

■■ / Table of Contents

0-intro.....	1-35
1-types-io.....	37-70
2-if-for-array.....	71-93
3-for-a-while.....	95-105
4-loops.....	107-124
5-function.....	125-130
6-data-types.....	131-164
7-pointers.....	165-178
8-pointers-strings.....	179-183
9-double-pointers.....	185-191
10-function-pointers.....	193-204
11-struct.....	205-211
12-linkedlist.....	213-221
13-recursion.....	223-239

0. INTRODUCING C

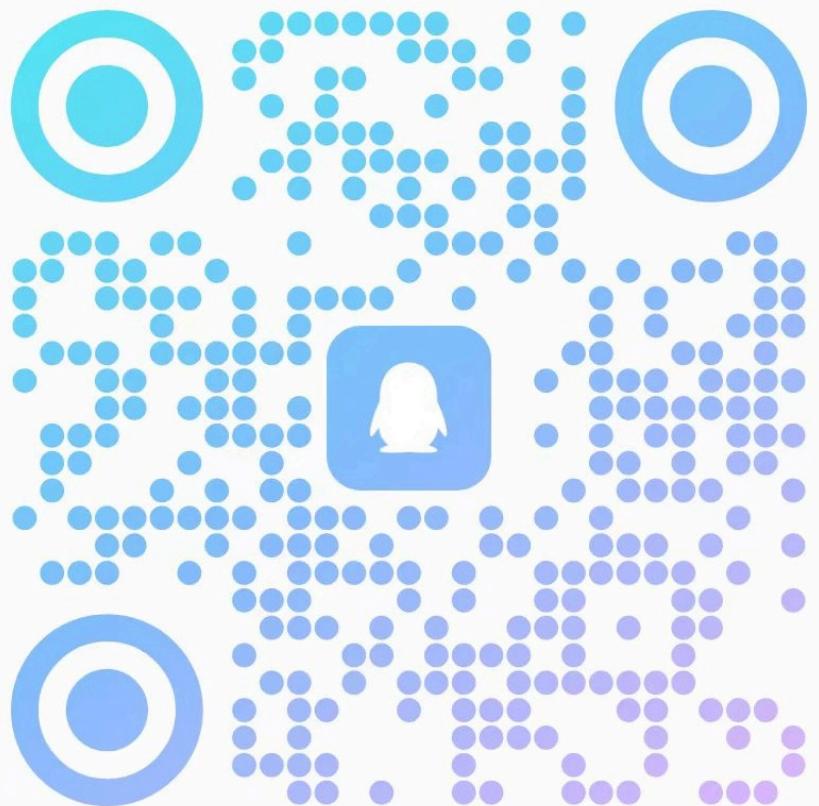
Hengfeng Wei(魏恒峰)
hfwei@nju.edu.cn





2024-CPL-6

群号: 1003531011

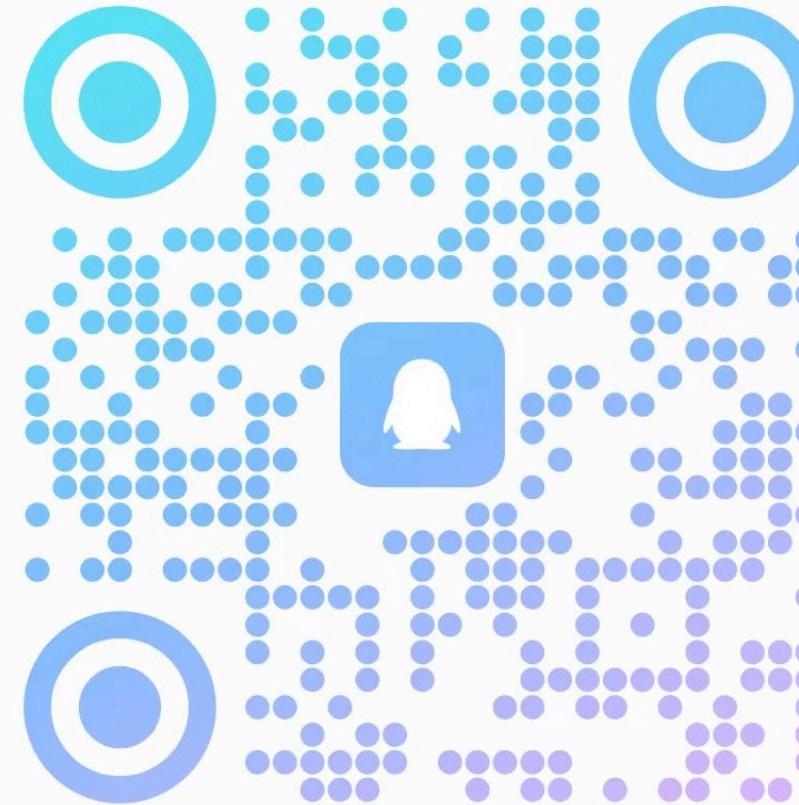


■ 2 ■ / ■ 240 ■

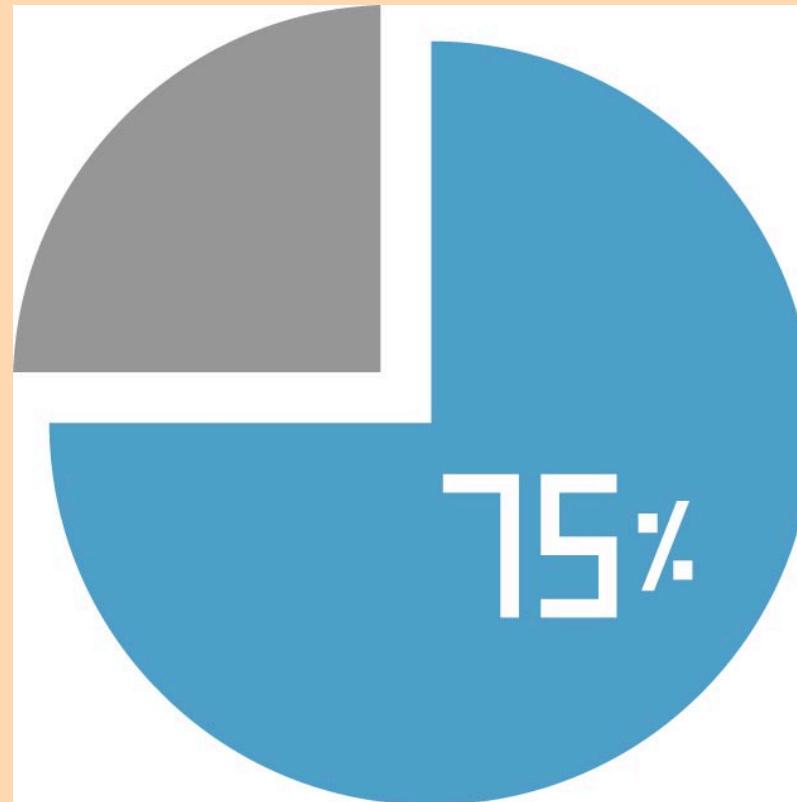


2024-CPL-答疑群

群号: 893689398

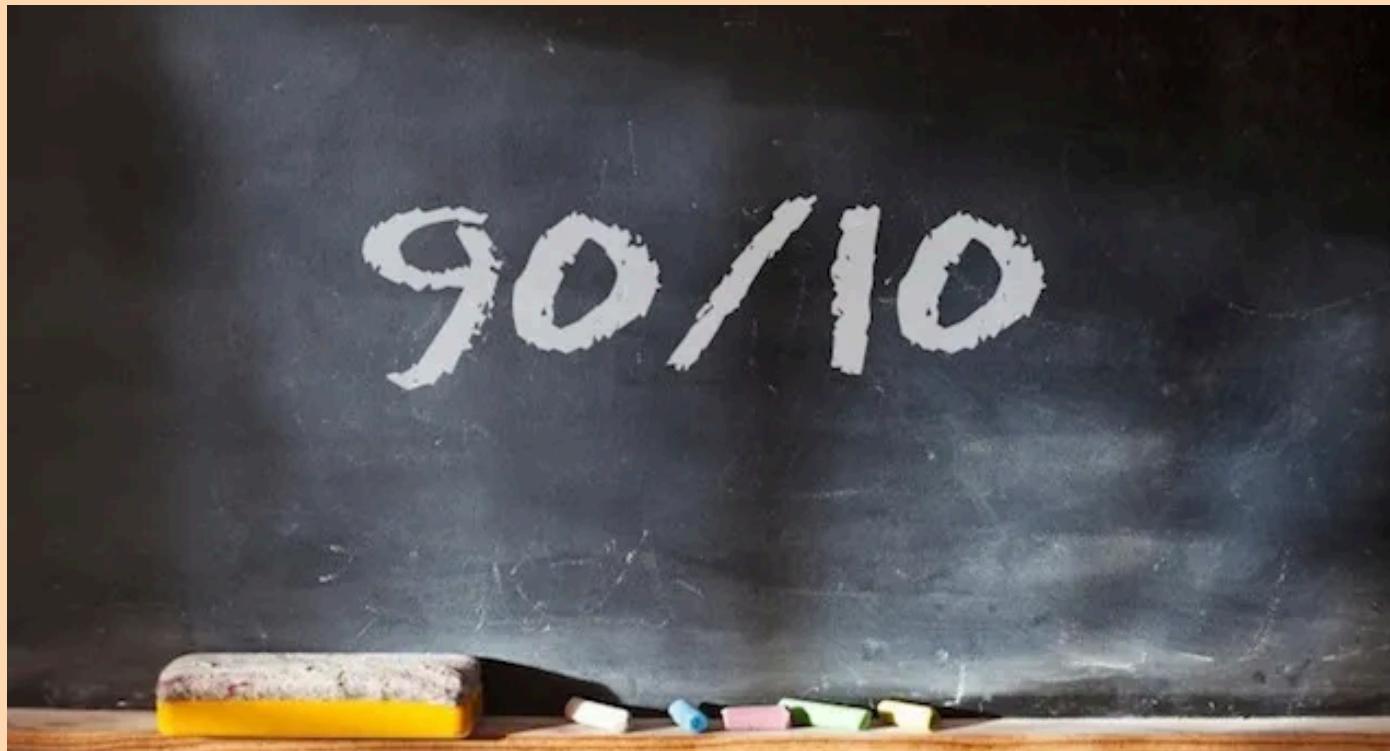


Questionnaire (1)



75% of students are new to programming.
3 / 240

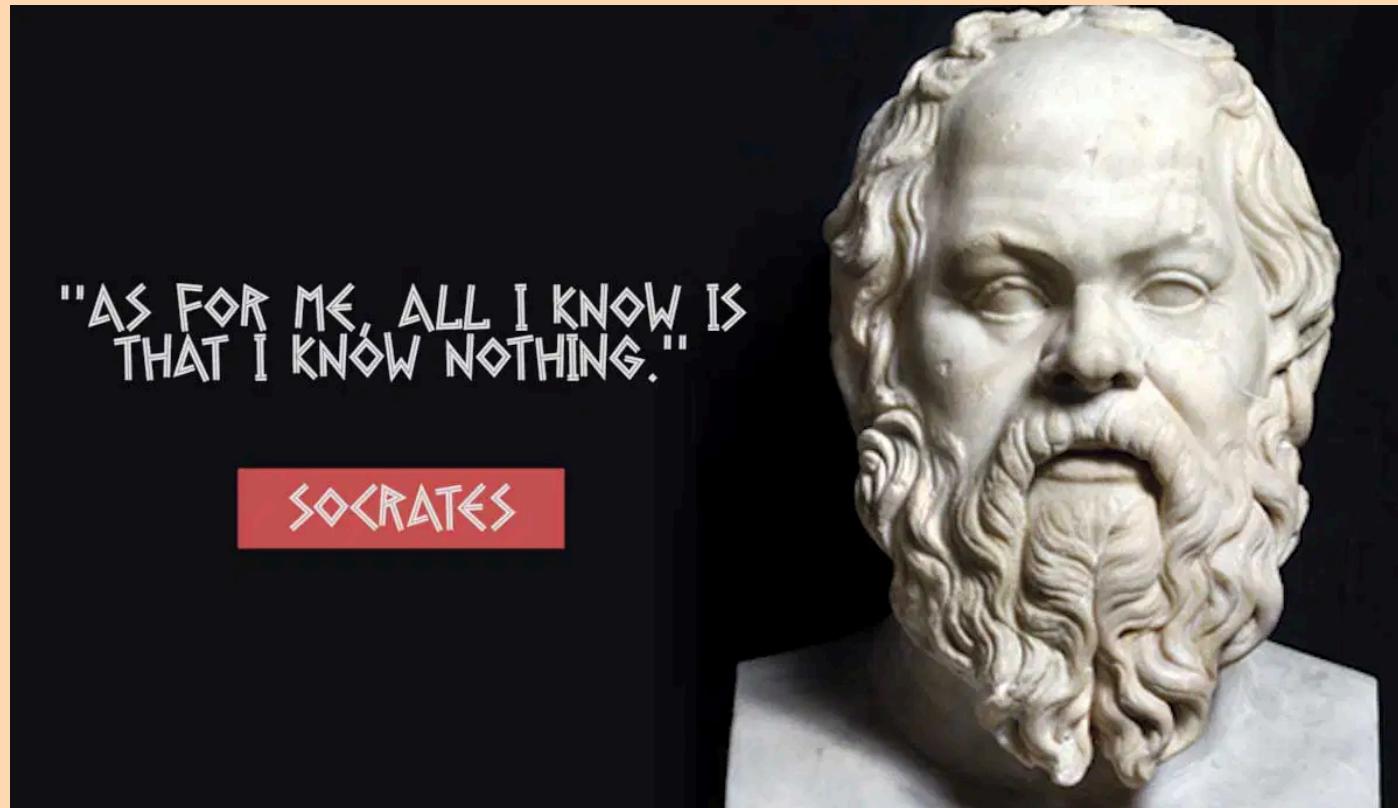
Questionnaire (2)



10% of students attended in some programming contests.



Questionnaire (3)



The C Beginners (know 0%)

5 / 240

The C Beginners



■ 6 ■ / ■ 240 ■

To The C Beginners

DON'T
PANIC!

From Beginners to Masters

Programming

De-Programming

cpl-docs @ docs.cpl.icu

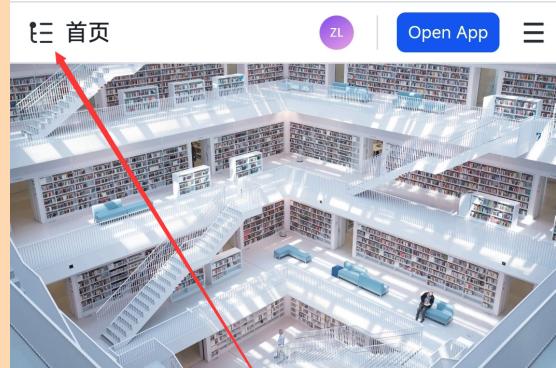


C 语言程序设计基础
C Programming Language

Q 搜索

- > 课程简介
- > 课程评价指标
- > 编程练习与题解
- > 抄袭与惩罚
- > 助教与答疑
- > C 语言开发环境
- > 课程资源
- > 编程风格
- > 往年 FAQ
- > 杂项

CPL Docs @ FeiShu



A screenshot of the CPL Docs @ FeiShu mobile application. At the top, there's a header with a library interior image, a '首页' (Home) button, a user icon labeled 'zl', an 'Open App' button, and a menu icon. A red arrow points from the '首页' button in the header down to the '首页' text on the main content page. Below the header, the word '首页' is displayed in large black font. Underneath is a box containing a red pushpin icon and the text '课程网站: <https://docs.cpl.icu>'. The main content area features a section titled '简介' (Introduction) with a detailed description of the course. At the bottom, there are sections for '须知' (Must-Know) and 'Open in FeiShu App' (with a blue button), along with social sharing icons for Favorite, Follow Updates, and Share.

10 / 240

首页

课程网站: <https://docs.cpl.icu>

简介

本系列讲义由南京大学软件学院C语言程序设计基础（CPL）课程的任课老师、助教、学生等共同撰写，面向不同基础的C语言学习者，旨在提供浅显易懂、简明扼要、深入浅出的教程，用以辅助课堂教学，降低学习门槛。

须知

Open in FeiShu App

Favorite Follow Updates Share

oj@oj.cpl.icu; oj@public.oj.cpl.icu

← 2024-0-introduction 2024-09-17 19:59:19 题目列表 评测情况 得分排名 admin ▾

2024-0-introduction
2024-09-03 08:00 - 2024-09-29 23:00

第 0 次作业, 不计分, 供各位同学测试环境使用 🎉

题目列表

#	名称	解决状态	通过人数
A	Hello C Programming!	✓ 已解决	144 / 151
B	a + b problem	✓ 已解决	124 / 137
C	Quine		44 / 71

导出互评结果到Excel 编辑作业



≡ Z ZULIP

- ▼ VIEWS
 - ✉️ Inbox
 - ⌚ Recent conversations
 - ≡ Combined feed
 - @ Mentions
 - 😊 Reactions
 - ★ Starred messages
 - 📄 Drafts
- ▶ DIRECT MESSAGES
- CHANNELS**
- Pinned

 - # Better-CPL
 - # cpl-resources
 - # gcc-mingw-clang**
 - # general
 - # git-github
 - # hello-world
 - # ide-clion
 - # ide-dev-c++
 - # ide-vs
 - # ide-vscode

- Active
- 🔒 TA

12 / 240

2024cpl@Zulip



Scores

- ~~考勤 (非必要不点名)~~
- 平时编程练习 (10 分)
- 阶段机试 1 (15 分)
- 阶段机试 2 (20 分)
- 期末机试 (30 分)
- 期末项目 (25 分)



No Plagiarism!!!



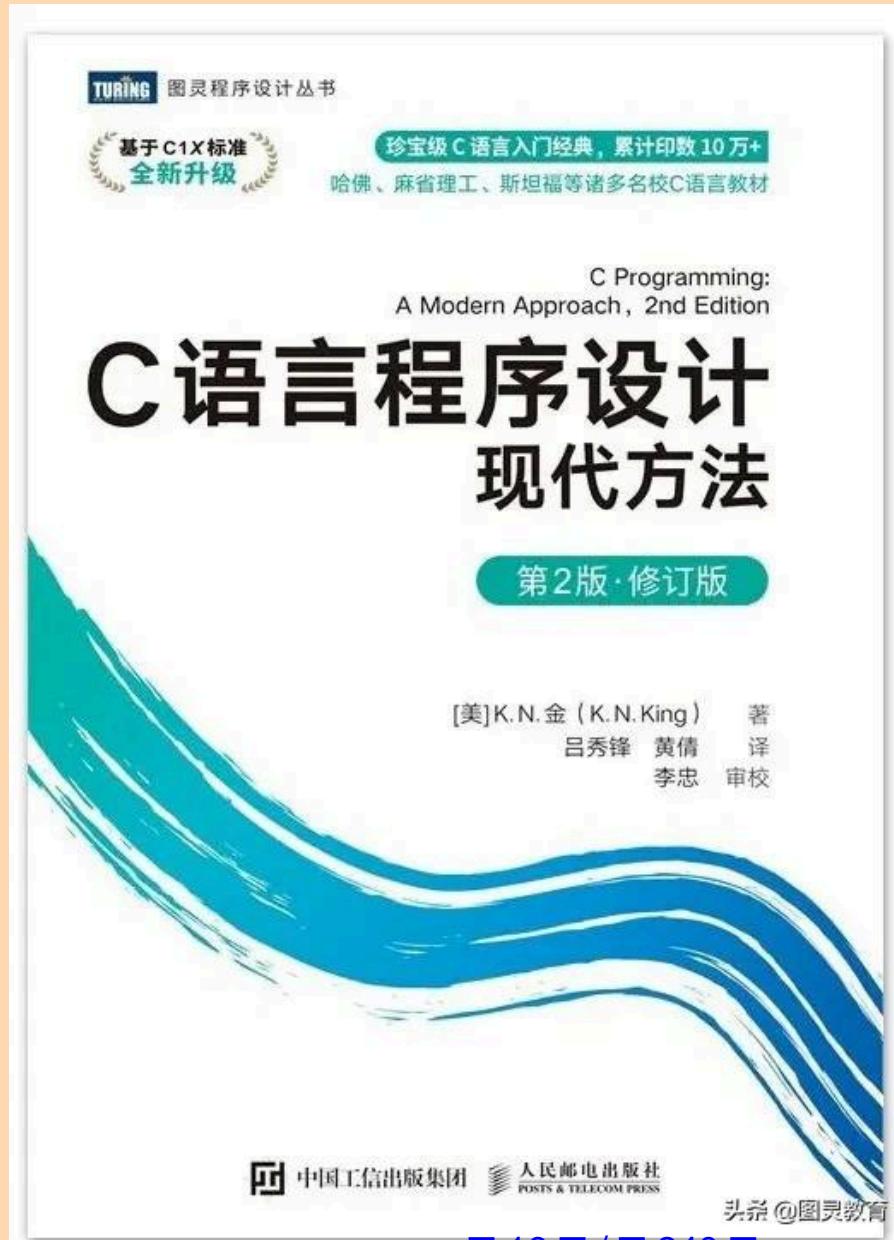
编程练习: 每次扣 5 分, 10 分扣完为止; **期末项目:** 项目分数清零

ASK ME ANYTHING

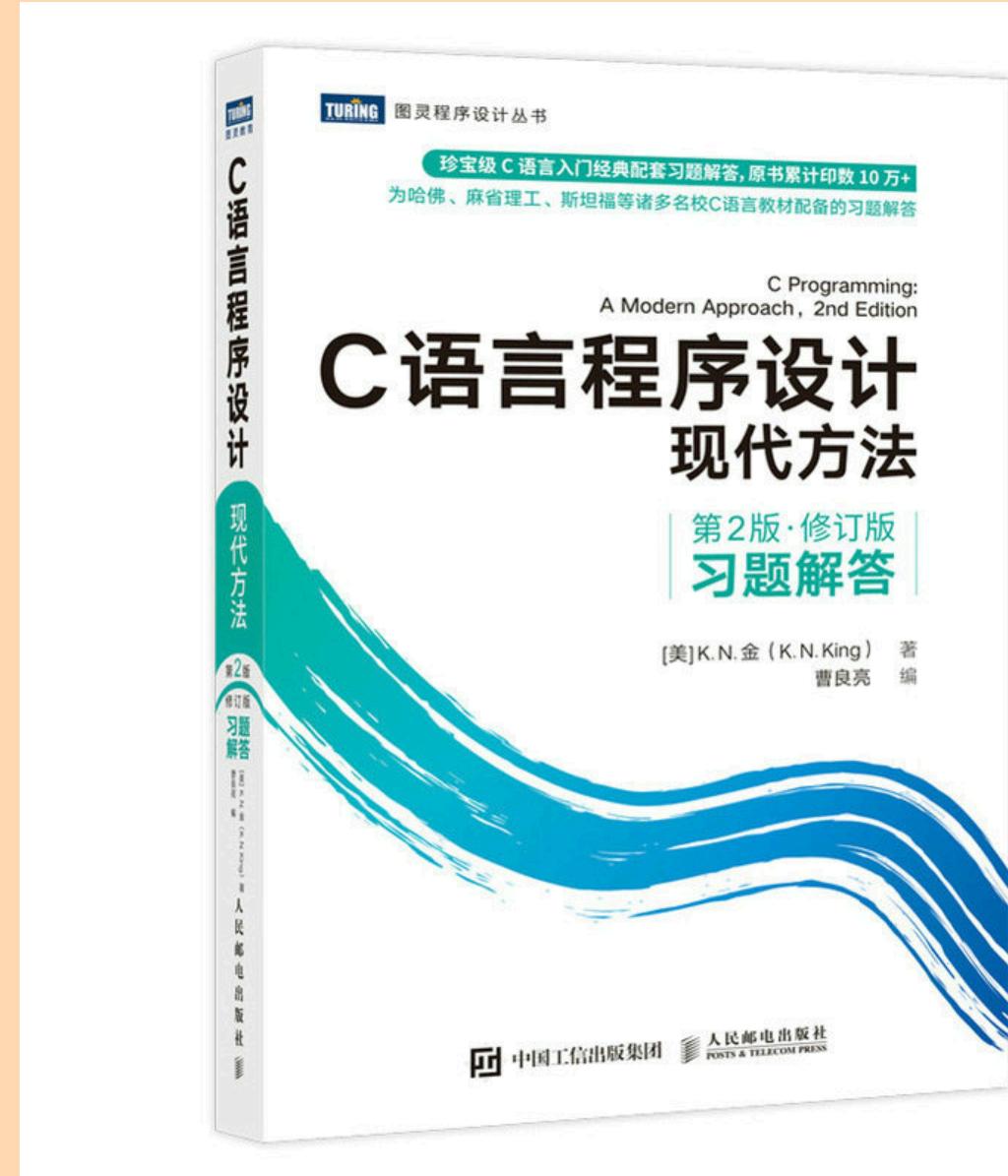
GREAT MINDS
DISCUSS IDEAS

About the 2024CPL Class

■ 15 ■ / ■ 240 ■



■ 16 ■ / ■ 240 ■



16



PEARSON

C和C++实务精选
品味岁月积淀，读享技术菁华

C Primer Plus

(第6版) 中文版

C Primer Plus
Sixth Edition

- 经久不衰的C语言畅销经典教程
- 针对C11标准进行全面更新



[美] Stephen Prata | 著
姜佑 | 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

■ 18 ■ / ■ 240 ■

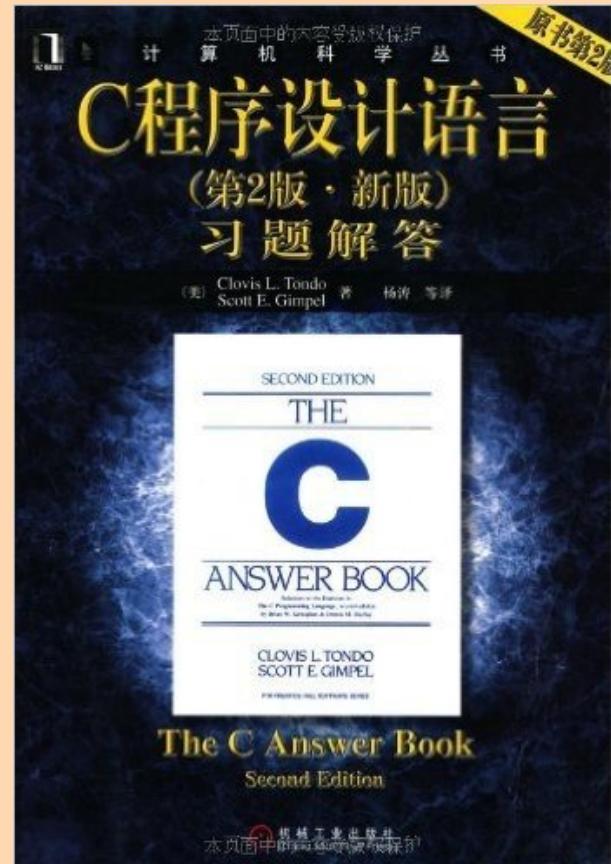
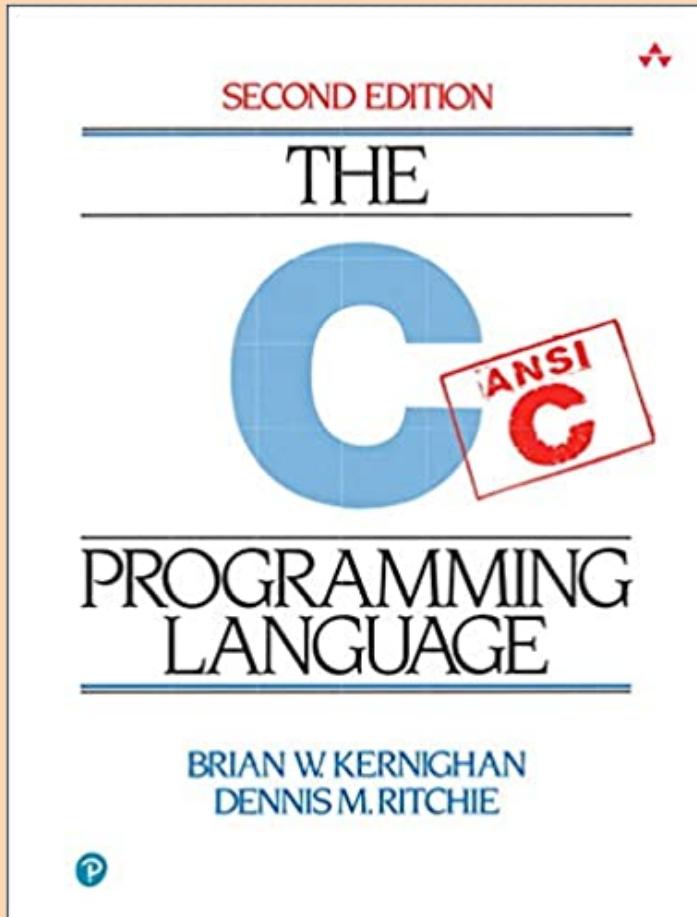
Stephen Prata

Sixth Edition

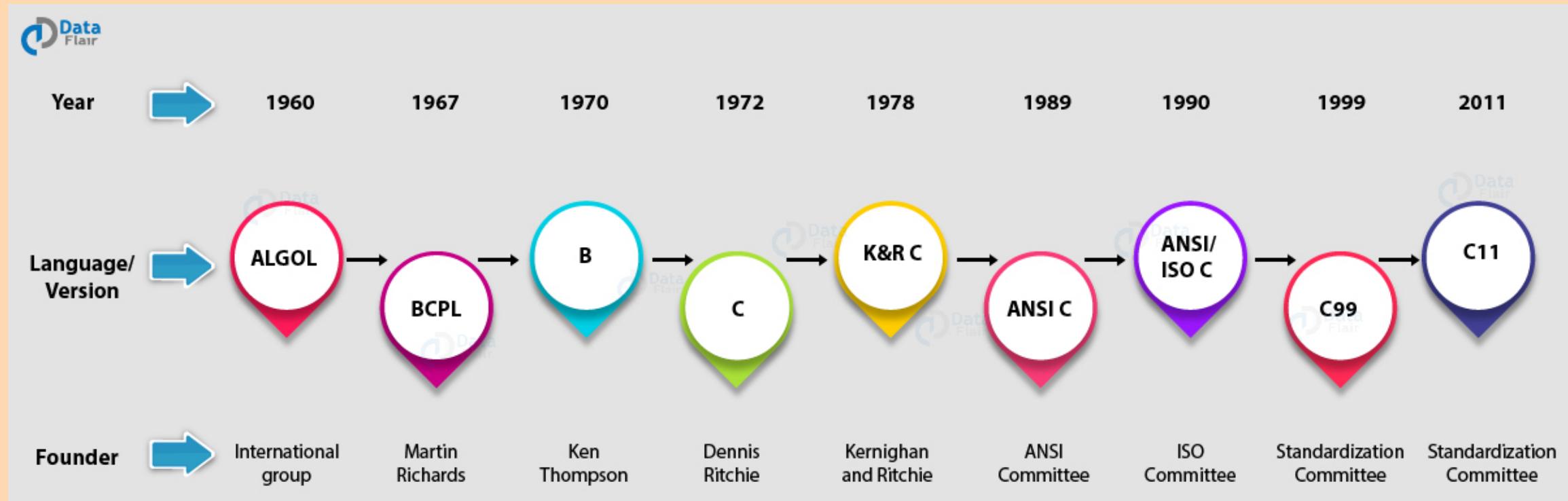
C Primer Plus



K&R C Bible (1978; 1988)



A Brief History of C @ cprefere~~n~~ce [C17; C23]



You do *NOT* need to be a **language lawyer!**

POISON



**If You're
Not Sure, Ask**





More Books in the Class . . .

