

# C语言与程序设计

## The C Programming Language

### 第1章 概述

华中科技大学计算机学院  
毛伏兵



群名称: C语言0910课堂2023秋  
群 号: 494790738



# 第1章 概论

## ■ 本章重点

1.2 学习C语言的第一个例子

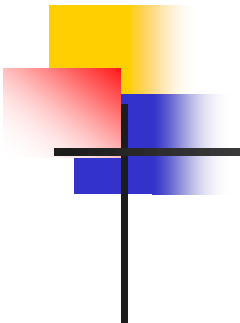
1.4 计算机系统及内存编址

1.5 数和字符的编码表示

1.6 算法及其表示

# 1.1 程序设计语言与程序设计

- 了解什么是程序设计语言，了解什么是程序，了解什么是程序设计，就是了解本门课程的基本研究对象。这些对于学习语言和程序设计来讲是非常必要的。
- 计算机语言包括机器语言、汇编语言和高级语言这三类语言。如果不涉及汇编语言，程序设计语言往往就是指高级语言。



## 1.2 例1--输出:Hello,world!

```
/* printf “Hello,world !” in screen */  
#include <stdio.h>  
int main( )  
{  
    printf ( "hello, world!\n " ) ;  
    return 0;  
}
```

```
/* printf “Hello,world !” in screen */
```

**Comment**  
注释

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

## Include information about standard library 文件包含

```
int main()
```

# Main function 主函数

{

## Body 函数体

```
printf ( "hello, world!\n " ) ;
```

}

## Library function 库函数

## Arguments 函数参数



进一步： 输入名字，计算机问候自己.

名字---字符串，

- 如何输入字符串？
- 如何保存？
- 如何输出保存的信息？



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

---

```
int main(void)
```

```
{
```

```
    char name[20];
```

```
    printf("Input your name please!\n");
```

```
    scanf("%s", name );    // vs(Visual Studio)  scanf_s
```

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

```
    return 0;
```

```
}
```

如何输入成绩?

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

```
int main(void)
```

```
{
```

```
    char name[20];
```

```
    int score;
```

```
    printf("Input your name and score, please!\n");
```

```
    scanf("%s", name );    // fgets(name,20,stdin); 最多读19个字符
```

```
    scanf("%d", &score); // decimal string
```

```
    printf("Hello %s!, your score is %d\n", name, score );
```

```
    return 0;
```

```
}
```

【思考】scanf的%s格式和fgets使用中的区别

【练习】不同类型数据的输入输出 %c %d %f %s

如何定义函数？让计算机问候某人。





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

```
void show(char str[ ]);           // 函数说明
```

---

```
int main(void)
```

```
{   char name[20];
```

```
    printf("Input your name please!\n");
```

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

```
    show(name);                   // 函数调用
```

```
    return 0;
```

```
}
```

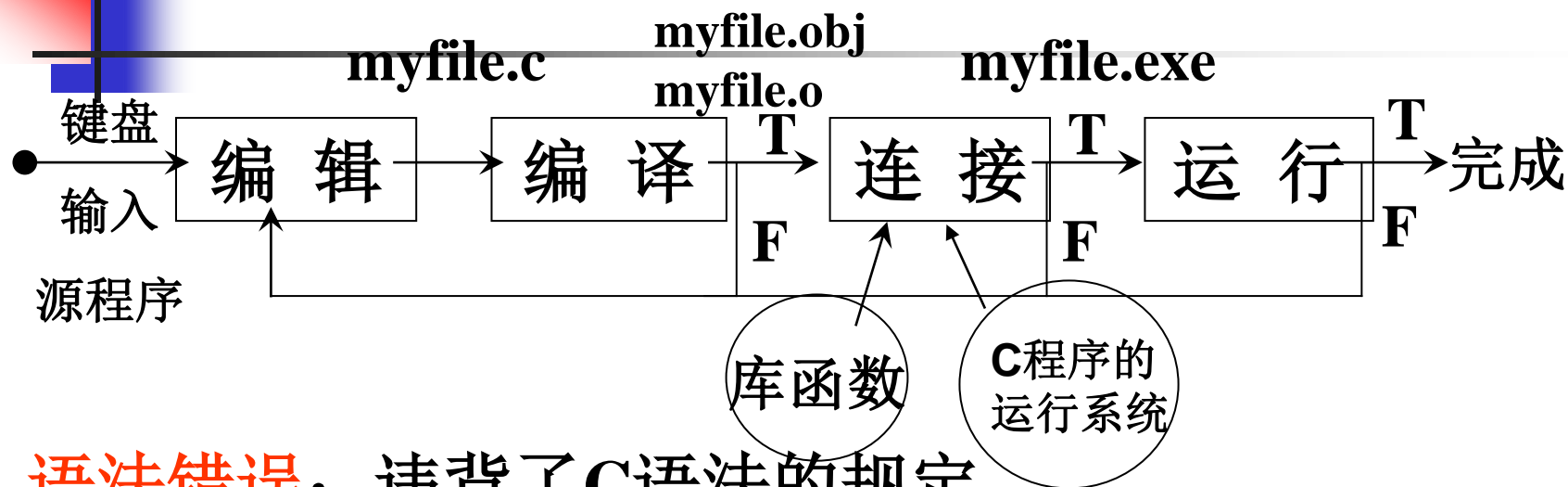
```
void show ( char str[ ] )        // 函数定义
```

```
{
```

```
    printf("Hello %s!\n", str);
```

```
}
```

