout << "BMI = " << bmi << "\n";

// 4) จัดหมวดหมู่
string category;
if (bmi < 18.5) category = "น้ำหนักน้อย/ผอม";
else if (bmi < 23.0) category = "สมส่วน (ปกติ)";
else if (bmi < 25.0) category = "น้ำหนักเกิน";
else if (bmi < 30.0) category = "อ้วน ระดับ 1";
else category = "อ้วน ระดับ 2";

cout << "ผลการประเมิน: " << category << "\n";
return 0;
}
```

ทำไมเลือกแบบนี้

- ใช้ `double` เพื่อความแม่นยำมากกว่า `float`
- ใช้ `fixed << setprecision(2)` เพื่อคงที่ 2 ทศนิยม (อ่านง่าย)
- เช็กอินพุตเบื้องต้น หากพิมพ์ผิด/เป็นค่าติดลบให้จบโปรแกรมทันที → ลด error time

4) ไชการทำงานแบบทีละบรรทัด (สิ่งที่นักศึกษาควรเข้าใจ)

- ประกาศตัวแปร `weightKg`, `heightM` เก็บน้ำหนักและส่วนสูง (หน่วยที่ถูกต้อง!)
- อ่านค่าจาก `cin`
 - ถ้า `cin` อ่านไม่ได้ (เช่น พิมพ์ตัวอักษร) หรือค่าน้อยกว่าหรือเท่ากับ 0 → แจ้งเตือนและ `return 1`
- คำนวณ `bmi = weightKg / (heightM * heightM);`
 - จุดพลาดยอดฮิต: เขียน `weightKg / heightM * heightM` (จะได้เท่ากับ `weightKg` เพราะคูณกลับ)
ต้องใส่วงเล็บให้ถูก!
- แสดงผล BMI แบบทศนิยม 2 ตำแหน่ง
- เข้าเงื่อนไข `if-else if` ไล่จากซึ่งเล็กไปใหญ่ $\text{ถ้า } \text{bmi} < 18.5 \rightarrow \text{กรอก} < 18.5$
 - เขียนให้ “อ่านซึ่ง” ได้ง่าย เช่น `else if (bmi < 23.0)` หมายถึงค่าจาก 18.5–22.9 เพราะกรณี `< 18.5` ถูกกรองไปก่อนแล้ว

5) ตัวอย่าง Dry Run (อธิบายเฉพาะกิจขั้นตอน)

อินพุต: น้ำหนัก = 70, ส่วนสูง = 1.75

- คำนวณส่วนสูงยกกำลังสอง: $1.75 \times 1.75 = 3.0625$
- BMI = $70 \div 3.0625 = 22.8571\dots \rightarrow$ ปีดสองทศนิยม = 22.86
- เงื่อนไข: $22.86 < 23.0 \Rightarrow$ "สมส่วน (ปกติ)"

6) จุดพลาดที่มักเจอ + วิธีแก้

- หน่วยผิด: ป้อน 170 โดยคิดว่าเป็นเมตร \rightarrow ต้องบังคับ/ย้ำว่า "เมตร" และยกตัวอย่าง 1.70
- ใช้ชนิดข้อมูลไม่เหมาะสม: ใช้ `int` ทำให้ทศนิยมหาย \rightarrow ใช้ `double`
- ลำดับการคำนวณผิด: ลืมวงเล็บ \rightarrow สาหริตร์ความต่างด้วยตัวเลขจริง
- ขอบเขตช่วง (Boundary): ตรวจให้ดีว่า 23.0 จะตกช่วงไหน เขียนช่วงแบบ "น้อยกว่า" ໄลชื่น อ่านง่ายกว่า
- หารด้วยศูนย์: กันไว้ด้วยการเช็ค `heightM > 0`

7) แบบฝึกหัดต่อ�อด (พาไปสู่ “Loop / Function” สักพานหนึ่ง)

1. **Formatting:** ให้พิมพ์ BMI 2 และ 1 หลักนิยม สลับกัน (ฝึก `setprecision`)
2. **แปลงหน่วย:** รับส่วนสูงเป็น “เซนติเมตร” และแปลงเป็นเมตรก่อนคำนวณ

cpp

```
double heightCm; cin >> heightCm;  
double heightM = heightCm / 100.0;
```

3. แนะนำช่วงน้ำหนักเหมาะสม (เพื่อ “ปกติ”):

- ช่วงน้ำหนักที่เหมาะสม = $18.5 \times h^2, 22.9 \times h^2$
ให้โปรแกรมคำนวณช่วงนี้และแสดงผล

4. **รับหลายคน:** ใช้ `for` หรือ `while` รับข้อมูลทีละคน คำนวณ BMI และ

- หากค่า เฉลี่ย สูงสุด ต่ำสุด (ฝึกตัวแปรสะสม/เบรียบเทียบ)

5. **แยกฟังก์ชัน:**

cpp

```
double calcBMI(double w, double h);  
string classify(double bmi);
```

■ บทนำสู่ while loop

1) Concept

- while (เงื่อนไข) = ทำซ้ำคราวที่เงื่อนไขเป็นจริง
- หมายความว่า ไม่รู้จำนวนรอบแน่นอน (เช่น รับข้อมูลจนกว่าจะเจอเงื่อนไขหยุด)

cpp

```
while (เงื่อนไข) {  
    // คำสั่งที่จะทำซ้ำ  
}  
End
```

คำสั่งวนลูปนี้
ดำเนินการ ไป จนกว่า ค่า ของ / จะ ไม่
Begin ดู ถูก ใจ ให้ ทำ คำ สั่ง ต่อ ไป
ต่อ จน ถึง เห็น ใจ

โปรแกรมที่ 2

◆ พิมพ์เลข 1 ถึง 5

cpp

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

int main() {
    int i = 1; ① initialize // กำหนดค่าเริ่มต้น
    while (i <= 5) { ② Update // ทำซ้ำครบที่ i <= 5
        cout << i << " "; ③ // แสดงค่า i
        i++; ④ Update // เพิ่มค่า i ทีละ 1
    }
    return 0;
}
```

คลิ๊ก Run Program แล้วอ่าน กด F5
เมื่อบนเครื่องคอมพิวเตอร์ต้องการคำนวณ

$i++$
 $i = i + 1$

i	$i < 5$	Edit loop	Display i	$i++$
1	๖๔๙๙	✓	1	2
2	๖๔๙๙	✓	2	3
3	๖๔๙๙	✓	3	4
4	๖๔๙๙	✓	4	5
5	๖๔๙๙	✓	5	6
6	๖๔๙๙	✗	—	—

ถ้าต้องแก้ไขให้เรา Run 1 - 100
*รับแจ้งข้อความจาก loop *

★ สิ่งที่ นศ. จะเห็น

- ถ้า ลีม $i++$ จะกลายเป็น loop ไม่รู้จบ
- ถ้า i เริ่มจาก 0 หรือ 2 ผลลัพธ์จะเปลี่ยน → เข้าใจการ initialize + update

กำหนดค่าผู้ตัวตัว
ให้กับ i

ถ้าจะสรุปเร็วๆ นี้ $while$ loop จะมี 4 อย่าง
① ค่าเริ่มต้น ② เงื่อนไข ③ ค่าสิ้น / ชุดค่าสิ้น ④ การอัพเดต
Update

