**多维数组 ≒ 一维数组 + 多重循环**

* 多维数组在内存中仍是线性排列（连续空间），只是通过多维的方式逻辑划分。
* 多用于地图、图像、坐标、状态表、敌人/弹幕管理等
* 屏幕上有字

  AI 生成的内容可能不正确。通常使用多重循环进行访问和管理：
  + 一维数组：1层循环
  + 二维数组：2层循环（行列）
  + 三维数组：3层循环（层、行、列）

① 全部初始化为相同值（如 0）

*int enemyPos[2][3] = { 0 };*

②　可分别给每个要素初始化值

*int enemyPos[2][3] = {*

*{10, 10, 10},*

*{10, 10, 10}};*

③　不指定要素数的话，由初始化值的数量决定数组长度

*int enemyPos[][3] = {*

*{10, 20, 30},*

*{40, 50, 60}}; // 推断为 2行3列*

④　初始化值数量少于要素数量：自动填充0

*int bulletPos[2][4] = {*

*{10, 20}, // → {10, 20, 0, 0}*

*{30} // → {30, 0, 0, 0}};*

⑤　初始化值数量多于要素数量：错误

int bulletY[2][3] = {

{10, 10, 10},

{10, 10, 10, 10} // ❌多了1个元素

};

**例：求第 1 列（column 0）的总和**

*for (int i = 0; i < 4; i++){*

*sum += array[i][0];}*

**求第 1 行（row 0）的总和**

*for (int i = 0; i < 4; i++){*

*sum += array[0][i];}*

**例：求所有行列的总和**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Array[][] | **j=0** | **j=1** | **j=2** | **j=3** |
| i=0 |  |  |  |  |
| i=1 |  |  |  |  |
| i=2 |  |  |  |  |

*for (int i = 0; i < 4; i++){*

*for (int j = 0; j < 4; j++){*

*num += array[i][j]);*

*}*

*}*