



# C语言程序设计基础

林川

# 第二章 用C语言编写程序



1. 在屏幕上显示 Hello World!
2. 求华氏温度 100°F 对应的摄氏温度
3. 计算分段函数
4. 输出华氏—摄氏温度转换表
5. 生成乘方表与阶乘表

本章目的:

- 通过实际例子让大家迅速熟悉程序设计的思路
- 能够动手写代码

## 2.1 在屏幕上显示Hello World!



例2-1 在屏幕上显示一个短句:

Hello World!



# 在屏幕上显示Hello World!

```
/* 显示 Hello World! */  
  
# include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World! \n");  
    return 0;  
}
```

Diagram annotations:

- Red arrow from `/* 显示 Hello World! */` to **注释文本** (Comment text)
- Red arrow from `# include <stdio.h>` to **编译预处理命令** (Compilation preprocessing command)
- Red arrow from `int main(void)` to **主函数** (Main function)
- Red arrow from `printf("Hello World! \n");` to **; 语句结束符** (Statement terminator)
- Red arrow from `\n` to **\n 换行符** (Newline character)
- Red arrow from `return 0;` to **输出函数** (Output function)

- 1.任何程序都有主函数
- 2.程序由若干语句组成
- 3.语句由;结束



# 问：下面语句的输出有何不同？

```
printf("Hello world\n");  
printf("Hello\n world\n");
```

- 不可见字符：
  - 换行符\n
  - 空格符
  - 制表符\t

# 练习2-3:输出一个倒三角图案



\* \* \* \*

\* \* \*

\* \*

\*

## 2.2 求华氏温度 $100^{\circ}\text{F}$ 对应的摄氏温度



摄氏温度  $c = 5 \times (f - 32) / 9$



# 程序解析: $C = 5(F - 32) / 9$

```
#include <stdio.h>
int main(void)
{
    int celsius, fahr;

    fahr = 100;
    celsius = 5 * (fahr - 32) / 9;
    printf("fahr = %d, celsius = %d\n", fahr, celsius);
    return 0;
}
```

**常量:**在程序运行过程中  
其值不能改变

**变量:**在程序运行过程中  
其值可以改变





# 变量的定义

变量定义的一般形式：

类型名 变量名表；

变量名字要合适，做到：简洁、顾名思义

例如：

`int` celsius, fahr; 定义整型变量

`float` x; 定义单精度浮点型变量

`double` area, length; 定义双精度浮点型变量

变量名代表内存中的一个存储单元

存储单元的内存大小由类型决定

`int`(4字节), `float`(4字节), `double`(8字节)

`double`比`float`字节多，精度高，取值范围大



# 变量命名规则

- 简洁、顾名思义
  - 单词, 通用缩写: year, abs (absolute缩写)
  - 整数: i, j, k, n 等 (小范围内的局部变量)
  - 浮点数: x, y, z 等
- min-length && max-information 原则
- 避免依赖大小写区分的相似标识符
  - 例如: float x, X; /\* 😞 不可取 \*/
- 变量名一般使用“名词”或“形容词+名词”
  - 例如: float value, oldValue, newValue;
- 函数名一般使用“动词”或“动词+名词”(动宾词组)
- 风格保持一致



# 变量的定义与使用

```
#include <stdio.h>
int main(void)
{
    int celsius, fahr;

    fahr = 100;
    celsius = 5 * (fahr - 32) / 9;
    printf("fahr = %d, celsius = %d\n",
           fahr, celsius);
    return 0;
}
```

变量只能定义一次  
应该先赋值，后使用



## 2.2.3 算术运算和赋值运算

- 算术运算

- +、-、\*、/、%(取模、余数)
- 算术表达式:

- 数学式:  $s(s-a)(s-b)(s-c)$

C表达式:  $s*(s-a)*(s-b)*(s-c)$

- 数学式:  $5(fahr-32)/9$

C表达式:  $5*(fahr-32)/9$

$$\text{celsius} = 5*(fahr-32)/9$$



# 算术运算

- 整数除整数, 结果为整数
  - 如:  $1/2 = 0$ ,  $9/4 = 2$ 
    - 没有四舍五入
  - $5 * (\text{fahr} - 32) / 9$  和  $5 / 9 * (\text{fahr} - 32)$  等价吗?
- 运算 **%** 仅仅适用于整型数据
  - 如:  $5\%6=5$ ,  $9\%4=1$ ,  $100\%4=0$
- 双目运算符两侧**操作数的类型要相同**