โปรแกรมที่ 3

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

int main() { input

```
    int n;
    cout << "กรอกค่า N: ";
    cin >> n;
```

ตัวแปร output ของขั้นตอนคำนวณ

```
    int i = 1, sum = 0; ①
    while (i <= n) { ②
        sum += i; ③ // บวกสะสม
        i++; ④
```

} process

```
cout << "ผลรวม 1 ถึง " << n << " = " << sum << endl;
return 0;          คำนวณหาผลรวมของตัวเลขที่ 1 ถึง N
```

output

n	$i \leq 6$	$sum = sum + i$	$i++$	Display (sum)
1	๖	$sum = 0 + 1 = 1$	2	—
2	๖	$sum = 1 + 2 = 3$	3	—
3	๖	$sum = 3 + 3 = 6$	4	—
4	๖	$sum = 6 + 4 = 10$	5	—
5	๖	$sum = 10 + 5 = 15$	6	—
6	๖	$sum = 15 + 6 = 21$	7	—
7	—	—	—	21

$i++$
 $i = i + 1$



ได้เลือกรับ

ให้ตัวที่คุณส่งมาเป็น ภาษา C++ ใช้ในการหาผลรวมตั้งแต่ 1 ถึง N โดยใช้ สูตร while

อธิบายที่ละเอียดอ้างอิง

cpp

- ต้อง imports iostream มาใช้สำหรับ cin และ cout
- ใช้ namespace std เพื่อไม่ต้องพิมพ์ std:: ทุกครั้ง

cpp

```
int main() {
    int n;
    cout << "กรอกค่า N: ";
    cin >> n;
```

- ประกาศตัวแปร n
- แสดงข้อความให้ผู้ใช้กรอกค่า N
- รับค่าที่ผู้ใช้พิมพ์มาเก็บใน n

cpp

```
int i = 1, sum = 0;
while (i <= n) {
    sum += i; // บวกสะสม
    i++;
}
```

- ประกาศตัวแปร i = 1 และ sum = 0
- ใช้ while (i <= n) หมายถึงทำ 반วนไปเรื่อยๆ จนกว่า i จะมากกว่า n
- sum += i; คือการบวกสะสมใน sum
- i++; เพิ่มค่า i ทีละ 1 ทุกครั้ง

ผลลัพธ์ sum จะได้ผลรวมของตัวเลขที่ 1 ถึง n

cpp

```
cout << "ผลรวม 1 ถึง " << n << " = " << sum << endl;
}
```

- แสดงผลลัพธ์ ผลรวมจาก 1 ถึง N เท่ากันเท่าไร
- return 0; จบโปรแกรม

ลักษณะผลลัพธ์

เข้าไปค่า N = 5

```
กรอกค่า N: 5
ผลรวม 1 ถึง 5 = 15
```

(พ Bere: 1+2+3+4+5 = 15)

สรุปง่าย ๆ: โปรแกรมนี้คือ หาผลรวมของตัวเลขจาก 1 ถึง N โดยใช้ สูตร while

โปรแกรมที่ 4

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

```
int main() {
    int num;
    cout << "ป้อนตัวเลข (0 = จบ): ";
    cin >> num;
```

1 รีไซเคิลชีวิต

```
2 จัดการข้อมูล
while (num != 0) {
    cout << "คุณป้อน: " << num << endl; 3 คำสั่ง/ธุลย์สั่ง
    cout << "ป้อนใหม่ (0 = จบ): ";
    cin >> num;
}
```

4 ติดตาม

```
cout << "จบโปรแกรม" << endl;
return 0;
}
```

✿ ใช้สาหร่ายที่ loop จะวนต่อจนกว่าจะเจอ 0 → นักศึกษาเข้าใจ “เงื่อนไขหยุด”

num	$n = 0$	$Cin = num$
5	จริง	1
1	จริง	0
0	偽	จบการทำงาน

เขียนและครับ โปรแกรมที่ดูคล้ายๆ เป็นตัวอย่าง while loop ที่ใช้วันรันค่าตัวเลขจากผู้ใช้ในเรื่องๆ จนกว่าจะป้อนเลข 0

อธิบายโค้ดที่ละเอียด

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

- ใช้ iostream สำหรับการรับค่า (cin) และแสดงผล (cout)

```
cpp
int main() {
    int num;
    cout << "ป้อนตัวเลข (0 = จบ): ";
    cin >> num;
```

- ประกาศตัวแปร num เก็บค่าตัวเลขที่ได้ใช้กรอก
- แสดงข้อความให้ป้อนตัวเลข (ใส่ 0 เพื่อจบ)
- รับค่าที่ผู้ใช้พิมพ์เข้ามาเก็บใน num

```
cpp
while (num != 0) {
    cout << "คุณป้อน: " << num << endl;
    cout << "ป้อนใหม่ (0 = จบ): ";
    cin >> num;
}
```

- เลื่อนไห้ while (num != 0) หมายถึง ถ้า num ไม่เท่ากับ 0 ให้ทำงานต่อ
- แสดงค่าที่ผู้ใช้ป้อน
- ให้ผู้ใช้ป้อนใหม่อีกครั้ง → วนค่าเก็บใน num
- วนซุบไปเรื่อยๆ จนกว่าจะป้อน 0

```
cpp
cout << "จบโปรแกรม" << endl;
return 0;
```

- เมื่อผู้ใช้ป้อนและ 0 เลื่อนไห้ while (num != 0) จะเป็น เฟลส (false) → ออกจากการloop
- แสดงข้อความ “จบโปรแกรม” และสิ้นสุดการทำงาน

ตัวอย่างการทำงาน

```
cpp
ป้อนตัวเลข (0 = จบ):
ป้อนใหม่: 5
ป้อนใหม่ (0 = จบ): 12
ป้อนใหม่: 12
ป้อนใหม่ (0 = จบ): -7
ป้อนใหม่ (0 = จบ): -7
ป้อนใหม่ (0 = จบ): 0
จบโปรแกรม
```

สรุปว่า ๆ โปรแกรมนี้ใช้ while loop เพื่อตรวจสอบว่า

- ถ้า ยังไม่ป้อน 0 → วนต่อ
- ถ้า ป้อน 0 → หยุด

คุณอาจได้พบเมื่อตอน do...while (วงกònเมื่อก็พัง) ของโปรแกรมที่ใช้กูบึร์บันทึกในครั้น?

โปรแกรมที่ 5 จงเขียนโปรแกรม หาผลรวมของเลขคู่ ตั้งแต่ 0 – 100 พร้อมคำอธิบายและบรรยาย

main.cpp

```

1 #include<iostream>
2 using namespace std; //เขียนบกต.
3 int main(){
4     int n;
5     int sum=0; //ลากมาต่อไปนี้ sum ให้ชื่อคล้ายๆกัน
6
7     for(int i=0;i<=100;i++){ //loop for วนลูป 1 ครั้ง 0-100
8         if (i%2==0){ //ถ้า i เป็น偶数
9             sum+=i; //เพิ่มค่า i ลงใน sum
10        } //แล้วก็ต้องมาค่า i ยังคงอยู่ใน sum
11    }
12
13    cout<<"ผลรวมเลขคู่ 0-100 คือ "<<sum<<endl;
14    return 0;
15 }
16 
```

ผลลัพธ์ที่ได้รับ

ผลรวมเลขคู่ 0-100 คือ 2550

Program finished with exit code 0
Press ENTER to exit console.