■ 22 ■ / ■ 240 ■

易步图书
www.eubit.com

MANNING

程序员 软技能

[美] 唐·琼斯 (Don Jones) 著
魏喆 译

Own Your
Tech Career

Soft skills for technologists



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

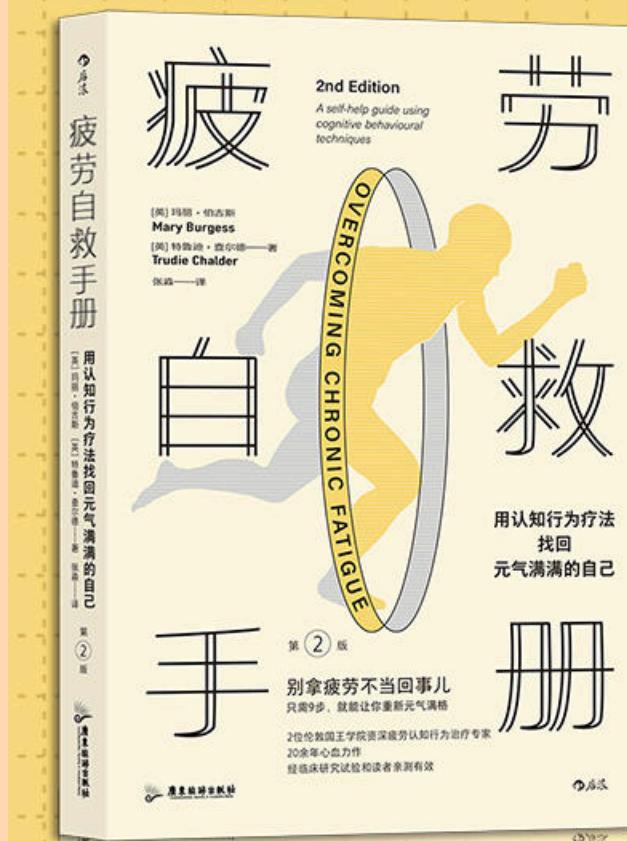
■ 23 ■ / ■ 240 ■

别拿疲劳不当回事儿



9步让你 重新元气满格

1. 观察你的活动与睡眠模式，了解疲劳是如何对你产生影响的
 2. 设定让自己生活更愉快、更平衡的目标
 3. 改善睡眠质量，稳定活动和休息模式
-



"TALK IS
CHEAP.
SHOW ME THE
CODE."
-LINUS TORVALDS



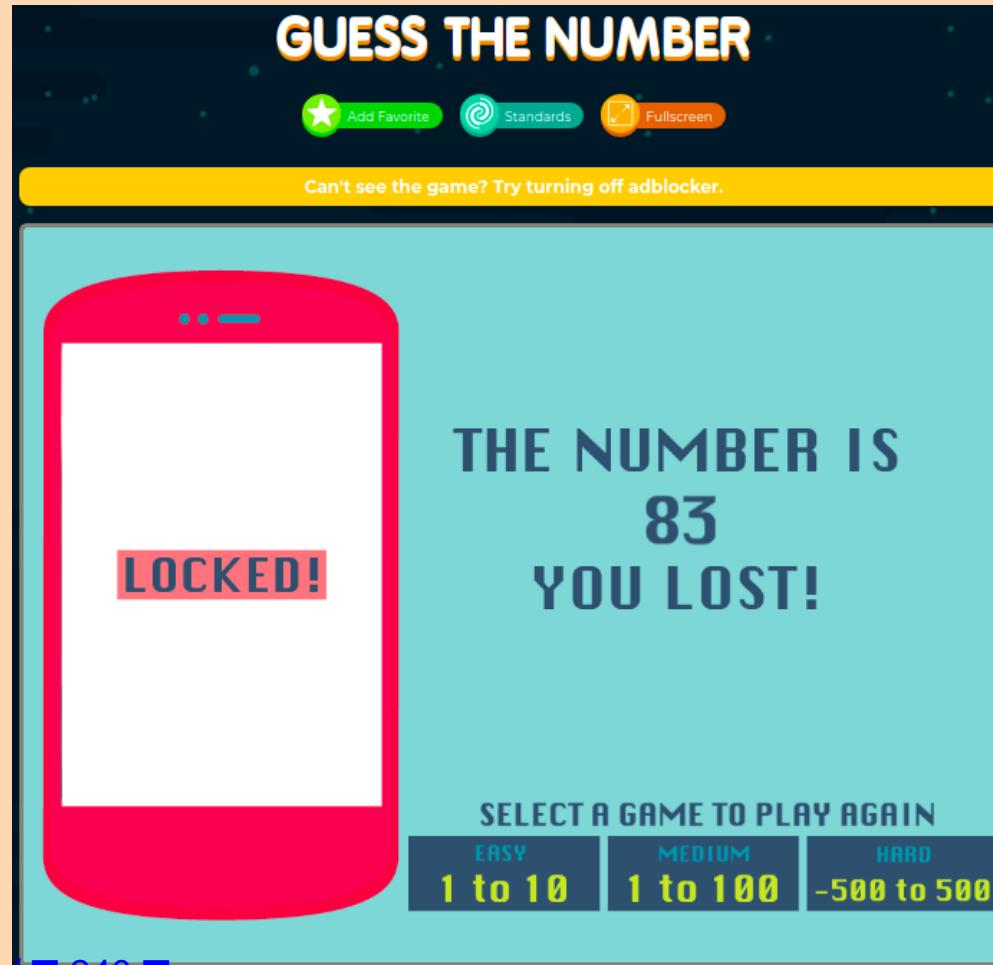


<Hello World/>

```
main()
{
    printf("Hello, world\n");
}
```

Zin Koht

Game: Guess the Number



Game: Guess the Number

Programming is *NOT*(only) about languages.

You learn C to express **YOUR IDEAS** with
COMPUTERS.

c reference

C reference

C89, C95, C99, C11, C17, C23

Language

Basic concepts
Keywords
Preprocessor
Expressions
Declaration
Initialization
Functions
Statements

Headers

Type support

Program utilities

Variadic functions

Error handling

Dynamic memory management

Date and time utilities

Strings library

Null-terminated strings:
byte – multibyte – wide

Algorithms

Numerics

Common mathematical functions

Floating-point environment (C99)

Pseudo-random number generation

Complex number arithmetic (C99)

Type-generic math (C99)

Input/output support

Localization support

Atomic operations library (C11)

Thread support library (C11)

Technical specifications

Dynamic memory extensions (dynamic memory TR)

Floating-point extensions, Part 1 (FP Ext 1 TS)

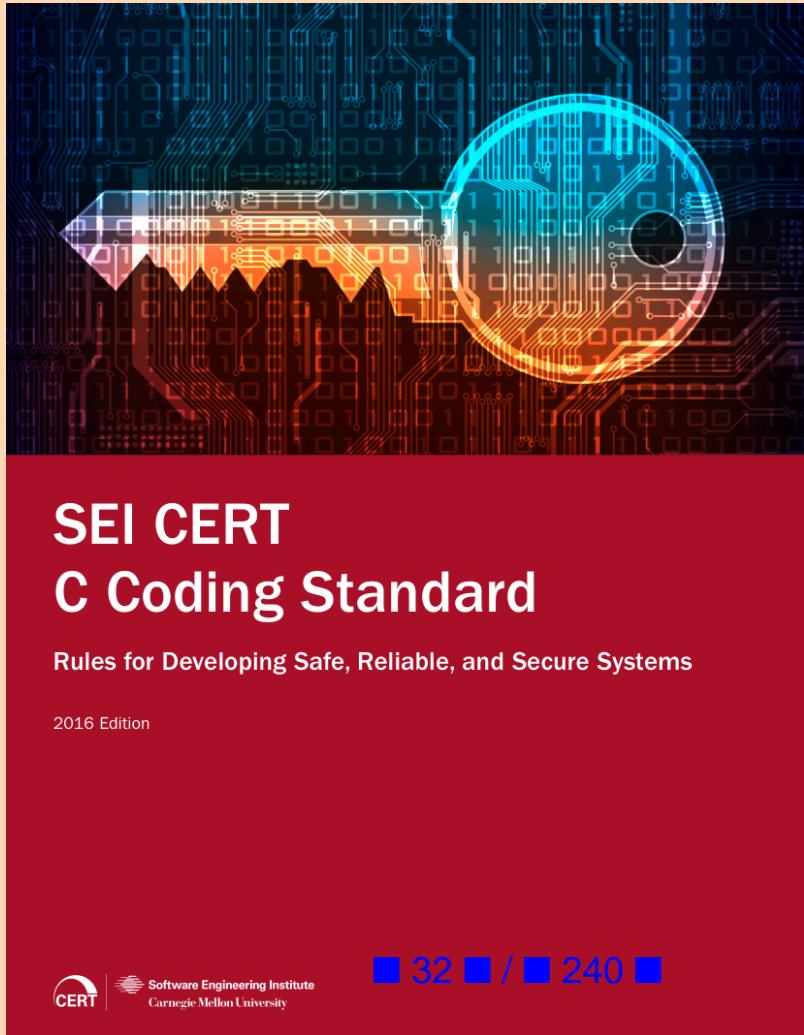
Floating-point extensions, Part 4 (FP Ext 4 TS)

External Links – Non-ANSI/ISO Libraries – Index – Symbol Index

Game: Guess the Number

Program = Input + Data + Operations + Output

Secure Coding in C



C语言编程指南 V1.0



ASK ME ANYTHING

GREAT MINDS
DISCUSS IDEAS



Thank you!

1. VARIABLES, TYPES, I/O

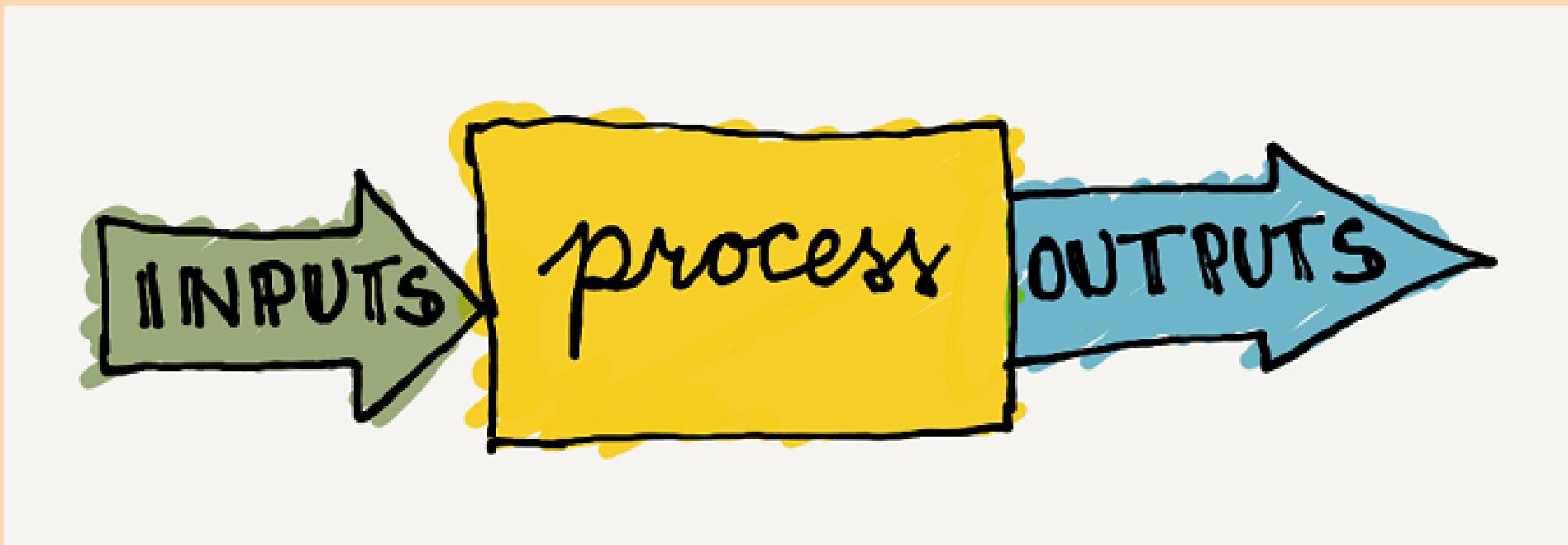
Hengfeng Wei(魏恒峰)

hfwei@nju.edu.cn



Review

Program = Input + Data + Operations + Output



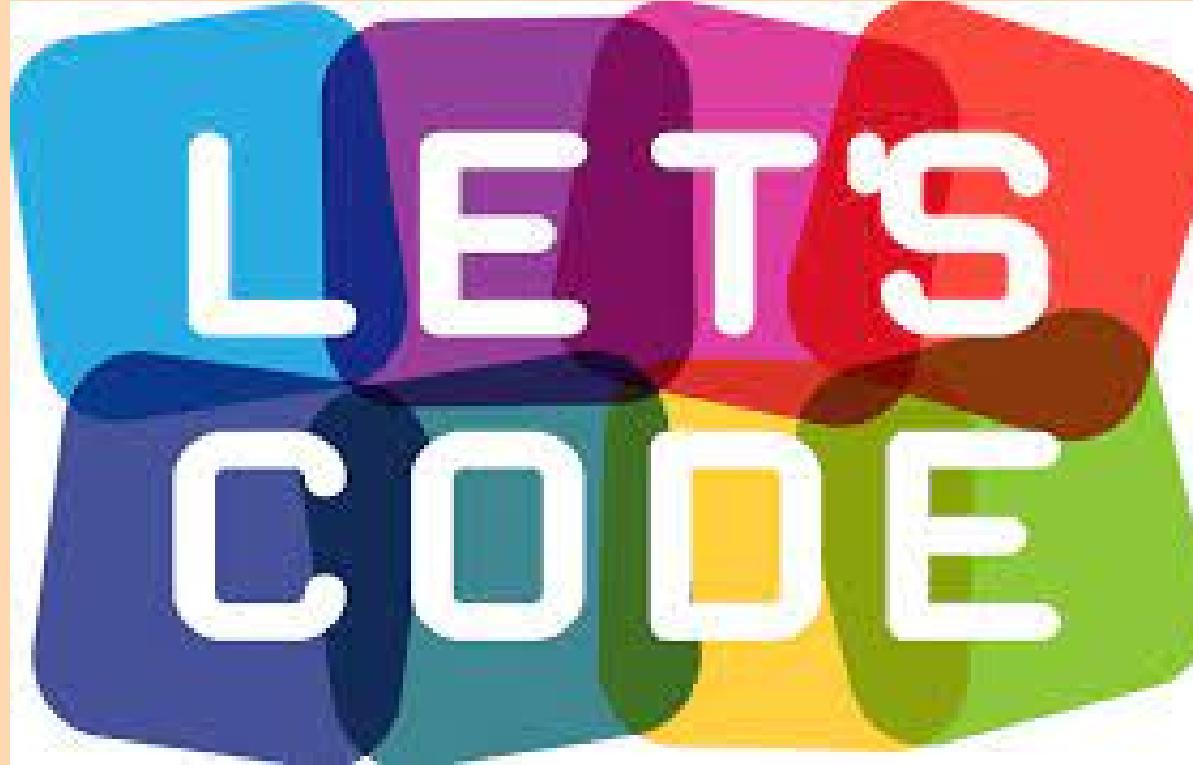
Overview

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)



circle.c **sphere.c** **mol.c**

admin.c **admin-scanf.c**

Circle

Given a **radius** (say 10) of a circle,
to compute its **circumference** and **area**.

$$L = 2\pi r \quad S = \pi r^2$$

- 每个结果各占一行
- 小数点后保留两位

Declaration/Definition (声明/定义)

```
int radius = 10;
```

- Declare/Define a *variable* called `radius`.
- The `type` of `radius` is `int` (integer).
- `radius` is `initialized` to 10.
- You can `assign` other integers to `radius`.
- `radius` refers to a `location` (`&radius`) in memory.

Identifiers (标识符)

```
int radius = 10;
```

radius is an *identifier*.

Warning: Do *not* start with `_`, which are reserved by C.

Always use meaningful identifiers in a uniform style!!!

Operators, Expressions, Assignment Statements

```
circumference = 2 * PI * radius;
```

Sphere

Given a **radius** (say 100) of a sphere,
to compute its **surface area** and **volume**.

$$A = 4\pi r^2 \quad V = \frac{4}{3}\pi r^3$$

- 每个结果占 1 行
- 小数点后保留 4 位
- 每个结果至少占 15 字符, 左对齐

○ : surface_area

■ 45 ■ Ø 240 ■ : volume

mol

6 克氧气的分子数是多少?

$$Q = 6/32 \times 6.02 \times 10^{23}$$

两种格式输出, 结果均使用科学计数法表示

- 第一行结果, 小数点后保留 3 位
- 第二行结果, 保留 5 位有效数字

A (Naive) Administration System

- Name (EN)
- Gender (F/M)
- Birthday (mm-dd-yyyy)
- Weekday (Xyz.)
- C
- Music
- Medicine
- Mean (.d)
- Standard Deviation (.dd)
- Ranking (%)



For 罗大佑 only:

- 每组信息占一行
- 各项信息使用 \t 间隔
- 各项信息遵循特定格式要求

char and <ctype.h>

	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>
<u>0_0</u>	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
<u>1_16</u>	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
<u>2_32</u>	SP 0020	! 0021	" 0022	# 0023	\$ 0024	% 0025	& 0026	' 0027	(0028) 0029	002A	002B 002C	002D 002C	002E 002D	002F 002E	
<u>3_48</u>	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
<u>4_64</u>	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
<u>5_80</u>	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C] 005D	^ 005E	_ 005F
<u>6_96</u>	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
<u>7_112</u>	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	 007C	{ 007D	~ 007E	DEL 007F

■ 49 ■ / ■ 240 ■ A **char** is actually an **int**.

C string

```
char first_name[ ] = "Tayu";
```

A C string is a null-terminated (`\0`) sequence of characters.

String literal: 'T', 'a', 'y', 'u', '\0'

- `char first_name[5] = "Tayu";`
- `char first_name[10] = "Tayu";`
- ~~`char first_name[2] = "Tayu";`~~

Conversion Specification

%[flags][width].[precision]specifier

- **%d** : decimal int
- **%f** : double
- **%e %E** : double ($-d.ddd \times 10^{\pm dd}$)
- **%C** : char
- **%S** : C string
- **%%**



It is up to you to ensure that the type of the actual argument matches the type expected by conversion specifiers.

[Undefined Behavior \(UB\)](#)

```
%[flags][width][.[precision]]specifier
```

- -: left-justified (otherwise, right-justified)
- +: always begin with a plus or minus sign

```
%[flags][width][.[precision]]specifier
```

- minimum field width
- padded with spaces if it has fewer characters

%[flags][width][.[precision]]specifier

- **%d** : **minimum number** of digits
 - expanded with leading zeros when needed
- **%f , %e , %E** : **number** of digits after **.**
 - default is 6
- **%s** : **maximum number** of characters

<https://en.cppreference.com/w/c/io/fprintf>

INTERNATIONAL STANDARD

©ISO/IEC

ISO/IEC9899:2017

Programming languages — C

(cover sheet to be replaced by ISO)

This is a working document of SC22/WG14

This version of the document is intended to be the version that is to go into ballot for C17.

- It is based on the transformed L^AT_EXversion of the document that has been proofread by the members of WG14 and that has been approved by teleconference in June 2017.
- It applies all TCs of closed DRs up to April 2017.
- It applies the changes that have been voted in Markham.
- It updates some normative references.
- It provides the minimal changes required for a new version of the standard.
- It integrates some editorial changes that had been found during the revision process.

A brief explanation of the changes could still be added to the foreword.

Document conventions

This document classifies identifiers into different categories. This categorization is important to produce a correct index.

The classes are

- Normal identifiers, `toto`.
- keywords, `while`
- symbols with external linkage of the C library, `malloc`
- types, `size_t`
- predefined macros that alias language features, `complex`
- other predefined macros, `EOF`
- pragmas and their particles, `STDC`
- tag names and members of `struct`, `union` or `enum`, `tv_sec`
- name fragments, usually reserved prefixes, `atomic_`

Section 7.21: <stdio.h>, P225--230

■ 57 ■ / ■ 240 ■

THE STANDARD

<time.h> * <limits.h> * <float.h>
<stddef.h> * <wctype.h> * <locale.h>
<stdio.h> * <ctype.h> * <string.h>
<math.h> * <stdlib.h> * <assert.h>
<stdarg.h> * <setjmp.h> * <signal.h>
<time.h> * <limits.h> * <float.h>
<stddef.h> * <errno.h> * <locale.h>
<stdio.h> * <ctype.h> * <string.h>
<math.h> * <stdlib.h> * <assert.h>
<stdarg.h> * <setjmp.h> * <signal.h>
<time.h> * <limits.h> * <float.h>
<stddef.h> * <errno.h> * <locale.h>

LIBRARY

P.J. PLAUCER



Chapter 12: <stdio.h>, P257--262

■ 58 ■ / ■ 240 ■

之乎者也



滾石唱片

RD-1008

羅大佑

■ 59 ■ / ■ 240 ■



%[*][width]specifier

- `%d`: skip white-spaces; match a decimal `int`
- `%lf`: skip white-spaces; match a `double`
- `%C`: match a `char` (do NOT skip white-spaces)
- `%S`: match a sequence of non-white-spaces
- `%%`: mathch a `%`

`%[*][width]specifier`

- *: assignment-suppressing

`%[*][width]specifier`

- maximum field width to scan

<https://en.cppreference.com/w/c/io/fscanf>

INTERNATIONAL STANDARD

©ISO/IEC

ISO/IEC9899:2017

Programming languages — C

(cover sheet to be replaced by ISO)

This is a working document of SC22/WG14

This version of the document is intended to be the version that is to go into ballot for C17.

- It is based on the transformed L^AT_EXversion of the document that has been proofread by the members of WG14 and that has been approved by teleconference in June 2017.
- It applies all TCs of closed DRs up to April 2017.
- It applies the changes that have been voted in Markham.
- It updates some normative references.
- It provides the minimal changes required for a new version of the standard.
- It integrates some editorial changes that had been found during the revision process.

A brief explanation of the changes could still be added to the foreword.

Document conventions

This document classifies identifiers into different categories. This categorization is important to produce a correct index.

