

字符串和格式化输入输出

www.codeisbook.com

字符串

- 字符串(character string)
 - 字符序列: “hello, world”
- char类型数组和空字符
 - C中用char类型数组存储字符串
 - 最后一个位置是空字符\0
 - 数组的单元数比字符数多1

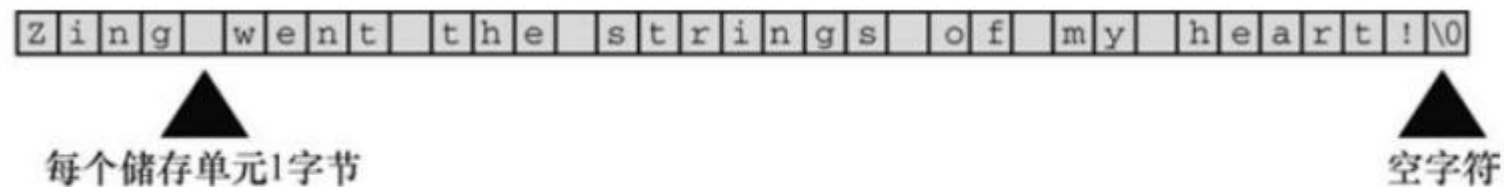


图4.1 数组中的字符串

字符串

- scanf:
 - 遇到空格(blank),制表符(tab),换行符(newline)停止读取

```
1  /* praise1.c -- 使用不同类别的字符串 */
2  #include <stdio.h>
3  #define PRAISE "What a super marvelous name"
4  int main(void)
5  {
6      char name[40];
7
8      printf("What is your name?\n");
9      scanf("%s", name);
10     printf("Hello, %s. %s\n", name, PRAISE);
11     return 0;
12 }
13
```

字符串

- strlen
 - sizeof 以字节为单位给出数据的大小
 - strlen 以字符个数为单位给出字符串的长度，不包括结尾的\0
 - <string.h>

常量和预处理器

- 常量：
 - 程序执行过程中，值不会变
- 预处理器定义常量
 - `#define PI 3.141593`
 - `#define BEEP '\a'`
 - `#define OOPS "Now you got it!"`
- `const`关键字定义常量
 - `const int MONTHS = 12`

printf

- 输出
- 转换说明符(conversion specification)
 - 以哪种数据类型打印变量
 - `printf("The value of pi is %f.\n", PI);`
 - 格式: `printf(Control-string, item1, item2, ...);`
 - 控制字符串：包括要打印的字符，转换说明符
 - 参数可以是常量、变量、表达式

printf

- 转换说明符列表

转换说明	输出
%a	浮点数、十六进制数和 p 记数法 (C99/C11)
%A	浮点数、十六进制数和 p 记数法 (C99/C11)
%c	单个字符
%d	有符号十进制整数
%e	浮点数, e 记数法
%E	浮点数, e 记数法
%f	浮点数, 十进制记数法
%g	根据值的不同, 自动选择%f 或%e。%e 格式用于指数小于-4 或者大于或等于精度时
%G	根据值的不同, 自动选择%f 或%E。%E 格式用于指数小于-4 或者大于或等于精度时
%i	有符号十进制整数 (与%d 相同)
%o	无符号八进制整数
%p	指针
%s	字符串
%u	无符号十进制整数
%x	无符号十六进制整数, 使用十六进制数 0f
%X	无符号十六进制整数, 使用十六进制数 0F
%%	打印一个百分号

printf

- 修饰符
 - 在%和转换说明符之间
 - 控制输出格式：字符宽度、精度、前导符号等
- `printf(" *%10d*", PAGES);` //宽度为10打印整数
 - `* 931*`
- `printf("%10.3f", RENT);` //宽度为10，精度为3打印浮点数
 - `* 3852.990*`

