

## 问题

数组指针和指针数组有什么区别？

## 数组指针

数组指针：**本质是一个指针，指向了一个数组**，数组中的每个元素都是某种数据类型的值（比如 int 类型）。

```
1 | int (*p)[n]; //定义了一个数组指针，指向一个大小为n的数组，数组中的每个元素都是int类型
```

数组指针也称**行指针**，也就是说，当指针p执行p+1时，指针会指向数组的下一行，如：

```
1 | int a[3][4];
2 | int (*p)[4]; //p是一个数组指针，指向了一个包含4个int型元素的数组
3 | p=a; //将二维数组的首地址赋给p，即a[0]或a[0][0]
4 | p++; //跨过第一行，p指向了a[1][0]
```

## 指针数组

指针数组：**本质是一个数组，该数组中的每个元素都是一个指针**。

```
1 | int *p[n]; //定义了一个指针数组，数组大小为n，数组中的每个元素都是一个int*指针
```

指针数组是一个包含若干个指针的数组，p是数组名，当执行p+1时，则p会指向数组中的下一个元素。

```
1 | int a[3][4];
2 | int *p[3]; //定义了一个数组，该数组中有3个int*指针变量，分别为p[0]、p[1]、p[2]
3 | //p++; //若执行此语句，则数组p指向下一个数组元素
4 | for(int i=0;i<3;i++){
5 |     p[i]=a[i]; //数组p中有3个指针，分别指向二维数组a的每一行
6 | }
```

## 访问数组

补充一下二维数组的访问方式：对于一个二维数组  $p[m][n]$ ，表示其第  $i$  行第  $j$  列的元素的方式有以下4种：

```
1 | p[i][j] //方式1
2 | *(p[i]+j) //方式2
3 | *(* (p+i)+j) //方式3
4 | (*(p+i))[j] //方式4
```

# 总结

---

总结一下数组指针和指针数组的区别：

## 定义不同

数组指针：**本质是一个指针，指向了一个数组**

指针数组：**本质是一个数组，该数组中的每个元素都是一个指针**

## 写法不同

数组指针：`int (*p)[n];`

指针数组：`int *p[n];`

区分方法：数组名带括号的就是数组指针，不带括号的就是指针数组。（这个类似于函数指针和指针函数的区别）

补充：同样类似函数指针和指针函数的区分方法，可以通过从右到左结合来区分

①对于数组指针 `int (*p)[n]`，因为括号优先级较高，因此\*号与数组名p先结合，也就是说p首先是一个指针，然后与[n]结合，表示指针p指向了一个大小为n的数组，数组的类型为int。

②对于指针数组 `int *p[n]`，p和[]先结合，因此p首先是一个大小为n的数组，剩下的部分是数组的类型，即int\*类型，也就是数组的每个元素都是一个int\*指针。

## 参考资料

---

[数组指针和指针数组的区别](#)