The classes are

- Normal identifiers, `toto`.
- keywords, `while`
- symbols with external linkage of the C library, `malloc`
- types, `size_t`
- predefined macros that alias language features, `complex`
- other predefined macros, `EOF`
- pragmas and their particles, `STDC`
- tag names and members of `struct`, `union` or `enum`, `tv_sec`
- name fragments, usually reserved prefixes, `atomic_`

Section 7.21: `<stdio.h>`, P231--P237

■ 64 ■ / ■ 240 ■

THE STANDARD

<time.h> * <limits.h> * <float.h>
<stddef.h> * <wctype.h> * <locale.h>
<stdio.h> * <ctype.h> * <string.h>
<math.h> * <stdlib.h> * <assert.h>
<stdarg.h> * <setjmp.h> * <signal.h>
<time.h> * <limits.h> * <float.h>
<stddef.h> * <errno.h> * <locale.h>
<stdio.h> * <ctype.h> * <string.h>
<math.h> * <stdlib.h> * <assert.h>
<stdarg.h> * <setjmp.h> * <signal.h>
<time.h> * <limits.h> * <float.h>
<stddef.h> * <errno.h> * <locale.h>

LIBRARY

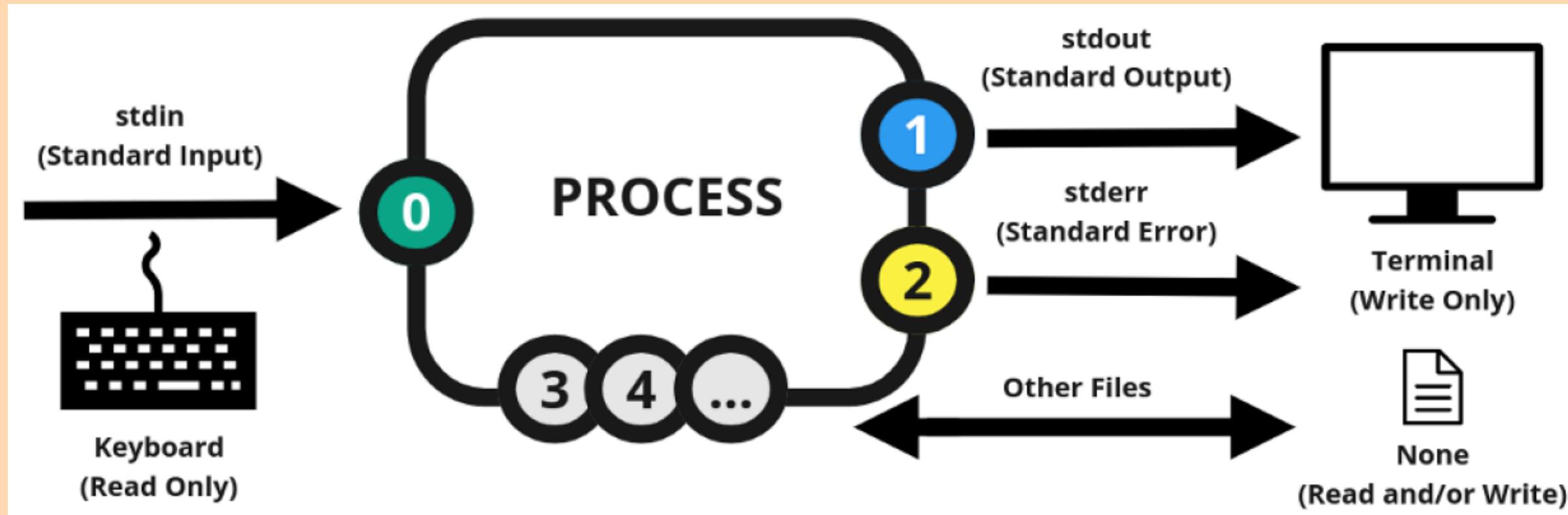
P.J. PLAUCER



Chapter 12: <stdio.h>, P263--P268

■ 65 ■ / ■ 240 ■

stdin, stdout, stderr



Input/Output Redirection



printf-error.c scanf-error.c

■ 67 ■ / ■ 240 ■

A beginners' guide away from `scanf`

Do NOT use `scanf`.

Why does everyone say not to use `scanf` ? What
should I use instead?



scanf-c17-ex2.c scanf-c17-ex3.c



■ 70 ■ / ■ 240 ■

2. If, For, Array

Hengfeng Wei(魏恒峰)

hfwei@nju.edu.cn



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)

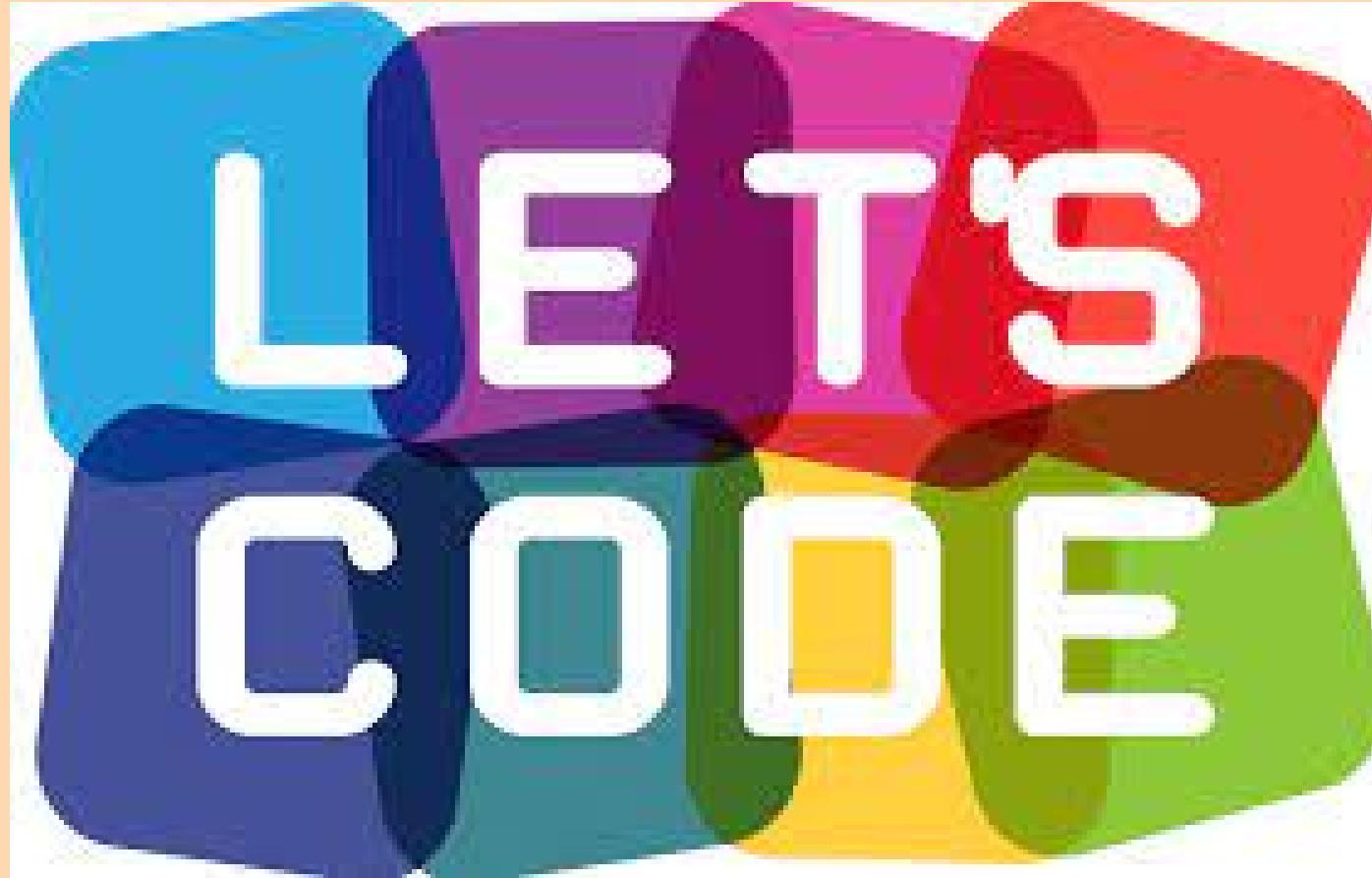
Overview

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 of Two

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

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

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

(三目运算符)

Do Not Use it Too Much!

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.

