

```
#include <iostream>
```

#include <iostream>

- เป็นการ include library iostream ซึ่งให้ส่วนรับการรับข้อมูล (`cin`) และแสดงผล (`cout`) ทางหน้าจอ
- ถ้าไม่ใส่บรรทัดนี้ จะไม่สามารถใช้ `cin` และ `cout` ได้

```
using namespace std;
```

- บอกคอมไไฟเลอร์ว่าให้ใช้ namespace std โดยตรง
- ทำให้เราไม่ต้องเขียน `std::cout` หรือ `std::cin` ทุกครั้ง สามารถเขียนล้วน ๆ เป็น `cout` และ `cin` ได้เลย

```
int main() {
```

- จุดเริ่มต้นของโปรแกรม C++ ทุกโปรแกรม
- ฟังก์ชัน `main()` คือที่คอมไไฟเลอร์จะเริ่มทำงาน

I. โปรแกรมพิมพ์ตัวอักษรเป็นรูปทรง (Flag)

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

int main() {
    int row, i, j;
    char sign;

    cin >> row;
    cin >> sign;

    for (i = row; i >= 1; i--) {
        for (j = 1; j <= i; j++) {
            cout << sign;
        }
        cout << "\n";
    }

    return 0;
}
```

*ลูป = Loop *

```
for (i = row; i >= 1; i--) {
    for (j = 1; j <= i; j++) {
        cout << sign;
    }
    cout << "\n";
}

return 0;
```

return 0; = ปิดงานแบบปกติ

- ตรงนี้คือคำสั่งบอกระบบว่า “โอเคจ้า โปรแกรมทำงานเสร็จแล้ว ไม่มี error อะ”

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

```
int main() {
    int row, i, j;
    char sign;

    cin >> row;
    cin >> sign;
```

ทำไมต้อง int?
 เพราะพวknี้เป็น “ตัวเลขันบอรอบลูป”
 อ่านแล้ว

- ใช้จำนวนเต็มคือเวิร์กสุด

สรุปแบบให้ง่ายสุดในจักรวาล

int = เก็บตัวเลขเทื่องบันลุป

char = เก็บตัวอักษร 1 ตัวไว้พิมพ์เป็นรูป

cin = รับค่าจาก keyboard เป้าหมายใส่ตัวแปร

ทั้งหมดนี้ถือ “การศัษย์มอง” ก่อนจะไปاردูรูปสามเหลี่ยม

ทำไมต้องใช้ < iostream >

- เพราะโปรแกรมนี้ต้องใช้ cin และ cout
- เอาไว้รับค่า row, sign จากผู้ใช้ แล้วพิมพ์ออกมาเป็นรูปดาว/สัญลักษณ์
- ดำเนินเมื่ ostream ตัวอื่น—โปรแกรมพูดกับ user ไม่ได้

ตัวนี้คือการประกาศ ตัวแปรชนิดจำนวนเต็ม (int) ทั้ง 3 ตัวพร้อมกันเพื่อ
 เอามาใช้แบบนี้:

- row = จำนวนแวกว่าที่ผู้ใช้ป้อนเข้ามา
- i = ตัวบันลุปนอก
- j = ตัวบันลุปใน

char sign;

- ประกาศตัวแปรชนิด ตัวอักษรตัวเดียว (char)
- ไว้เก็บสัญลักษณ์ที่ user จะให้พิมพ์ เช่น * หรือ # หรือ \$
- ดำเนิน string จะเก็บความจำเป็นไปหน่อย
- เพราะโจทย์พิมพ์ที่ลักษณะตัวอยู่แล้ว
- ใช้ char คือเบาสุด สะดวกสุด

cin >> row;

รับค่า “จำนวนแวก” กี่ผู้ใช้จะพิมพ์ เช่น ถ้าพิมพ์ 5 ก็เอาไปใช้ในลูป i

cin >> sign;

รับสัญลักษณ์ 1 ตัว เพื่อบอกว่าในแต่ละแวกจะพิมพ์ตัวอะไร

ลูปนอก (for i = row; i >= 1; i--)

- อันนี้คือวนคุณจำนวนแวกเริ่มจากค่ามากสุด แล้วลดลงเรื่อยๆ

ถ้า row = 5

ลูปอกจะวิ่ง i = 5, 4, 3, 2, 1

เหตุผลที่ i--

- เพราะต้องการ “ลดจำนวนสัญลักษณ์ลงเรื่อยๆ” ให้ได้สามเหลี่ยม
 กลับหัวแบบนี้:



ลูปใน (for j = 1; j <= i; j++)

- ลูปนี้เป็นคนพิมพ์สัญลักษณ์นินแต่ละแวกจำนวนกี่พิมพ์เพื่อกัน
 ค่า i ของลูปนอกตอนนั้น

เช่น ถ้า i = 4

ก็พิมพ์ 4 ตัวแบบนี้

ทำไมต้องเริ่ม j = 1?

- เพราะบันคือ “ตัวบันทึกจำนวนสัญลักษณ์”
 บันตังแต่ตัวแรกจนถึงตัวที่ i

cout << sign;

บีก้อค่าสั่งยังตัวอักษรออก
 มา
 เช่น sign = *\n ก็จะพิมพ์ตามจำนวนรอบ
 ของลูปใน

cout << "\n";

หลังพิมพ์ครบที่แล้วแวก ต้องขึ้นบรรทัดใหม่
 ไม่ขึ้นตูกว่าช่วงเวลาที่กันเป็นช่วง ๆ แบบงั้น
 เช่นไม่ใช่
 จะคือเป็น ***** ต้องกันหมด ไม่
 เป็นรูป

2. โปรแกรมหาค่าผลรวมและค่าที่มากที่สุด (summax)

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

int main() {
    int n, x, i, sum, max;

    cin >> n;
    cin >> x;

    sum = x;
    max = x;

    for (i = 2; i <= n; i++) {
        cin >> x;
        sum += x;

        if (x > max) {
            max = x;
        }
    }

    cout << sum << endl;
    cout << max << endl;

    return 0;
}
```