# 运行C程序的步骤



## 语法错误：违背了C语法的规定

实际错误可能出现在编译程序指定错误位置前面很远处；一个实际错误有时会使编译程序产生许多出错信息行。

应当注意系统给出的警告信息（**WARNING**）。警告常预示着隐藏较深的实际错误，必须认真弄清原因

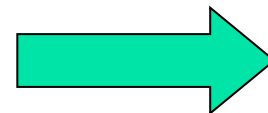
**逻辑错误**：程序无违背语法规则，但程序执行结果与原意不符，这是因为程序设计人员通知给系统的指令与原意不相同，即出现了逻辑上的混乱



# 作业

---

- 1、安装VC或C:B编译器或Dev C++，熟悉运行C程序的步骤。

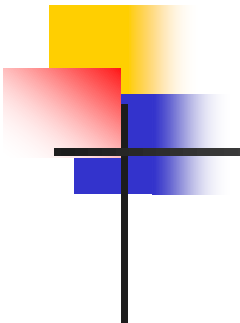


# 1.3 C语言的产生、发展与语言特征

## 1.3.1 C语言的产生与发展



图1.1 C语言的继承、产生与发展历程



## 1.3.2 C语言的标准化

- 以1978年K&R C为代表的C语言被称为**传统的C语言**
- 1989年底公布美国第一个C语言的国家标准ANSI 89，简称**C89**
- 1990年，国际标准化组织ISO将其接受为C语言的国际标准，称为ISO/IEC 9899-1990。它是C语言的第一个国际标准，也称为标准C，简称**C90**。
- ISO/IEC在1995年公布了一个新的C语言标准草案，称为C95，供讨论和征求意见。
- 接着ISO/IEC在1998年又公布新标准的草案WG14/N843和WG14/N897，进一步就C语言标准的完善征求意见。
- ISO/IEC于1999年12月公布了C语言国际标准ISO/IEC 9899:1999 (E)，**C99**。它是C语言国际标准9899的第二版。
- ISO/IEC于2011年12月公布了C语言国际标准ISO/IEC 9899:2011，**C11**。它是C语言国际标准9899的第三版。