$$\min = \min A$$

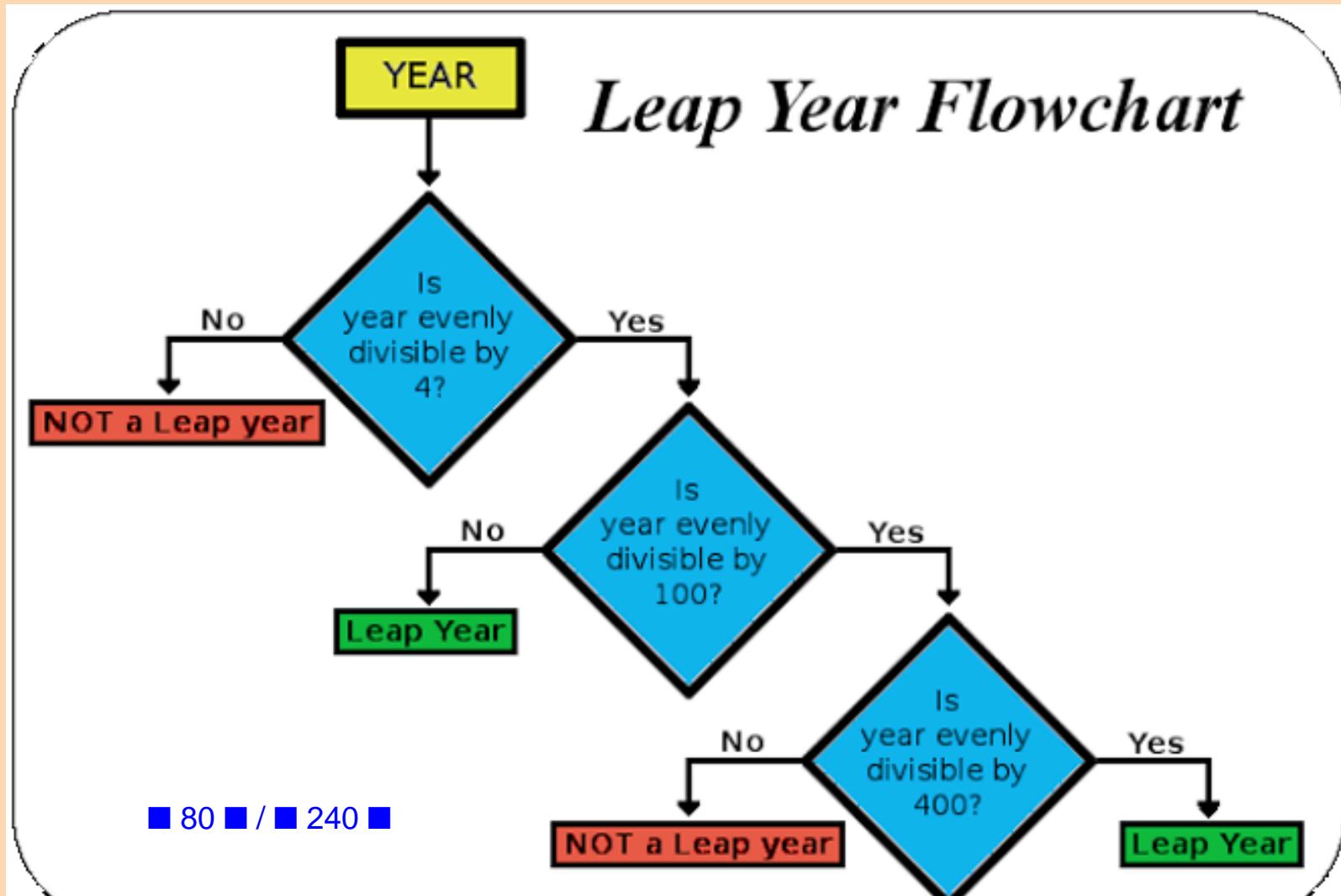


$$\min\{3, 5, 2, 7\} = \min(\min(\min(3, 5), 2), 7)$$

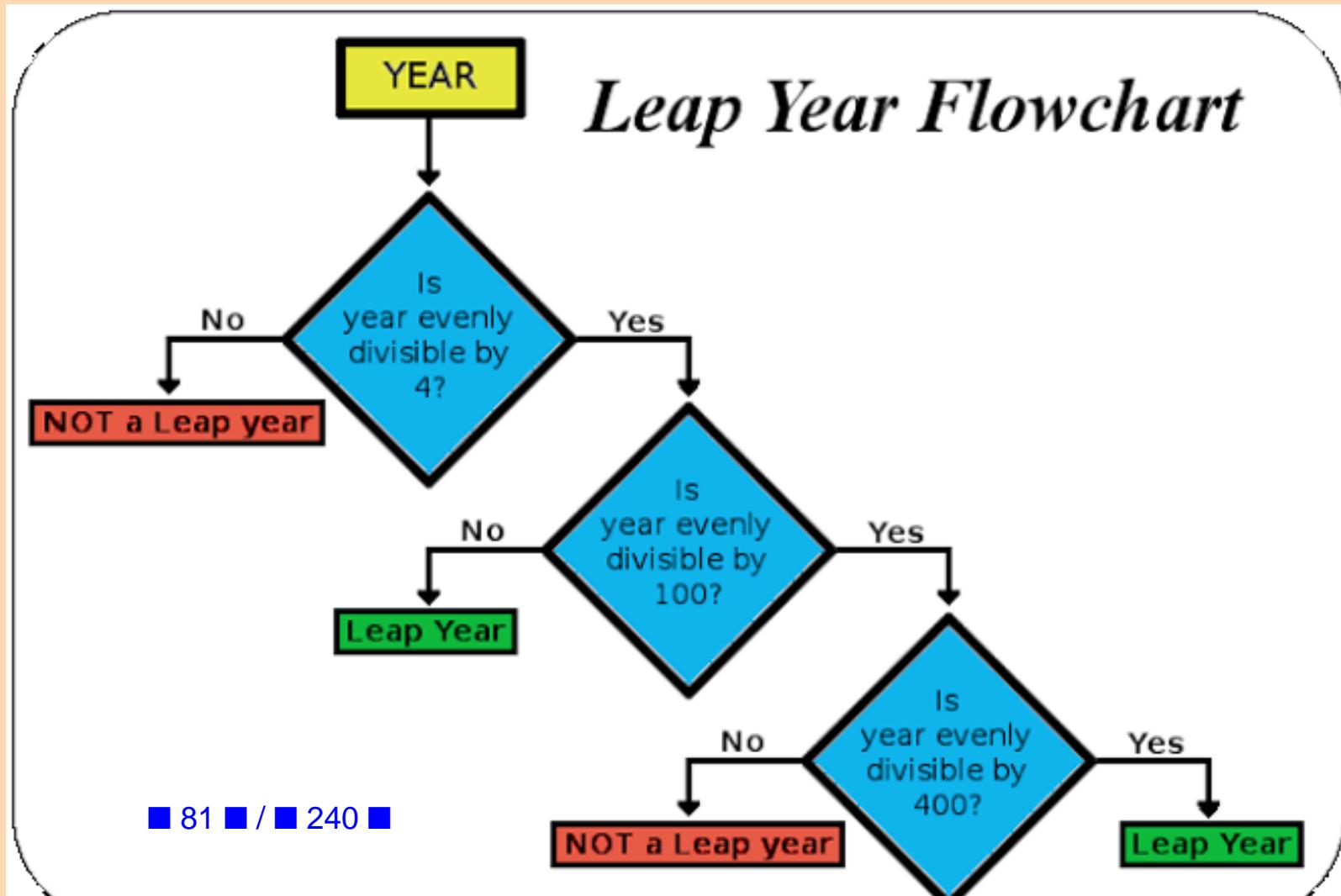
Leap Year



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
```

```
#define NUM 5
```

NUM is known at *compiler* time.

Array Initializer

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

$$\min = \min_i A_i$$



■ 93 ■ / ■ 240 ■

3. FOR A WHILE

Hengfeng Wei(魏恒峰).
hfwei@nju.edu.cn



Review

If Statement (if 语句)

For Statement (for 语句)

Logical Expressions (逻辑表达式)

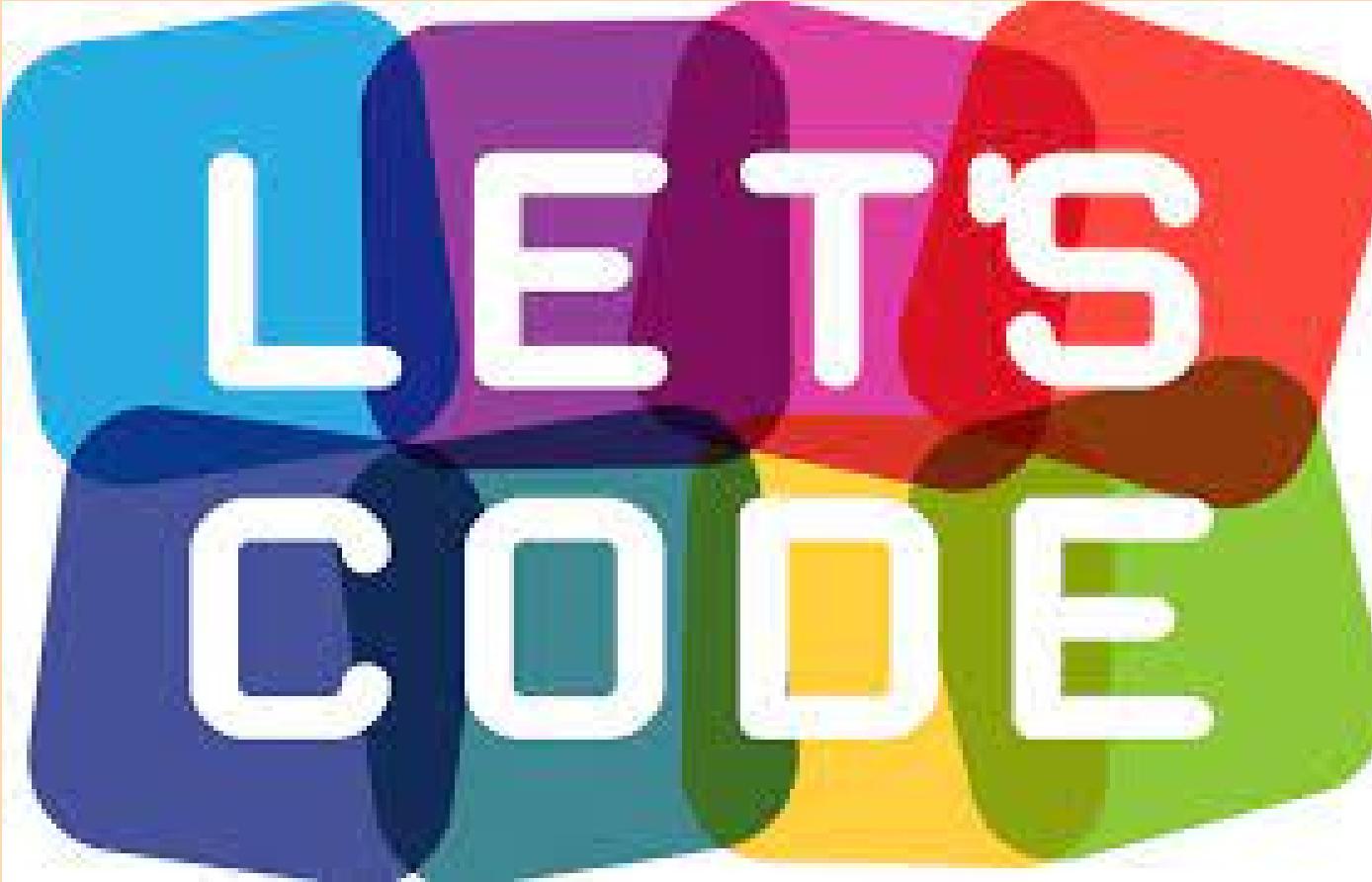
Array (数组)

Overview

For Statement (More Examples)

While (Do-While) Statement

break Statement

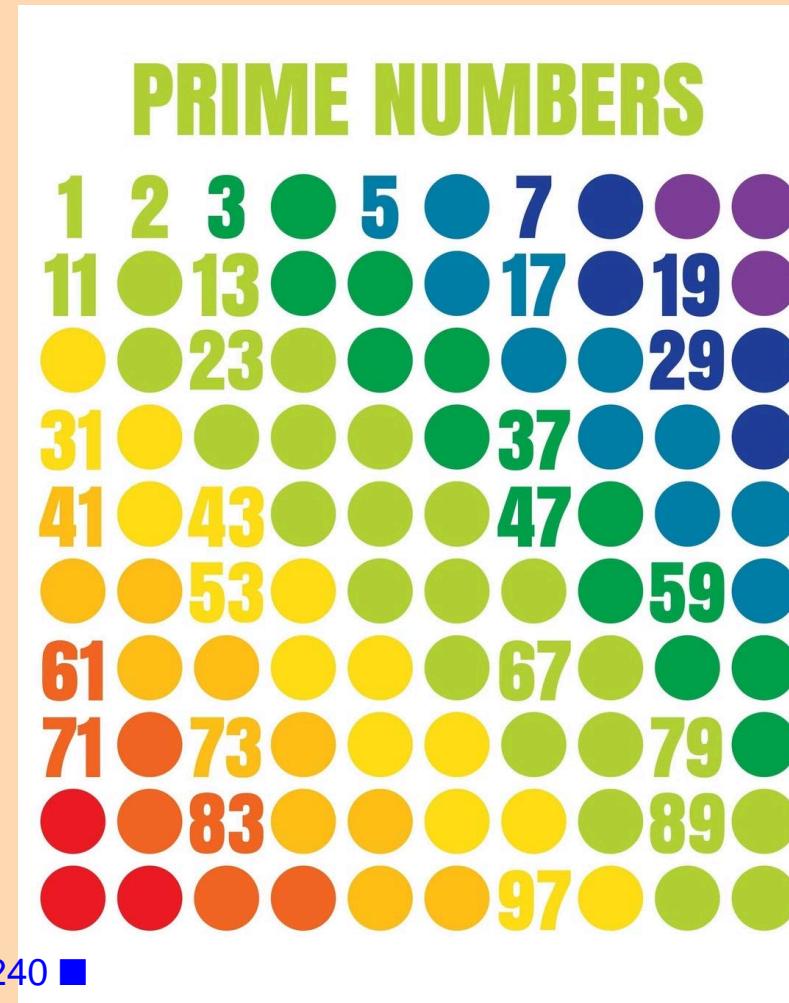


stars.c **primes.c** **binary-search.c** **digits.c**
selection-sort.c **palindrome.c**

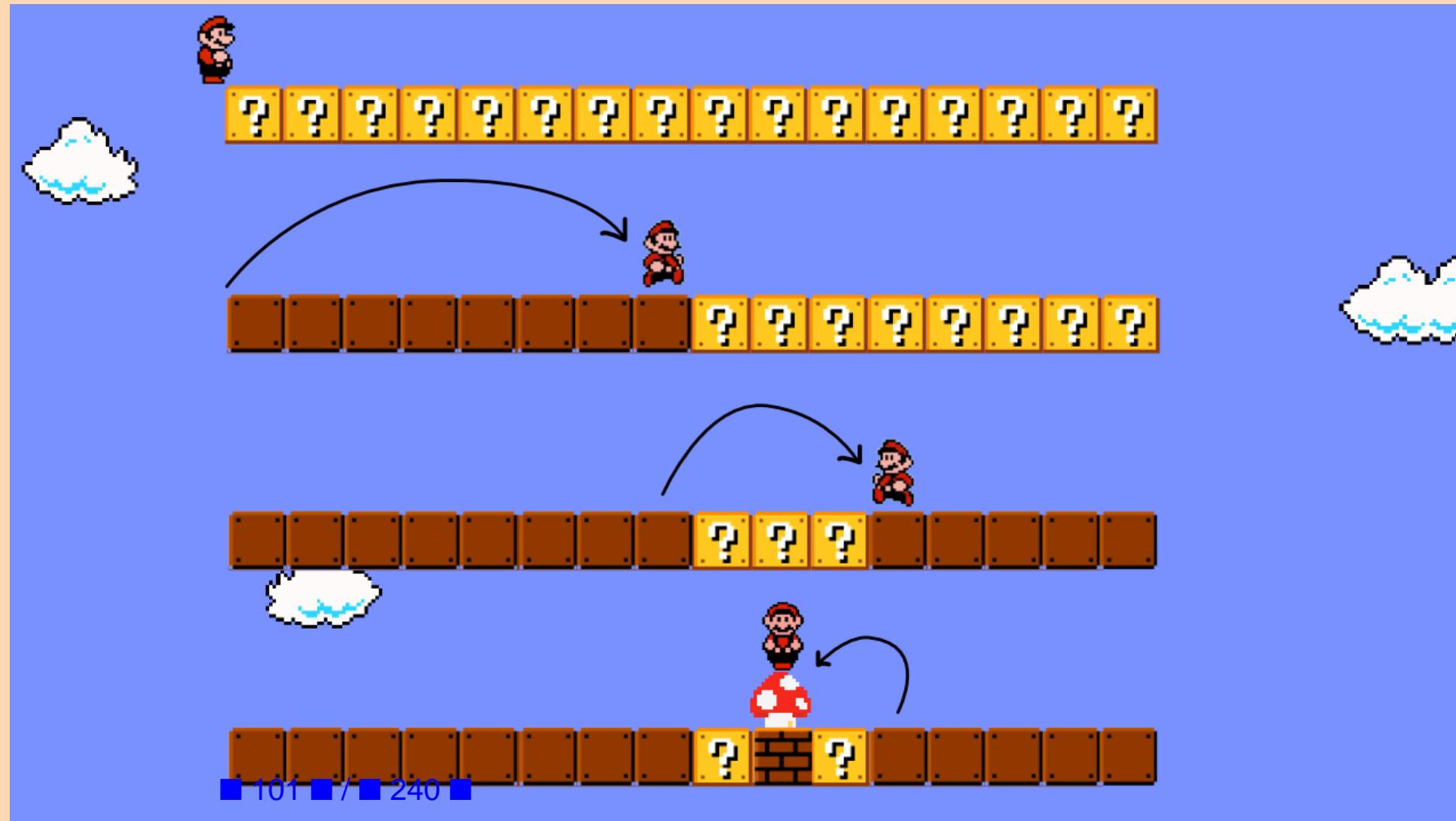
Stars Pyramid (stars.c)



Prime Numbers (primes.c)



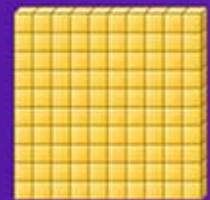
Binary Search (binary-search.c)



Digits (digits-while.c , digits-do-while.c)



Identify the Place & Value
of Digits in a
Three-Digit Number



1 Hundred

127

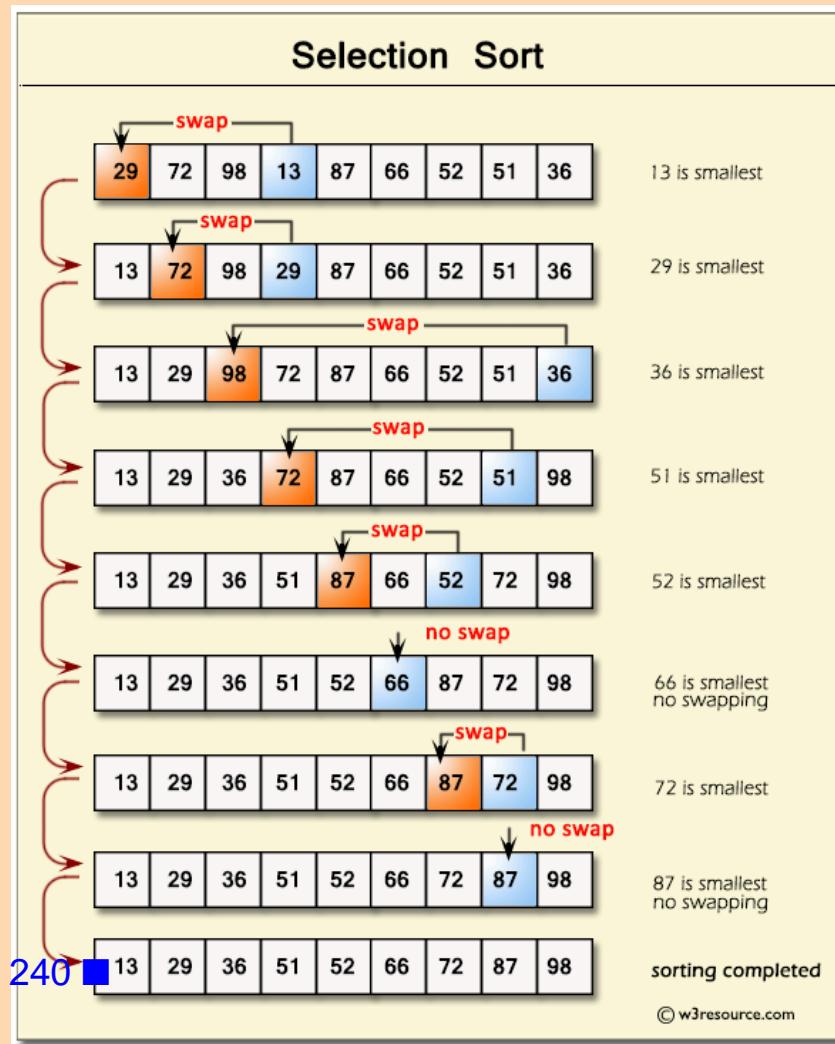


2 Tens

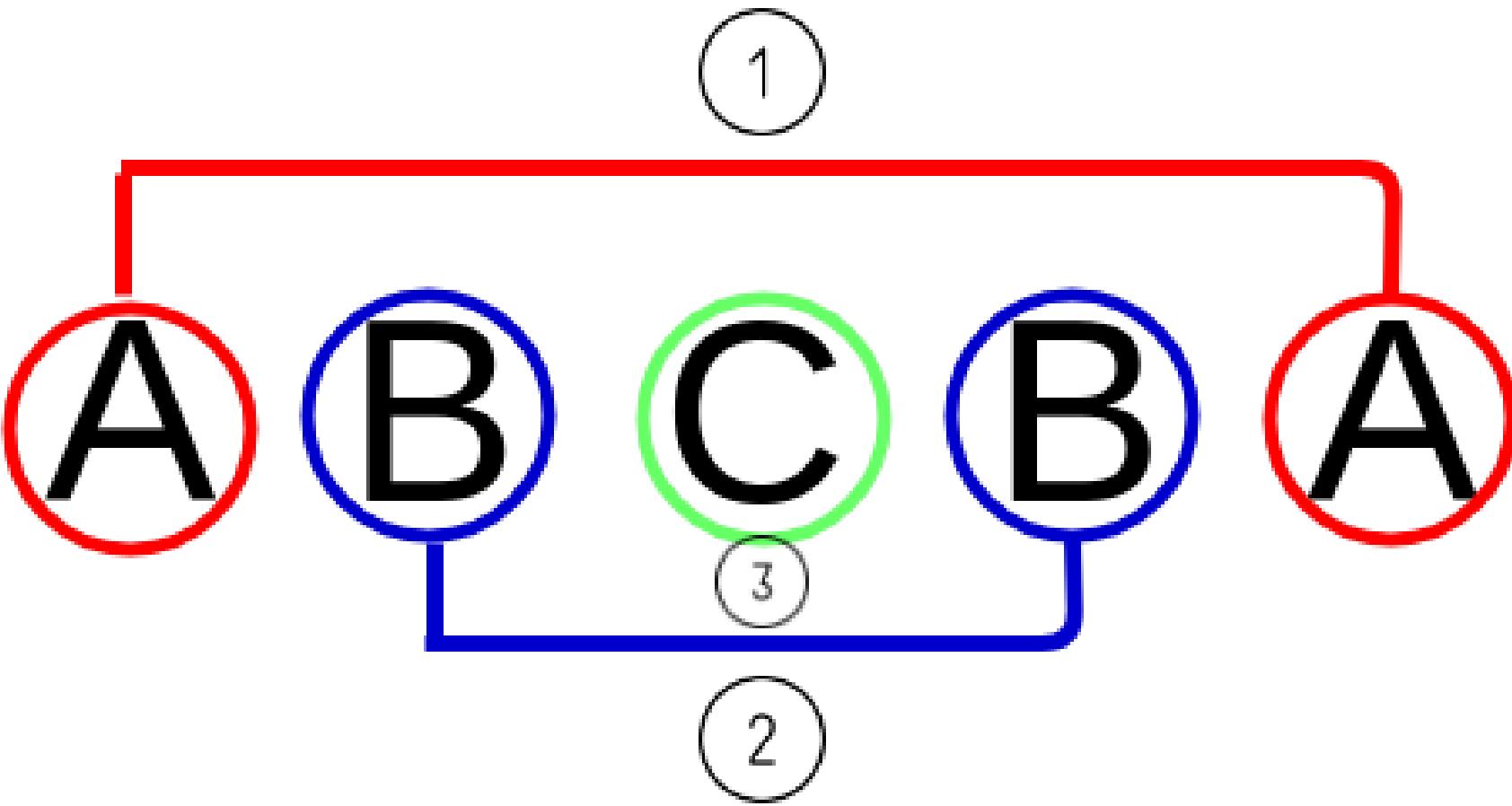


7 Ones

Selection Sort (selection-sort.c)



Palindrome (palindrome.c)





■ 105 ■ / ■ 240 ■

4. LOOPS

Hengfeng Wei(魏恒峰).
hfwei@nju.edu.cn



Review

For Statement (More Examples)

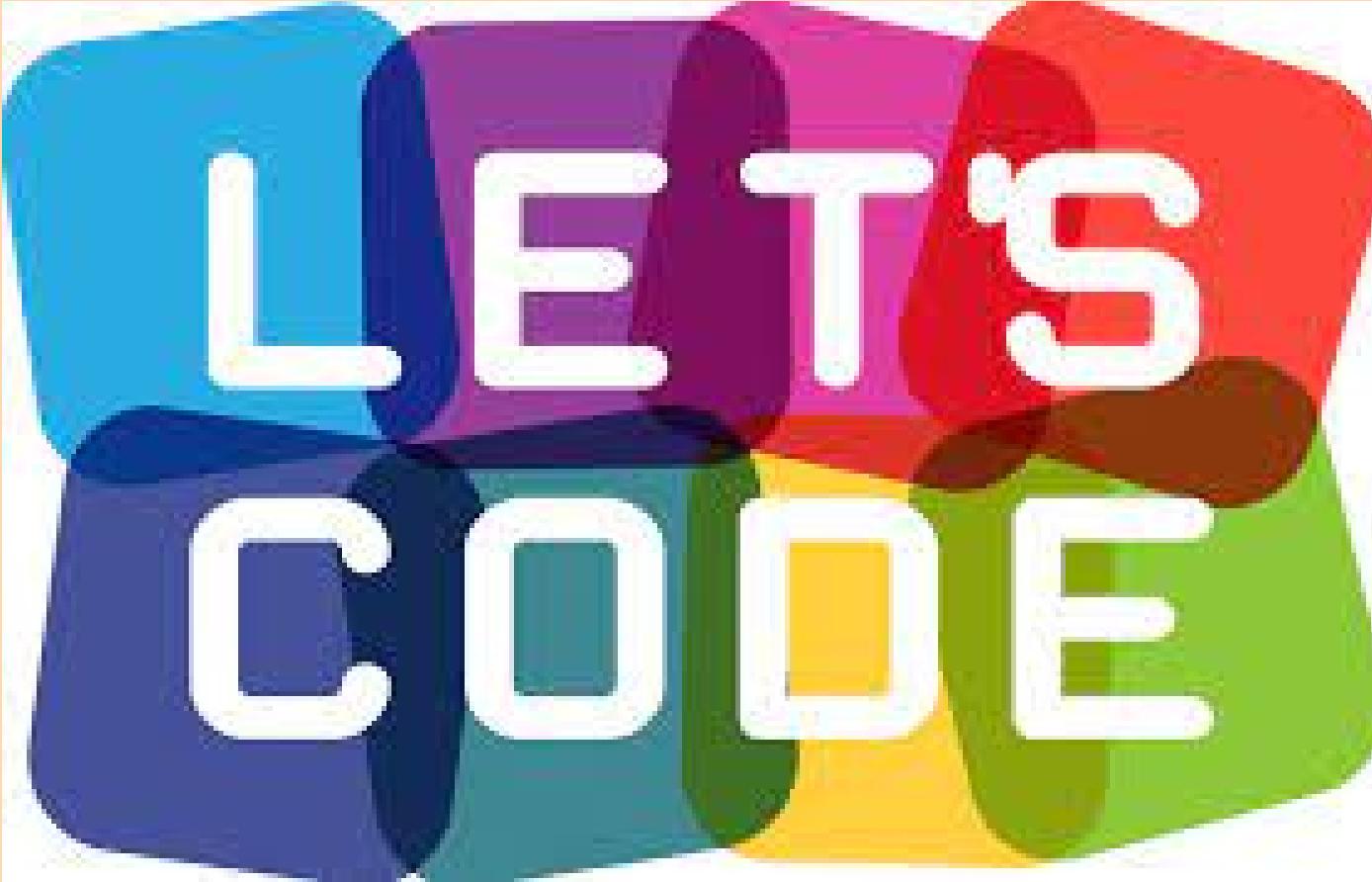
While (Do-While) Statement

break Statement

Overview

Loops (More Examples)

Multi-dimensional Arrays (多维数组)



game-of-life.c merge.c insertion-sort.c

■ 110 ■ / ■ 240 ■

Conway's Game of Life @ wiki

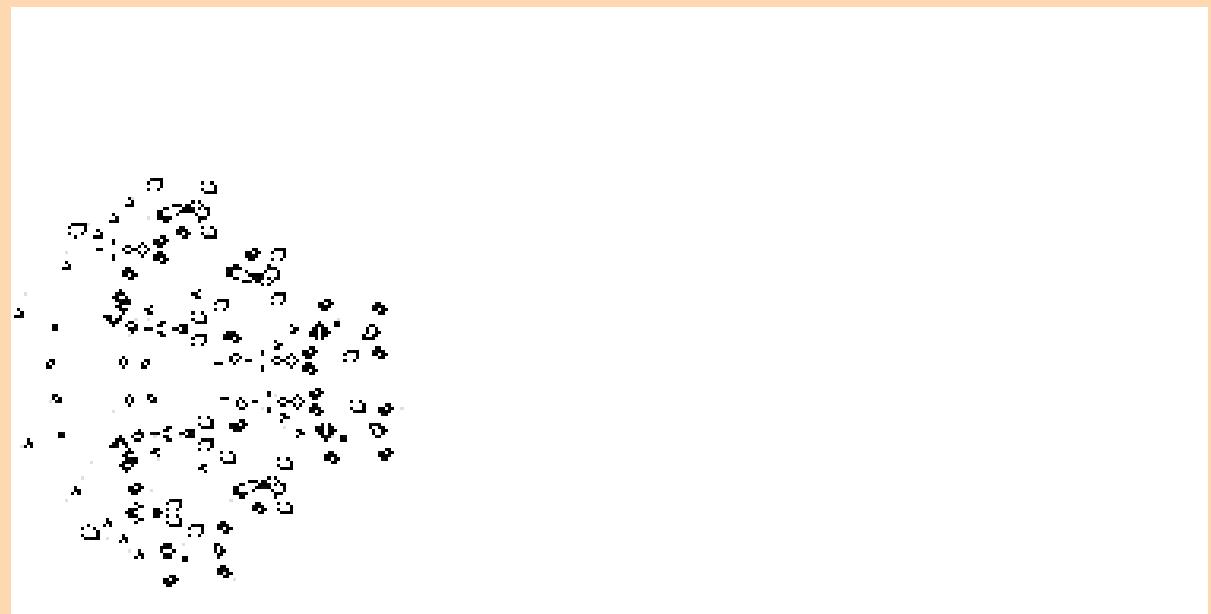
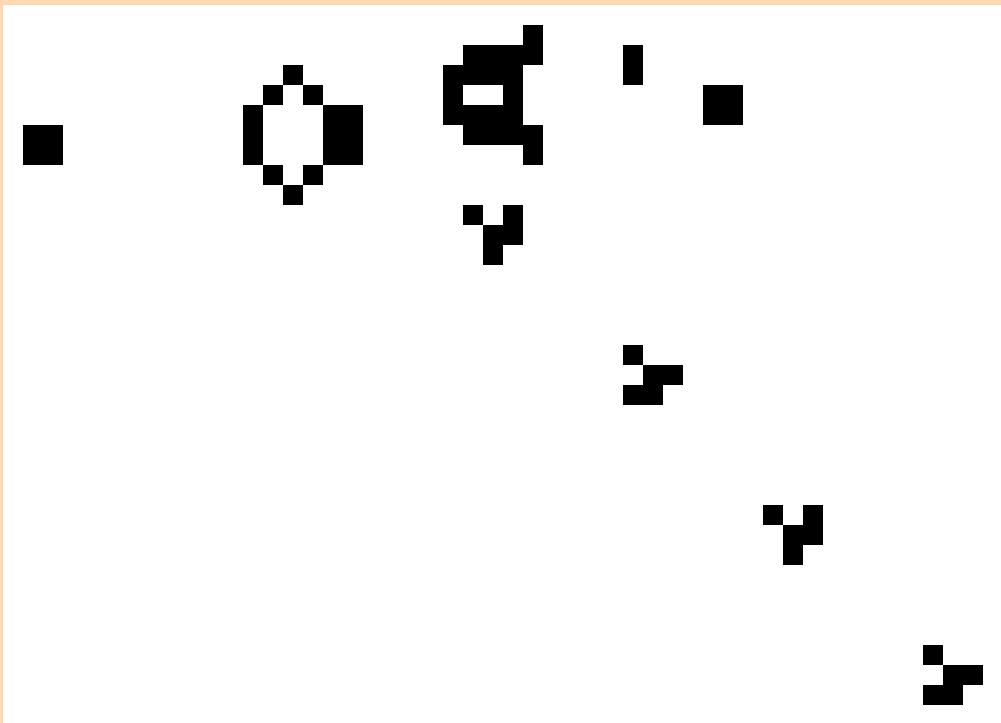


John Horton Conway (1937 ~ 2020)

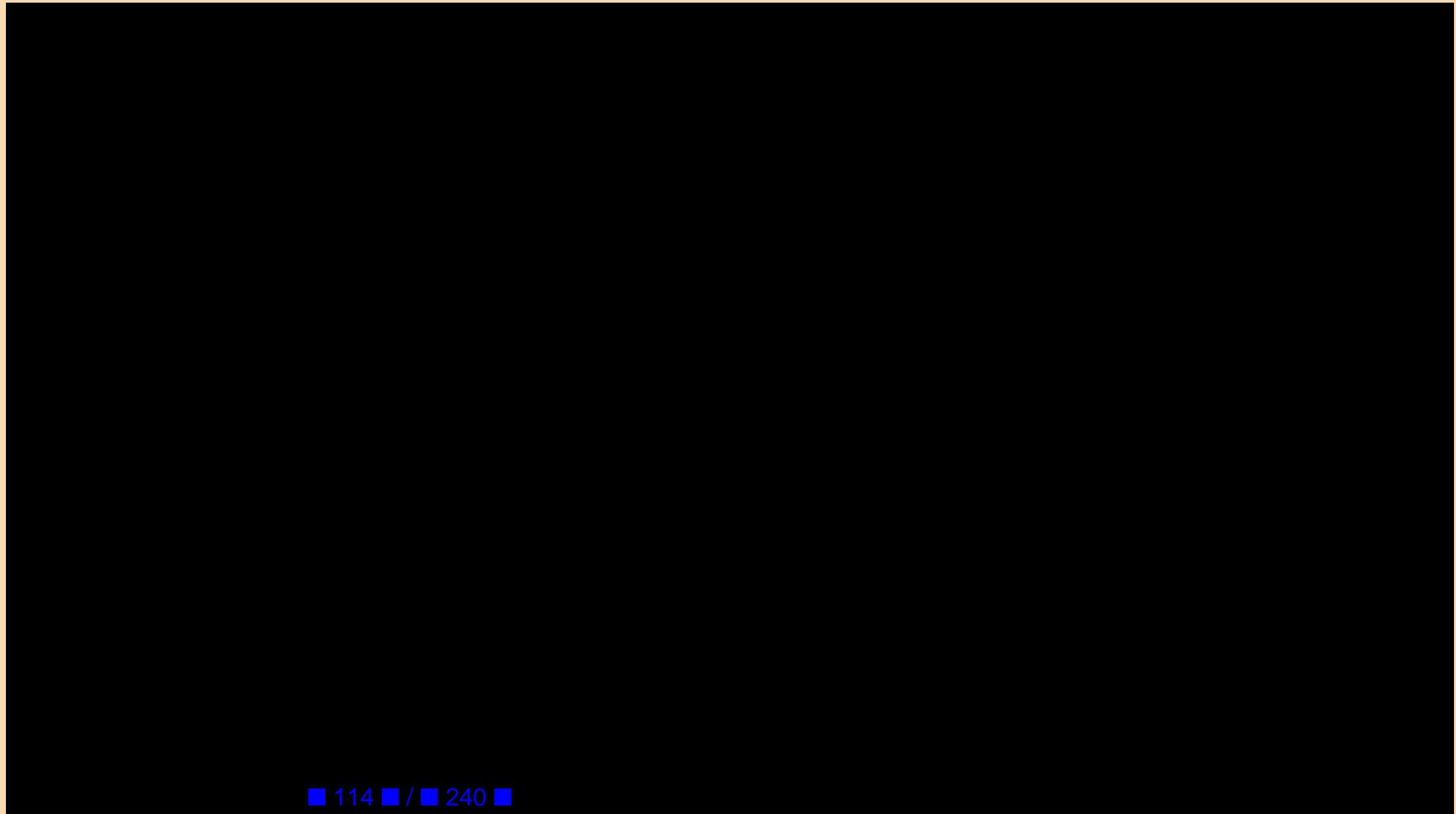
■ 11 ■ / ■ 240 ■

playgameoflife.com (Cellular Automata; 元胞自动机)

- Any **live** cell with two or three live neighbours survives.
- All other **live** cells die in the next generation.
- Any **dead** cell with three live neighbours becomes a live cell.
- All other **dead** cells stay dead.

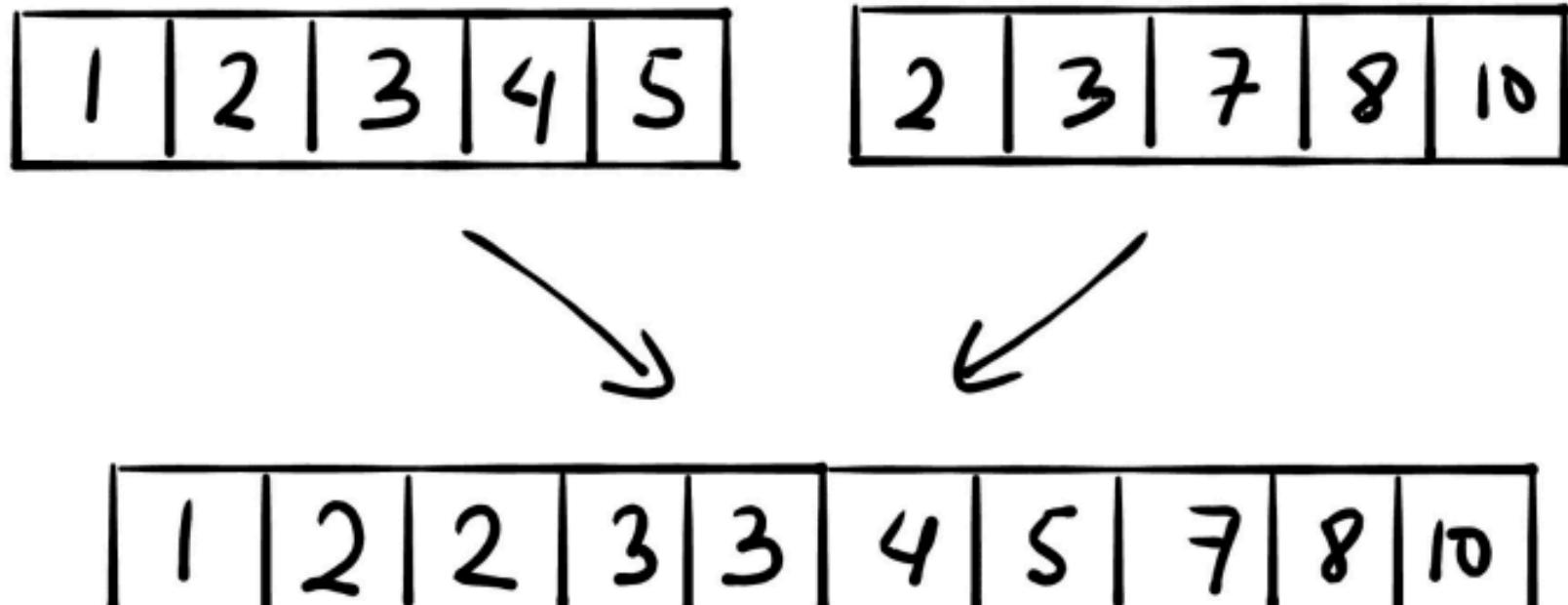


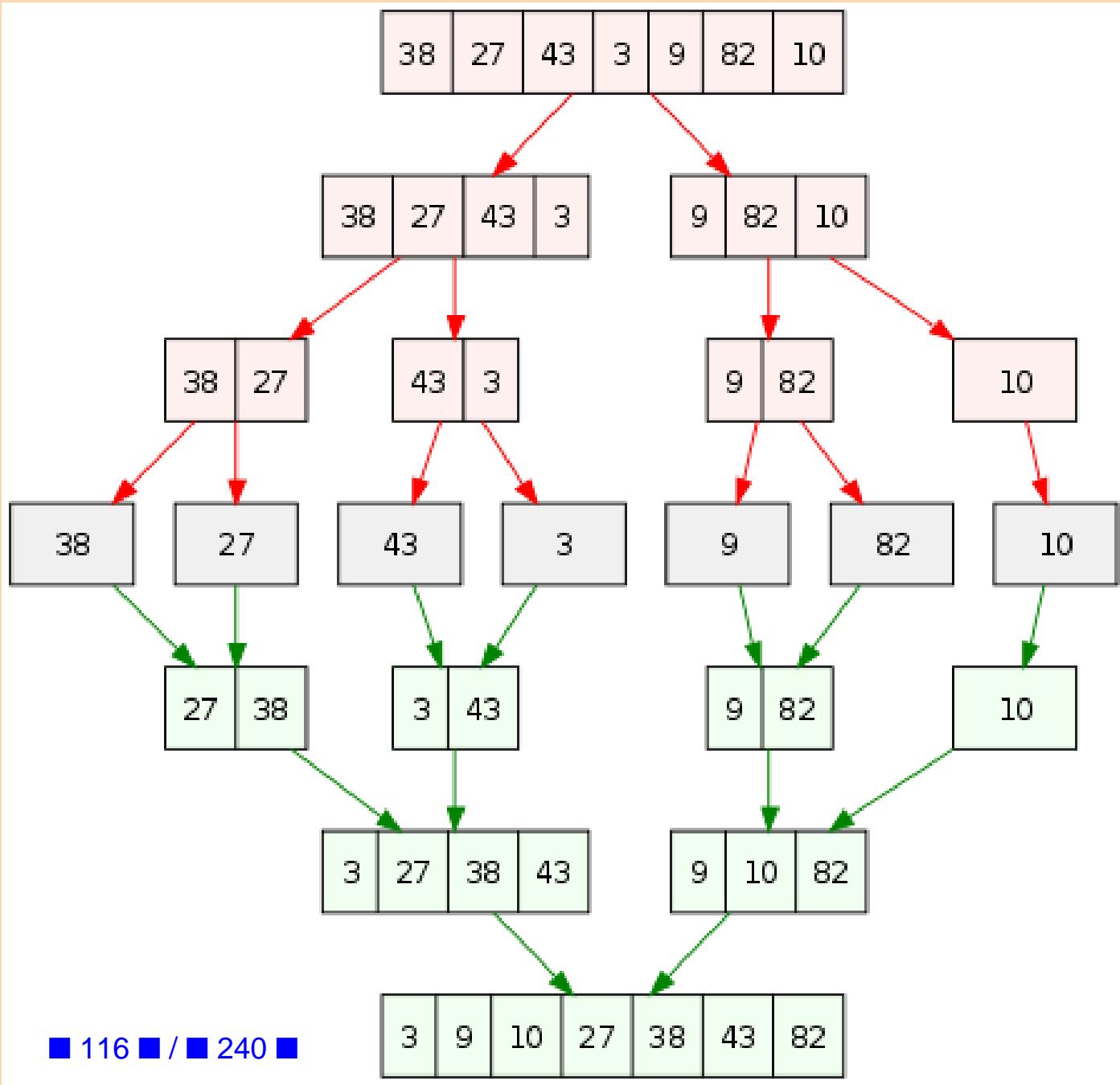
■ 113 ■ / ■ 240 ■



Merge (merge.c)

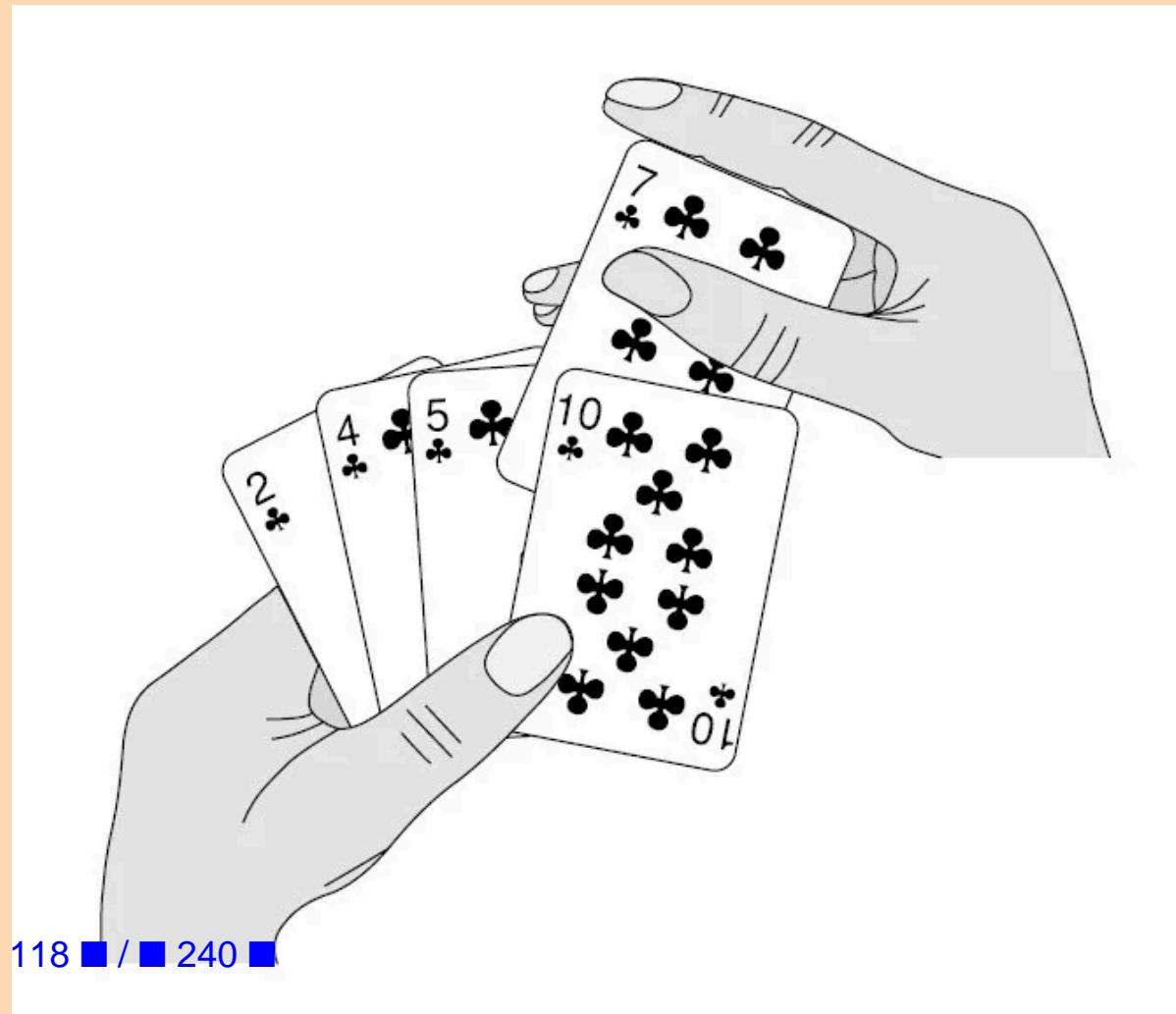
Merge Two Sorted Arrays





6 5 3 1 8 7 2 4

Insertion Sort



■ 118 ■ / ■ 240 ■

今日天气百一十度。打牌。

七月四日 (星二)

读 Plato's "Apology of Socrates"。

今日为美国独立纪念日，夜八时至湖上观此间庆祝会。士女来游者无算，公园中百戏俱陈，小儿女燃花炮为乐。既而焰火作矣，五光十色，备极精巧。九时半始归。

七月五日 (星三)

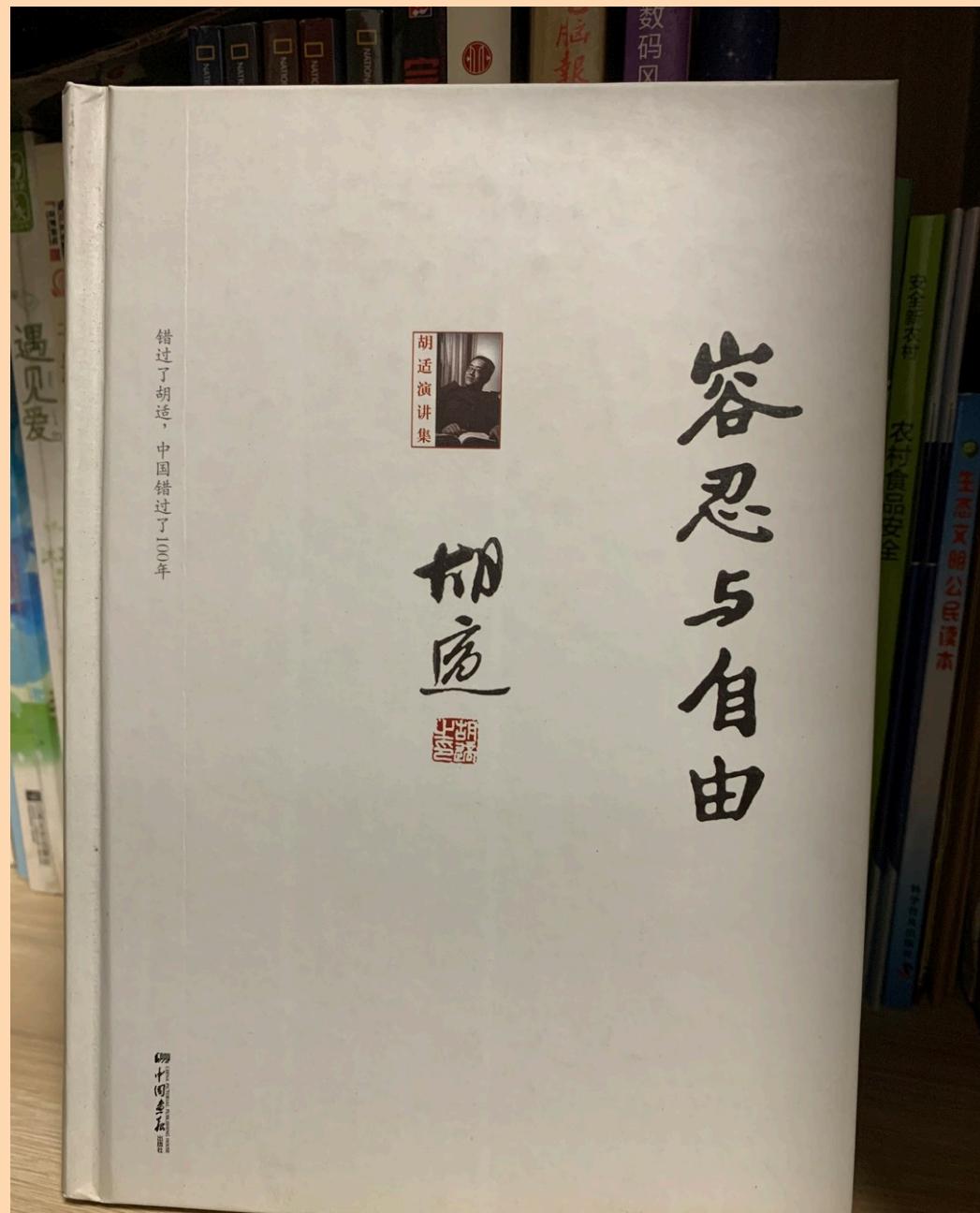
往暑期学校注册。下午打牌。

七月六日 (星四)

暑期学校第一日，化学 (八时至一时)。打牌。

七月七日 (星五)

上课。打牌。



6 5 3 1 8 7 2 4

Sorted partial result

Unsorted data

$\leq x$

$> x$

x

Sorted partial result

Unsorted data

$\leq x$

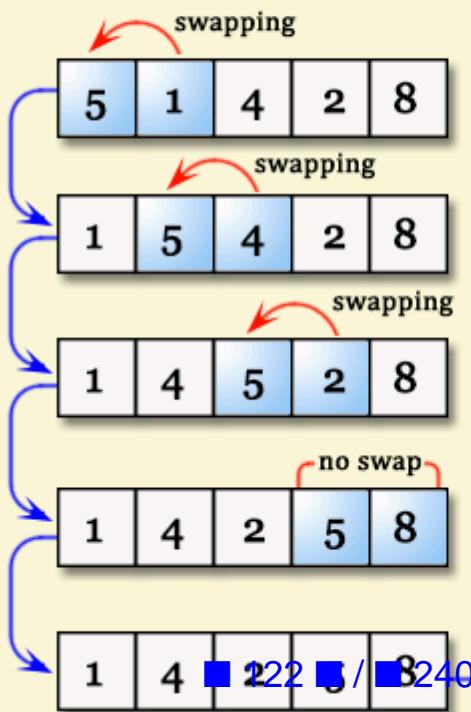
$> x$

x

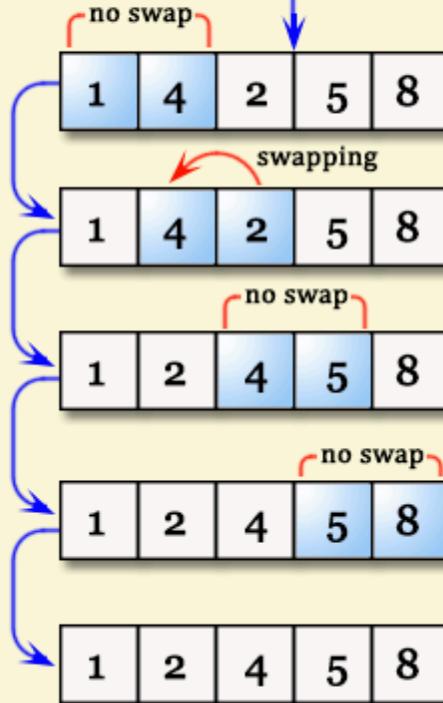
Bubble Sort

Bubble Sorting

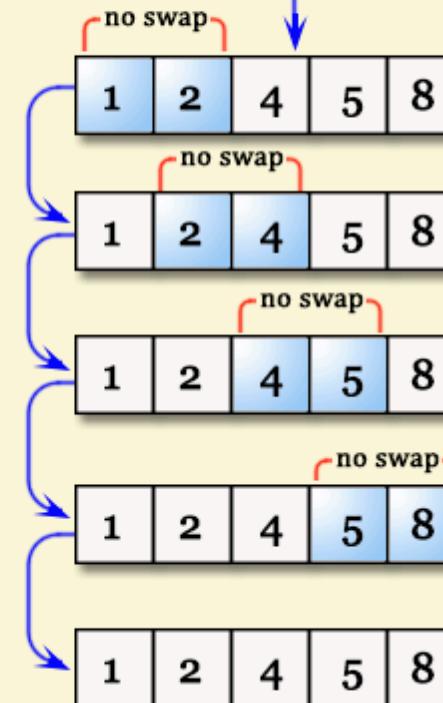
First Pass



Second Pass



Third Pass



6 5 3 1 8 7 2 4



■ 124 ■ / ■ 240 ■

5. FUNCTION

Hengfeng Wei(魏恒峰).
hfwei@nju.edu.cn



Review

Loops (More Examples)

Multidimensional Arrays (多维数组)

Overview

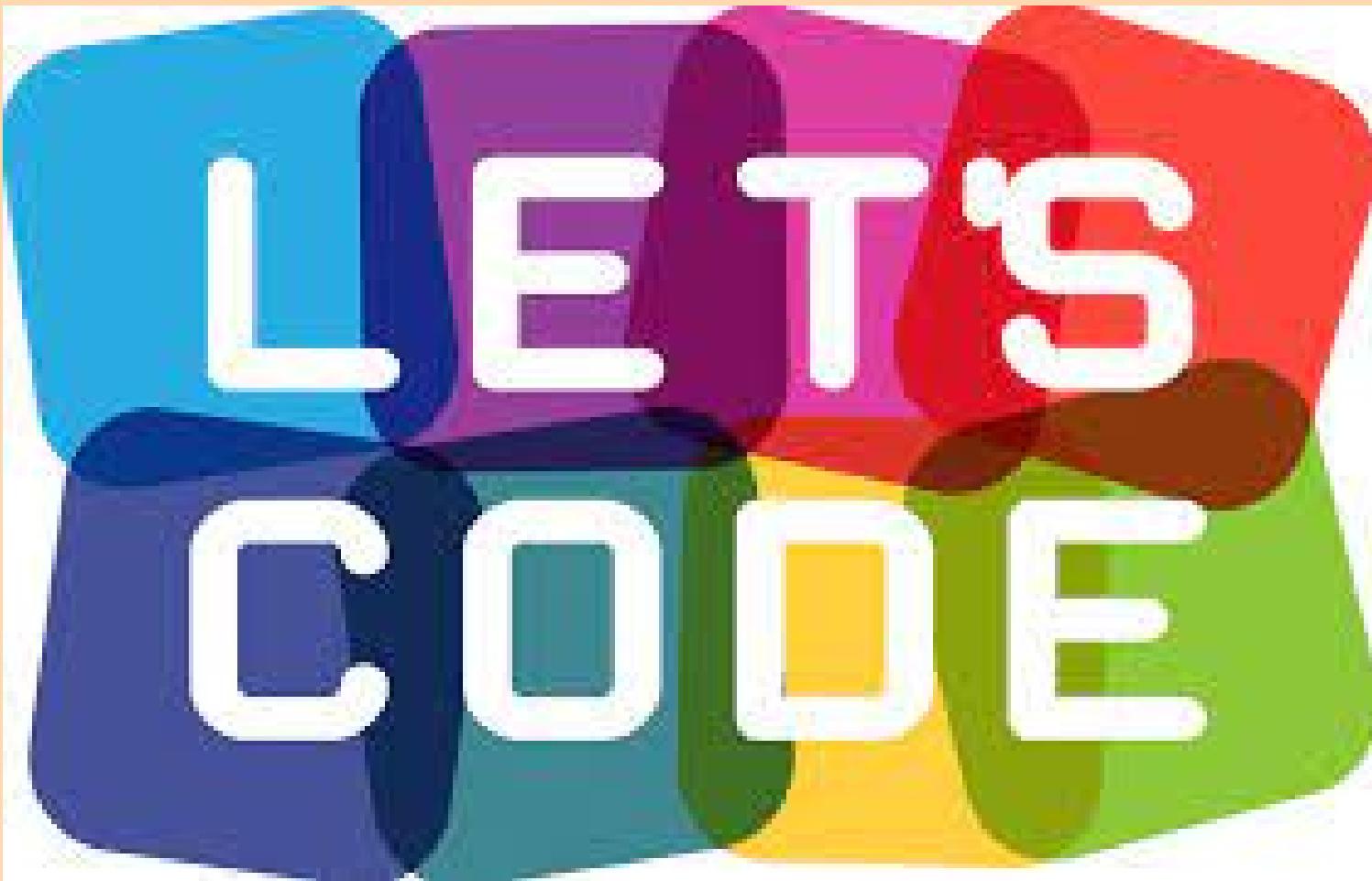
Functions

Function Definition

Function Declaration

Arrays as Parameters

Pass by Value





RE-WRITING PROGRAMS USING FUNCTIONS

129 / 240



■ 130 ■ / ■ 240 ■

6. DATA TYPES

Hengfeng Wei(魏恒峰)

hfwei@nju.edu.cn



Review

Functions

Function Definition

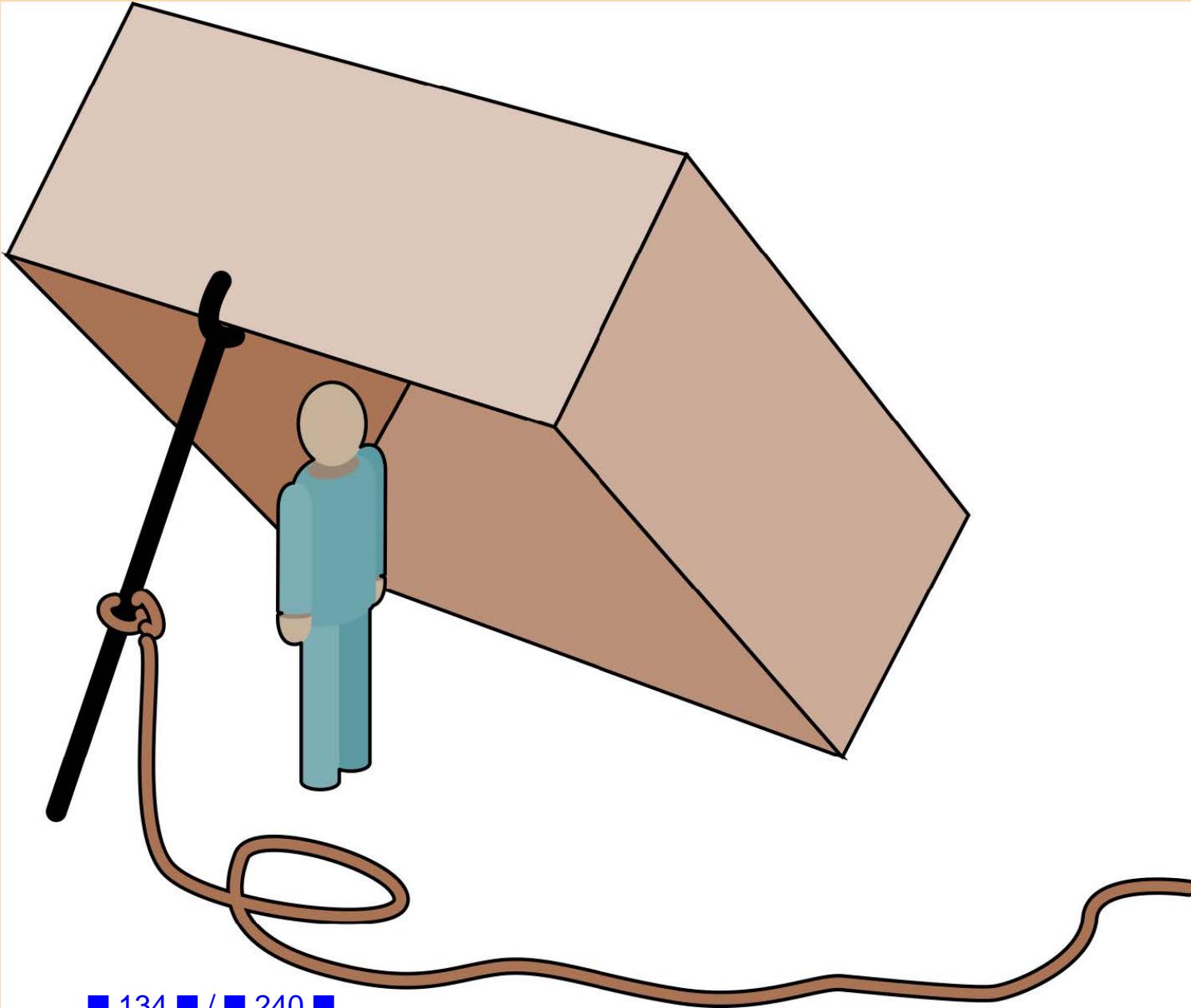
Function Declaration

Arrays as Parameters

Pass by Value

Overview

(Basic) Data Types



■ 134 ■ / ■ 240 ■

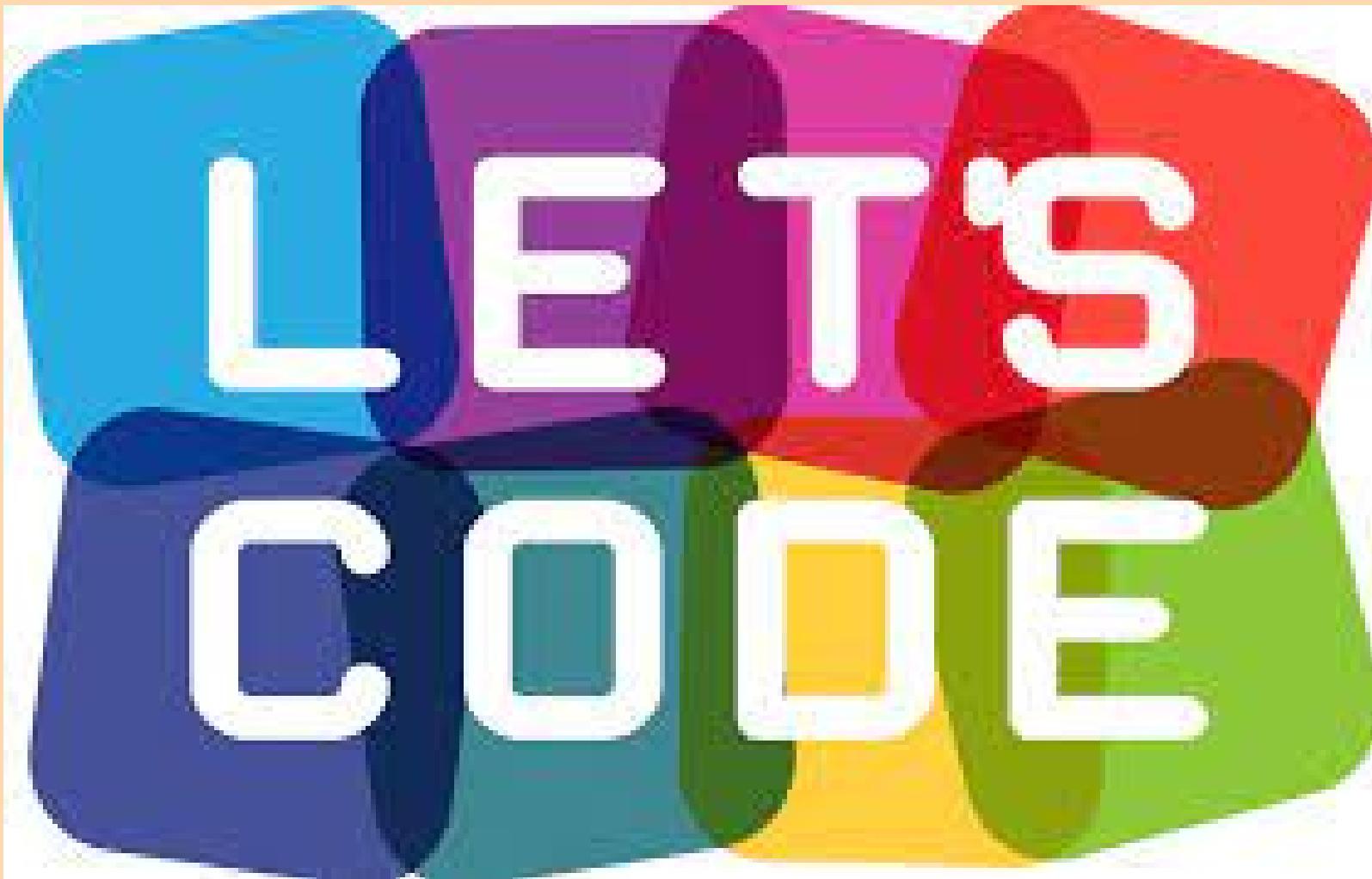