$i++$
 $i = i + 1$

n	$i \leq 6$	$sum = sum + i$	$i++$	Display(sum)
1	๖	$sum = 0 + 1 = 1$	2	—
2	๖	$sum = 1 + 2 = 3$	3	—
3	๖	$sum = 3 + 3 = 6$	4	—
4	๖	$sum = 6 + 4 = 10$	5	—
5	๖	$sum = 10 + 5 = 15$	6	—
6	๖	$sum = 15 + 6 = 21$	7	—

$$2 + 4 + 6$$

โปรแกรมที่ 6 จงเขียนโปรแกรมพิมพ์เลขคู่ตั้งแต่ 2-100

โปรแกรมที่ 7

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

int main() {
    double weight, height;

    cout << "===== โปรแกรมคำนวณ BMI หลายคน =====\n";
    cout << "(ป้อนน้ำหนัก = 0 เพื่อจบโปรแกรม)\n\n";

    cout << "กรอกน้ำหนัก (kg): ";
    cin >> weight;

    while (weight != 0) {
        cout << "กรอกส่วนสูง (m): ";
        cin >> height;

        double bmi = weight / (height * height);
        cout << fixed << setprecision(2);
        cout << "BMI = " << bmi << " ";
```

```
        if (bmi < 18.5) cout << "(ผอม)";
        else if (bmi < 23.0) cout << "(สมส่วน)";
        else if (bmi < 25.0) cout << "(น้ำหนักเกิน)";
        else if (bmi < 30.0) cout << "(อ้วน ระดับ 1)";
        else cout << "(อ้วน ระดับ 2)";
        cout << "\n\n";

        cout << "กรอกน้ำหนัก (kg): ";
        cin >> weight;
    }

    cout << "\nจบการทำงานโปรแกรม" << endl;
    return 0;
}
```

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

- `#include <iostream>` : เรียกใช้ **ไลบรารีมาตรฐาน** สำหรับรับค่า (`cin`) และแสดงผล (`cout`)
- `#include <iomanip>` : ใช้สำหรับควบคุมรูปแบบการแสดงผล เช่น `setprecision(2)` เพื่อกำหนดทศนิยม
- `using namespace std;` : ทำให้เราเขียน `cin`, `cout` ได้เลย ไม่ต้องพิมพ์ `std::cin`, `std::cout`

```
int main() {
    double weight, height;
```

- `int main()` : จุดเริ่มต้นของการทำงานโปรแกรม
- `double weight, height;` : ประกาศตัวแปร **น้ำหนัก** และ **ส่วนสูง** เป็นชนิด `double` (ทศนิยม)

```
cout << "===== โปรแกรมคำนวณ BMI หลายคน =====\n";
cout << "(ป้อนน้ำหนัก = 0 เพื่อจบโปรแกรม)\n\n";
```

- แสดงข้อความแนะนำผู้ใช้
- \n = ขึ้นบรรทัดใหม่
- \n\n = เว้นบรรทัดว่าง 1 บรรทัด เพื่อให้ข้อความดูเป็นระเบียบ

```
cout << "กรอกน้ำหนัก (kg): ";
cin >> weight;
```

- แสดงข้อความ “กรอกน้ำหนัก”
- รับค่าที่ผู้ใช้กรอกเข้ามาเก็บในตัวแปร weight

```
while (weight != 0) {
```

- ใช้ while loop เพื่อทำซ้ำ
- เงื่อนไขคือ วนซ้ำจนกว่าน้ำหนัก ≠ 0
- ถ้าผู้ใช้กรอก 0 จะหยุด loop และข้ามไปทำงานบรรทัดหลังวงล็บปิด

```
cout << "กรอกส่วนสูง (m): ";
cin >> height;
```

- รับค่าความสูงจากผู้ใช้ในหน่วย เมตร (m)

```
double bmi = weight / (height * height);
```

- คำนวณค่า BMI ตามสูตร

$$\text{BMI} = \frac{\text{weight}}{(\text{height})^2}$$

- เก็บผลลัพธ์ไว้ในตัวแปร `bmi`

```
cout << fixed << setprecision(2);
cout << "BMI = " << bmi << " ";
```

- `fixed + setprecision(2)` : แสดงผลทศนิยม 2 ตำแหน่ง (เช่น 22.86)
- แสดงค่า BMI ที่คำนวณได้

```
if (bmi < 18.5) cout << "(ผอม)";  
else if (bmi < 23.0) cout << "(สมส่วน)";  
else if (bmi < 25.0) cout << "(น้ำหนักเกิน)";  
else if (bmi < 30.0) cout << "(อ้วน ระดับ 1)";  
else cout << "(อ้วน ระดับ 2)";
```

- เป็น เงื่อนไข if-else สำหรับจัดหมวดหมู่ BMI
- ໄລ່ตรวจທີ່ລະເງື່ອນໄຂ ເຮັດຈາກນ້ອຍໄປນາກ
- ເນື້ອເຈືອເງື່ອນໄຂທີ່ຕຽງແລ້ວຈະຫຼຸດຕຽບທັນທີ

```
cout << "\n\n";
```

- ເວັນນຽທັດ 1 ນຽທັດໜັງແສດງຜລ ເພື່ອຄວາມອ່ານຸ່າຍ

```
    cout << "กรอกน้ำหนัก (kg): ";
    cin >> weight;
}
```

- หลังคำนวณเสร็จ 1 รอบ จะกลับมาถามน้ำหนักใหม่อีกครั้ง
- ถ้าผู้ใช้ใส่ค่า $\neq 0 \rightarrow$ loop จะทำงานต่อ
- ถ้าใส่ค่า $= 0 \rightarrow$ loop จะหยุด

