

lab挑战实验：多维数组

学号：19373075

姓名：许天识

思路方法

用一维数组模拟高维数组

这种方法核心在于编译出来的数组只有一维类型，其重点主要在于如何转换，以及下标计算。

可以将工作分为3部分：数组定义初始化、数组重赋值、数组调用。

具体细节

符号定义

数组 $a[n_1][n_2]...[n_k]$ ，其中 k 是它的维度， n_k 是每一维的维度大小。

记 $total = n_1 * n_2 * ... * n_k$

1.数组定义及初始化

数组定义

识别出每一个 n_k 后，将其相乘（因为定义时一定能求出值），即全部化为一维数组：

```
[n1*n2*...*nk x i32]
```

具体定义语法根据不同类型（全局、局部、const、非const）写即可(allocat或其他)

初始化

重点在于读取后面用花括号嵌套的初始值，读出来后，再一个个赋值。首先直接new一个一维数组出来，维度是total，初始时元素全为null。然后引入两个变量i和index，i表示嵌套的层数，index表示当前下标，那么根据i的大小可判断是否要赋值（因为条件给定花括号嵌套数量一定与数组维数对应），根据index得知赋值位置。

根据推理得index递推公式为：

$$index = index + n_{i+1} * n_{i+2} * ... * n_k$$

当进入下一层{}时，index不变，i++，进入同级的下一个{}时，i不变，index根据递推公式计算。

关键代码如下：

```

InitVal(i+1,index);
int total=1;
for(int j=i+1;j<tmp_shape.size();j++){
    total*=tmp_shape.get(j);
}
while (token==Tokens.Comma){
    getSym();
    index+=kk;
    InitVal(i+1,index);
}

```

得到这些值后，用for循环遍历赋值，不存在的就赋值为0或者不赋值(是否是全局数组)

对于没有初始化的全局数组，初始化为全0

2.数组重赋值

难点在于如何转换下标，赋值只需要一个store即可。

根据推理，下标index公式如下：

对于数组访问 $a[x_1][x_2]...[x_k]$:

$$index = x_k + x_{k-1} * n_k + x_{k-2} * n_k * n_{k-1} + \dots + x_1 * n_k * n_{k-1} * \dots * n_2$$

每计算一步，输出一步计算代码即可。

3.数组调用

跟2类似，先计算出下标，再用gep取得下标，再load加载出来即可。

注意调用的维数一定要小于等于数组维数，否则报错。