



# 知识点总结



- 在C语言中，**字符串**是由一系列字符组成的字符数组，以**空字符'\0'**作为结束标志。

- 定义一个字符串：**char 字符串名[数组长度];**

- 定义并初始化字符串时，通常是将**字符串字面量赋值给一个字符数组**：

```
char str[6] = "Hello"; // 程序会自动在末尾添加空字符'\0'
```

上面这行代码等价于 `char str[6] = {'H', 'e', 'l', 'l', 'o', '\0'};`

- 注意：定义字符串时，数组长度至少为字符串 + 1（为**空字符'\0'**预留空间）。

- 使用scanf、printf输入输出字符串时，用到的格式符是**%s**。

- 使用scanf读取字符串到数组str里：

```
// 数组名本身就是一个地址，不需要再用取地址符&
```

```
scanf("%s", str);
```

注意：



1、scanf使用%s读取字符串时，遇到空白字符（空格、换行符等）会停止读取，并在把字符串存储到字符数组后，在字符串末尾自动添加空字符'\0'。

例：scanf("%s", str); // 输入 "ab cd" → str 内容为 {'a','b','\0'}

2、输入长度超过数组容量，会导致缓冲区溢出。为了防止输入的字符串超出数组长度，可以在格式符%s中用一个整数指定宽度，来限制scanf最多能够读取的字符个数。

例：

```
char str[10];
```

// 要预留一个位置给'\0'，限制scanf最多读取9个字符

```
scanf("%9s", str);
```

使用printf打印字符串str：

```
printf("%s", str);
```

printf使用%s输出字符串时，会从字符数组的起始地址开始打印，直到遇到第一个空字符'\0'为止。



为了给字符串结尾的空字符预留空间，定义字符串时，数组长度至少是字符串的长度再加1。

题目1、下面是有关C语言字符数组的描述，其中错误的是（D）

- A. 定义字符串的时候，数组长度至少应该是字符串的长度再加1
- B. 可以用输入语句把字符串整体输入给字符数组
- C. 字符数组中的内容不一定是字符串
- D. 字符数组只能存放字符串



没有指定长度，程序会根据数组里的元素个数来确定数组长度

题目2、设有以下定义：

```
char s1[] = "0123";
```

```
char s2[] = {'0', '1', '2', '3'};
```

则以下叙述正确的是（ C ）

- A. 数组s1的长度小于s2的长度
- B. 数组s1和s2的长度相同
- C. 数组s1的长度大于s2的长度
- D. 数组s1和s2完全等价



不能直接把一个字符串赋值给一个char类型变量

不能用赋值语句把字符串赋给字符数组名

不能直接把一个数组名赋给另一个数组名

题目3、以下选项中，合法的是（ A ）

A. `char str3[] = {'d', 'e', 'b', 'u', 'g', '\0'}`

B. `char str4; str4 = "hello world";`

C. `char name[10]; name = "china";`

D. `char str1[5] = "pass", str2[6]; str2 = str1;`

题目4、有如下程序：



ASCII码值为0的字符是是'\0'



```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    char name[10] = {'S', 'T', 'R', 'I', 'N', 'G'};
```

```
    name[3] = 'E';
```

```
    name[5] = 0;
```

```
    printf("%s\n", name);
```

```
}
```

程序运行后的输出结果是 ( C )

A. STRENG

B. STRIEG

C. STREN

D. STREN0

题目5、有以下程序：

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int i, j = 0;
```

```
    char a[] = "How are you!", b[10] = {0};
```

```
    for (i = 0; a[i]; i++)
```

```
        if (a[i] == ' ')
```

```
            b[j++] = a[i-1];
```

```
    printf("%s\n", b);
```

```
}
```

程序运行后的输出结果是 ( A )

A. we

B. How are you!

C. ay

D. we!



ASCII码值为0的字符是'\0'

