# 2. If, For, Array

Hengfeng Wei (魏恒峰)
hfwei@nju.edu.cn

Oct. 08, 2022

# Review

Variables (变量)　　Data Types (数据类型)

Operators (运算符)　　Expressions (表达式)

Assignment Statements (赋值语句)

I/O (Input/Output; 输入输出)

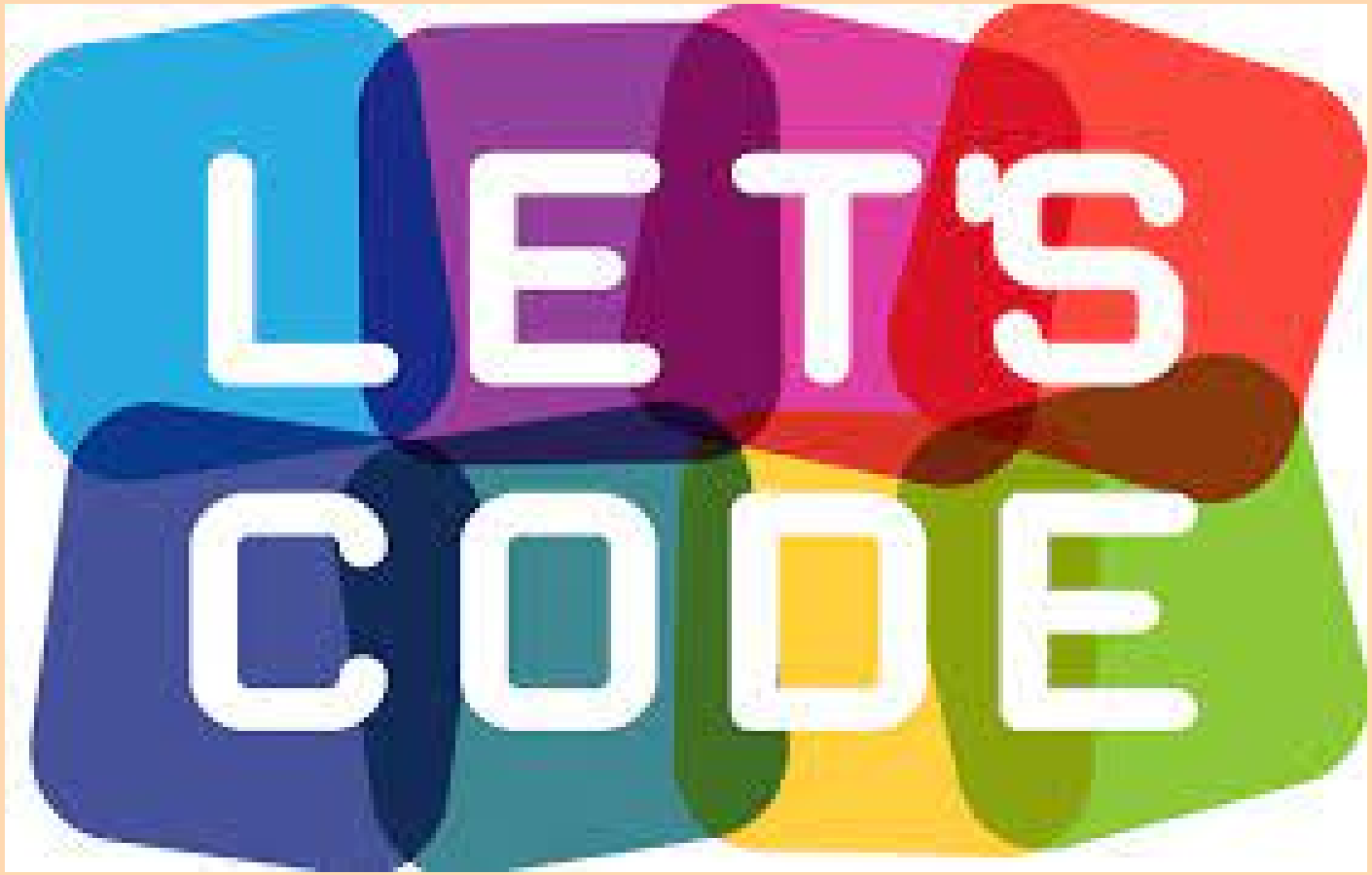
# Overview

If Statement ( `if` 语句)

For Statement ( `for` 语句)

Logical Expressions (逻辑表达式)

Array (数组)




**min.c leap.c min-array.c**

# Min

# Min of Two

**Given two integers $a$ and $b$, to compute their minimum.**

$$min = \min\{a, b\}$$

# Min of Two

```
min = a >= b ? b : a;
```

(条件表达式; 三目运算符)

