

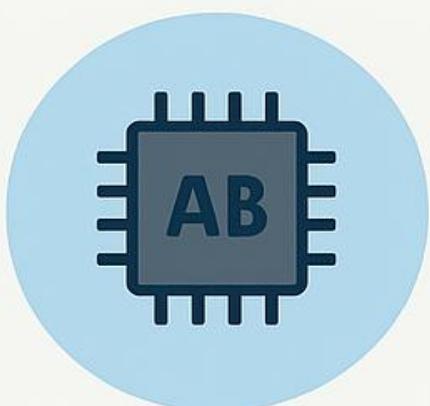
# COMPUTER PROGRAMMING

## C++



### INPUT

Receive data from the user



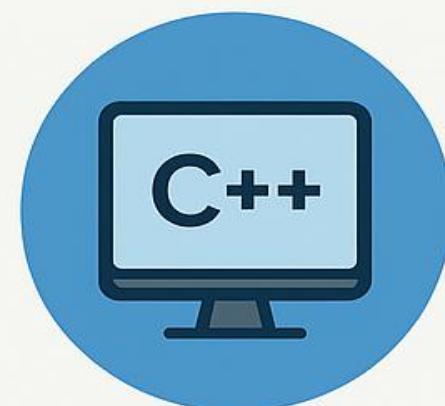
### VARIABLES

Store and manage data



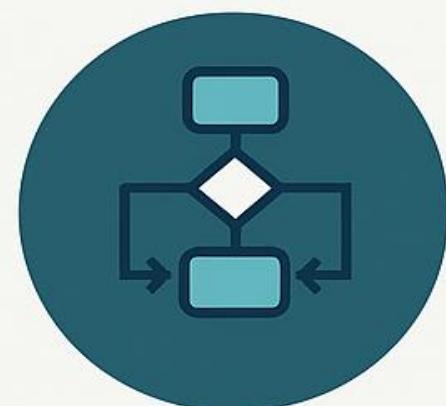
### PROCESS

Perform operations on data



### OUTPUT

Produce results for the user



### ALGORITHM

Define steps to solve a problem

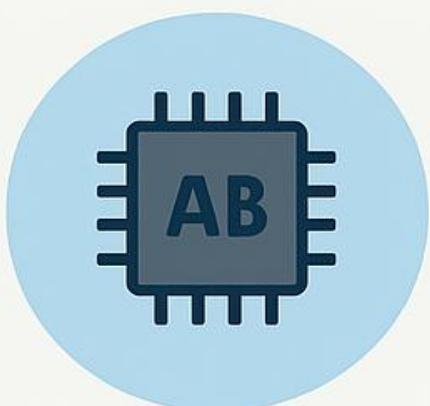
# COMPUTER PROGRAMMING

## C++



### INPUT

Receive data from the user



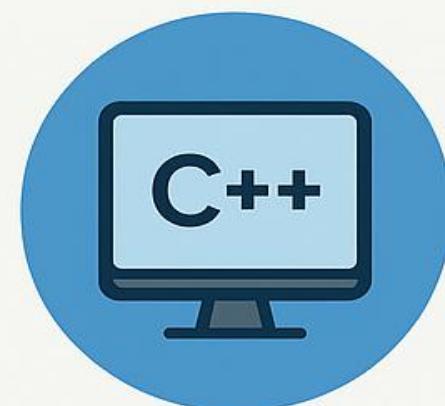
### VARIABLES

Store and manage data



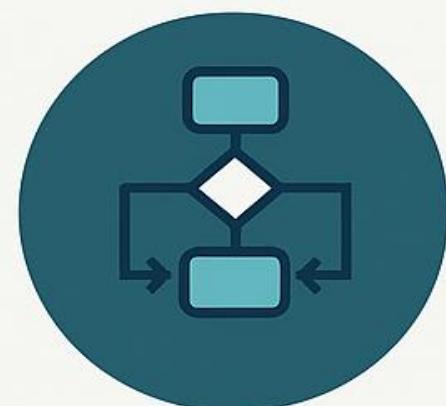
### PROCESS

Perform operations on data



### OUTPUT

Produce results for the user

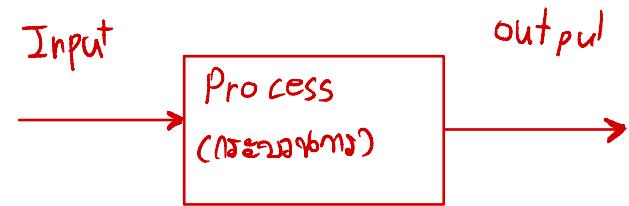


### ALGORITHM

Define steps to solve a problem

# โปรแกรมที่ 1 ให้ นศ. เขียนโปรแกรม Input–Process–Output ดังต่อไปนี้

รับชื่อ → รับอายุ → แสดงข้อความ



Block Diagram  
โครงสร้างของ Input, output และ process

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

```
int main()
```

จุดเด่น  
จุดเด่น  
จุดเด่น

```
string name;  
int age;
```

ส่วนของภาษา C++ ที่ใช้คือ string  
กำหนดชื่อตัวแปร name  
เป็นตัว type string

```
cout << "Enter your name: ";
```

cin >> name; คำสั่งรับตัวแปร (name) / cin คือค่าสั่งรับค่า คำสั่ง Input ทางแป้นพิมพ์

```
cout << "Enter your age: ";
```

cin >> age; คำสั่งที่ 2  
ส่วนของตัวแปร 2 (age)

```