Two Major Reasons

Architectures May Vary

Finite vs. Infinite



■ 136 ■ / ■ 240 ■



Object

3.15

1 **object**

region of data storage in the execution environment, the contents of which can represent values

2 Note 1 to entry: When referenced, an object may be interpreted as having a particular type; see 6.3.2.1.

Object Types

■ 138 ■ / ■ 240 ■

Function Types

Data Types

The **type** of a variable determines

- the set of **values** it may take on and
- what **operations** can be performed on them.

int **char** **bool** **(_Bool)** **double**

[]

Integral Types (`size.c`)

- `(unsigned) short (int)`
- `unsigned int`
- `unsigned long (int)`
- `unsigned long long (int)`

When `sizeof` is applied to an operand that has type `char, unsigned char, or signed char`, (or a qualified version thereof) the result is 1. When applied to an operand that has array type, the result is the total number of bytes in the array.¹⁰⁵⁾ When applied to an operand that has structure or union type, the result is the total number of bytes in such an object, including internal and trailing padding.

The value of the `result of sizeof` operators is implementation-defined, and its type (an unsigned integer type) is `size_t`, defined in `<stddef.h>` (and other headers).

Integral Types

- (unsigned) short (int)
- unsigned int
- unsigned long (int)
- unsigned long long (int)

The *precision* of an *integer type* is the number of bits it uses to represent values, excluding any sign and padding bits. The *width* of an *integer type* is the same but including any sign bit; thus for unsigned integer types the two values are the same, while for signed integer types the width is one greater than the precision.

Integral Types (`int-limits.c`)

- (unsigned) short (int)
- unsigned int
- unsigned long (int)
- unsigned long long (int)

Integral Types ([exact-width.c](#))

int8_t int16_t int32_t int64_t

7.20.1.1 Exact-width integer types

The typedef name **intN_t** designates a signed integer type with width N , no padding bits, and a two's complement representation. Thus, **int8_t** denotes such a signed integer type with a width of exactly 8 bits.

The typedef name **uintN_t** designates an unsigned integer type with width N and no padding bits. Thus, **uint24_t** denotes such an unsigned integer type with a width of exactly 24 bits.

These types are optional. However, if an implementation provides integer types with widths of 8, 16, 32, or 64 bits, no padding bits, and (for the signed types) that have a two's complement representation, it shall define the corresponding typedef names.

Signed and Unsigned (`unsigned.c`)

Be careful when **MIXING** signed and unsigned types.

typedef

typedef unsigned __int64 size_t

#define __int64 long long

typedef long long time_t

char (char.c)

Use `char` only for representing characters.

Do NOT assume `signed char` or `unsigned char`.

Overflow

(**unsigned-wrap.c** **for-unsigned.c**
unsigned-wrap-fix.c)

无符号整数运算中没有溢出, 取而代之的是**回绕 (wrap)**现象

Overflow

(signed-overflow-fix.c)

有符号整数运算中发生溢出, 程序的行为是未定义的

Implicit Conversion

(implicit-conversion.c)

- 算术表达式、逻辑表达式 (先做整值提升)
- 定义初始化、赋值 (类型转换)
- 函数调用时 (类型转换)
- 函数返回时 (类型转换)

Be careful about narrowing conversions!!!

Implicit Conversion

[Integer promotions \(integer-promotion.c\)](#)

[Integer conversion rank \(Section 7.4.3\)](#)

[Usual arithmetic conversions \(Section 7.4.1\)](#)

Explicit Conversion

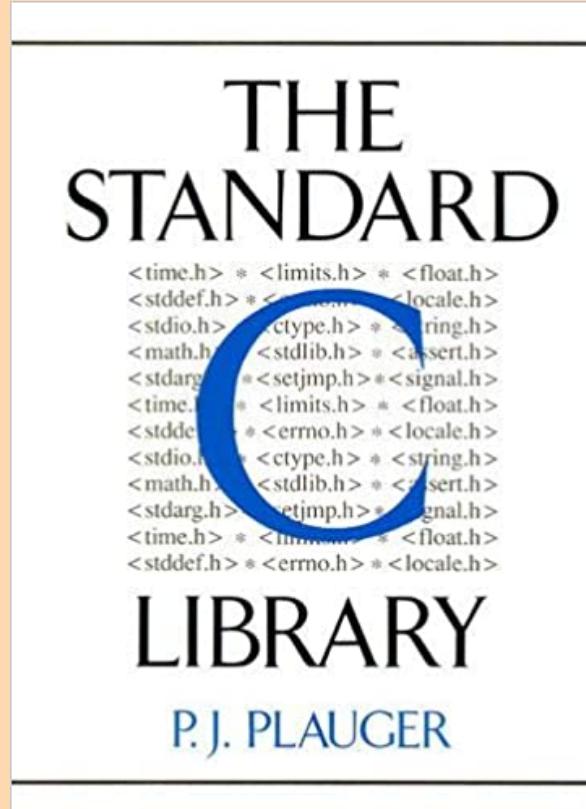
(**explicit-conversion.c**)

(type) expression

Floating-point Numbers

