

```
#include <iostream>
```

#include <iostream>

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

```
using namespace std;
```

- บอกคอมไพเลอร์ว่าให้ใช้ **namespace std** โดยตรง
- ทำให้เราไม่ต้องเขียน `std::cout` หรือ `std::cin` ทุกครั้ง สามารถเขียนสั้น ๆ เป็น `cout` และ `cin` ได้เลย

```
int main() {
```

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

1. โปรแกรมพิมพ์ตัวอักษรเป็นรูปธง (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 "โอเคจ้ะ โปรแกรมทำงานเสร็จแล้ว ไม่มี error นะ"

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

int main() {
```

```
    int row, i, j;
    char sign;

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

ทำไมต้อง int?
เพราะพวกนี้เป็น "ตัวเลขนับรอบลูป"
อยู่แล้ว

- ใช้จำนวนเต็มคือเวิร์กสุด

สรุปแบบให้ง่ายสุดในจักรวาล

int = เก็บตัวเลขเพื่อนับลูป
char = เก็บตัวอักษร 1 ตัวไว้พิมพ์เป็นรูป
cin = รับค่าจาก keyboard เข้ามาใส่ตัวแปร

ทั้งหมดนี้คือ "การเตรียมของ" ก่อนจะไปวาดรูปสามเหลี่ยม

ทำไมถึงต้องใช้ < iostream >

- เพราะโปรแกรมนี้นี้ต้องใช้ cin และ cout
- เอาไว้รับค่า row, sign จากผู้ใช้ แล้วพิมพ์ออกมาเป็นรูปดาว/สัญลักษณ์
- ถ้าไม่มี iostream คือจบ—โปรแกรมพูดกับ user ไม่ได้

ตัวนี้คือการประกาศ ตัวแปรชนิดจำนวนเต็ม (int) ทั้ง 3 ตัวพร้อมกันเพื่อ
เอามาใช้แบบนี้:

- row = จำนวนแถวที่ผู้ใช้ป้อนเข้ามา
- i = ตัวนับลูปนอก
- j = ตัวนับลูปใน

char sign;

- ประกาศตัวแปรชนิด ตัวอักษรตัวเดียว (char)
- ไว้เก็บสัญลักษณ์ที่ user จะให้พิมพ์ เช่น * หรือ # หรือ \$
- ถ้าใช้ string จะเกิดความจำเป็นไปหน่อย
- เพราะโจทย์พิมพ์ทีละตัวอยู่แล้ว
- ใช้ char คือเบาสุด สะอาดสุด

cin >> row;

รับค่า "จำนวนแถว" ที่ผู้ใช้จะพิมพ์ เช่น ถ้าพิมพ์ 5 ก็เอาไปใช้ในลูป i

cin >> sign;

รับสัญลักษณ์ 1 ตัว เพื่อบอกว่าในแต่ละแถวจะพิมพ์ตัวอะไร

ลูปนอก (for i = row; i >= 1; i--)

- อันนี้คือคนคุมจำนวนแถวเริ่มจากค่ามากที่สุด แล้วลดลงเรื่อย ๆ

ถ้า row = 5

ลูปนอกจะวิ่ง i = 5, 4, 3, 2, 1

เหตุผลที่ i--

- เพราะต้องการ "ลดจำนวนสัญลักษณ์ลงเรื่อย ๆ" ให้ได้สามเหลี่ยม
กลับหัวแบบนี้:

```
*****
****
***
**
*
```

ลูปใน (for j = 1; j <= i; j++)

- ลูปนี้เป็นคนพิมพ์สัญลักษณ์ในแต่ละแถวจำนวนที่พิมพ์เท่ากับ
ค่า i ของลูปนอกตอนนั้น

เช่น ถ้า i = 4

ก็พิมพ์ 4 ตัวแบบนี้

```
****
```

ทำไมต้องเริ่ม j = 1?

- เพราะมันคือ "ตัวนับจำนวนสัญลักษณ์"
นับตั้งแต่ตัวแรกจนถึงตัวที่ i

cout << sign;

นี่คือคำสั่งยังตัวอักษรออก
มา
เช่น sign = *
ก็จะพิมพ์ตามจำนวนรอบ
ของลูปใน

cout << "\n";

หลังพิมพ์ครบหนึ่งแถว ต้องขึ้นบรรทัดใหม่
ไม่เช่นนั้นจะต่อกันเป็นยาว ๆ แบบงง ๆ
เช่น ไม่ใส่
จะกลายเป็น ***** ติดกันหมด ไม่
เป็นรูป

2. โปรแกรมหาผลรวมและค่าที่มากที่สุด (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;
}
```

```
#include <iostream>
```

เรียกใช้ไลบรารีสำหรับรับ/แสดงผลข้อมูล เช่น `cin`, `cout`

```
using namespace std;
```

ทำให้ใช้ `cin`, `cout` ได้โดยไม่ต้องพิมพ์ `std::cin`, `std::cout`

```
int main() {
```

จุดเริ่มต้นของโปรแกรม

```
int n, x, i, sum, max;
```

ประกาศตัวแปร:

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

```
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 แล้วบวกเพิ่มเข้าไปใน `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. โปรแกรมพิมพ์ตัวเลขศูนย์-หนึ่ง (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;
```

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

```
int main() {
```

- จุดเริ่มต้นของโปรแกรม

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

- ประกาศตัวแปร:
 - `n`: จำนวนรอบที่ต้องการพิมพ์
 - `i`: ตัวนับใน loop
 - `k`: ไม่ได้ใช้ในโค้ดนี้ (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