cout << "Hello " << name << ", you are " << age << " years old." << endl;
```

return 0; คำสั่ง Cout طبعค่าที่รับเข้ามานี้ในลักษณะภาษาไทย หรือภาษาไทย (Display)

## ◆ คำถ้าที่ออกแบบให้นักศึกษาอธิบาย

1. เมื่อโปรแกรมเริ่มต้นทำงาน เกิดอะไรขึ้นบ้าง ก่อนที่จะเจอ main() ?

(เขียนโดยกับการ include ไลบรารี `#include <iostream>` และการใช้ namespace std)

2. การประกาศตัวแปร

- string name; int age; หมายถึงอะไร?

- แต่ละตัวแปรเก็บข้อมูลชนิดใด และทำไว้ในต้องเลือกใช้ชนิดนั้น?

สร้างพื้นที่เก็บชื่อ Name (ตัว)  
เก็บอายุ

สร้างพื้นที่เก็บชื่อ Age (ตัว)  
เก็บอายุไว้

3. การทำงานของ `cin`

- บรรทัด `cin >> name;` ทำงานอย่างไร? คำสำคัญ Input มาเก็บไปที่哪儿

- ข้อมูลที่ผู้ใช้พิมพ์เข้าไปถูกเก็บไว้ที่ไหน ก่อนส่งไปยังตัวแปร?

4. การทำงานของ `cout`

- `cout << "Hello" << name << ...` ทำงานทีล่ะขั้นตอนอย่างไร?

- ทำไมต้องใช้ `<<` หลายครั้งเชื่อมกันได?

คำสำคัญ cout คือการพิมพ์ผลลัพธ์  
`<< "Hello"` `<< name`

5. ลำดับ Input–Process–Output

- ให้นักศึกษาเขียนอธิบายเป็น Step 1 → Step 2 → Step 3 ตั้งแต่รับค่า → เก็บค่าในหน่วยความจำ → แสดงผล

อกมา 1. รับค่าจากผู้ใช้ `Cin`

2. Process หัวฟัง

3. แสดงผลป้องกัน

ให้คนรู้สึก

int  
↓  
integer

## 6. ทดลองค่าด้วยผลลัพธ์

- ถ้าผู้ใช้พิมพ์ Alice และ 20 จะได้ output ว่าอย่างไร?
- ถ้าผู้ใช้พิมพ์ชื่อที่มีช่องว่าง เช่น Somchai Phuengsungwan จะเกิดอะไรขึ้น? (ฝึกคิดว่าทำใน cin รับแค่คำแรก)

## 7. บทบาทของ return 0;

- ทำไมต้องใส่?
- ถ้าไม่ใส่จะเกิดผลต่างอย่างไร?