# 赋值运算

- 赋值表达式：用 **=** 将一个**变量**和一个**表达式**连接起来的式子

**变量** = **表达式**

左边必须是一个变量

- 例如：

```
fahr = 100;
```

```
celsius = 5 * (fahr - 32) / 9;
```

- 计算赋值运算符**右侧表达式**的值
- 将结果值赋给**左侧变量**



# 格式化输出函数printf

printf(格式字符串, 输出参数1, ... ,输出参数n);

例如:

```
printf("Hello World! \n");
```

```
printf("fahr = %d, celsius = %d\n", fahr, celsius);
```

- 用,号隔开控制字符串和输出参数
- 控制字符串用双引号""

注意不要中文的全角符号



# printf—格式控制字符串

- 普通字符：原样输出

```
printf("Hello World!\n");
```

Hello World !

- 格式控制字符：

- **%d**: 输出参数为**int**

- **%f**: 输出参数为**float**或**double**

```
printf("fahr = %d, celsius = %d\n", fahr, celsius);
```

- 更多的...(一边用, 一边学)





# 小结

- 数据类型
  - int, float, double
- 变量的定义和使用
- 输出函数
  - printf( 格式控制字符串, 参数表);
  - 格式
    - %d – int
    - %f – float
    - %lf – double
    - \n – 换行符



## 2.3 计算分段函数

### 分段计算水费

- 输入  $x$
- 计算函数  $f(x)$ 
  - 设置一个变量  $y$ , 保存  $f(x)$  的值
- 输出, 并保留2位小数
  - 格式控制 **`%.2f`**

$$y = f(x) = \begin{cases} \frac{4}{3}x & x \leq 15 \\ 2.5x - 10.5 & x > 15 \end{cases}$$



# 程序解析—求分段函数

```
#include <stdio.h>
int main(void)
{
    double x, y;

    printf("Enter x (x>=0):\n"); /* 输入提示 */
    scanf("%lf", &x);           /* 输入数据 */
    if( x <= 15 )
        y = 4 * x / 3;
    else
        y = 2.5 * x - 10.5;
    printf("f(%f) = %.2f\n", x, y);
    return 0;
}
```

演示:

$f(9.500000)=12.67$

$f(15.000000)=20.00$

$f(21.300000)=42.75$

## 测试程序

- 1) 输入测试数据, 验证结果正确性
- 2) 覆盖每一个分支(情况)
- 3) 执行到每一种情况
- 4) 分块测试(复杂、多模块程序)



# 知识要点

- 关系运算
  - 比较大小
- 分支语句
  - if – else
- 输入函数
  - scanf



# 关系运算

$x \leq 15$

比较  $x$  和 15 的大小关系

结果为真(1)或假(0)

当  $x$  为 9.5 时,  $x \leq 15$  的结果是

真(1)

当  $x$  值为 21 时,  $x \leq 15$  的结果是

假(0)

关系运算有:  $>$ 、 $<$ 、 $>=$ 、 $<=$ 、 $==$



## 2.3.2 关系运算

关系运算有：

>          例如  $a > b$

<          例如  $a < b$

>=        例如  $a \geq b$

<=        例如  $a \leq b$

==        例如  $a == b$     **注意区分 = 和 ==**

!=        例如  $a != b$



# 运用关系表达式

- 判断  $x$  是否为负数

$$x < 0$$

- 判断  $x$  是否为零

$$x == 0$$

- 判断  $x$  是否不为零

$$x != 0$$



## 2.3.3 if-else语句

if ( 表达式 )

    语句1

else

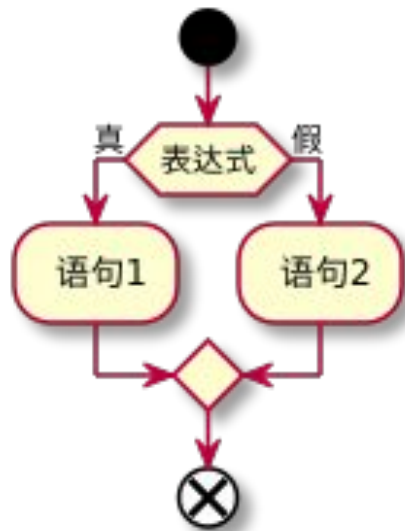
    语句2

if( $x \leq 15$ )

