

If, For, Array

王慧妍

why@nju.edu.cn

南京大学



计算机科学与技术系



计算机软件研究所



回顾

Variables Constants Data Types

Operators Expressions Assignment Statements

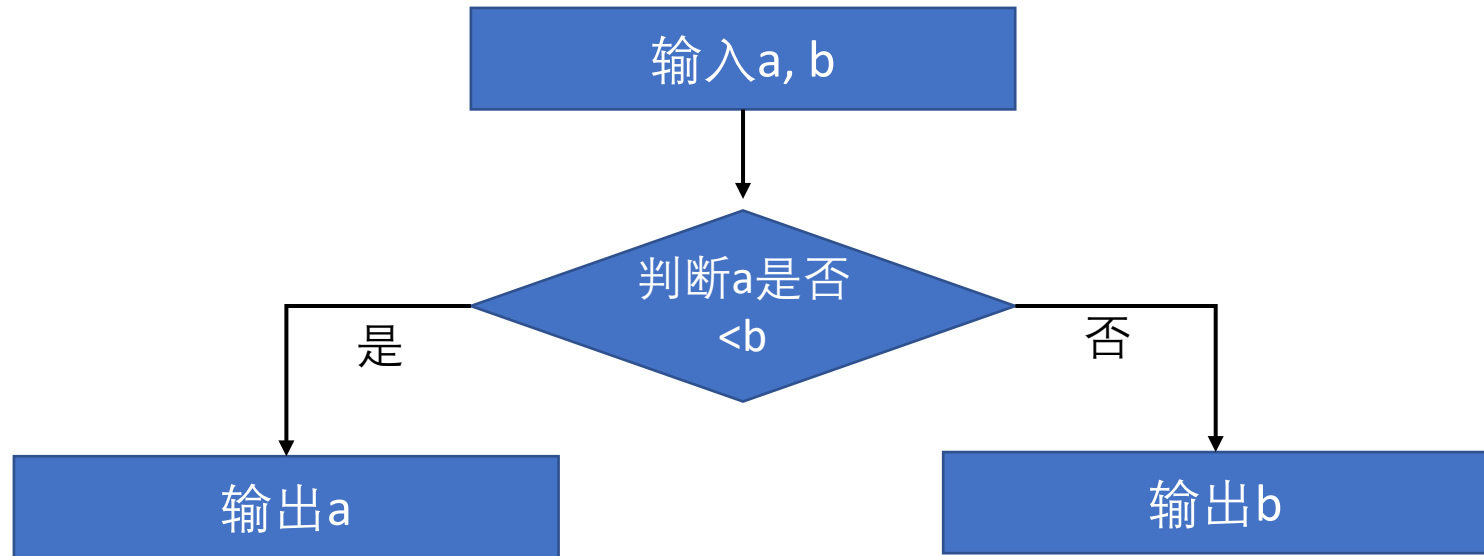
I/O (Input/Output)

本讲

- If
- For
- Array

MIN

- $\text{result} = \min\{a, b\}$



if条件语句

```
if(...) {  
    ...  
}
```

```
if(...) {  
    ...  
}  
else {  
    ...  
}
```

if条件语句

- else总是匹配前面最近的if
 - 无大括号隔开
- 缩进并不能暗示匹配情况
 - 新手Tips: 始终加上{}, 自我检查后的缩进style

```
if(a<b)//<=
|   if(a<10)
|       printf("min1: %d\n", a);
else
|   printf("min2: %d\n", b);
```

级联的if

```
if(...) {  
    ...  
}  
else{  
    if(){  
        ...  
    }  
    else(){  
        ...  
    }  
}
```



```
if(...) {  
    ...  
}  
else if(){  
    ...  
}  
else(){  
    ...  
}
```

MIN

- $\text{result} = \min\{a, b, c\}$
- min.c



逻辑表达式

- 关系运算符

- $<$, $>$, $<=$, $>=$

- (优先级低于算术运算符)

$i + j < j * k$

$(i + j) < (j * k)$

$i > j > k$

$(i > j) > k$

- 判等运算符

- $==$, $!=$

- (优先级低于关系运算符)

$i < j == j < k$

$(i < j) == (j < k)$

- 逻辑运算符

- $\&\&$ (与), $\|\|$ (或), $!$ (非)