จะ return 0; เป็นเหมือนปั๊บปั๊บ กด 0 รองรับ  
กรณีที่ต้องการ แต่ ลืมป้อนชื่อ ไม่สามารถรัน

## ◆ เฉลยคำถาม

### 1. เมื่อโปรแกรมเริ่มต้นทำงาน ก่อนถึง main()

- `#include <iostream>` → คือการบอกคอมไไฟเลอร์ให้นำเอาฟังก์ชันมาตรฐานที่ใช้ input/output (เช่น `cin`, `cout`) มาใช้ได้
- `using namespace std;` → ทำให้เราไม่ต้องเขียน `std::cout` หรือ `std::cin` ทุกครั้ง
- เมื่อคอมไไฟล์แล้ว โค้ดตรงนี้จะถูกแทนที่ด้วย “คำสั่งยาว ๆ” จากไลบรารี C++ standard library

### 2. การประกาศตัวแปร

**ชนิดตัวแปร ชื่อตัวแปร**

- `string name;` → ประกาศตัวแปรชื่อ `name` มีชนิดเป็น `string` (ข้อความ ตัวอักษรต่อเนื่อง)
- `int age;` → ประกาศตัวแปรชื่อ `age` มีชนิดเป็น `integer` (จำนวนเต็ม)

👉 เหตุผลที่เลือกใช้:

- `string` เหมาะกับชื่อ เพราะประกอบด้วยอักษรหลายตัว
- `int` เหมาะกับอายุ เพราะอายุไม่เป็นทศนิยม

ចុច្ចាល់ថ្មីទេនា វ៉ា កាមុជ្រក្ប និងឱឡូរាជីវិកាងាល់ប្រឈម  
ឱចិត្តលើកំណែ

សារិនិទ្ទេនឹងទៅលើ នៅថ្ងៃ ១០ ខែ

នូវការឱ្យក្រុងការបង្កើតការងារ និងការអនុវត្តន៍ និងការអនុវត្តន៍



### 3. การทำงานของ cin

- บรรทัด `cin >> name;` → รอให้ผู้ใช้พิมพ์ข้อมูลใน console และกด Enter
- ข้อมูลที่พิมพ์เข้าไป จะถูกเก็บไว้ที่ input buffer (หน่วยความจำชั่วคราวของโปรแกรม)
- จากนั้นจะถูกส่งไปเก็บในตัวแปร `name`

ถ้าหากเราพิมพ์ชื่อ "John" ลงใน input buffer แล้วเราเรียกใช้ `cout << name;` คำสั่งนี้จะนำชื่อ "John" ที่อยู่ใน input buffer ไปแสดงบนหน้าจอ

ข้อมูล

### 4. การทำงานของ cout

ค่าที่ได้รับ

- บรรทัด `cout << "Hello " << name << ", you are " << age << " years old.;"`
  - Step 1: ส่งข้อความ `"Hello "` ไปยัง output stream
  - Step 2: ต่อด้วยค่าที่อยู่ในตัวแปร `name`
  - Step 3: ต่อด้วยข้อความ `", you are "`
  - Step 4: ต่อด้วยค่าของ `age`
  - Step 5: ต่อด้วย `" years old."` และขึ้นบรรทัดใหม่ด้วย `endl`

👉 ทำไมใช้ `<<` ได้หลายครั้ง?

เพราะ `<<` ใน C++ ถูก Overload ให้เชื่อมต่อข้อมูลได้หลาย ๆ ส่วนในบรรทัดเดียว

## 5. ลำดับ Input–Process–Output

- Step 1: Input → `cin >> name; cin >> age;` รับค่าจากผู้ใช้
- Step 2: Process → เก็บค่าที่รับมาไว้ในหน่วยความจำ (ตัวแปร `name` และ `age`)
- Step 3: Output → ใช้ `cout` แสดงผลข้อความ + ค่าที่เก็บในตัวแปรออกทางหน้าจอ

ໃນໂປຣແກຣມໂຄສະນຸ້າໄຟ ຄົວຂອງພວກເຮົາ

## 6. ทดลองคาดการณ์ผลลัพธ์

กรณี 1: ผู้ใช้พิมพ์

nginx

Alice      **name**  
20            **age**

Output:

sql

Hello Alice, you are 20 years old.