```
cout << "\nจบการทำงานโปรแกรม" << endl;
return 0;
}
```

- เมื่อผู้ใช้กรอก 0 \rightarrow โปรแกรมออกจากการ while loop
- แสดงข้อความว่า “จบการทำงานโปรแกรม”
- `return 0;` = ส่งค่ากลับว่าโปรแกรมทำงานสำเร็จ

📌 สรุป Flow ของโปรแกรม

1. โปรแกรมถ่าน้ำหนัก → ถ้า 0 จบเลย
2. ถ้าไม่ใช่ 0 → รับส่วนสูง
3. คำนวณ BMI และแสดงผล พร้อมเกณฑ์
4. กลับไปถ่าน้ำหนักใหม่อีกครั้ง (วน loop)
5. วนซ้ำเรื่อย ๆ จนกว่าน้ำหนัก = 0 → โปรแกรมจบ

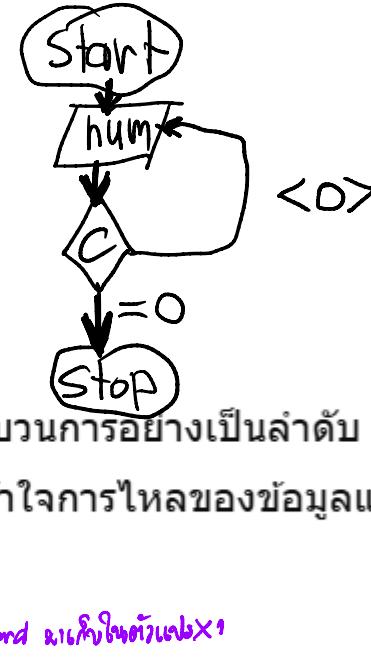
⭐ เกริ่นนำเรื่องผังงาน (Flowchart)

1) ผังงานคืออะไร?

- ผังงาน (Flowchart) คือแผนภาพที่ใช้แสดงขั้นตอนการทำงานของโปรแกรมหรือกระบวนการอย่างเป็นลำดับ
- ใช้สัญลักษณ์มาตรฐาน (เช่น วงรี สี่เหลี่ยม ข้าวหลามตัด) เชื่อมด้วยลูกศร → เพื่อให้เข้าใจการไหลของข้อมูลและ| | | |
| --- | --- | --- |
| - Start - process - stop - Equation - display | - Decision ตรวจสอบ - condition | - Read (x_1) รับค่าจาก key board บันทึกไว้ในตัวแปร x_1 |
| | | |
กระบวนการ

2) ทำไมต้องใช้ผังงาน? (ความสำคัญ)

- เข้าใจง่าย : แทนที่จะมองเป็นโค้ดยาว ๆ การมองเป็นภาพทำให้เข้าใจโครงสร้างได้รวดเร็ว
- ช่วยออกแบบ : ก่อนเขียนโค้ดจริง ถ้าเขียนผังงาน จะช่วยวางแผนลำดับความคิดไม่ให้หลงทาง
- ตรวจสอบข้อผิดพลาด : เห็นเงื่อนไข, วนน, จุดเริ่ม–จบ ชัดเจน → ลดความผิดพลาดจากการลืมหรืออุบัติไม่จบ
- สื่อสารร่วมกัน : นักศึกษากับเพื่อน, หรือนักพัฒนาหลายคน สามารถใช้ผังงานอธิบายแทนคำพูดหรือโค้ด
- เป็นพื้นฐานของวิศวกรรมซอฟต์แวร์ : เก็บทุกขั้นตอนใน SDLC (Software Development Life Cycle) ใช้ผังงานหรือโปรแกรมช่วยอธิบาย



ทักษะทางวิชาชีพ
Block Diagram
ผังงานทาง邏輯
ผังงาน (อัลกอริズึม)

3) ความสัมพันธ์กับการเขียนโปรแกรม Algorithm

- โคด = รายละเอียด (Implementation)
- ผังงาน = ภาพรวม (Logic/Algorithm)
- ตัวอย่าง: โปรแกรม BMI ที่เขียนไปแล้ว มี Input → Process → Output + Loop
→ เมื่อแปลงเป็นผังงาน นักศึกษาจะเห็นภาพ “โปรแกรมทำงานซ้ำไปเรื่อย ๆ จนกว่าผู้ใช้จะใส่ 0” ได้ทันที

4) ประโยชน์สำหรับนักศึกษา

- ช่วยให้นักศึกษาที่เพิ่งเริ่มเขียนโคด เข้าใจ “ตรรกะ” มากกว่า “ไวยากรณ์”
โครงสร้างทางตรรกะ ไม่ว่าจะดูยังไงก็ตาม
- เวลาทำโจทย์ซับซ้อน เช่น มีทั้ง if และ while ผังงานจะช่วยไม่ให้สับสน
- เวลา สอน/ทำงานกลุ่ม อาจารย์ขอดูแค่ผังงานก็พอเข้าใจได้ว่าคิดอย่างไร

FLOWCHART SYMBOLS



Oval
Start/End



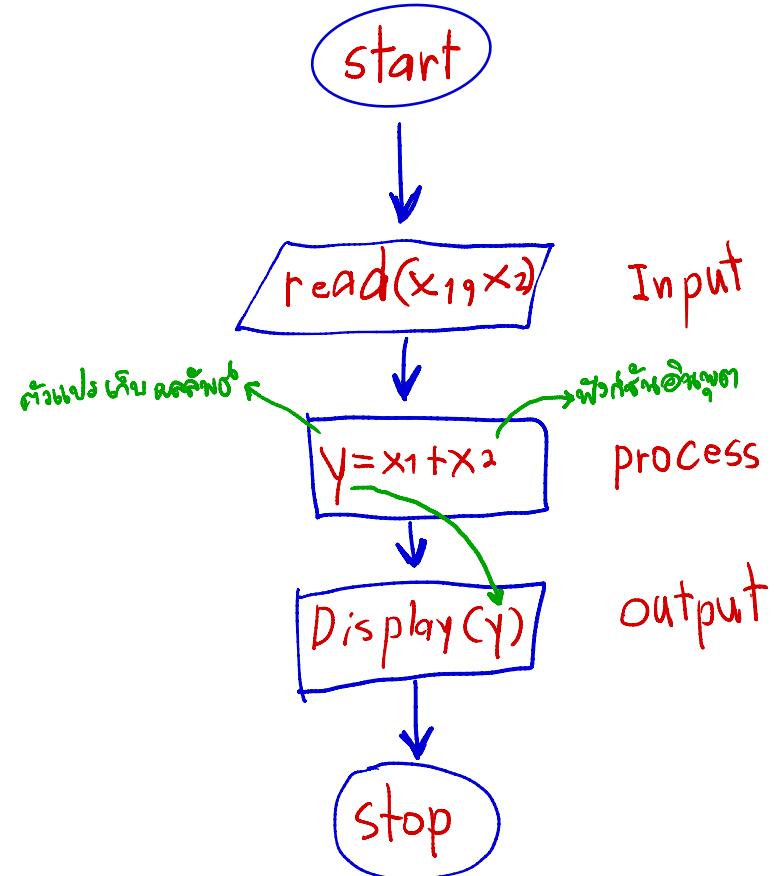
Rectangle
Process
-Display



Parallelogram
Input/Output



Diamond
Decision



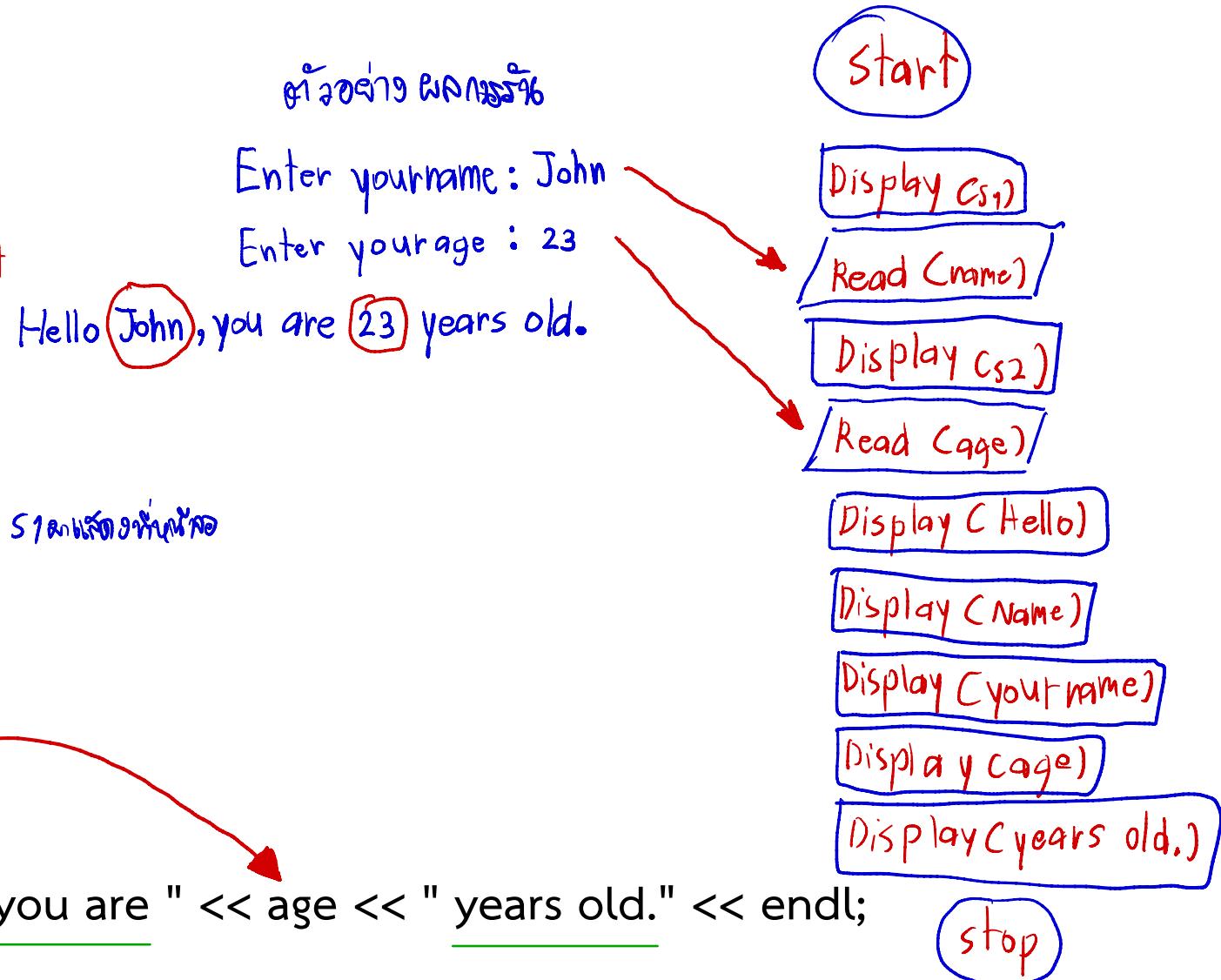
ເວັບໄຈສິນ
Enter x₁:5
Enter x₂:10
Result :15

ผังงานที่ 1 ให้ นศ. เขียนผังงานของโปรแกรม Input–Process–Output ดังต่อไปนี้

รับซื้อ → รับอายุ → แสดงขอความ