(**float-limits.c**)

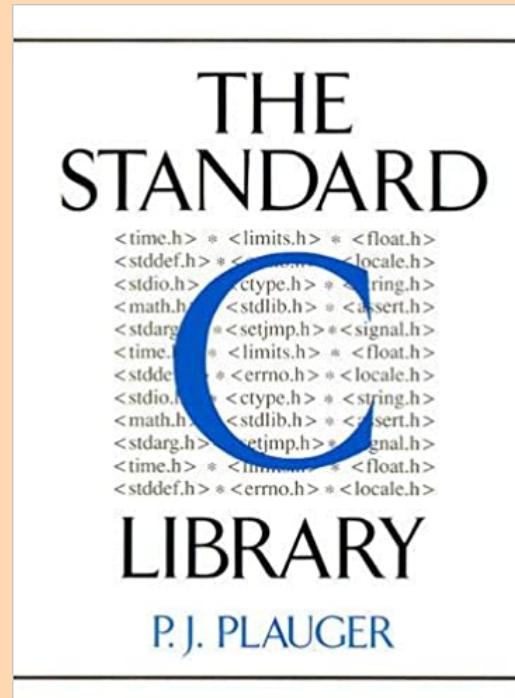
- float (F)
- double
- long double (L)



"Floating-point Arithmetic is Hard."

(Section 23.1 float.h)

"Many applications don't need floating-point arithmetic at all."



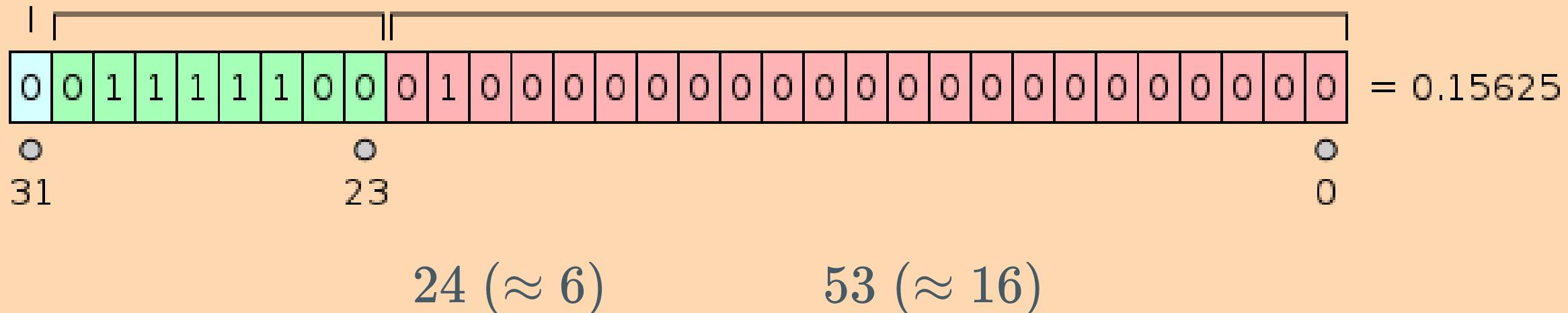
Use **math.h** (Section 23.3) whenever possible.
■ 155 ■ / ■ 240 ■

7.12 Mathematics <math.h>

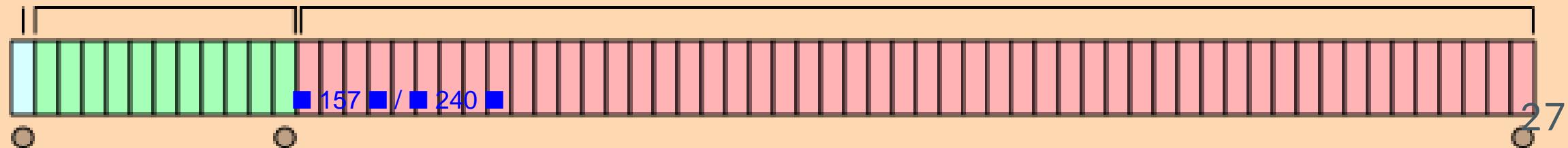
- 7.12.1 Treatment of error conditions
- 7.12.2 The FP_CONTRACT pragma
- 7.12.3 Classification macros
- 7.12.4 Trigonometric functions
- 7.12.5 Hyperbolic functions
- 7.12.6 Exponential and logarithmic functions
- 7.12.7 Power and absolute-value functions
- 7.12.8 Error and gamma functions
- 7.12.9 Nearest integer functions
- 7.12.10 Remainder functions
- 7.12.11 Manipulation functions
- 7.12.12 Maximum, minimum, and positive difference functions
- 7.12.13 Floating multiply-add
- 7.12.14 Comparison macros

IEEE 754

sign exponent (8-bit)



exponent
sign (11 bit)



5.2.4.2.2 Characteristics of floating types <float.h>

The characteristics of floating types are defined in terms of a model that describes a representation of floating-point numbers and values that provide information about an implementation's floating-point arithmetic.²¹⁾ The following parameters are used to define the model for each floating-point type:

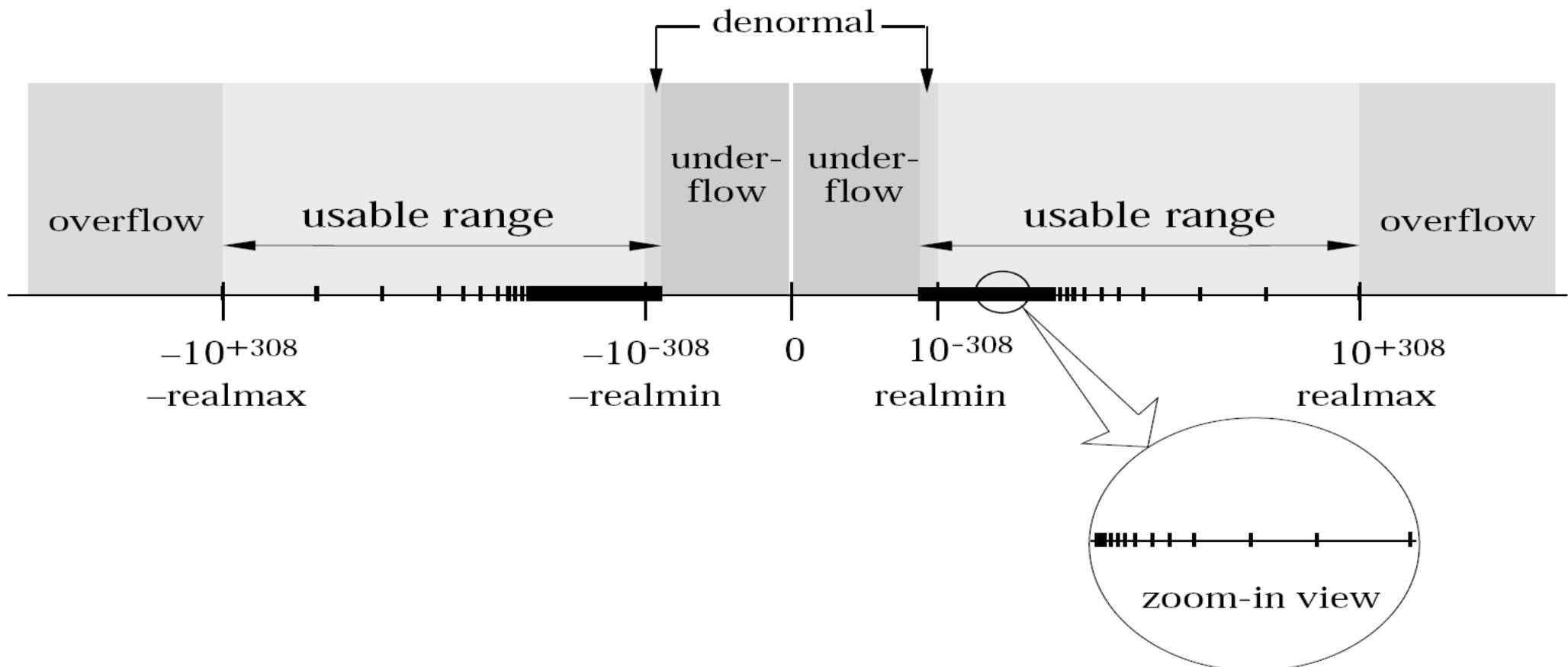
- s sign (± 1)
- b base or radix of exponent representation (an integer > 1)
- e exponent (an integer between a minimum e_{\min} and a maximum e_{\max})
- p precision (the number of base- b digits in the significand)
- f_k nonnegative integers less than b (the significand digits)

A *floating-point number* (x) is defined by the following model:

$$x = sb^e \sum_{k=1}^p f_k b^{-k}, \quad e_{\min} \leq e \leq e_{\max}$$

In addition to normalized floating-point numbers ($f_1 > 0$ if $x \neq 0$), floating types may be able to contain other kinds of floating-point numbers, such as subnormal floating-point numbers ($x \neq 0$, $e = e_{\min}$, $f_1 = 0$) and unnormalized floating-point numbers ($x \neq 0$, $e > e_{\min}$, $f_1 = 0$), and values that are not floating-point numbers, such as infinities and NaNs. A NaN is an encoding signifying Not-a-Number. A quiet NaN propagates through almost every arithmetic operation without raising a floating-point exception; a signaling NaN generally raises a floating-point exception when occurring as an arithmetic operation.²²⁾ ■ 240 ■

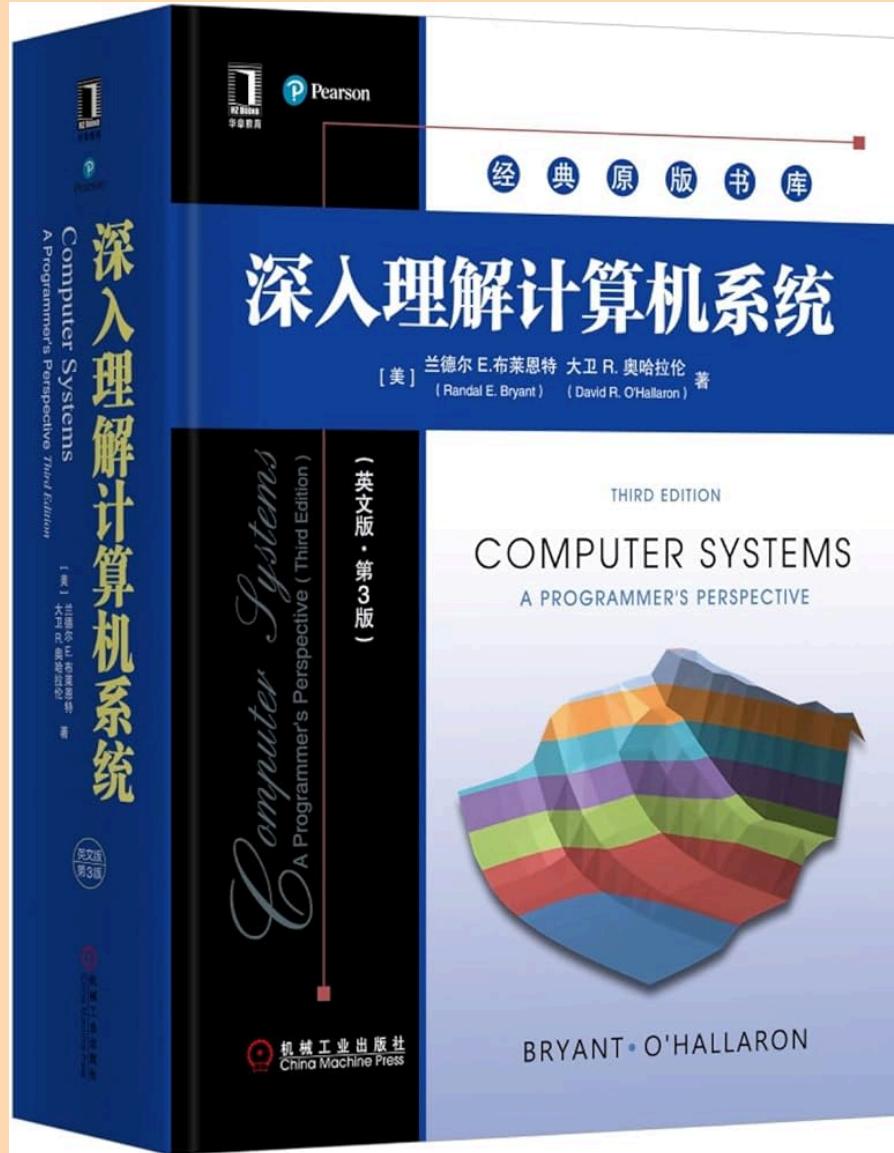
Floating Point Number Line



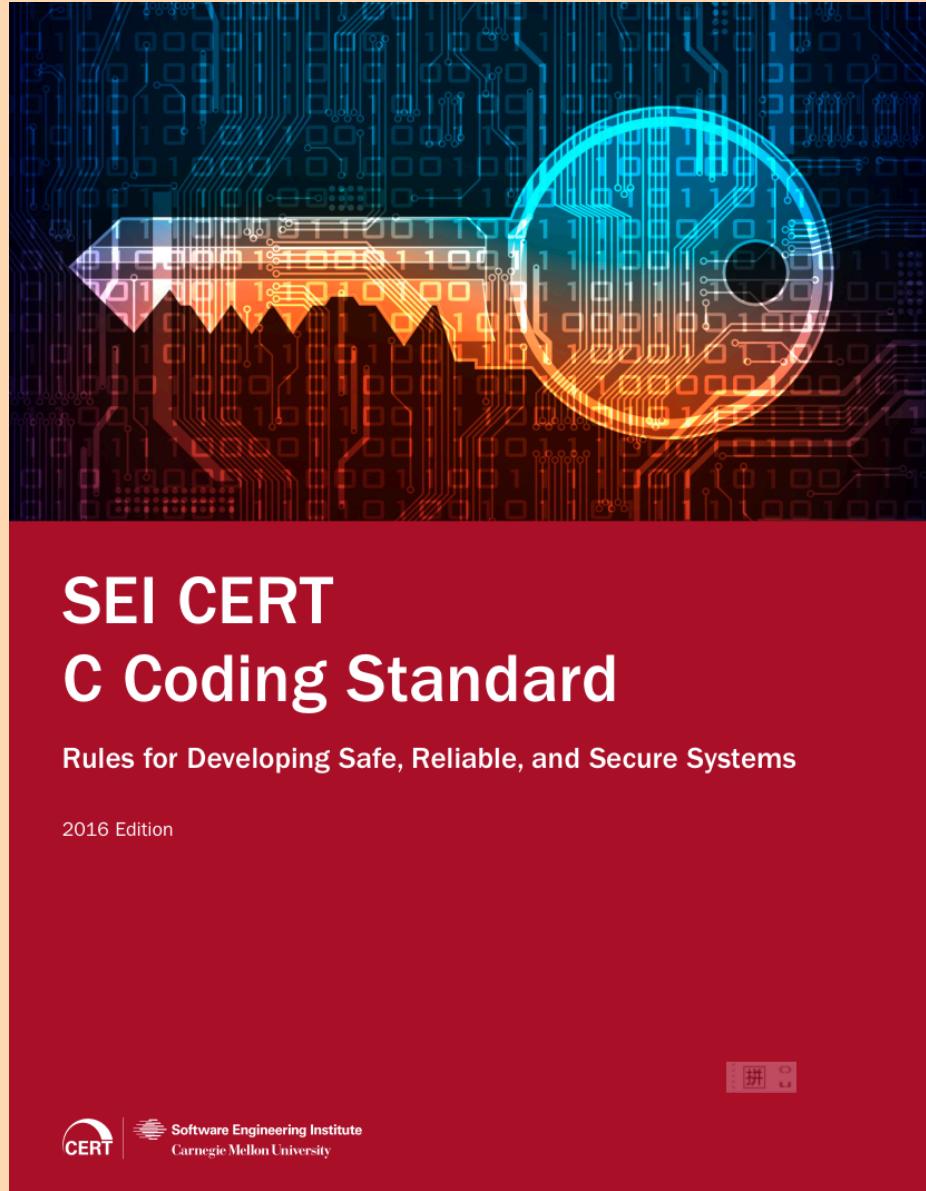
What is a subnormal floating point
number? @stackoverflow

implicit-conversion.c

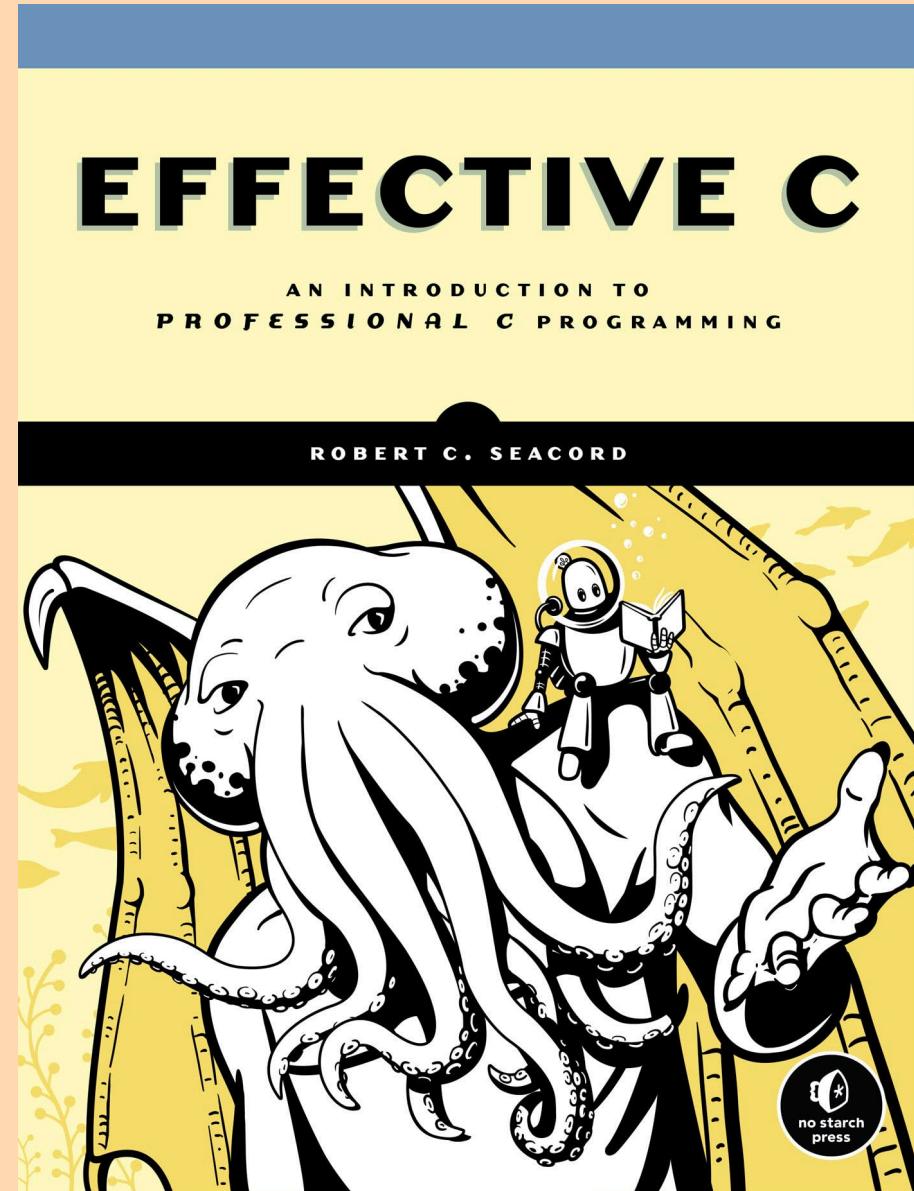
sum-product.c loop.c compare.c



■ 162 ■ / ■ 240 ■



■ 163 ■ / ■ 240 ■





■ 164 ■ / ■ 240 ■

7. POINTERS AND ARRAYS

Hengfeng Wei(魏恒峰)

hfwei@nju.edu.cn



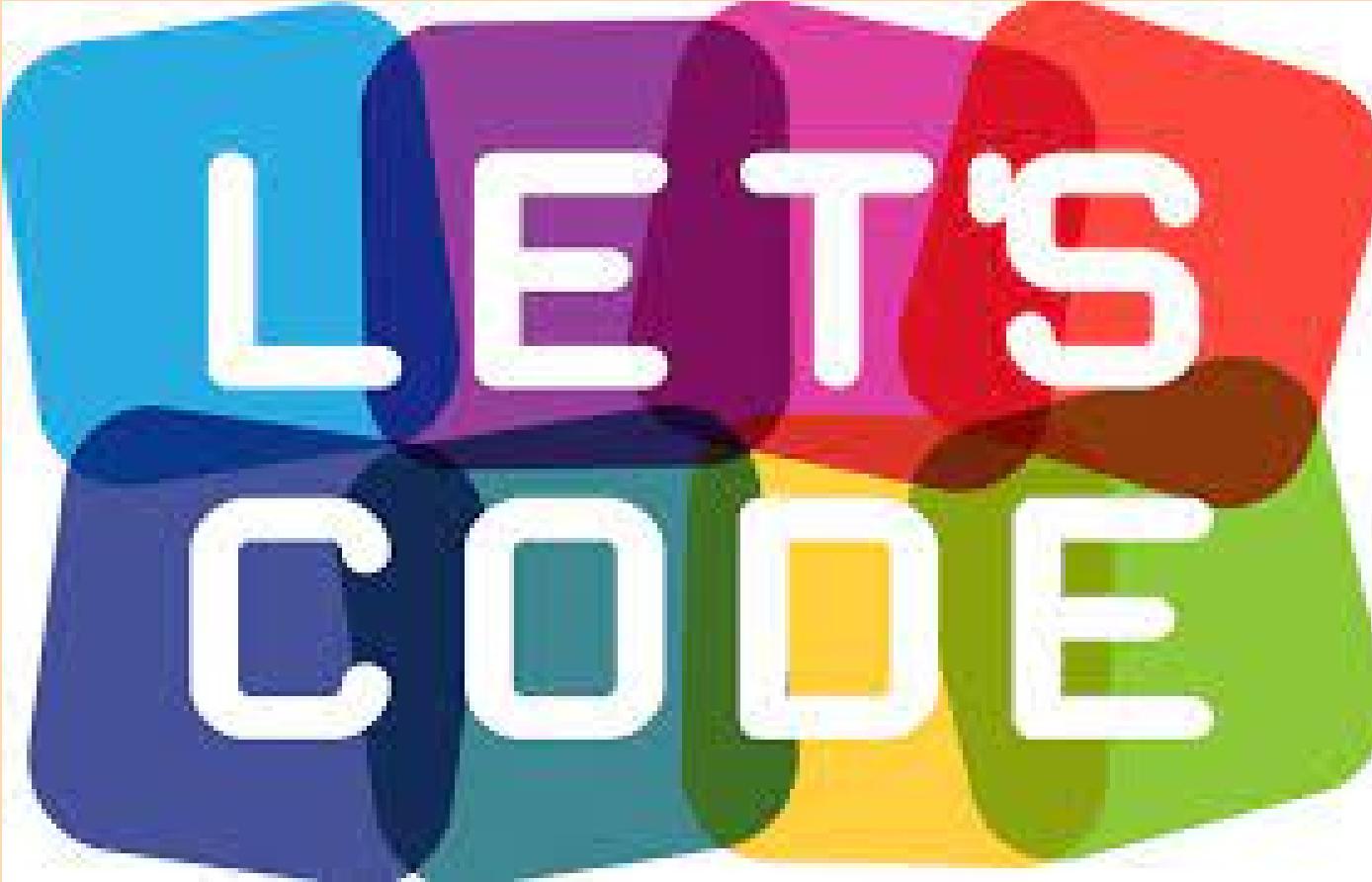
Overview



Pointers and Arrays (7 sentences = 4 + 3)

Dynamic Memory Management

• 66 • 7 • 240



[pointer.c](#)

[selection-sort.c](#)

[pointer-array.c](#)

[pointer-const.c](#)

Pointers provide an abstraction to memory.



■ 168 ■ / ■ 240 ■

Memory Address

Object

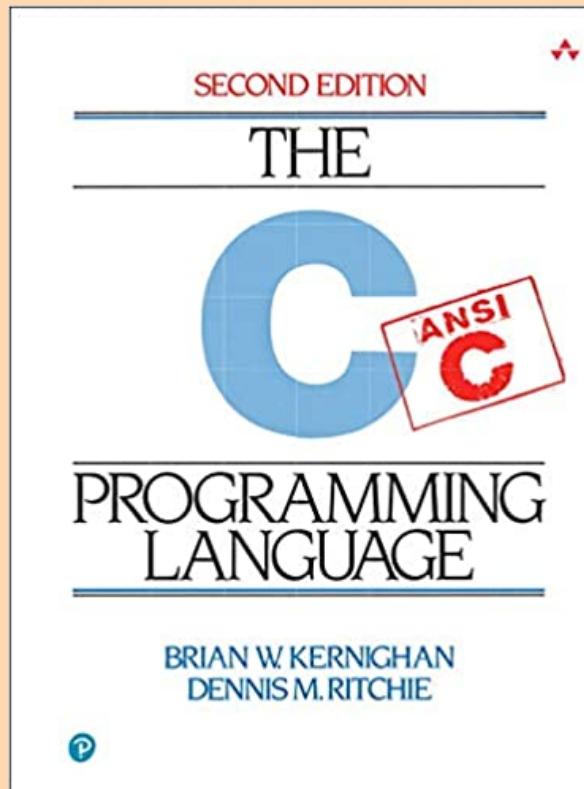
Variable (*identifier*)

Variables (pointer.c)

A **variable** has its *type*, *value*, and *address*.

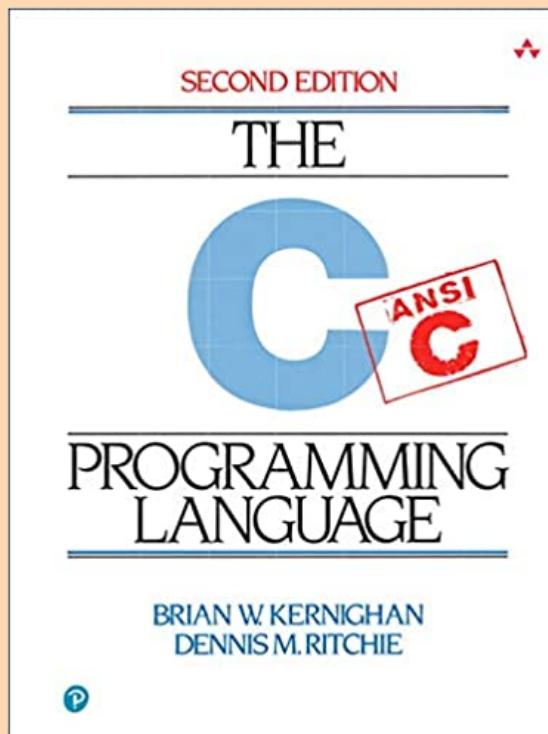
A **variable** can be used as a *lvalue* or a *rvalue*.

"A *pointer* is a *variable* that contains the *address* of a variable."

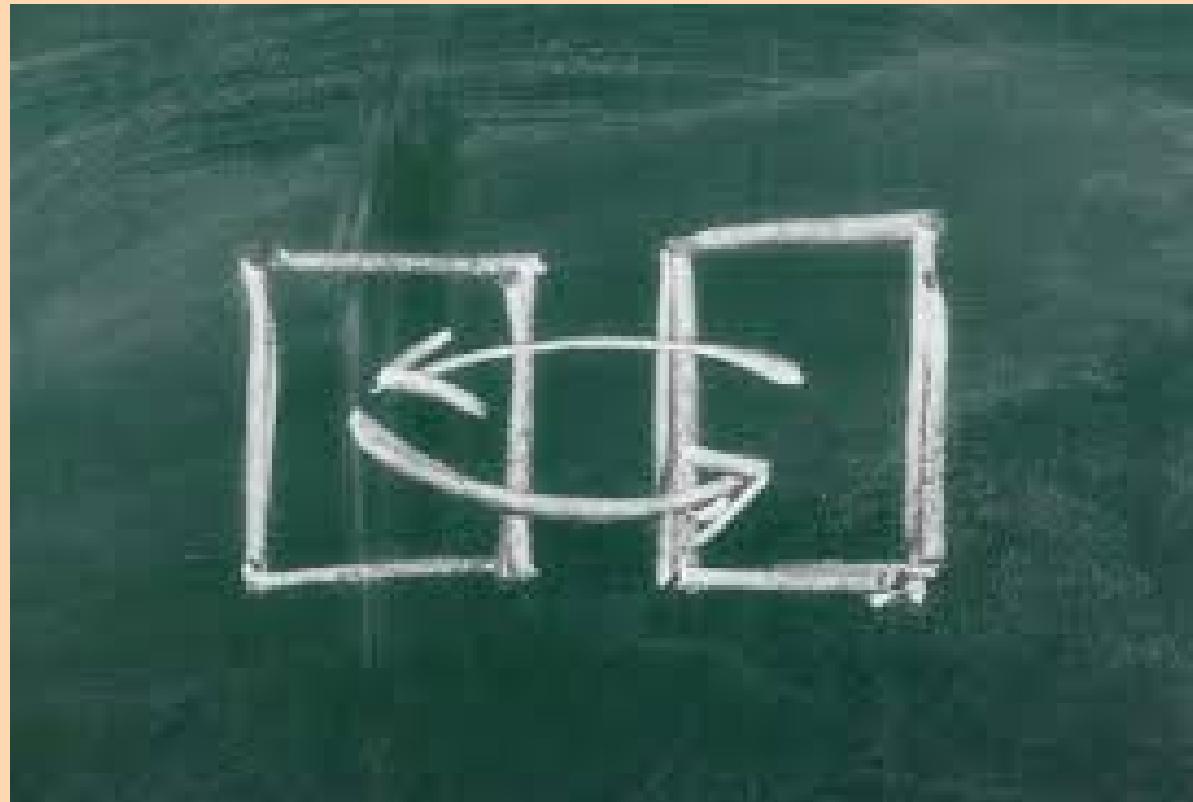