ຈົດການ

ຕົວແລ້ວ

## กรณี 2: ผู้ใช้พิมพ์

```
nginx
```

```
Somchai Phuengsungwan
```

```
20
```

Output:

```
sql
```

```
Hello Somchai, you are 20 years old.
```

!  
! สังเกตว่า `cin` จะอ่านชื่อแค่ คำแรกก่อนช่องว่าง (`Somchai`) → เพราะ `cin` ใช้ช่องว่าง/enter/tab เป็นตัวแบ่ง  
ข้อมูล

## 7. บทบาทของ `return 0;`

- บอกว่าโปรแกรมทำงานเสร็จสมบูรณ์และ คืนค่ากลับไปยังระบบปฏิบัติการ (OS)
- `0` หมายถึงสำเร็จ, ส่วนค่าที่ไม่ใช่ 0 หมายถึง error

👉 ถ้าไม่ใส่ `return 0;`

- ใน C++ เวอร์ชันใหม่ ๆ คอมไพล์เลอร์จะใส่ให้เองโดยอัตโนมัติ (ถือว่า main คืนค่า 0)
- แต่เพื่อมาตรฐานและความชัดเจน → **ควรใส่เสมอ**

## โปรแกรมที่ 2 ให้ นศ. เขียนโปรแกรม Input–Process–Output ดังต่อไปนี้

คง แuren ที่ 2 ทำ g++ นำเข้าผลรวมของตัวเลข 3 จำนวนเต็ม 2 จิต 1 จิต

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int x = 5;
```

```
    int y = 2;
```

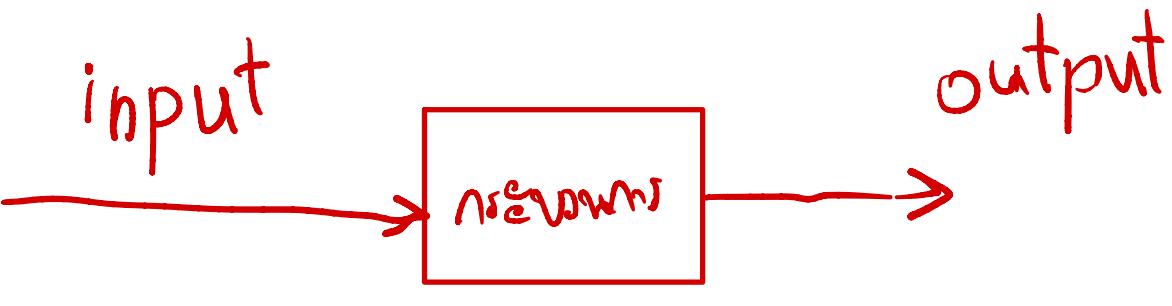
```
    int z = x + y;
```

```
    cout << "Result = " << z << endl;
```

```
    return 0;
```

```
}
```

End



โปรแกรมที่ 2  
เขียนโดย นศ.  
ผลรวมของตัวเลข  
3 จำนวนเต็ม 2 จิต 1 จิต

คำถามที่ให้นักศึกษาอธิบาย

1. เมื่อโปรแกรมเริ่มทำงาน ตัวแปรแต่ละตัว ( $x, y, z$ )

เก็บค่าอะไร?

2. บรรทัด `int z = x + y;` ทำงานอย่างไร? เกิดการ

คำนวณในหน่วยประมวลผล (CPU) แบบไหน?

3. คำสั่ง `cout` ทำหน้าที่อะไร/อย่างไร

## โปรแกรมที่ 3 ให้ นศ. เขียนโปรแกรม Input–Process–Output ดังต่อไปนี้

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

int main() {
    float celsius, fahrenheit; In      out
    cout << "Enter temperature in Celsius: ";
    cin >> celsius; รับค่าอุณหภูมิจากผู้ใช้งานเป็นหน่วย Celsius แล้ว Input
    fahrenheit = (celsius * 9 / 5) + 32;
    cout << "Temperature in Fahrenheit = " << fahrenheit << endl;
    return 0;
}
```

กฤษฎีกา √ รุ่น Input, output

คำถามที่ให้นักศึกษาอธิบาย

1. เมื่อผู้ใช้พิมพ์ค่า `cin >> celsius;` โปรแกรมนำ

ข้อมูลจากไหน?