```
#include <iostream>
using namespace std; // input, output
int main() { start
    string name; // ព័ត៌មានអ្នកសរុប
    int age; // អាយុអ្នកសរុប
    cout << "Enter your name: "; S1
    cin >> name; // ទូទាត់ព័ត៌មានអ្នកសរុប
    cout << "Enter your age: "; S2
    cin >> age; // ទូទាត់អាយុអ្នកសរុប
    cout << "Hello " << name << ", you are " << age << " years old." << endl;
    return 0;
} stop
```

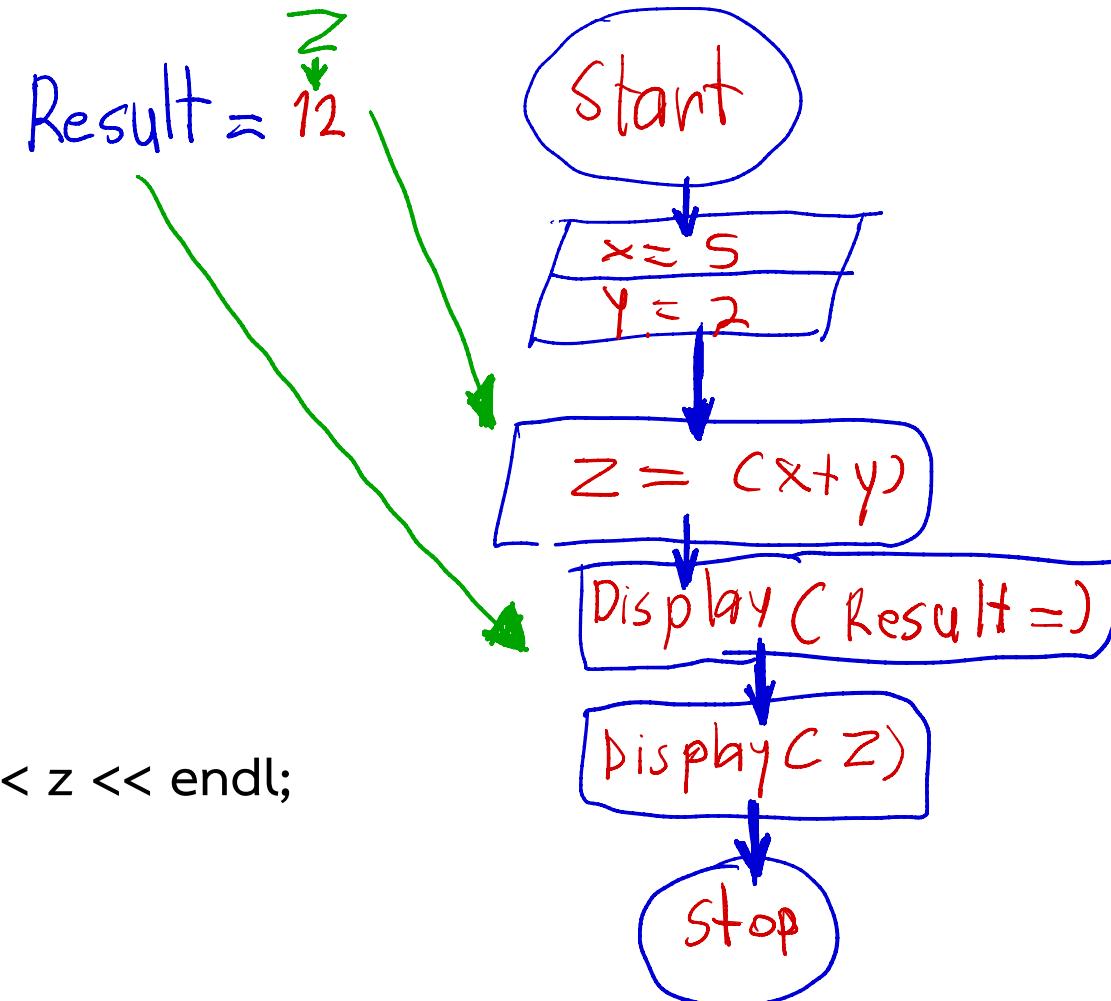
Enter your name : John
Enter your age : 23
Hello John, you are 23 years old.



ผังงานที่ 2 ใน นศ. เขียนผังงานของโปรแกรม Input–Process–Output ดังต่อไปนี้

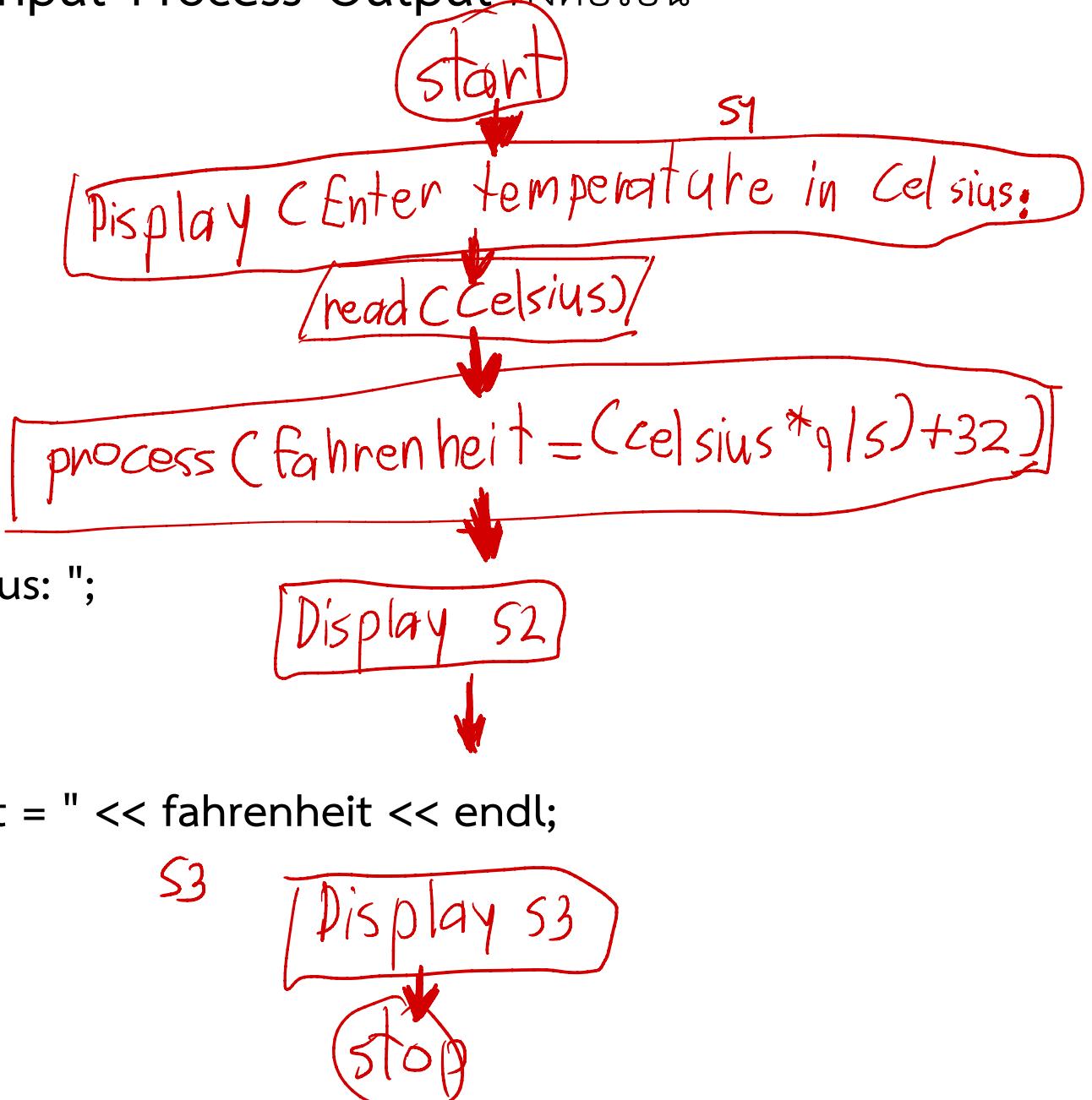
```
#include <iostream>
using namespace std;

