# 近两日关于数组名与指针学习汇总

一，数组名与常量指针区别

数组名不是常量指针，从两方面：

1.类型方面，数组的类型是type[size]，和常量指针类型type\* const不同

2.使用方面，sizeof(数组名)等于数组所有元素的大小，而不是sizeof(指针)；对数组取地址，得到的指针进行加减，增减字节数是sizeof(数组)；你可以用字符串字面量初始化一个字符数组，但是不能用常量指针来初始化一个字符数组。

从类型系统和隐式转换规则来讲：

* 数组名的类型就是数组类型type[size] （参见[Array declaration](http://link.zhihu.com/?target=http://en.cppreference.com/w/c/language/array" \t "/home/rilan/文档\\x/_blank)）
* 对数组取址，得到的类型是指向数组的指针type(\*)[size] （参见[Member access operators](http://link.zhihu.com/?target=http://en.cppreference.com/w/c/language/operator_member_access" \t "/home/rilan/文档\\x/_blank)）
* 数组名在大部分情况下会隐式转换为首元素指针右值，除了（参见 [Implicit conversions](http://link.zhihu.com/?target=http://en.cppreference.com/w/c/language/conversion" \t "/home/rilan/文档\\x/_blank)）
  1. sizeof运算的时候
  2. 取址的时候
  3. 用字符串字面量初始化字符数组的时候
  4. 以及C11的\_Alignof运算符（这个本文不考虑）

其他情况下都会发生数组到指针隐式转换，比如你用方括号取值，如果方括号左边是个数组，那么数组会被隐式转换为首元素指针右值，然后对这个值进行的解引用。（参见[Member access operators](http://link.zhihu.com/?target=http://en.cppreference.com/w/c/language/operator_member_access" \t "/home/rilan/文档\\x/_blank)）

[更进一步](https://www.baidu.com/s?wd=%E6%9B%B4%E8%BF%9B%E4%B8%80%E6%AD%A5&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "/home/rilan/文档\\x/_blank)，二维数组type[size1][size2]，实际上就是元素类型是数组的数组，进行隐式转换后得到的首元素指针类型是type(\*)[size2]，指向数组的指针，对这个指针加减，得到的也是数组的指针，对它解引用，得到数组。这一套过程，用那个什么数组名是常量指针的说法，是解释不来的。

以上学习自https://zhuanlan.zhihu.com/p/24799071

二，数组名a和&a区别

对数组a而言，数组名a是指向数组的特殊指针，&a相当于指向a的二级指针，（&a）相当于一个二维数组，在其共同指向的内存区域中，&a与a的值没有区别，但当对其进行指针的加法操作时，a+1相当于a[1]，而&a+1相当于a[n+1]（数组长度为n），相当于即在二维数组[中移动](https://www.baidu.com/s?wd=%E4%B8%AD%E7%A7%BB%E5%8A%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "/home/rilan/文档\\x/_blank)了行级元素，即相比而言超出一个数组长度。