**String[ ]**

什么是字符串？

在 C 语言中，字符串就是 char 类型的数组，此数列以 '\0' 终止。

‘/0’占用一个字符，它的作用是？

'\0' 是字符串的终止标志（null 终止符），相当于句号，告诉 C 语言字符串已经结束。

没有 '\0'，printf() 等函数无法判断字符串在哪结束。为了让字符串正常识别终点，必须在数组中预留存放 '\0' 的空间，或者在写入字符时显式添加该终止符。

一维字符串初始化方式：

1. 手动添加终止符：

char str[5] = { 'A', 'B', 'C', 'D', '\0' }; // OK：预留了终止符空间

1. 使用字符串字面量自动添加 '\0'

char str[7] = "kamata"; // 编译器自动在末尾添加 '\0'

1. 让编译器自动推导数组长度

char str[ ] = "kamata"; // 实际长度为 7（含 '\0'）

char str[ ] = "あいうえお"; // UTF-8 编码下长度为 11（每个假名占 3字节+1个终止符）

PG课题中得到需要注意的点！！

在程序运行过程中，如果你通过逐个字符赋值的方式或循环构造字符串内容，**编译器不会自动添加终止符 '\0'**，此时你必须**手动在末尾加上 '\0'**，否则这个“字符串”在 C 的标准库函数中无法被识别为合法字符串。

例：进制转换程序：

int num = 25; // 要转换的十进制数

char binary[32]; // 最多支持32位二进制数

int index = 0;

while (num > 0) { // 十进制转二进制循环（从低位到高位）

// 取余(余只可能是0或1)

// ascII码计算'0' == 48 ；'0' + 1 == 49 == '1' ；0位就是binary[ 0 ] 最高位是binary[ index - 1 ]

binary [index++] = (num % 2) + '0';

num /= 2;

}

binary[index] = '\0'; // 手动添加终止符， binary [index] = ‘/0’：表示字符串结束

printf("Binary: ");

for (int i = index - 1; i >= 0; i--) { // 用for，最高位index-1递减来反向输出

putchar(binary [i]);

}

**Table[ ]**

什么是二维字符串？

在 C 语言中，**二维字符串**就是**多个一维字符串组成的数组**，也称为**字符串数组**，

本质上是：char table [行数（字符串数量）] [列数（字符串长度）];

二维字符串的主要用途是什么？

表格化字符串数据的管理

二维字符串初始化方式：

你可以不指定字符串数量（行数），但必须指定每个字符串的长度（列宽）才能分配内存

char students [3(可省略)] [11] = { "Alice", "Bob", "Charlie" };

例：计算元素数量以依次打印程序：

char fruits [ ][10] = { "Apple", "Banana", "Grape" };

int rowCount = sizeof(fruits) / sizeof(fruits[0]); //二维数组大小/单个元素分配的大小 = 元素个数

for (int i = 0; i < rowCount; i++) {

printf("fruits[%d] = %s\n", i, fruits[i]); //利用得到的元素个数快捷依次输出字符串

}