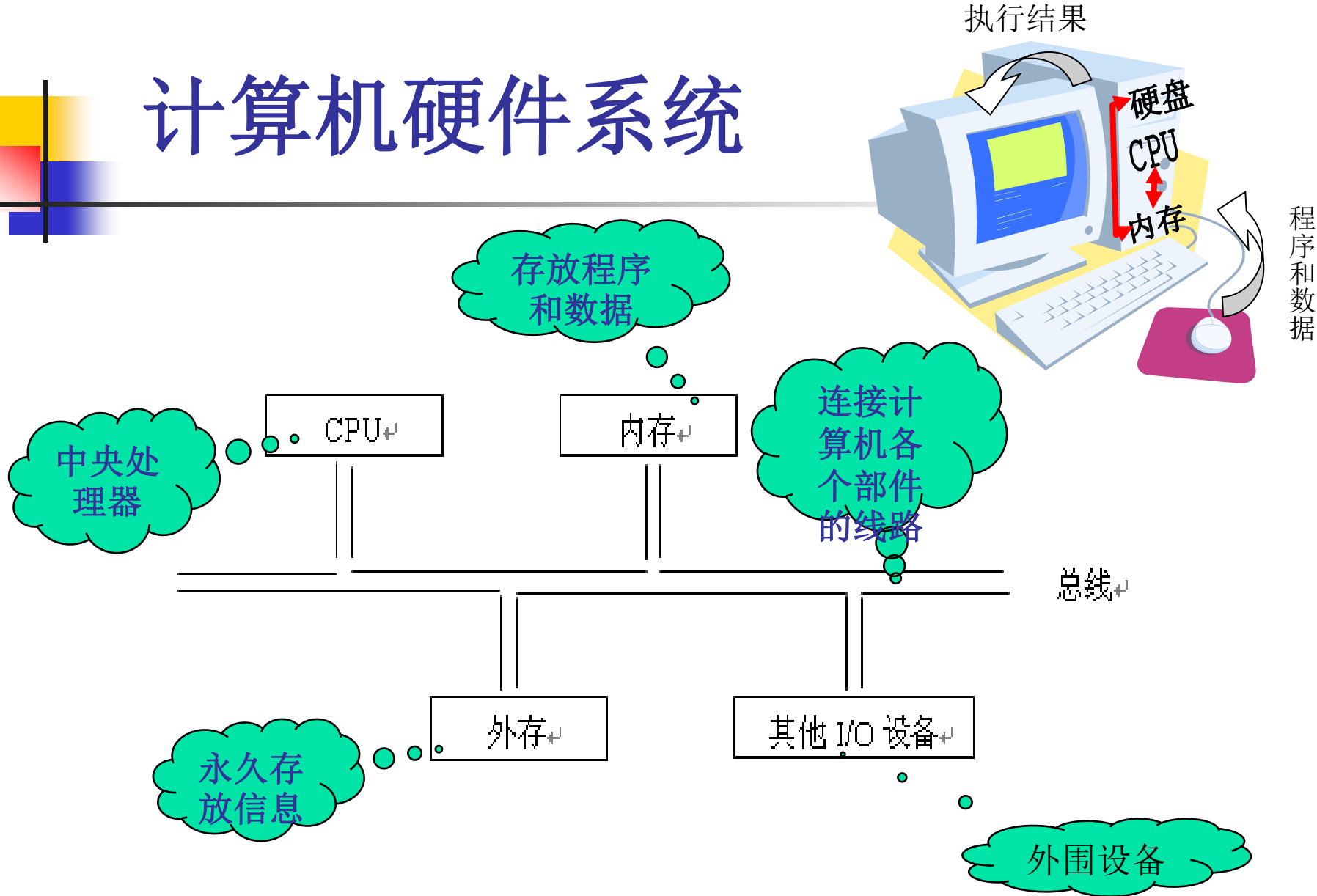
# 1.3.3 C语言的特征

- 1) 语言简洁紧凑
- 2) 目标代码质量高
- 3) 语言表达能力强
- 4) 流程控制结构化
- 5) 弱类型
- 6) “中级语言”特性
- 7) 书写自由、使用灵活
- 8) 可移植性好

## 1.4 计算机系统及内存编址

- 一台计算机硬件系统由**CPU**、**内存**、外存、其他I/O设备和总线组成。
- **CPU** 又称为中央处理器 。 **CPU** 又是由运算器、控制器、指令计数器、内部寄存器、标志寄存器等部件组成。
- **内存** 又称为**内存存储器**，用以存放程序和数据。
- 总线是用以连接计算机各个部件的线路，它完成各个部件之间的信息传送。根据传送信号种类的不同，总线又分成地址总线、数据总线和控制总线。

# 计算机硬件系统





# 内存地址

- 内存以字节为单位线性连续编址。即按照0x0000, 0x0001, 0x0002, ...的方式; 从低地址端开始向高地址端为每一个内存字节进行顺序连续编号。

0x0000	
0x0001	
0x0002	
.	
.	
.	

# 字长

- CPU数据总线的宽度（bit数）称为计算机的机器字长。对8位CPU，机器字长为1字节；对16位CPU，机器字长为2字节；对32位CPU，机器字长为4字节。

0x0000	
0x0001	
0x0002	
.	
.	
.	