เรียกใช้ไลบรารีสำหรับรับ/แสดงผลข้อมูล เช่น `cin`, `cout`

ใช้ให้ใน `cin`, `cout` ได้โดยไม่ต้องพิมพ์ `std::cin`, `std::cout`

```
int main() {
    // จุดเริ่มต้นของโปรแกรม
```

ประกาศตัวแปร:

- `n`: จำนวนตัวเลขทั้งหมดที่ผู้ใช้จะป้อน
- `x`: ตัวเลขแต่ละตัวที่รับเข้ามา
- `i`: ตัวนับใน loop
- `sum`: ผลรวมของตัวเลขทั้งหมด
- `max`: ค่ามากที่สุดในชุดข้อมูล

```
cin >> n;
cin >> x;
```

รับค่าจากผู้ใช้:

- จำนวนตัวเลขทั้งหมด `n`
- ตัวเลขตัวแรก `x`

```
sum = x;
max = x;
```

กำหนดค่าเริ่มต้นให้ `sum` และ `max` เป็นค่าของตัวเลขตัวแรก

```
for (i = 2; i <= n; i++) {
```

เริ่ม loop ตั้งแต่ตัวที่ 2 ถึงตัวที่ `n`

```
    cin >> x;
    sum += x;
```

รับค่าตัวเลขใหม่ และบวกเพิ่มเข้าไปใน `sum`

```
    if (x > max) {
        max = x;
    }
```

ถ้าตัวเลขใหม่มากกว่าค่า `max` เดิม → ปรับ `max` เป็นค่านั้น

```
}
```

จบ loop

```
cout << sum << endl;
cout << max << endl;
```

แสดงผลรวมทั้งหมด และค่ามากที่สุด

```
return 0;
}
```

จบโปรแกรม

3. โปรแกรมพิมพ์ตัวเลขคูนย์-กันง (zeroone)

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

int main() {
    int n, i;
    cin >> n;

    for (i = 1; i <= n; i++) {
        if (i % 2 == 1) {
            cout << 0;
        } else {
            cout << 1;
        }
    }

    return 0;
}
```

```
#include <iostream>
```

- ใช้สำหรับเรียกฟังก์ชัน input/output เช่น `cin` และ `cout`

```
using namespace std;
```

- หากให้เราใช้ `cin`, `cout` ได้โดยไม่ต้องพิมพ์ `std::cin`, `std::cout`

```
int main()
```

- จุดเริ่มต้นของโปรแกรม

```
int n, i, k;
```

- ประกาศตัวแปร:

- `n`: จำนวนรอบที่ต้องการพิมพ์
- `i`: ตัวนับใน loop
- `k`: ไม่ได้ใช้ในโค๊ดนี้ (สามารถลบออกได้)

```
cin >> n;
```

รับค่าจำนวนรอบจากผู้ใช้

```
for (i = 1; i <= n; i++) {
```

วนลูปตั้งแต่ `i = 1` ถึง `i = n`

```
    if (i % 2 == 1) {
        cout << 0;
    } else {
        cout << 1;
    }
```

- ถ้า `i` เป็นเลขคี่ → พิมพ์ `0`

- ถ้า `i` เป็นเลขคู่ → พิมพ์ `1`

```
}
```

- จบ loop

```
Cpp ^
```

```
    return 0;
}
```

- จบโปรแกรมและคืนค่ากลับไปยังระบบ