## Do Not Use it Too Much!

## Do Not Be Too Clever!

# Min of Three

**Given three integers $a$, $b$, and $c$, to compute their minimum.**

$$min = \min\{a, b, c\}$$

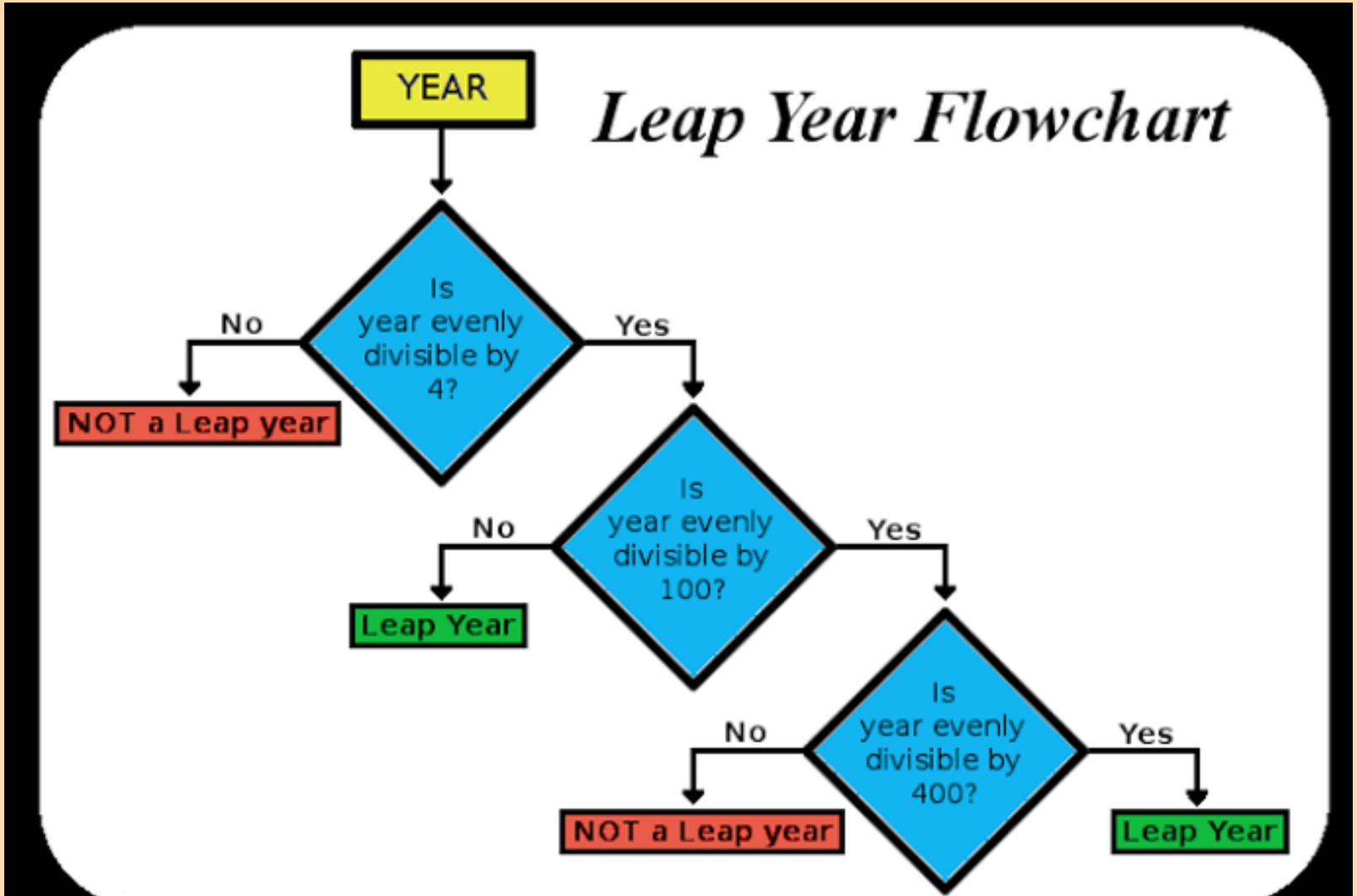
# Min of a Set of Numbers

**Given a set $A$ of integers, to compute their minimum.**

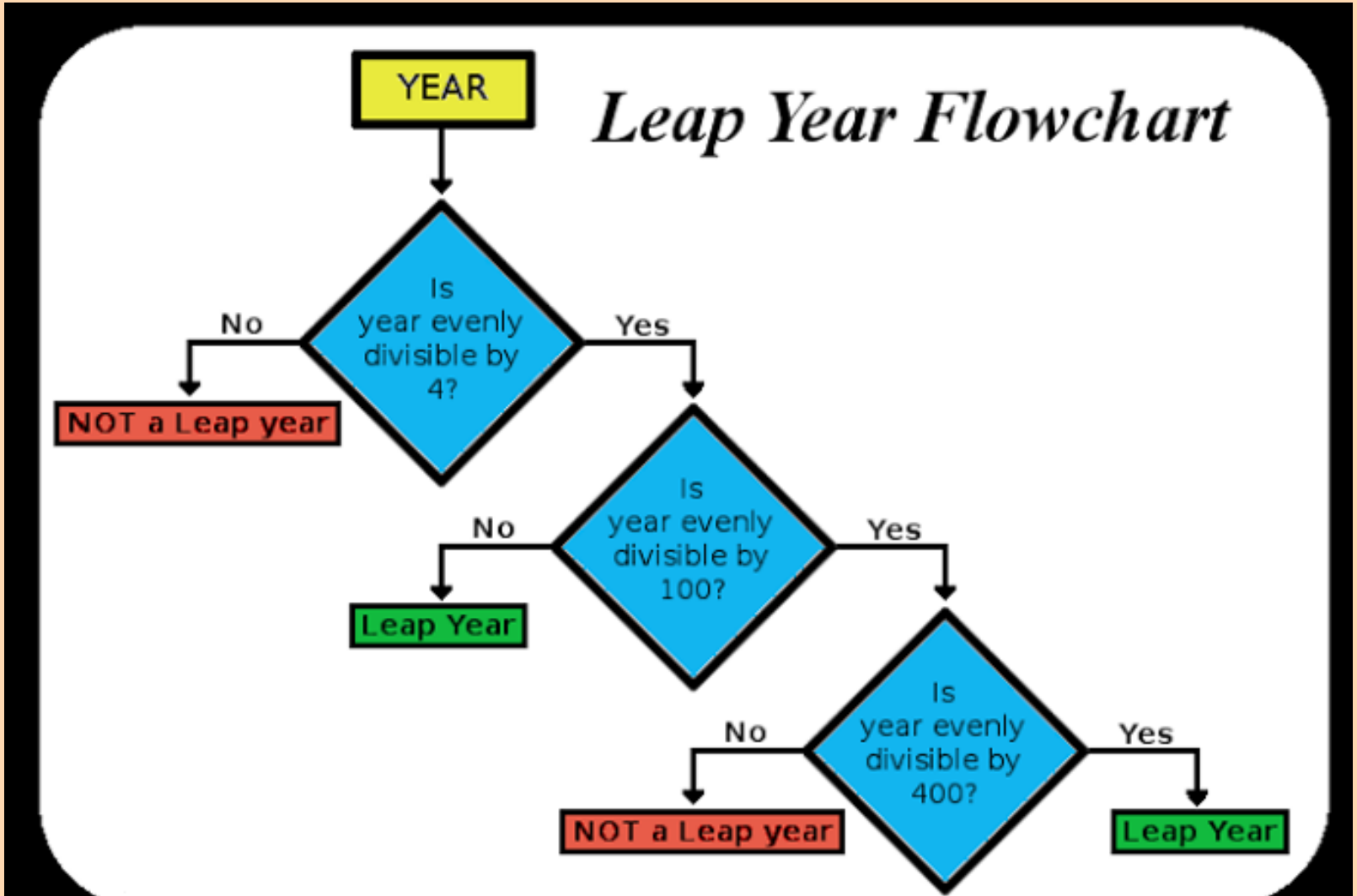
$$min = \min A$$

# Leap Year

# Leap Year (1): Nested `if/else` (YES)



# Leap Year (2): Nested `if/else` (NO)



# Leap Year (3): `else if`

**if** (*year* is not divisible by 4) **then** (it is a common year)

**else if** (*year* is not divisible by 100) **then** (it is a leap year)

**else if** (*year* is not divisible by 400) **then** (it is a common year)

**else** (it is a leap year)

# Leap Year (4): The Ultimate Version

**A year is a leap year if**

- it is divisible by $4$ but not by $100$,
- except that years divisible by $400$ are leap years.

# Short-circuit Evaluation (短路求值)

```
// test: year = 25
// test: year = 80
// test: year = 100
// test: year = 400
if ((year % 4 == 0 && year % 100 != 0) ||
    (year % 400 == 0)) {
  leap = 1;
}
```