```
int *ptr = &var;
```

"**ptr* can occur in any context where *var* could"



Swap (selection-sort.c)



■ 173 ■ / ■ 240 ■

Pointers and Arrays ([selection-sort.c](#))

In *expressions*, the *name* of an array is a synonym
for the *address of its first element*.

Pointers and Arrays ([selection-sort.c](#))

arr[i] is an *lvalue*.

Pointers and Arrays (`selection-sort.c`)

But an *array name* is **NOT** a *variable*.

(*unmodifiable lvalue*)

Dynamic Memory Management

(selection-sort.c)

void *malloc(size_t size);

void free(void *ptr);



■ 178 ■ / ■ 240 ■

8. POINTERS AND C STRINGS

Hengfeng Wei(魏恒峰)

hfwei@nju.edu.cn



Review

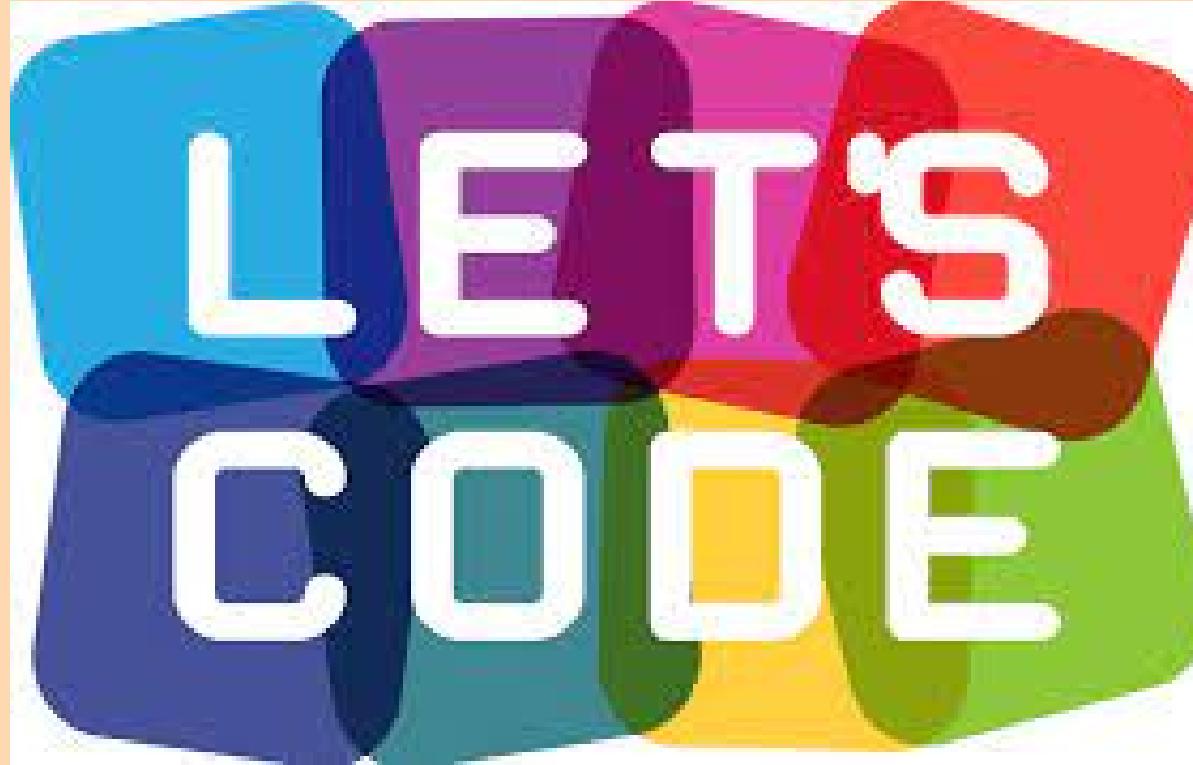


Pointers and Arrays ($7 = 4 + 3$)

Overview

Pointers and C Strings

Null-terminated byte strings (NTBS)



str-literals.c

strlen.c

strcmp.c

strcpy.c

strcat.c

■ 182 ■ / ■ 240 ■



■ 183 ■ / ■ 240 ■

9. DOUBLE POINTERS

Hengfeng Wei(魏恒峰)

hfwei@nju.edu.cn



Review

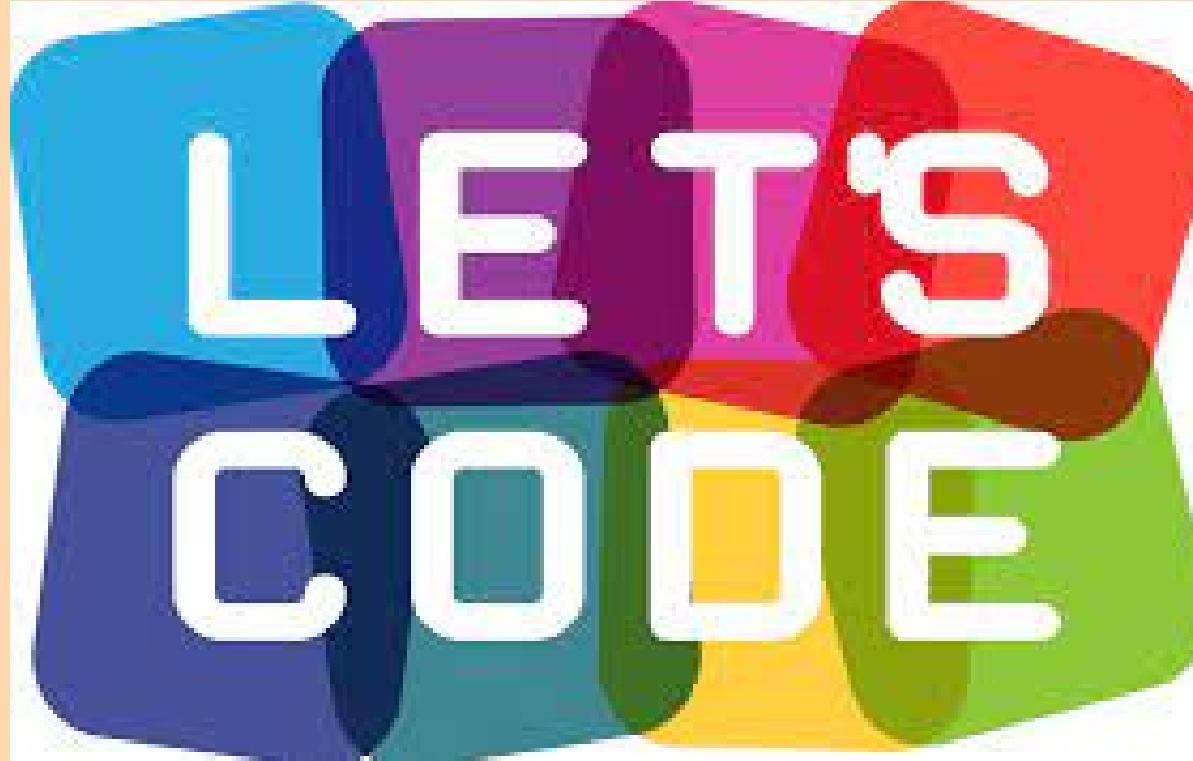
Pointers and C Strings

Overview

Pointer Arrays (`char *names[]`)

Pointers and 2D Arrays (`int (*scores)[]`)

Pointer Types Greatly Matter!!!



selection-sort-strings.c

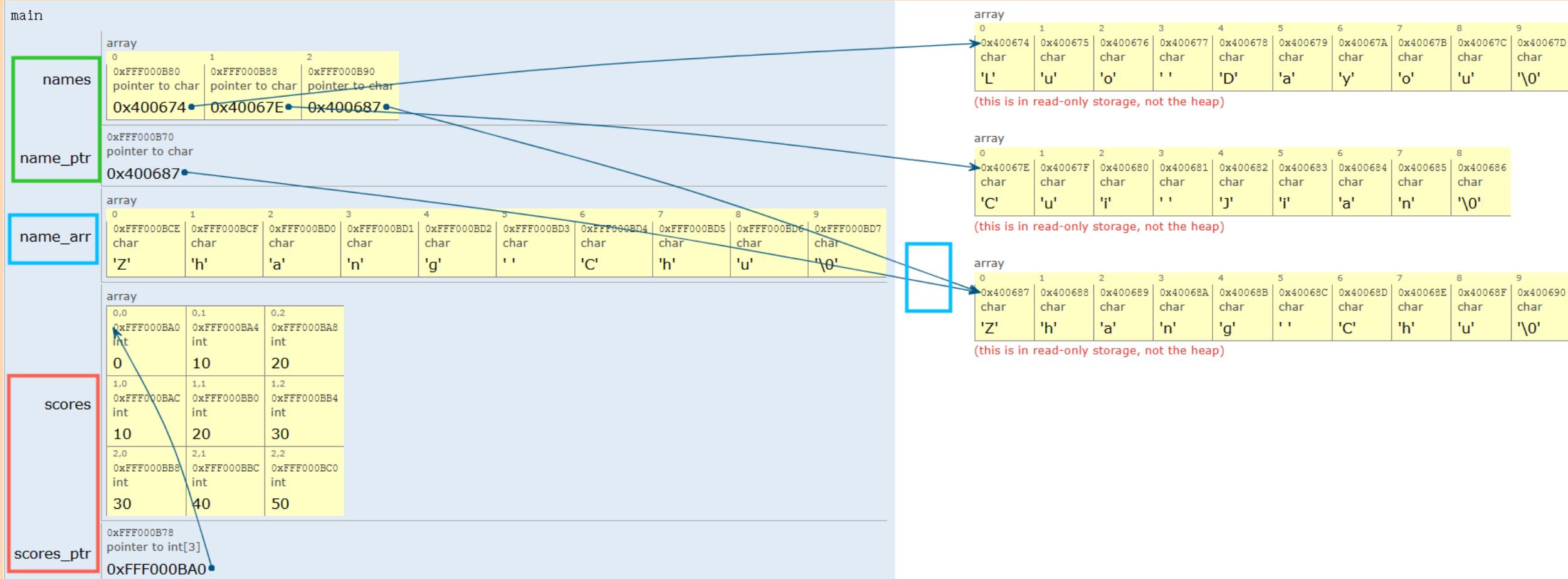
scores.c

echo.c

■ 188 ■ / ■ 240 ■



■ 189 ■ / ■ 240 ■





■ 191 ■ / ■ 240 ■

10. FUNCTION POINTERS

Hengfeng Wei(魏恒峰)

hfwei@nju.edu.cn



Review

Pointer Arrays (`char *musicians[]`)

Pointers and 2D Arrays (`int (*scores)[]`)

Overview

Function Pointers

Why Function Pointers?

3.15

1 object

region of data storage in the execution environment, the contents of which can represent values

2 Note 1 to entry: When referenced, an object may be interpreted as having a particular type; see 6.3.2.1.

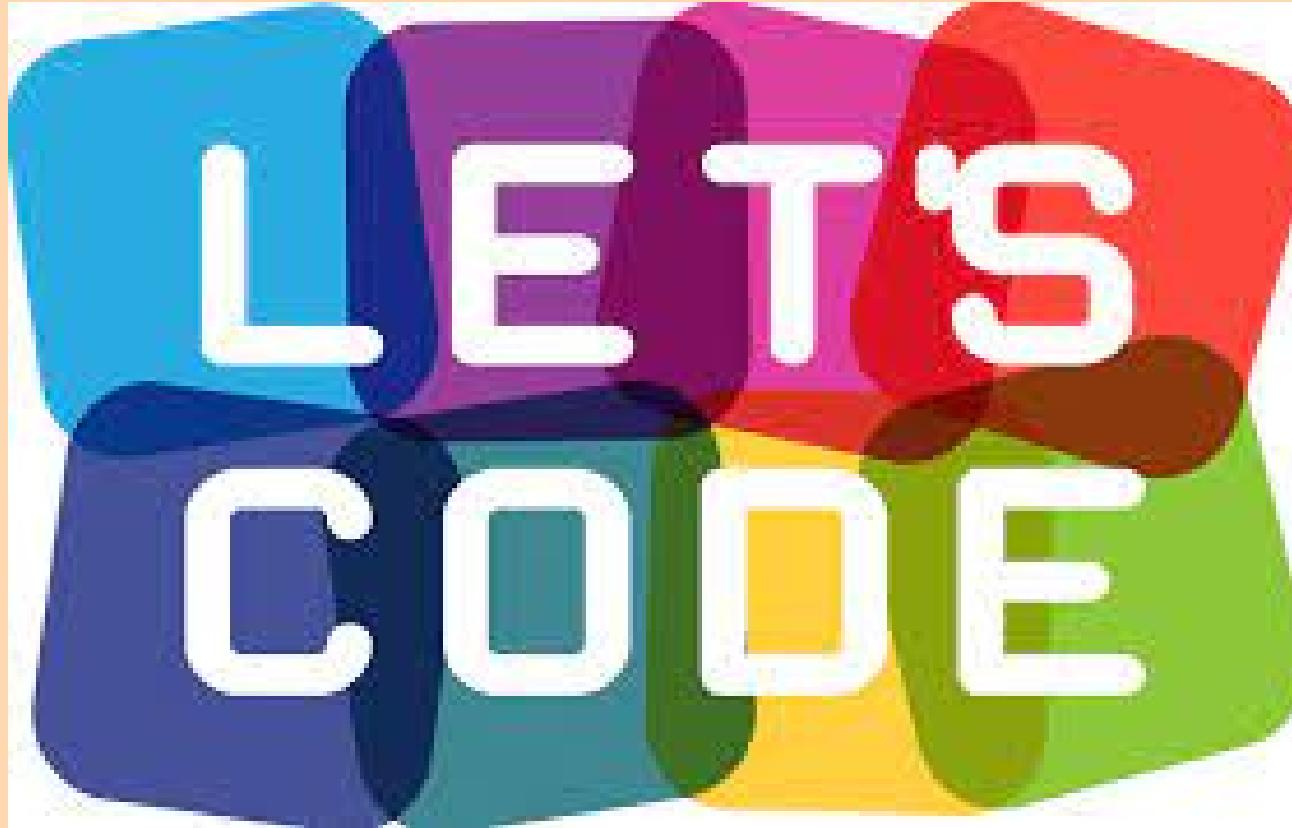
6.2.5 Types

1 The meaning of a value stored in an object or returned by a function is determined by the *type* of the expression used to access it. (An identifier declared to be an object is the simplest such expression; the type is specified in the declaration of the identifier.) Types are partitioned into object types (types that describe objects) and function types (types that describe functions). At various points within a

Why they say...

**Functions are
first class
citizens?**

args return vals assignment in array



integrate.c

sort.c

bsearch.c

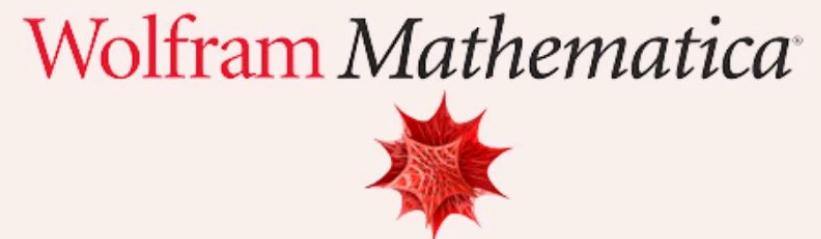
$$\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} f(x_i)\left(\frac{b-a}{n}\right)$$

$$a = x_0 < x_1 < \cdots x_i < \cdots < x_{n-1} < x_n = b$$

$$x_i = a + \frac{b-a}{n} \cdot i$$

It is an Abstraction!!!





■ 202 ■ / ■ 240 ■



啥啥啥 写的这是啥

decl.c

■ 203 ■ / ■ 240 ■



■ 204 ■ / ■ 240 ■

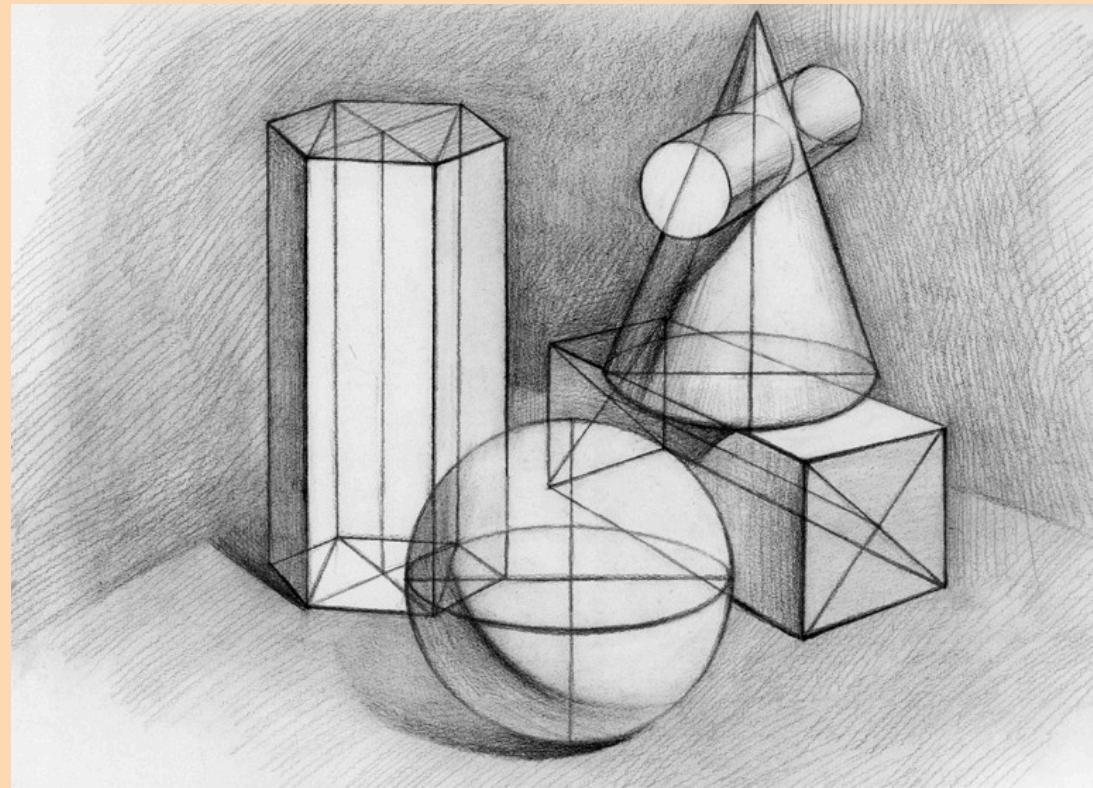
11. STRUCT

Hengfeng Wei(魏恒峰)

hfwei@nju.edu.cn



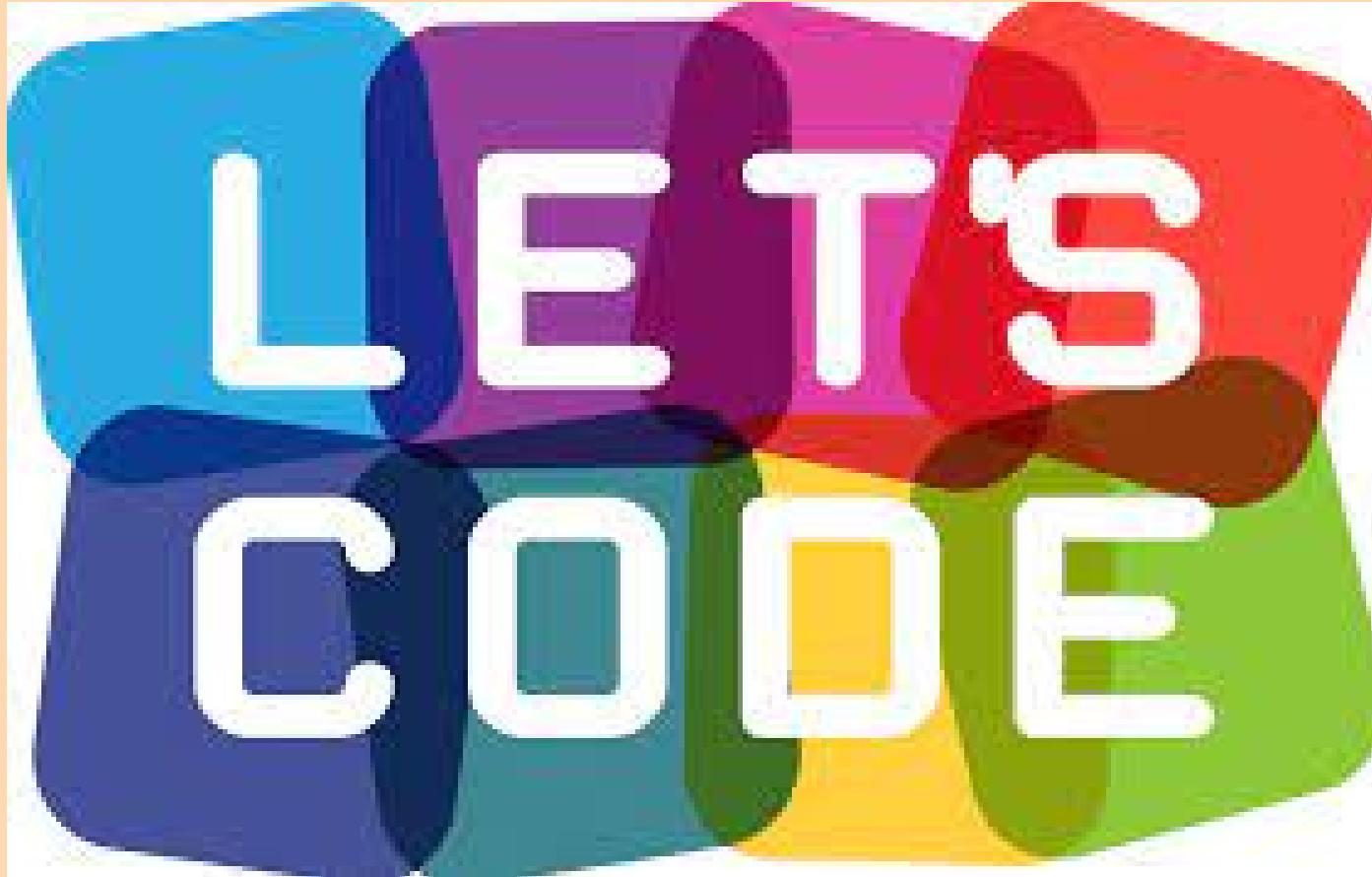
Overview



.Struct
■ 206 ■ / ■ 240 ■

Union

Enum



musician.c

offset.c

sds.c

■ 207 ■ / ■ 240 ■

南京大学摇滚联盟

Nanjing University Rock Republic





■ 209 ■ / ■ 240 ■



For 罗大佑 only:

- 每组信息占一行
- 各项信息使用 \t 间隔
- 各项信息遵循特定格式要求



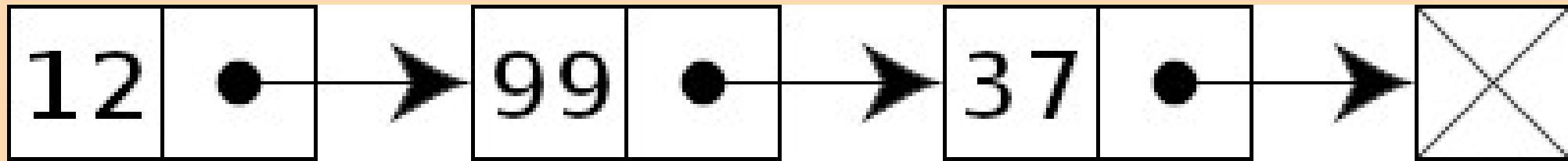
■ 211 ■ / ■ 240 ■

12. LINKED LISTS

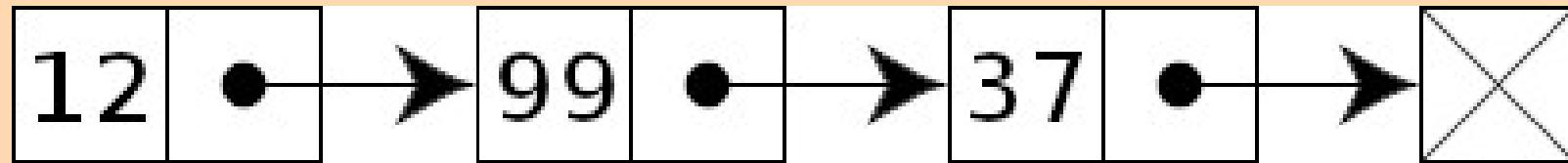
Hengfeng Wei(魏恒峰).
hfwei@nju.edu.cn



Overview



Linked List



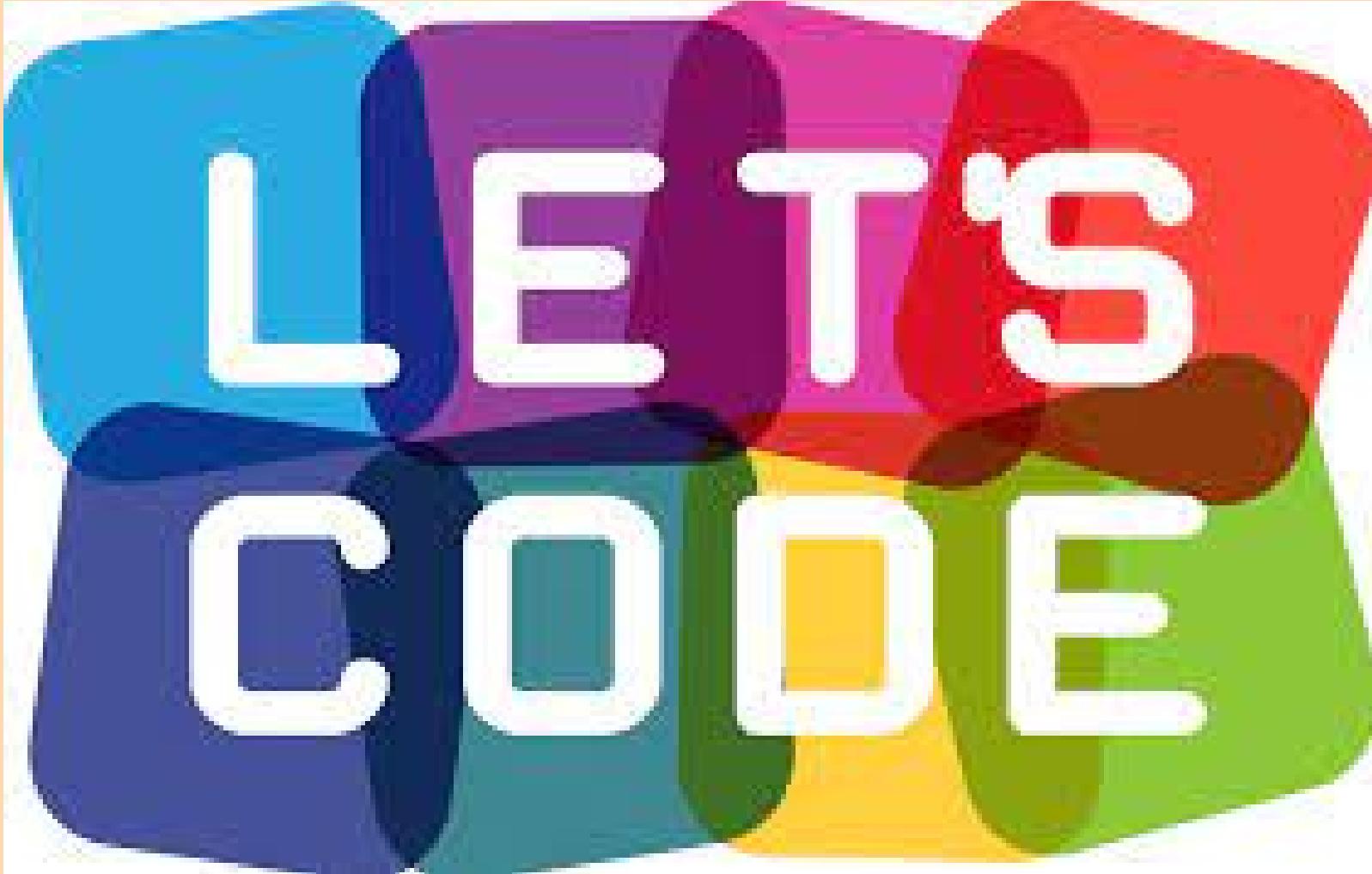
Singly Linked List



Doubly Linked List

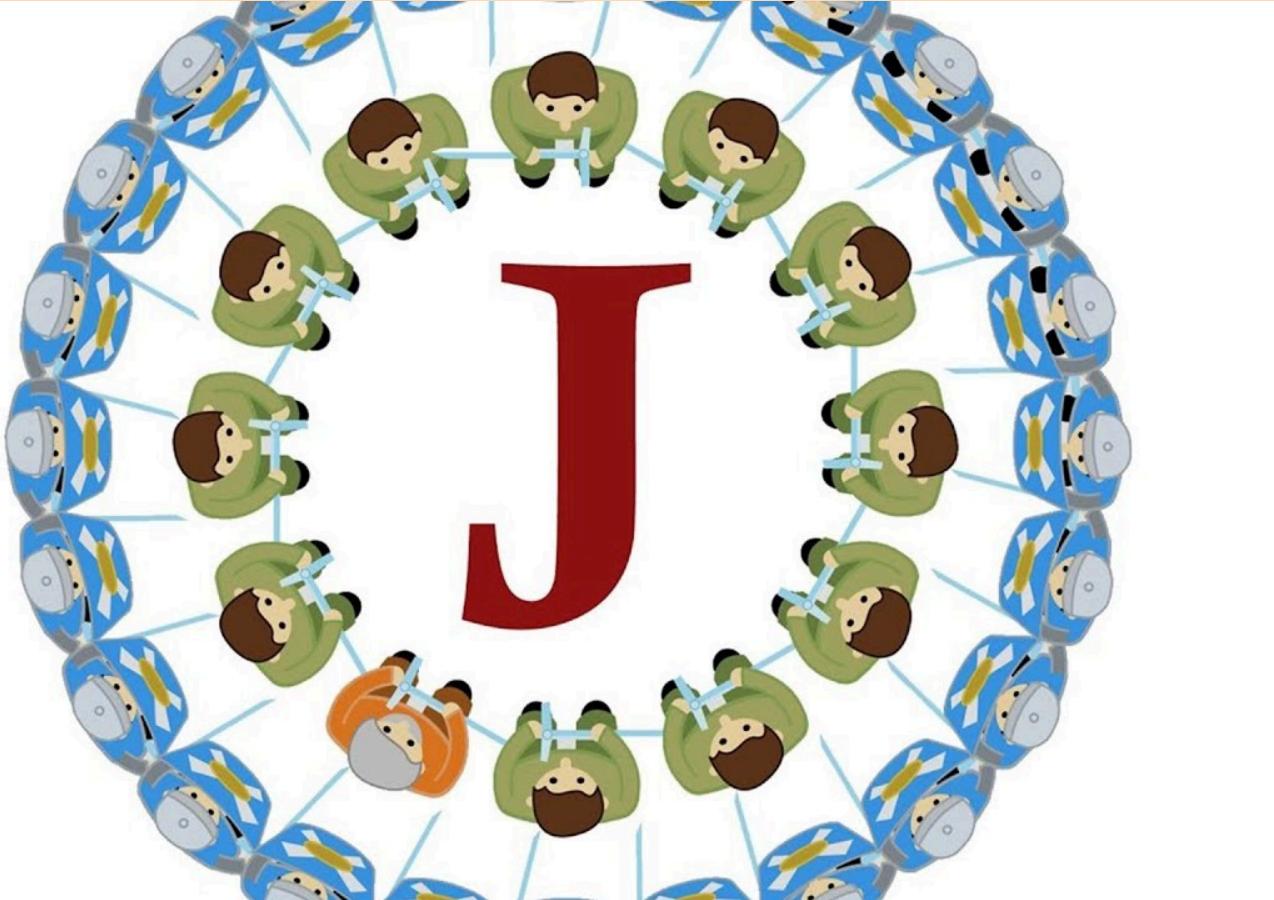


Circular (Singly) Linked List



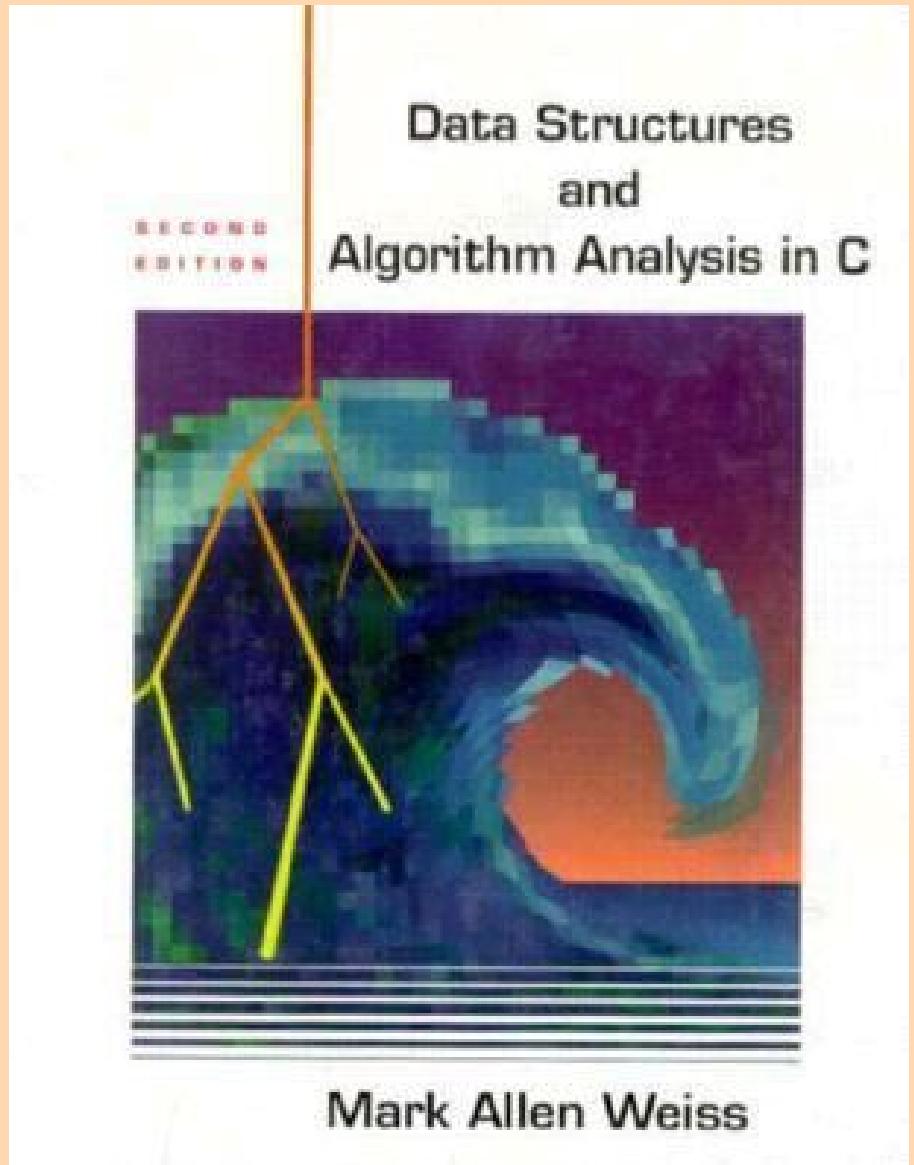
■ 217 ■ / ■ 240 ■

josephus.c

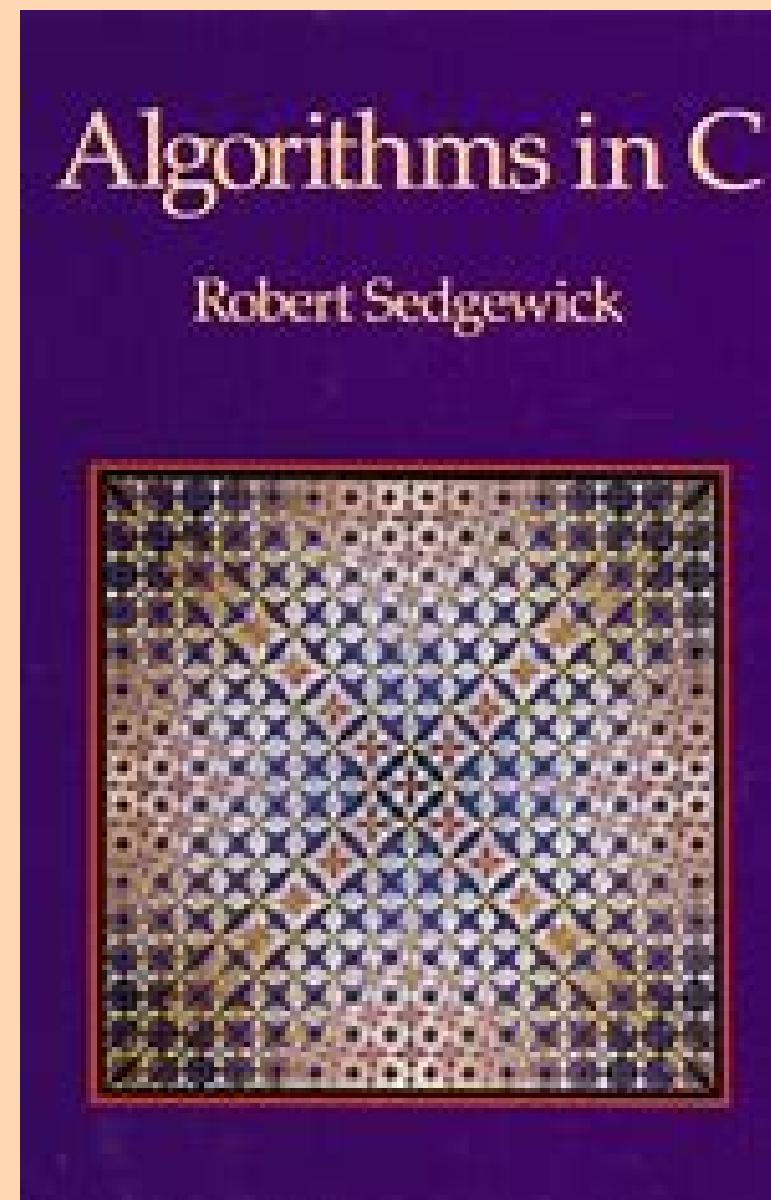


"I hate the Josephus Game!"

$$J(2^m + l) = 2l + 1 \quad (m \geq 0 \wedge 0 \leq l < 2^m)$$



■ 220 ■ / ■ 240 ■





■ 221 ■ / ■ 240 ■

13. RECURSION

Hengfeng Wei(魏恒峰).
hfwei@nju.edu.cn



Overview

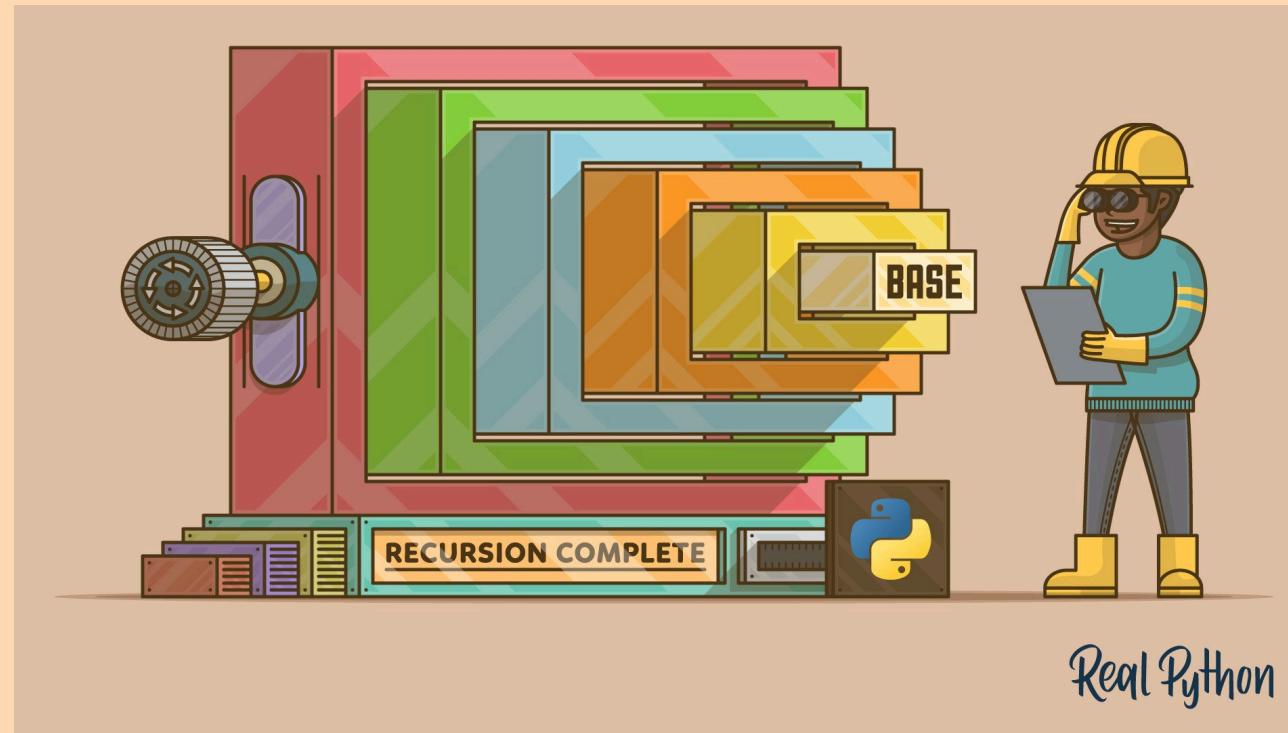
Recursive Functions (Recursion)



A function that calls itself.

■ 224 ■ / ■ 240 ■

(1) Thinking like a Computer Scientist



Solving a task by first solving its smaller subtasks

Mathematician: mathematical induction!!!

(1) Thinking like a Computer Scientist

You want to solve a task and suppose you have the Mirror.



The Mirror can solve the smaller task for you magically.

26 - 240

(1) Thinking like a Computer Scientist

- What is a smaller task? (★★★★★)
- How to reduce the original task into the smaller task? (★★★)
- How to solve the task given the solution to the smaller one? (★★★)
- What is the smallest task? (★)

Thinking Recursively

It will be a loooooooooong way to go to
master RECURSION!!!

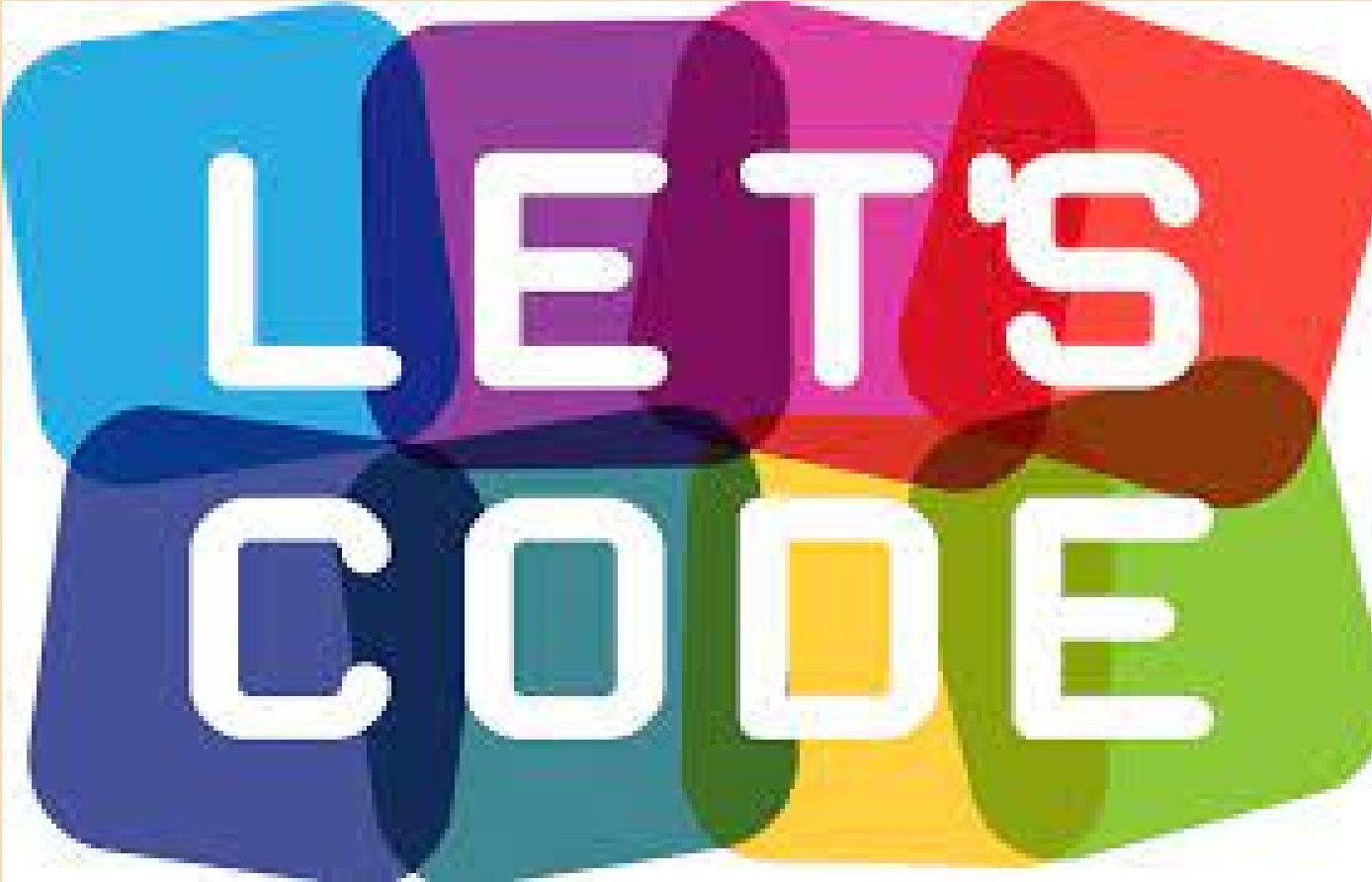
(2) Thinking like a Computer



 MANHATTAN PREP

■ 229 ■ / ■ 240 ■

min.c



min-re.c **sum-re.c** **fib-re.c** **gcd-re.c**

bsearch-re.c **mergesort.c**

Min (min-re.c)



■ 231 ■ / ■ 240 ■

Min (min-re.c)

