# C语言中关于数组名什么情况为首元素地址,什么情况为整个数组的地址的问题。

其实数组名就是数组首元素(第一个元素)的地址是对的,但是有两个例外: • sizeof(数组名), sizeof中单独放数组名,这里的数组名表示整个数组,计算的是整个数组的大小,单位是字节 • &数组名,这里的数组名表示整个数组,取出的是整个数组的地址(整个数组的地址和数组首元素的地址是有区别的)除此之外,任何地方使用数组名,数组名都表示首元素的地址。

## 那么请看下面的代码:

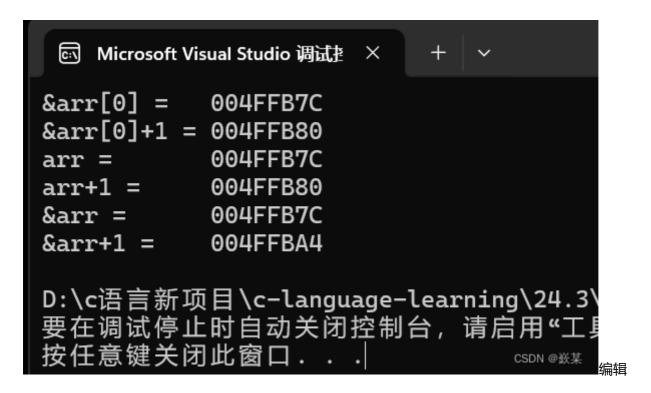
```
#include <stdio.h>
int main()
{
    int arr[10] = { 1,2,3,4,5,6,7,8,9,10 };
    printf("&arr[0] = %p\n", &arr[0]);
    printf("arr = %p\n", arr);
    printf("&arr = %p\n", &arr);//这三个printf的输出都一样
    return 0;
}
```

### 那么怎么解释这三个printf的输出都一样的事实呢?

#### 来,继续:

```
#include <stdio.h>
int main()
{
    int arr[10] = { 1,2,3,4,5,6,7,8,9,10 };
    printf("&arr[0] = %p\n", &arr[0]);
    printf("&arr[0]+1 = %p\n", &arr[0]+1);
    printf("arr = %p\n", arr);
    printf("arr+1 = %p\n", arr+1);
    printf("&arr = %p\n", &arr);
    printf("&arr+1 = %p\n", &arr+1);
    return 0;
}
```

### 输出结果:



这里我们发现&arr[0]和&arr[0]+1相差4个字节, arr和arr+1相差4个字节, 是因为&arr[0]和arr都是 首元素的地址, +1就是跳过一个元素。 但是&arr和&arr+1相差40个字节, 这就是因为&arr是数组的地址, +1操作是跳过整个数组的。到这里大家应该搞清楚数组名的意义了吧。

数组名是数组首元素的地址,但是有2个例外。

本期博客到这里就结束了,如果有什么错误,欢迎指出,如果对你有帮助,请点个赞,谢谢!