int main() {
    int x = 5;
    int y = 2;
    int z = x + y;
    cout << "Result = " << z << endl;
    return 0;
}
```



ผังงานที่ 3 ให้ นศ. เขียนผังงานของโปรแกรม Input–Process–Output ดังต่อไปนี้

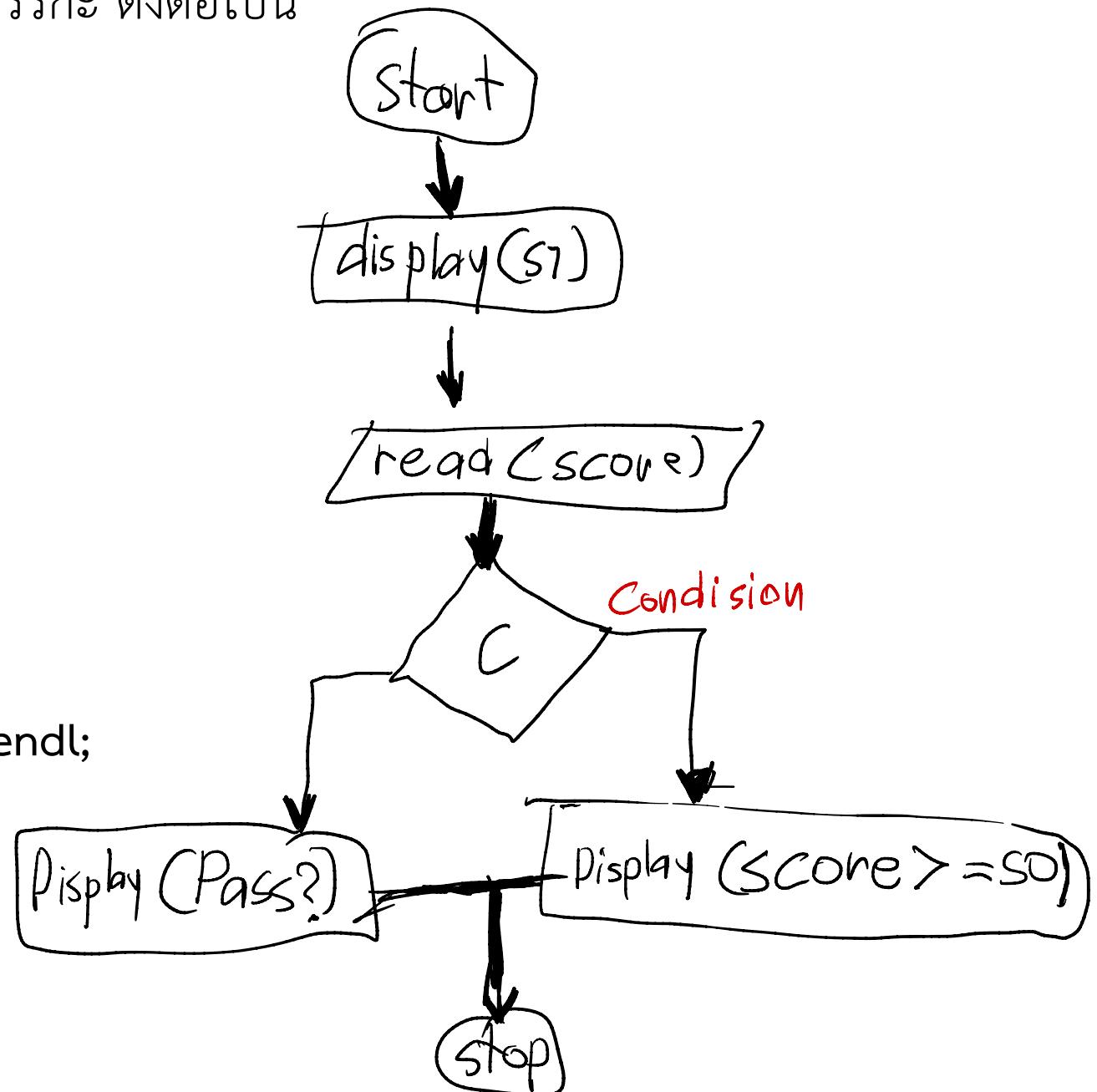
```
#include <iostream>  
using namespace std;  
  