2. บรรทัด `fahrenheit = (celsius * 9 / 5) + 32;`

เกิดกระบวนการใด?

3. ถ้าใส่ค่า 0 องศาเซลเซียส จะได้ผลลัพธ์เท่าไร?

## ໂປຣແກຣມທີ 4 ໃຫ້ ນສ. ເຂີຍນໂປຣແກຣມເຊີງຕຽບຮະ ດັ່ງຕໍ່ໄປນີ້

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

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

$\geq 50 \rightarrow 1$

$< 50 \rightarrow 0$

คำถามที่ให้นักศึกษาอธิบาย

1. ทำไมโปรแกรมใช้ ( $score \geq 50$ ) ถึงแสดงผลเป็น 1

หรือ 0?

2. ในเชิงตรรกะ ตัวเลข 1 และ 0 สื่อความหมายอะไร?

3. ถ้าต้องการให้แสดงว่า Pass / Fail แทน 1 / 0 ต้อง<sup>ชี้</sup>  
แก้โค้ดอย่างไร?

## โปรแกรมที่ 5 ให้ นศ. เขียนโปรแกรมให้เป็นไปตาม ผลการรัน

ดังต่อไปนี้ โดยกำหนดเงื่อนไขของโปรแกรมดังนี้

In 2 = kg cm  
Out 4

### ▶ โปรแกรมคำนวณค่า BMI และประเมินผลสุขภาพ

โจทย์:

เขียนโปรแกรมรับค่า น้ำหนัก (kg) และ ส่วนสูง (cm) จากผู้ใช้ และคำนวณค่า BMI ด้วยสูตร

$$BMI = \frac{\text{น้ำหนัก}}{(\text{ส่วนสูง}/100)^2}$$

จากนั้นแสดงผลลัพธ์พร้อม คำอธิบาย เช่น

- น้อยกว่า 18.5 → “น้ำหนักน้อยเกินไป”
- 18.5 – 22.9 → “น้ำหนักปกติ”
- 23.0 – 24.9 → “น้ำหนักเกิน”
- 25.0 ขึ้นไป → “อ้วน”

ตัวอย่างผลการรันโปรแกรม

Enter weight (kg): 60

Enter height (cm): 165

Your BMI = 22.04

Status: Normal





## ເລືອຢໂປຣແກຣມທີ 5

```
#include <iostream>
using namespace std;
int main() {
    float weight, height, bmi;
    cout << "Enter weight (kg): ";
    cin >> weight;
    cout << "Enter height (cm): ";
    cin >> height;
    // ດຳນວນ BMI
    bmi = weight / ((height / 100) * (height / 100));
    cout << "Your BMI = " << bmi << endl;
    // ປະເມີນຜລສຸຂາພ
    if (bmi < 18.5) {
        cout << "Status: Underweight" << endl;
    } else if (bmi < 23) {
        cout << "Status: Normal" << endl;
    } else if (bmi < 25) {
        cout << "Status: Overweight" << endl;
    } else {
        cout << "Status: Obese" << endl;
    }
    return 0;
}
```

Enter weight (kg): 60

Enter height (cm): 165

Your BMI = 22.0386

Status: Normal

Enter weight (kg): 45

Enter height (cm): 170

Your BMI = 15.5709

Status: Underweight

## ◆ เฉลยการทำงานทีละส่วน

### 1. ประกาศตัวแปร

cpp

```
float weight, height, bmi;
```

- `weight` = น้ำหนัก (kg)
- `height` = ส่วนสูง (cm)
- `bmi` = ค่า BMI ที่คำนวณได้