$y = 4 * x / 3;$

else

$y = 2.5 * x - 10.5;$







# 计算二分段函数

$$f(x) = \begin{cases} \frac{1}{x} & x \neq 0 \\ x & x = 0 \end{cases}$$

if (表达式)

    语句1

else

    语句2

if( x != 0 )

    y = 1/x;

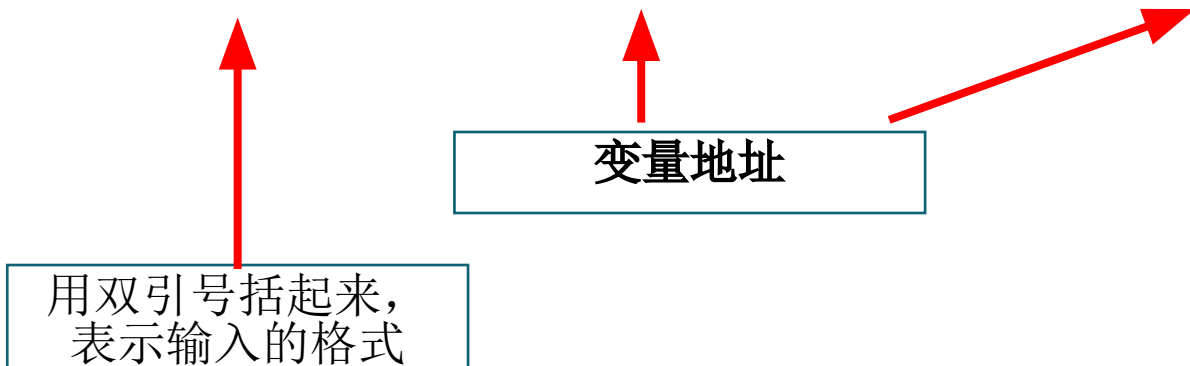
else

    y = 0;



## 2.3.4 格式化输入函数scanf

scanf(格式字符串, 输入参数1, ... , 输入参数n);



总体上和输出函数printf类似:

printf(格式字符串, 输出参数1, ... , 输出参数n);



## scanf—格式控制字符串

- int型 : %d
- float型 : %f
- double型 : %lf                      lf ==> long float  
scanf("%lf", &x);  
输入的时候, 输入x的值即可。例如 : 9.5  
注意 : 运算符 & 作用是取变量地址
- 普通字符 : 原样输入  
scanf("x=%lf", &x);  
输入的时候, 需要输入 : x=9.5
  - 尽量不要出现普通字符



## 2.3.5 常用数学库函数

- 库函数

- C语言处理系统提供事先编好的函数，供用户在编程时调用。scanf(), printf(), exp()
- 在相应的系统文件(头文件)中定义一些必需的信息。

- #include命令

- 用户调用库函数时，将相应的头文件包含到源程序中。

例如

- 调用scanf, printf, 需要 #include <stdio.h>
- 调用sqrt, 需要 #include <math.h>



# 常用数学库函数

- 平方根函数  $\text{sqrt}(x)$
- 绝对值函数  $\text{fabs}(x)$   
 $\text{fabs}(-3.56)$  的值为3.56
- 幂函数  $\text{pow}(x, n) : x^n$   
 $\text{pow}(1.1, 2)$  的值为1.21 (即1.1<sup>2</sup>)
- 指数函数  $\text{exp}(x) : e^x$   
 $\text{exp}(2.3)$  的值为 $e^{2.3}$
- 以e为底的对数函数  $\text{log}(x) : \ln x$   
 $\text{log}(123.45)$  的值为4.815836
- 以10为底的对数函数  $\text{log10}(x) : \log_{10} x$   
 $\text{log10}(123.45)$  的值为2.091491