int main() {  
    float celsius, fahrenheit;      S1  
    cout << "Enter temperature in Celsius: ";  
    cin >> celsius;  
    fahrenheit = (celsius * 9 / 5) + 32;      S2  
    cout << "Temperature in Fahrenheit = " << fahrenheit << endl;  
    return 0;                            S2  
}
```



ผังงานที่ 4 ให้ นศ. เขียนผังงานของโปรแกรมเชิงตรรกะ ดังต่อไปนี้

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

int main() {
    int score;
    cout << "Enter your score: ";
    cin >> score;
    cout << "Pass? " << (score >= 50) << endl;
    return 0;
}
```

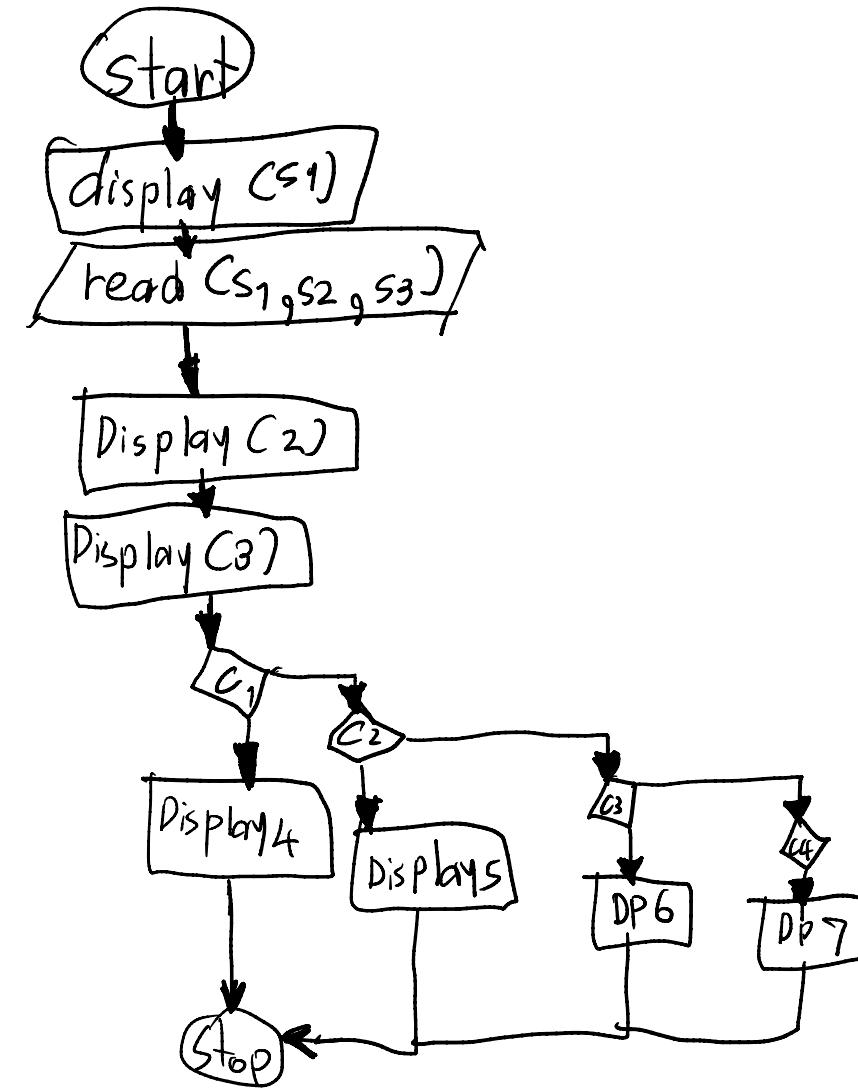


ເອກະລານ ຈົດ

ສັນຕິພາບ ທາງເລືອດ

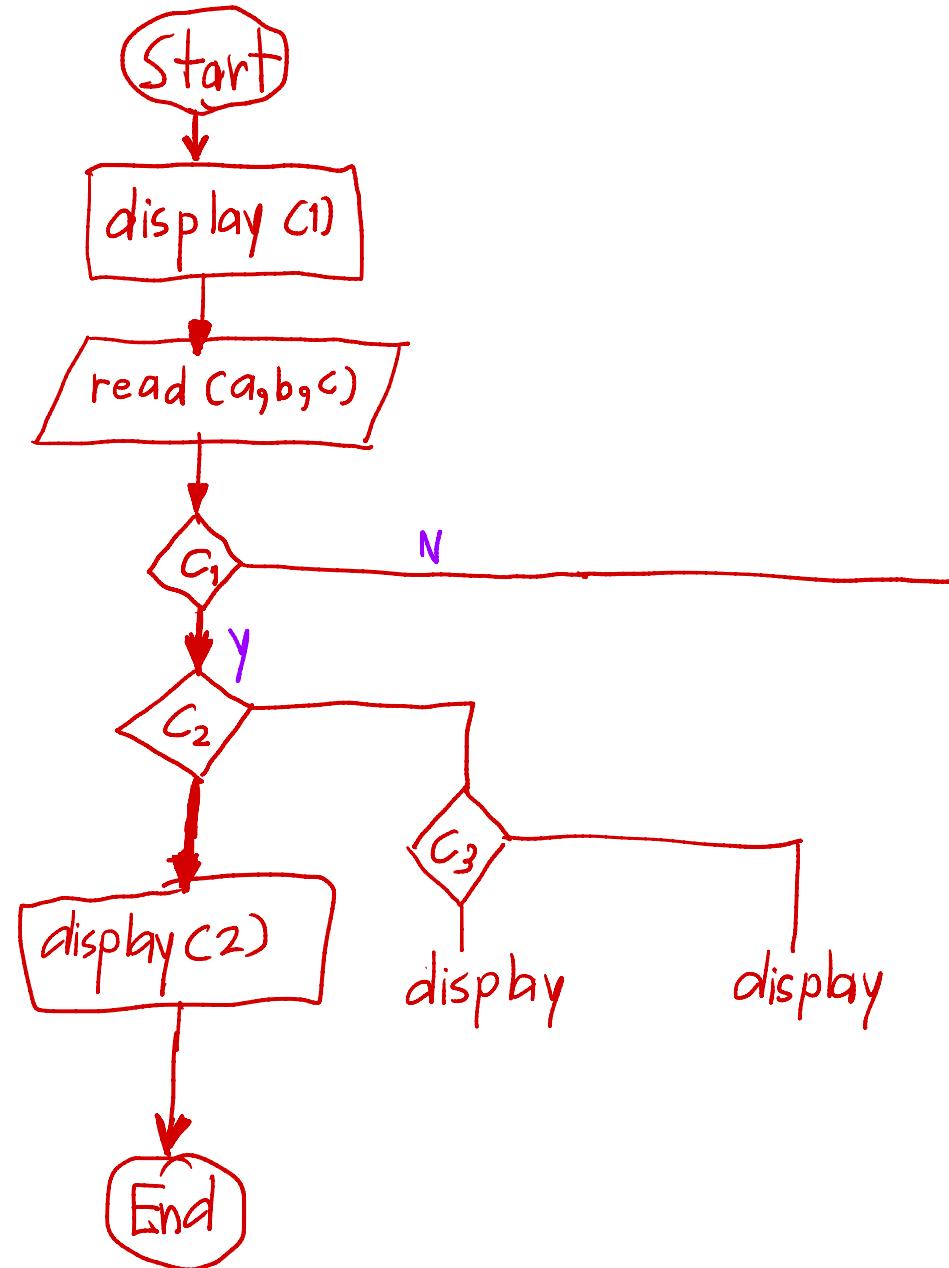
ຜັງງານທີ 5 ໃຫ້ ນສ. ເບີຍນຜັງງານຂອງໂປຣແກຣດັ່ງຕ້ອໄປນີ້

```
#include <iostream>
using namespace std;
int main() {
    float s1, s2, s3;           s1
    cout << "Enter 3 scores: ";
    cin >> s1 >> s2 >> s3;      2
    float avg = (s1 + s2 + s3) / 3.0;
    cout << "Average = " << avg 3 << endl;
    if (avg >= 80) cout << "Grade = A\n"; 4
    else if (avg >= 70) cout << "Grade = B\n"; 5
    else if (avg >= 60) cout << "Grade = C\n"; 6
    else if (avg >= 50) cout << "Grade = D\n"; 7
    else cout << "Grade = F\n"; 8
    return 0;
}
```



