数组定义方式：

数组中的每个数据元素都是相同的数据类型；且处于连续的内存位置

- 数据类型 数组名[数组长度]；
- 数据类型 数组名[数组长度] = {值1，值2，值3…}；
- 数据类型 数组名[] = {值1，值2，值3…}；

定义数组的时候必须有初始长度

访问数组（可以通过下标访问数组元素）：数组名[0]，数组名[1]……

一维数组的用途：

- 可以统计数组在内存中的长度：用sizeof（数组名）命令
- 获取数组在内存中的首地址:cout<<数组名<<endl;这样输出16进制,加int转变为十进制。
- 数组名是一个常量，不能再进行赋值

二维数组

- 数据类型 数组名[行数][列数]；
- 数据类型 数组名[行数][列数]：{{数据1，数据2}，{数据3，数据4}}；
- 数据类型 数组名[行数][列数]：{数据1，数据2，数据3，数据4…}；
- 数据类型 数组名[][列数]：{数据1，数据2，数据3，数据4…}；

可以不写行数，但是需要指定列数。可用双层循环打印数组

二维数组的用途：与一维数组类似。但是其数据地址递增的方式是，先行增加，再列增加。

地址递增方式如下：

有一个比较特殊的是： &arr[2][3],类似于特定某个地址，需要加寻址符 &。