



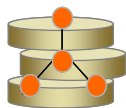
第五单元（5）：字符型数组

| | |
|------|---|
| 教学内容 | character array 字符型数组 |
| 教学目标 | |
| 应知 | 掌握一维字符型数组的定义、引用、初始化及字符串与字符串结束标志（采用与整型数组对照的方法讲解） |
| 应会 | 有关字符串的常用程序、算法：统计字符串中各种字符的个数，大、小写字母转换，字符串复制、连接等 |
| 难点 | 字符串结束标志的使用 |

教学方法

- 采用对比法，通过与一维数组的定义、输入输出、初始化等操作的对比，引出字符数组的定义、输入输出、初始化等操作

教学过程



1. 定义一个整型数组 `a[10]`，输入输出数据，将整型数组改为字符型数组，输入输出数据（引入字符型数组的定义，逐个输入输出方式）（基本概念，务必清楚）

Define a character array and input data from the keyboard and output to the screen one by one .

- 一维数组定义：类型、数组名、长度： `int a[10];`
字符数组定义：类型、数组名、长度： `char s[10];`
- 一维数组输入： `for(i=0;i<10;i++) scanf("%d" , &a[i]);`
字符数组输入： `for(i=0;i<10;i++) scanf("%c" , &s[i]);`
- 一维数组输出： `for(i=0;i<10;i++) printf("%d" , a[i]);`
字符数组输入输出： 要求学生自己写出
- 一维数组初始化： `int a[10] = {1,2,5,3,7,8}`
字符数组初始化： `char s[10] = { 's' , 't' , 'u' , 'd' , 'e' , 'n' , 't' }`
- 一维数组元素引用：下标从 0 开始，不可以整体引用（学生易错）
字符数组元素引用：下标从 0 开始，可以整体引用

2. 字符数组的整体引用



将第 1 步字符数组的输入输出方式进行优化（引入字符数组的整体操作方式及字符串及字符串结束标志的概念）

3. 将第 2 步输入的字符串

（从键盘输入 20 个字符），统计其中字母、数字及其他字符的个数并输出结果（掌握算法与字符串结束标志的实际应用）

4. 在第 3 步的基础之上，将该字符串中的大、小写字母互相转换

（掌握算法，大小写字母的 `ascii` 码，巩固字符串结束标志的应用，复习循环语句）

5. 有 m 个学生 n 门课程，编程分别统计每个学生的总成绩和每门课程的平均成绩。

定义两个字符型数组变量 `s1[20]`、`s2[10]`、从键盘输入字符串（均在 10 字符以内），在屏幕上输出；使用库函数 `strcmp()`、`strcpy()`、`strcat()`、`strlen()` 分别对字符串 `s1` 和 `s2` 进行比较、复制、连接、求长度的运算，并输出结果



学生容易出错的地方

- ❑ 将字符数组定义为一个字符
- ❑ 混淆 ‘`c`’ 和 “`c`” 的意义
- ❑ 字符数组整体输入时用取地址符号是不对的

问题与讨论

- ❑ 为什么字符数组输入时不用取地址符号？
- ❑ 字符串结束标志有什么意义？



小结（可由问题与讨论方式给出）

- ❑ 字符数组可以通过数组名进行整体引用
- ❑ C 语言没有字符串类型数据，字符串的操作通过一维字符数组（和字符型指针）实现
- ❑ C 语言规定以 ‘`\0`’ 作为字符串结束标志
- ❑ 字符数组的输入输出分别可使用 `getchar()`、`scanf()`、`gets()` 和 `putchar()`、



printf()、puts()方式输出，在用 scanf()或 printf()时又可使用“%c”或“%s”格式，但多数时候使用“%s”



课后任务



不使用库函数，编程实现两个数组的复制、连接。