### 2. รับค่า Input จากผู้ใช้

cpp

```
cin >> weight;  
cin >> height;
```

→ ผู้ใช้กรอกข้อมูล น้ำหนัก และ ส่วนสูง

### 3. คำนวณค่า BMI

cpp

```
bmi = weight / ((height / 100) * (height / 100));
```

- แปลงส่วนสูงจาก cm → m (height/100)
- ยกกำลังสอง → (height/100) \* (height/100)
- หารน้ำหนักด้วยค่าที่ได้ → ได้ค่า BMI

### 4. แสดงผลลัพธ์ BMI

cpp

```
cout << "Your BMI = " << bmi << endl;
```

### 5. ใช้ if-else ประเมินผลสุขภาพ

cpp

```
if (bmi < 18.5) → Underweight  
else if (bmi < 23) → Normal  
else if (bmi < 25) → Overweight  
else → Obese
```

### 6. สิ้นสุดโปรแกรม

cpp

```
return 0;
```

- ◆ วิธีสรุปหลังจบโปรแกรมที่ 5

## 1. เชื่อมโยงหลักการ Input–Process–Output

- โปรแกรม 1–2 → Input/Output พื้นฐาน (`cin` / `cout`)
- โปรแกรม 3–4 → คำนวนเชิงสูตร (เช่น ค่าเฉลี่ย, การแปลงหน่วย)
- โปรแกรม 5 → คำนวน + การตัดสินใจ (if-else)

👉 ทำให้นักศึกษารู้ว่า “ทุกโปรแกรมคือกระบวนการเดียวกัน”

## 2. หบทวน Data Types และ Variables

- `int` → จำนวนเต็ม เช่น อายุ
  - `float/double` → ทศนิยม เช่น BMI
  - `string` → ข้อความ เช่น ชื่อ
  - `bool` → ค่าจริง/เท็จ (เตรียมต่อยอดไปเงื่อนไข)
- 

## 3. Operators ที่ใช้ไปแล้ว

- คณิตศาสตร์: `+ - * / %`
- เปรียบเทียบ: `< > == != <= >=`
- ตรรกะ: `&& || !` (แนะนำฯ ได้เลยตอนปิดท้าย)

## 4. บทเรียนสำคัญจากโปรแกรม 5 (BMI)

- เริ่มมี การตัดสินใจด้วยเงื่อนไข → ทำให้โปรแกรมตอบสนองตามสถานการณ์จริง
  - บุพ्पื้นไปสู่บทเรียนต่อไป: การใช้ if-else หลายเงื่อนไข และ loop
- 

## 5. สรุปให้เห็นการต่อ�อด

“ที่ผ่านมาเราทำโปรแกรมที่ รับค่า → คำนวณ → แสดงผล แต่ยังทำงานแบบ ตรงไปตรงมา ‘ไม่มีการเลือกเงื่อนไข หรือวนซ้ำ’

สุดท้ายต่อไป เราจะทำให้โปรแกรม ‘ฉลาดขึ้น’ โดยให้มัน ตัดสินใจ และ ทำงานช้า ๆ ได้ โดยไม่ต้องเขียนโค้ดช้า เอง”

## โปรแกรมที่ 6 ให้ นศ. อธิบายการทำงานของโปรแกรมดังต่อไปนี้ จากนั้นเขียนโปรแกรมด้วย DEV C++

```
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    int minutes;
    cout << "Enter minutes: ";
    cin >> minutes;
    int h = minutes / 60;
    int m = minutes % 60;
    cout << "Time = " << h << ":" << setw(2) << setfill('0') << m << endl;
    return 0;
}
```





## โปรแกรมที่ 7 ให้ นศ. อธิบายการทำงานของโปรแกรมดังต่อไปนี้ จากนั้นเขียนโปรแกรมด้วย DEV C++

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





## โปรแกรมที่ 8 ให้ นศ. อธิบายการทำงานของโปรแกรมดังต่อไปนี้ จากนั้นเขียนโปรแกรมด้วย DEV C++

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