- $(i != 0) \&\& (j / i > 0)$

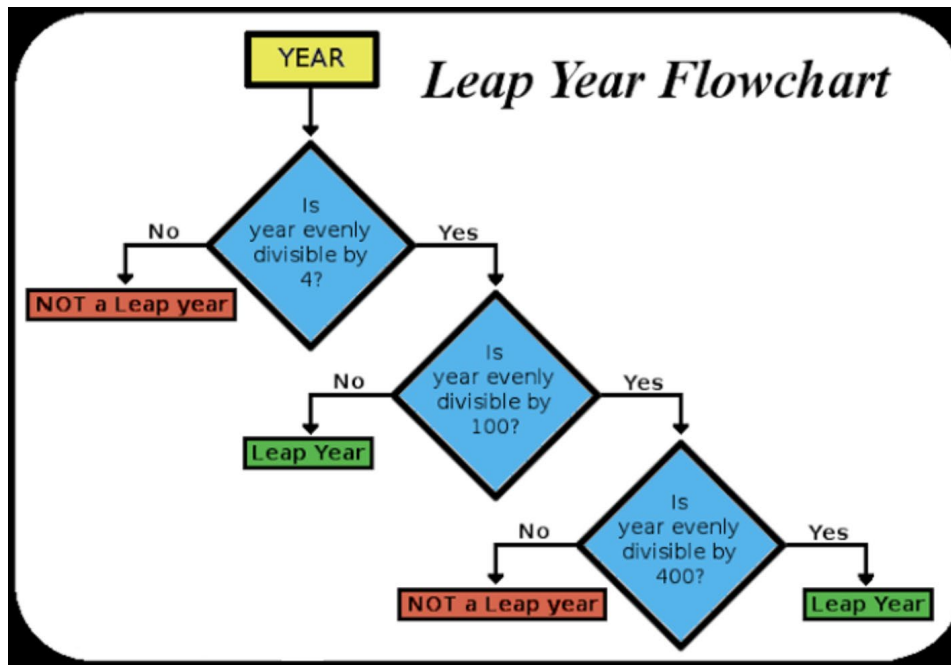
$\min\{a, b, c\}$

- [min.c](#)



LEAP year

- 如何判断输入是否是闰年？
 - 4的倍数，且不是100的倍数的，为闰年
 - 400的倍数，为闰年



[leap.c](#)

条件表达式

- 三元运算符

- 表达式1 ? 表达式2 : 表达式3
- 将简单if语句简化为一条表达式

- $i > j ? i : j$

- [ternary.c](#)

```
int x = a > b ? a : b;  
int y = (a > 0 ? a : 0) + b;
```

switch

```
switch (expression)
{
case /* constant-expression */:
    /* code */
    break;
case /* constant-expression */:
    /* code */
    break;

default:
    break;
}
```

switch

- 输入1-12数字，输出对应当前月份

[month.c](#)

[month1.c](#)

For循环

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



```
for ( <expression> ; <expression> ; <expression> )  
    <statement>
```

循环开始前的准备

循环结束条件

每轮循环的最后
一个执行
惯用法i++

For循环

- For循环就像一个计数循环，设定一个计数器，初始化后，在计数器到达某个值之前，重复循环体，每执行一轮，计数器更新
- 一个简单点的任务
 - 输入正整数n，计算n!
 - [fact.c](#)
- for语句三个表达式的省略
 - [simplefor.c](#)

For循环

- For循环就像一个计数循环，设定一个计数器，初始化后，在计数器到达某个值之前，重复循环体，每执行一轮，计数器更新
- 一个简单点的任务
 - 输入正整数 n ，计算 $n!$
 - [fact.c](#)
- 略复杂一点点的任务
 - Now, given a set A of integers, to compute their minimum.

Array

- `#define NUM 5`

- 初始化

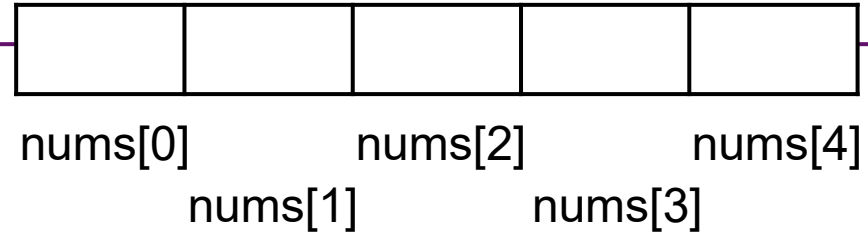
- `int nums[NUM] = {0};`
 - `{0,0,0,0,0}`
- `int nums[] = {0};`
 - `{0}`
- `int nums[NUM] = {1};`
 - `{1,0,0,0,0}`
- `int nums[NUM] = {[2]=1};`
 - `{0,0,1,0,0}`

• ~~`int numbers[NUM] = {};`~~

- `//forbidden by C99, allowed in GCC by default`

• ~~`int numbers[NUM];`~~

• ~~`int numbers[];`~~



`[]`: *subscript operator* (下标运算符)

- Now, given a set A of integers, to compute their minimum.
- [minInArray.c](#)



Star Pyramid

- [stars.c](#)



Keep coding!