ผังงานที่ 6 ให้ นศ. เขียนผังงานของโปรแกรมดังต่อไปนี้

```
#include <iostream>
using namespace std;
int main() { Start
    int a, b, c;           1
    cout << "Enter three sides: ";
    cin >> a >> b >> c;
    // ตรวจว่าเป็นสามเหลี่ยมได้ไหม
    if (a + b > c && a + c > b && b + c > a) { Begin
        if (a == b && b == c) cout << "Equilateral triangle\n"; 2
        else if (a == b || a == c || b == c) cout << "Isosceles triangle\n";
        else cout << "Scalene triangle\n"; 9 A
    } else {
        cout << "Not a valid triangle\n";
    }
    return 0;
} Stop
```



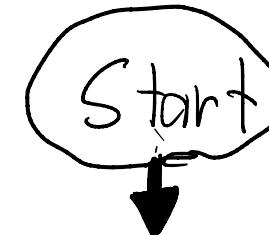
ผังงานที่ 7 ให้ นศ. เขียนผังงานของโปรแกรมดังต่อไปนี้

◆ พิมพ์เลข 1 ถึง 5

cpp

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

int main() {
    int i = 1; // กำหนดค่าเริ่มต้น
    while (i <= 5) { // ทำซ้ำครบที่ i <= 5
        cout << i << " "; // แสดงค่า i
        i++; // เพิ่มค่า i ทีละ 1
    }
    return 0;
}
```



✿ สิ่งที่ นศ. จะเห็น

- ถ้า ลีม `i++` จะกลายเป็น loop ไม่รู้จบ
- ถ้า `i` เริ่มจาก 0 หรือ 2 ผลลัพธ์ก็เปลี่ยน → เข้าใจการ initialize + update

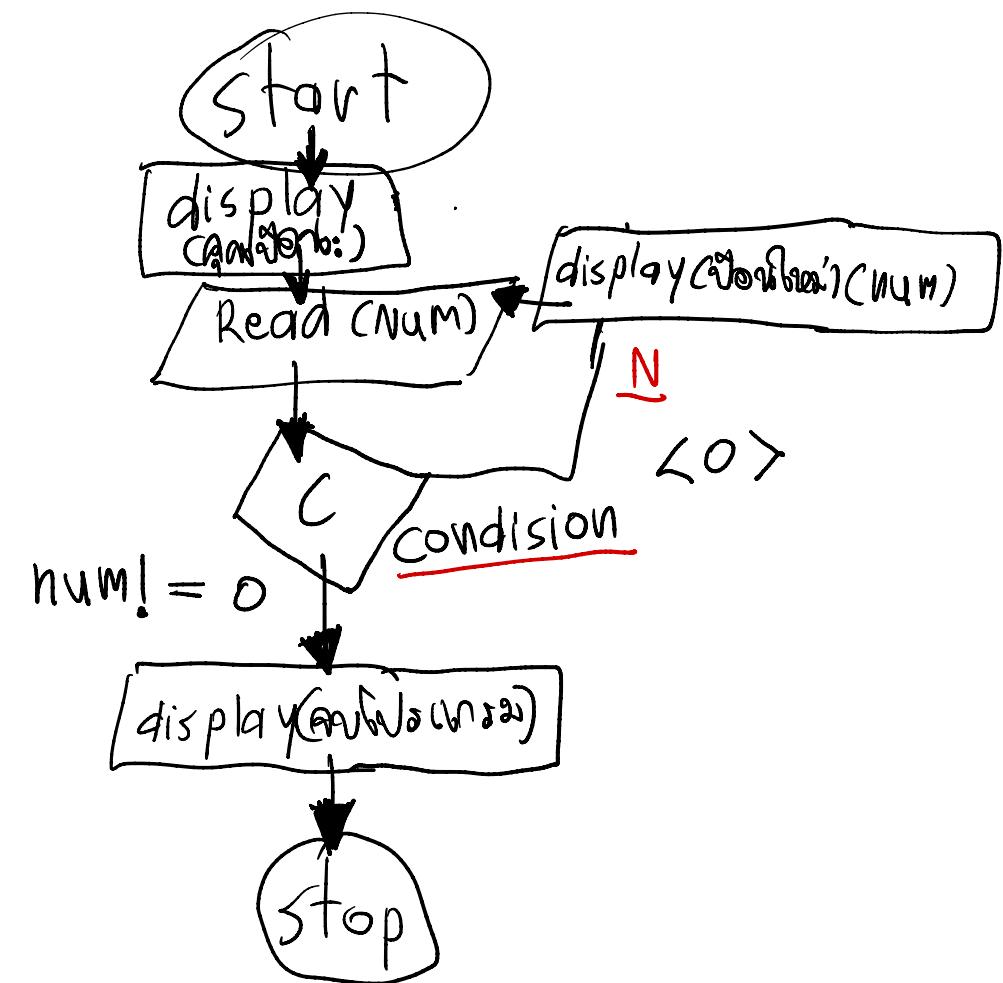
ผังงานที่ 8 ให้ นศ. เขียนผังงานของโปรแกรมดังต่อไปนี้

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

int main() {
    int num; DP1
    cout << "ป้อนตัวเลข (0 = จบ): ";
    cin >> num; Read

    while (num != 0) { DP2
        cout << "คุณป้อน: " << num << endl;
        cout << "ป้อนใหม่ (0 = จบ): "; DP3
        cin >> num; Read
    }

    cout << "จบโปรแกรม" << endl; DP4
    return 0;
}
```



✿ ใช้สาหร่ายว่า loop จะวนต่อ จนกว่าจะเจอ 0 → นักศึกษาเข้าใจ “เงื่อนไขหยุด”