# Min of a Set of Numbers

**Given a set $A$ of integers, to compute their minimum.**

$$min = \min_i A_i$$

$$\min\{3, 5, 2, 7\} = \min(\min(\min(3, 5), 2), 7)$$

# For Statement

```
for ( init-clause ; cond-expression ; iteration-expression ) loop-statement
```

```
for (initialization; condition; increment/decrement)
    statement
```

```
#define NUM 5
```

Symbolic Constants (符号常量)

`int numbers[NUM] = {0};` has a *fixed* size.

# Array Initializer

- `int numbers[NUM] = {1};`
- `int numbers[] = {0};`
- `int numbers[NUM] = {[2] = 1};`

# Array Initializer (DON'T)

```
int numbers[NUM] = {};
```

**Forbidden in C99 (Unfortunately!)**

**Allowed by GCC by default (Unfortunately!!)**

**Allowed in C23 (Fortunately or not???)**

# Array Initializer (DON'T)

```
int numbers[NUM];
```

`numbers` may contain garbage values;

always initialize it

# Array Initializer (DON'T)

```
int numbers[];
```

**You must specify the size so that the compiler/runtime can allocate memory for it.**

# Min of a Set of Input Numbers

**Input a set $A$ of $n \geq 1$ integers, to compute their minimum.**

$$min = \min_i A_i$$


SEE
YOU
NEXT
WEEK