```
Min(3, 5, 2, 7) = min(7, Min(3, 5, 2))  
                  = min(7, min(2, Min(3, 5)))  
                  = min(7, min(2, min(5, Min(3))))  
                  = min(7, min(2, min(5, 3)))  
                  = min(7, min(2, 3))  
                  = min(7, 2)  
                  = 2
```

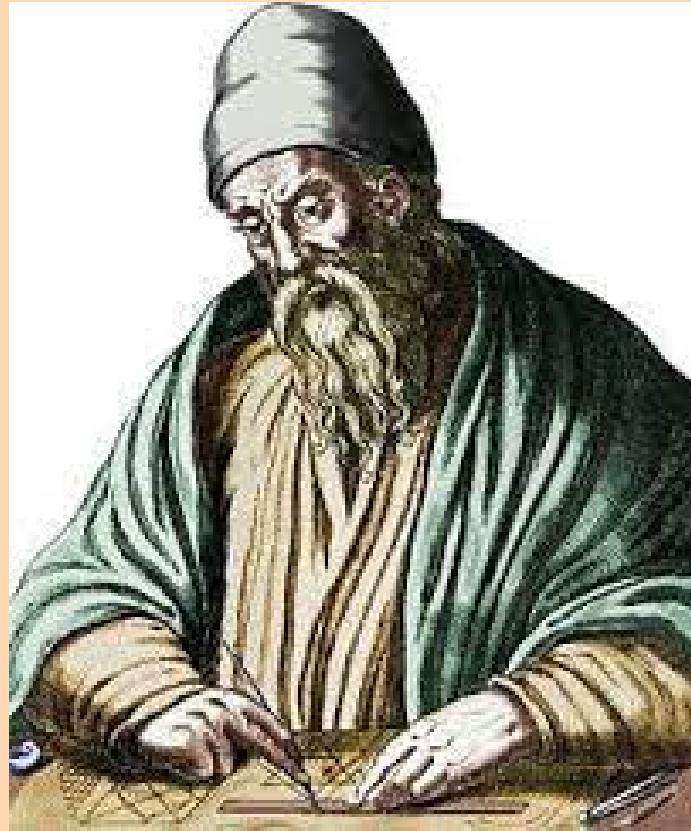
Fibonacci Sequence (**fib-re.c**)

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad (n > 1)$$

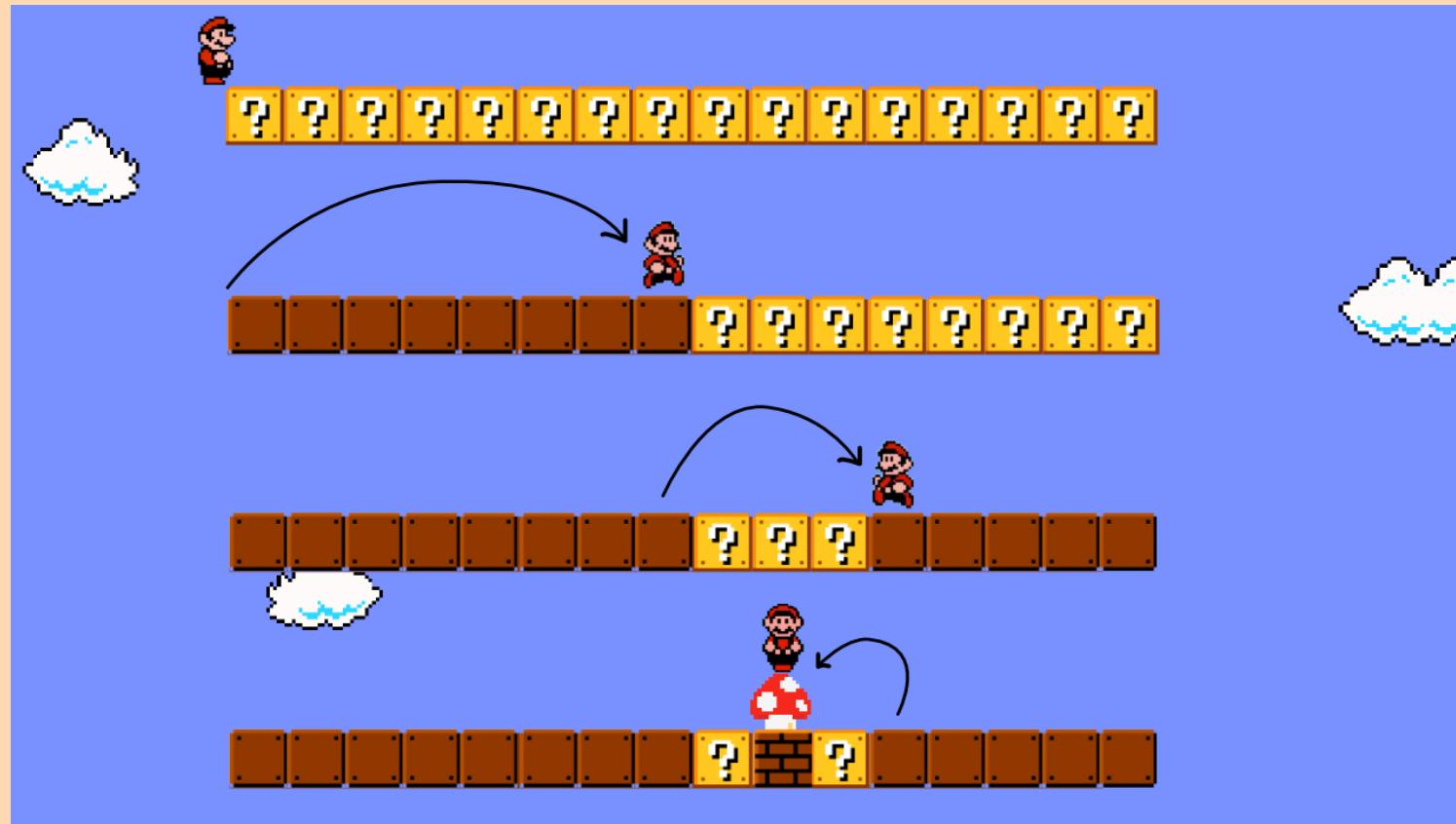
Greatest Common Divisor (gcd-re.c)



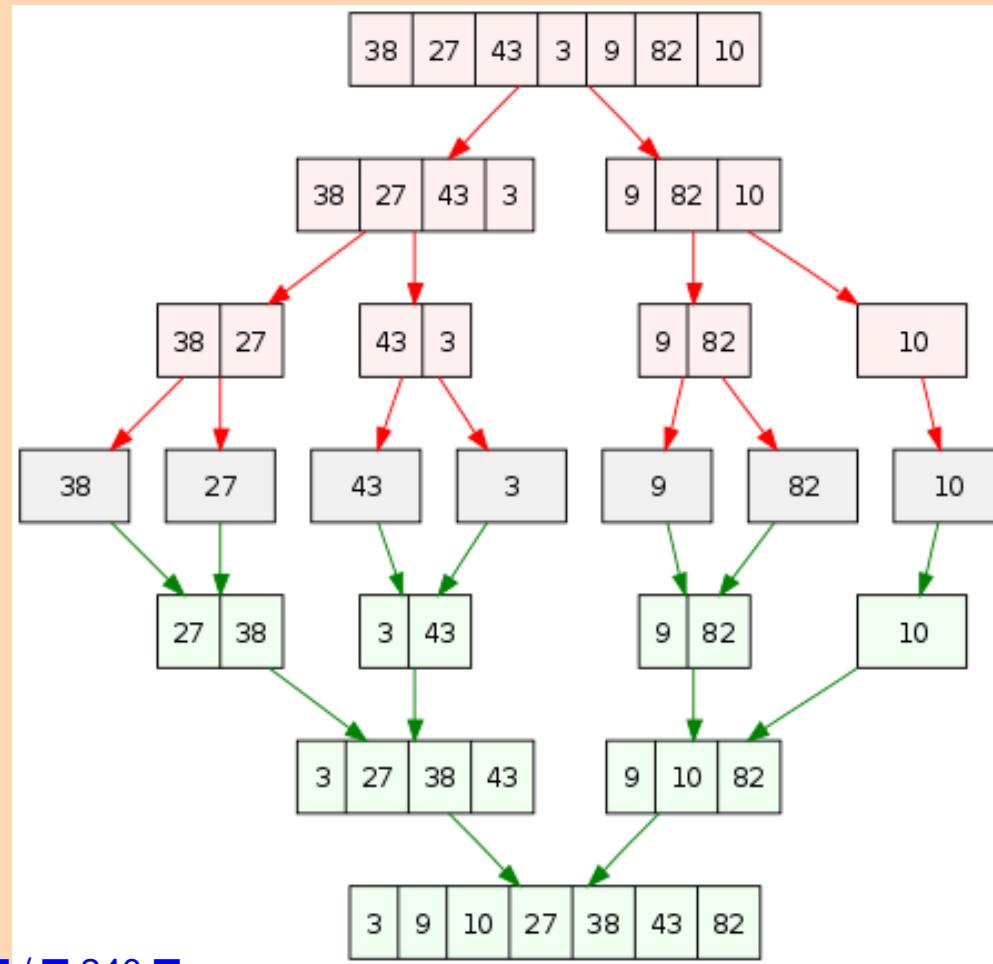
$$\text{gcd}(a, b) = \text{gcd}(b, a \% b)$$

■ 234 ■ / ■ 240 ■

Binary Search (`bsearch-re.c`)



MergeSort (mergesort.c)



■ 236 ■ / ■ 240 ■

6 5 3 1 8 7 2 4



■ 238 ■ / ■ 240 ■



■ 239 ■ / ■ 240 ■

