



# 复习

通用考点：

- 1.基本语法（变量、常量、运算符、表达式、变量存储类型）
- 2.程序结构（选择，循环）
- 3.排序（选择排序、冒泡排序），查找
- 4.字符串处理
- 5.数组、指针、结构
- 6.函数的声明定义与使用

## 一 判断题

- a) 考点：语法、数据类型的值域、指针的使用
- b) 注意：是否正确不是以编译是否通过为依据  
判断地址的级与指针的使用方法是否正确  
常量不能作为表达式的左值

## 二 计算表达式的值

- a) 考点：优先级、数据类型的转换
- b) 注意：无符号整形的运算

### 三 改错

- a) 考点：语法错误，函数的声明、定义与使用
- b) 注意：如果使用strcpy等字符串处理函数需要加  
`#include<string.h>`
  - 变量需要先定义后使用
  - 字符串循环处理结束条件一般以 ‘\0’ 为标志
  - 累加的变量考虑是否需要初始化
  - 碰到scanf都要考虑格式是否正确
  - 常量（比如数组名）不能作为表达式的左值

### 四 填空题

- a) 注意：联系程序上下文

## 五 写程序结果

- a) 考点：通用考点、变量的作用域
- b) 注意：严格按照输出格式书写，比如回车等

## 六 编程题

- a) 考点：函数必考，结构必考
- b) 注意：采用自顶向下的设计方式，（1）首先分析程序功能；（2）将功能分解为多个模块（函数）；（3）写main函数、函数声明、调用用户自定义函数；（4）定义用户自定义函数

## 1. 指针的级别判断

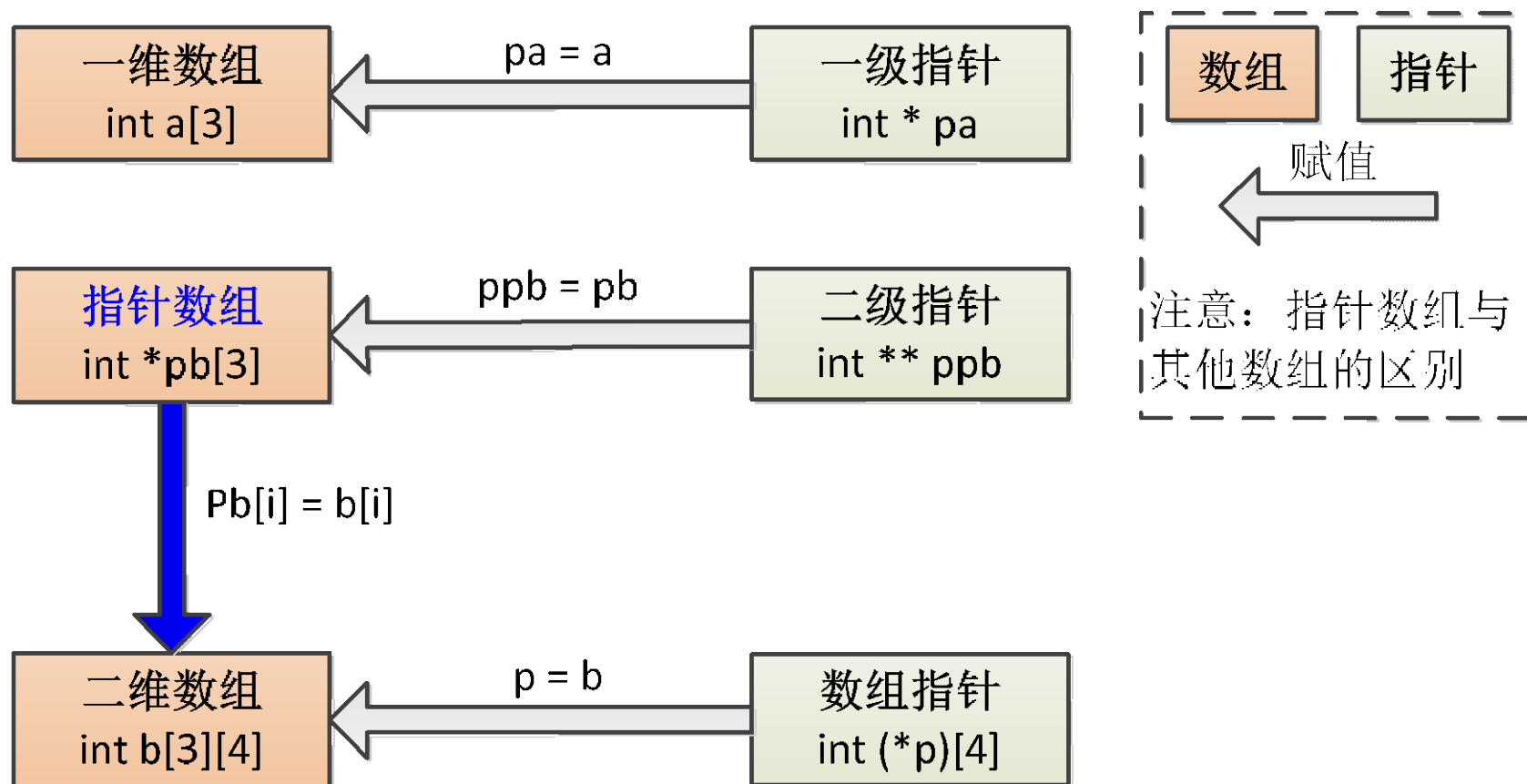
表 1 指针级别判断

	升级	降级
变量定义	* []	
变量使用	&	* []

int a[3][4] (二级):	a[1] 一级	&a[0][0] 一级
int a; (一级)	:	&a 一级
int *a[3] (二级) :	a[i] 一级	*a 一级

## 2. 指针的使用，指针与数组

图 1 指针与数组



比如二级指针不能直接指向二维数组，但是可以通过下面方式实现：定义指针数组指向二维数组，二级指针指向指针数组

### 3. 数组指针与指针数组比较

表 2 数组指针 vs 指针数组 (b[3][4])

	指针数组	数组指针
定义方式	<code>int *pb[3]</code>	<code>int (*p)[4]</code>
数据类型本质	数组：数组的元素为指针	指针：指针指向的数据类型为数组
赋值	<code>pb[i] = b[i]</code>	<code>p=b</code>
数组元素访问	<code>pb[i][j]</code> <code>*(pb[i] + j)</code> <code>(* (pb+i))[j]</code> <code>*(* (pb+i)+j)</code>	<code>p[i][j]</code> <code>*(p[i] + j)</code> <code>(* (p+i))[j]</code> <code>*(* (p+i)+j)</code>



#### 4. 指针与数组相关的实参与形参

本质是函数调用时实参给形参赋值，所以参照图 1

表 3 指针与数组相关的实参与形参对应关系

实参（int）	形参
a[3]	int *p            或    int p[]
a[3][4]	int (*p)[4]    或    int p[][4]
*a[3]	int **p        或    int *p[]