

# C/C++ 程序设计

## 字符串与格式化输入/输出

---

张晓平

武汉大学数学与统计学院

## 1. 格式化输入

## 格式化输入

同 `printf()` 一样, `scanf()` 也使用控制字符串和参数列表, 主要区别在参数列表。`printf()` 使用变量名、常量和表达式; 而 `scanf()` 使用指向变量的指针。

## 格式化输入：scanf() 的参数列表

- 若使用 `scanf()` 来读取某种基本类型的值，请在变量名前加一个 `&`。
- 若使用 `scanf()` 把一个字符串读入一个字符数组，请不要使用 `&`。

## 格式化输入：scanf() 的参数列表

```
// input.c:
#include <stdio.h>
int main(void) {
    int age;
    double weight;
    char name[20];
    printf("Enter your name, age and weight:\n");
    scanf("%s", name);
    scanf("%d , %lf", &age, &weight);
    printf("%s: %d %f\n", name, age, weight);
    return 0;
}
```

## 格式化输入：scanf() 的参数列表

```
Enter your name , age and weight:
```

```
Xiaoming
```

```
23 100
```

```
Xiaoming: 23 100.000000
```

## 格式化输入：scanf() 的参数列表

- scanf() 使用空格（换行、制表符和空格）来决定如何把输入分成几个字段。它依次把格式说明符与字段相匹配，并跳过它们之间的空格。
- 也可以分一行或多行输入，只要每个输入项目之间至少有一个换行符、空格或制表符。
- %c 是个例外，即使下一个字符是空白字符，它也会读取。



## 格式化输入：printf() 与 scanf() 格式说明符的区别

- printf() 把%f、%e、%E、%g 和 %G 同时用于 float 和 double 类型
- scanf() 只把它们用于 float 类型，而用于 double 类型时要求加上 l 修饰符。

## 格式化输入：scanf 格式说明符

格式说明符	意义
%c	把输入解释成一个字符
%d	把输入解释成一个有符号十进制数
%e, %f, %g, %a	把输入解释成一个浮点数
%E, %F, %G, %A	把输入解释成一个浮点数
%i	把输入解释成一个有符号十进制数
%o	把输入解释成一个有符号八进制数
%p	把输入解释成一个指针

## 格式化输入：scanf 格式说明符

格式说明符	意义
%s	把输入解释成一个字符串：输入内容以第一个非空白字符作为开始，并且包含到下一个空白字符的全部字符
%u	把输入解释成一个无符号十进制数
%x, %X	把输入解释成一个有符号十六进制数

## 格式化输入：scanf 格式说明符

可在格式说明符中使用修饰符，修饰符出现在 % 与格式字符之间。

## 格式化输入：scanf 格式说明符

可在格式说明符中使用修饰符，修饰符出现在 % 与格式字符之间。

修饰符	意义
*	滞后赋值，如 %*d
digit	最大字段宽度：在达到最大字段宽度或遇到第一个空白字符时停止对输入项的读取，如 %10s
hh	把整数读作 <b>signed char</b> 或 <b>unsigned char</b> ，如 %hhd 或 %hhu
ll	把整数读作 <b>long long</b> 或 <b>unsigned long long</b> ，如 %lld 或 %llu

## 格式化输入：scanf() 格式说明符

修饰符	意义
%hd, %hi	以 short 存储
%ho, %hx, %hu	以 unsigned short 存储
%ld, %li	以 long 存储
%lo, %lx, %lu	以 unsigned long 存储
%le, %lf, %lg	以 double 存储
%Le, %Lf, %Lg	以 long double 存储

## 格式化输入：scanf() 格式说明符

修饰符	意义
%hd, %hi	以 short 存储
%ho, %hx, %hu	以 unsigned short 存储
%ld, %li	以 long 存储
%lo, %lx, %lu	以 unsigned long 存储
%le, %lf, %lg	以 double 存储
%Le, %Lf, %Lg	以 long double 存储

若没有这些修饰符，则%d, %i, %o 和%x 指示 **int** 类型，而%e, %f, %g 指示 **float** 类型。

## 格式化输入：格式字符串中的常规字符

`scanf()` 允许把普通字符放在格式字符串中，除了空格字符之外的普通字符一定要与输入字符串准确匹配。



## 格式化输入：格式字符串中的常规字符

```
scanf ("%d, %d", &n, &m);
```

---

### 合法的输入方式

12, 23

12, 23

12, 23

12,

23

## 格式化输入：格式字符串中的常规字符

```
scanf ("%d_and_%d", &n, &m);
```

---

### 合法的输入方式

```
12_and_23
```

```
12_and_23
```

```
12and23
```

## 格式化输入：格式字符串中的常规字符

除了 %c 之外的说明符会自动跳过输入项之前的空格，故以下两条语句的效果相同：

```
scanf ("%d%d", &n, &m);
```

```
scanf ("%d %d", &n, &m);
```

## 格式化输入：格式字符串中的常规字符

对于 %c 来说，向格式字符串中添加一些空格将导致一些差别。如：

```
scanf ("%c", &ch)
```

读取在输入中遇到的第一个字符，而

```
scanf ("_%c", &ch)
```

则读取遇到的第一个非空白字符。

## 格式化输入：scanf() 的返回值

scanf() 返回成功读入的项目个数。

- 若没有读取任何项目，则返回 0；
- 若检测到文件结尾 (end of file)，则返回 EOF。(EOF 是 stdio.h 中定义的特殊值，一般为-1)

## 格式化输入：printf() 的 \* 修饰符 i

```
// varwidth.c:
#include <stdio.h>
int main(void)
{
    unsigned width, precision;
    int number = 256;
    double weight = 123.5;
    printf("What field width?\n");
    scanf("%d", &width);
    printf("The number is:%*d\n", width, number);
    printf("Now enter a width and a precision:\n");
    ;
    scanf("%d %d", &width, &precision);
```

## 格式化输入：printf() 的 \* 修饰符 ii

```
printf("Weight=%*.*f\n", width, precision,  
weight);  
return 0;  
}
```

## 格式化输入：scanf() 的 \* 修饰符

在 scanf() 中，把 \* 放在 % 与格式字符之间时，会使函数跳过相应的输入项目。



## 格式化输入：scanf() 的 \* 修饰符

```
// skip2.c:
#include <stdio.h>
int main(void)
{
    int n;
    printf("Please enter three integers:\n");
    scanf("%*d %*d %d", &n);
    printf("The last integer was %d\n", n);
    return 0;
}
```

## 格式化输入：scanf() 的 \* 修饰符

```
// skip2.c:
#include <stdio.h>
int main(void)
{
    int n;
    printf("Please enter three integers:\n");
    scanf("%*d %*d %d", &n);
    printf("The last integer was %d\n", n);
    return 0;
}
```

```
Please enter three integers:
10 20 30
The last integer was 30
```