printf

- 修饰符列表

修饰符	含义
标记	表 4.5 描述了 5 种标记 (-、+、空格、#和 0)，可以不使用标记或使用多个标记 示例: "%-10d"
数字	最小字段宽度 如果该字段不能容纳待打印的数字或字符串，系统会使用更宽的字段 示例: "%4d"
.数字	精度 对于 %e、%E 和 %f 转换，表示小数点右边数字的位数 对于 %g 和 %G 转换，表示有效数字最大位数 对于 %s 转换，表示待打印字符串的最大数量 对于整型转换，表示待打印数字的最小位数 如有必要，使用前导 0 来达到这个位数 只使用 , 表示其后跟随一个 0，所以 %.f 和 %.0f 相同 示例: "%5.2f" 打印一个浮点数，字段宽度为 5 字符，其中小数点后有两位数字
h	和整型转换说明一起使用，表示 short int 或 unsigned short int 类型的值 示例: "%hu"、"%hx"、"%6.4hd"
hh	和整型转换说明一起使用，表示 signed char 或 unsigned char 类型的值 示例: "%hhu"、"%hhx"、"%6.4hhd"
j	和整型转换说明一起使用，表示 intmax_t 或 uintmax_t 类型的值。这些类型定义在 stdint.h 中 示例: "%jd"、"%8jx"
l	和整型转换说明一起使用，表示 long int 或 unsigned long int 类型的值 示例: "%ld"、"%8lu"
ll	和整型转换说明一起使用，表示 long long int 或 unsigned long long int 类型的值 (C99) 示例: "%lld"、"%8llu"
L	和浮点转换说明一起使用，表示 long double 类型的值 示例: "%Ld"、"%10.4Le"
t	和整型转换说明一起使用，表示 ptrdiff_t 类型的值。ptrdiff_t 是两个指针差值的类型 (C99) 示例: "%td"、"%12ti"
z	和整型转换说明一起使用，表示 size_t 类型的值，size_t 是 sizeof 返回的类型 (C99) 示例: "%zd"、"%12zd"

表4.5 printf()中的标记

标记	含义
-	待打印项左对齐。即，从字段的左侧开始打印该项 示例: "%-20s"
+	有符号值若为正，则在值前面显示加号；若为负，则在值前面显示减号 示例: "%+6.2f"
空格	有符号值若为正，则在值前面显示前导空格（不显示任何符号）；若为负，则在值前面显示减号 + 标记覆盖一个空格 示例: "%6.2f"
#	把结果转换为另一种形式。如果是 %o 格式，则以 0 开始；如果是 %x 或 %X 格式，则以 0x 或 0X 开始； 对于所有的浮点格式，# 保证了即使后面没有任何数字，也打印一个小数点字符。对于 %g 和 %G 格式，# 防止结果后面的 0 被删除 示例: "%#o"、"%#B.0f"、"%#+#10.3e"
0	对于数值格式，用前导 0 代替空格填充字段宽度。对于整数格式，如果出现 - 标记或指定精度，则忽略该标记

printf

- 匹配
 - 参数个数匹配
 - `printf("%d %d", n1);`
 - 参数类型匹配
 - `printf("%d", 3.14159)`

printf

- 打印长字符串

```
1  /* longstrg.c -- 打印较长的字符串 */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Here is one way to print a ");
7      printf("long string.\n");
8
9      printf("Here is another way to print a \
10 long string.\n");
11
12      printf("Here is the newest way to print a "
13 "long string.\n");
14
15      return 0;
16 }
```

scanf

- 将输入字符串转换成各种形式：整数，浮点数，字符，字符串
- 使用变量的指针
 - 读取基本类型变量，变量前加取地址符&
 - 读取字符串，使用字符串变量名，不加&
 - `int n; scanf("%d", &n)`
 - `char a[30]; scanf("%s", a)`
- 使用空格(换行、制表符、空格)来决定把输入分成几段

scanf

- 转换说明符

转换说明	含义
%c	把输入解释成字符
%d	把输入解释成有符号十进制整数
%e、%f、%g、%a	把输入解释成浮点数（C99 标准新增了%a）
%E、%F、%G、%A	把输入解释成浮点数（C99 标准新增了%A）
%i	把输入解释成有符号十进制整数
%o	把输入解释成有符号八进制整数
%p	把输入解释成指针（地址）
%s	把输入解释成字符串。从第 1 个非空白字符开始，到下一个空白字符之前的所有字符都是输入
%u	把输入解释成无符号十进制整数
%x、%X	把输入解释成有符号十六进制整数

scanf

- %d, %f:
 - 跳过空格，读取数字或符号，到下一个空格或非数字结束
- 例子
 - -13.45e12# 0
 - %d -13
 - %f -13.45e12
 - %s -13.45e12#
- 格式字符串中的常规字符
 - scanf("%d, %d", &n, &m)
 - 88, 121
 - 88, 121