## 例2-5 计算存款的本息

- 输入存款金额 `money`、存期 `year` 和年利率 `rate`，根据公式计算存款到期时的本息合计`sum`。输出时保留2位小数。

$$\text{sum} = \text{money} (1 + \text{rate})^{\text{year}}$$

$$\text{sum} = \text{money} * \text{pow}((1 + \text{rate}), \text{year})$$



## 例2-5 程序

```
# include <stdio.h>
# include <math.h>
int main(void)
{
    int money, year;
    double rate, sum;
    printf("Enter money:");
    scanf("%d", &money);
    printf("Enter year: ");
    scanf("%d", &year);
    printf("Enter rate:");
    scanf("%lf", &rate);
    sum = money * pow((1 + rate), year);
    printf("sum = %.2f", sum);
    return 0;
}
```



# 调用scanf函数输入多个数据

```
scanf("%d%d%lf", &money, &year, &rate);
```

用户只需输入: 1000 3 0.025

输入数据之间用“空格”隔开

- 输入参数的类型、个数和位置要与格式控制说明一一对应
  - 否则, 程序很可能意外崩溃 ! ! !





## 2.4 循环控制结构(for 循环)

```
for(表达式1; 表达式2; 表达式3) {  
    循环体语句  
}
```

表达式1只执行一次





## 2.4.3 指定次数的循环程序设计

求  $1 + 2 + \dots + n$

变量设置

```
int i, n, sum;
```

初值:

```
i=1; sum=0;
```

输入 n: `scanf("%d",&n)`

工作

```
sum = sum + i
```

```
i ++; 或者 i = i+1;
```

工作条件

```
i<=n
```

```
int i, n, sum;  
sum = 0;  
for( i=1; i<=n; i++ )  
    sum = sum + i;
```



for(表达式1 ; 表达式2 ; 表达式3)

## 循环体语句

**表达式1** : 给循环变量赋初值, 指定循环的起点。

`i = 1`

**初始化**

**表达式2** : 给出循环的条件, 决定循环的继续或结束。

`i <= n`

**条件判断**

**表达式3** : 设置循环的步长, 改变循环变量的值, 从而可改变表达式2的真假性。

`i ++`

**更新控制变量**

**循环体语句** : 被反复执行的语句。 **主要任务**

`sum = sum + i;`



## 2.4.3 指定次数的循环程序设计

求  $1 - 1/3 + 1/5 - 1/7 + \dots$  的前n项和

变量设置

`int i, n, sum, item, flag;`

初值：

输入 n: `scanf("%d",&n)`

`sum=0; i=1; flag=1;`

工作

`item = flag*1.0/(2*i-1)`

`sum = sum + item`

`i ++; 或者 i = i+1;`

`flag = - flag`

工作条件

`i<=n`

```
sum = 0;
flag = 1;
for( i=1; i<=n; i++ )
{
    item = flag*1.0/(2*i-1);
    sum = sum + item;
    flag = - flag;
}
```



## 例2-9 求 $n! = 1*2*...*n$

### 变量设置

int n, i;

double product;

### 初值:

输入 n: scanf("%d",&n)

product=1; i=1;

### 工作

product = product \* i;

i ++; 或者 i = i+1;

### 工作条件

i<=n

```
product = 1;  
for( i=1; i<=n; i++ )  
    product = product * i;
```



## 2.5 生成阶乘表

输入一个正整数 $n$ ，输出 $0!$ 到 $n!$ 的值。

变量设置

```
int n, i;
```

初值：

```
输入 n: scanf("%d",&n);
```

```
i = 0
```

工作

计算 $i$ 的阶乘;

输出阶乘值;

```
i++;
```

工作条件

```
i<=n
```

定义函数：**double fact( int i )**  
计算 $i$ 的阶乘



## 例2-11 生成阶乘表

```
#include <stdio.h>
double fact(int n);
int main(void)
{
    int i, n;
    printf("Enter n:");
    scanf("%d",&n);
    for( i=0; i<=n; i++ )
        printf("%d!=%.0f\n", fact(i));
    return 0;
}
```

```
double fact(int n)
{
    int i;
    double product = 1;
    for ( i = 1; i <= n; i++ )
        product = product * i;
    return product;
}
```



# 结构化程序设计思想

- 结构化程序设计(Structured Programming)
  - 程序设计技术
  - C语言是结构化程序设计语言
- 强调程序设计的风格和程序结构的规范化, 提倡清晰的结构
- 基本思路: 自顶向下, 复杂问题 □ 划分为若干**更为简单的**阶段





# 本章要点

- 输入输出函数printf, scanf
- 变量定义、使用、赋值
- 算术运算
- if 语句
- for 语句