# 1.5 数和字符的编码表示 (自学为主)

## 计算机采用二进制表示法

- 计算机为什么用二进制呢？
- 为什么不用我们日常熟悉的十进制呢？
  - 二进制在电器元件中容易实现
  - 计算机进行二进制运算比进行十进制运算要简单得多

但是C中没有二进制数（不能直接输入输出二进制数），只支持十进制、八进制和十六进制数的表示。

- 进位制数的运算
- 进位制数之间的转换？



## 1.5.1 进位计数制

二进制数: 只有0、1, 逢2进位1

八进制数: 0~7, 逢8进位1

在C语言中, 通过加前导零的方式来表示一个数是八进制数。如: **0136**表示的是八进制数  $(136)_8$ 。

十六进制数: 0~9 a~f(或A~F), 逢16进位1

在C语言中, 通过加前导0x或前导0X的方式来表示一个数是十六进制数。如: **0x2d5b**表示的是十六进制数  $(2d5b)_{16}$ 。

## 1.5.2 进位制数之间的转换

### 1、二、八、十六进制转换为十进制

用按权展开的方式，以二进制为例：

$$\begin{array}{ccccccc} 1 & 1 & 0 & 1 & 1 & . & 1 & 0 & 1 & (2) \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & & \uparrow & \uparrow & \uparrow & \end{array}$$

$$2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \ 2^{-1} \ 2^{-2} \ 2^{-3} \quad \text{权值}$$

$$= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + \\ 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

$$= 27.625$$

## 2. 十进制整数转换为二、八、十六进制整数

### 用除2（8、16）取余法

**例1.6** 用除2取余法求十进制整数 189 的二进制表示。

**解**

$2 \overline{) 189}$	
$2 \overline{) 94}$	……余 1 (最低位)
$2 \overline{) 47}$	……余 0
$2 \overline{) 23}$	……余 1
$2 \overline{) 11}$	……余 1
$2 \overline{) 5}$	……余 1
$2 \overline{) 2}$	……余 1
$2 \overline{) 1}$	……余 0
0	……余 1

$\therefore (189)_{10} = (10111101)_2$



### 3、二、八、十六进制整数的相互转换

---

二  $\rightarrow$  八进制数 : 3位分组法

二  $\rightarrow$  十六进制 : 4位分组法

$(10111101)_2 : (275)_8$

$(10111101)_2 : (bd)_{16}$



## 1.5.3 数的机器码表示

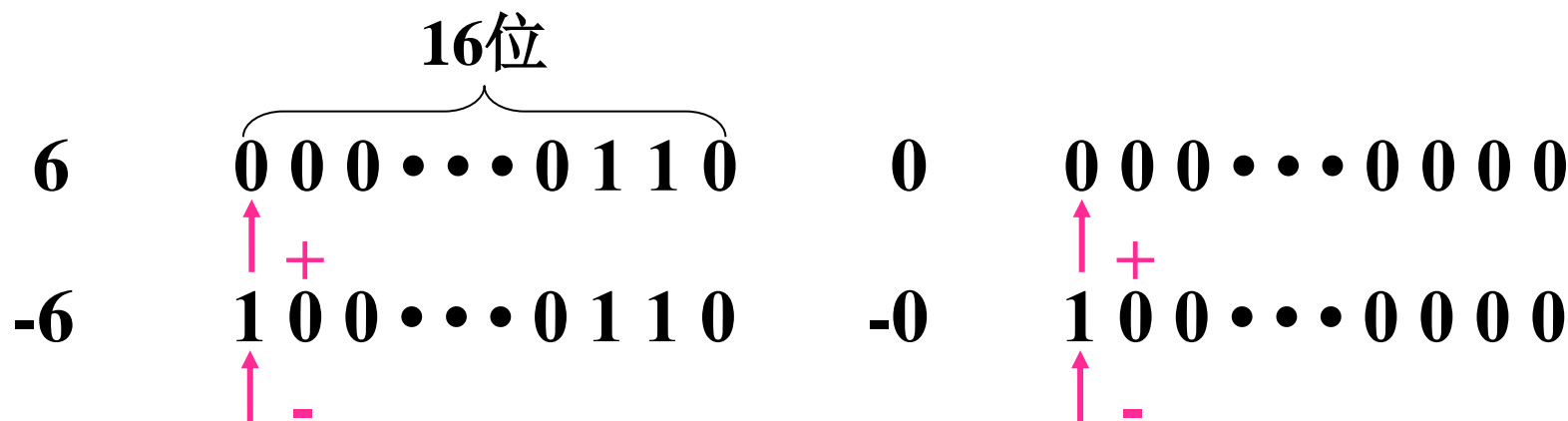
### 1. 机器数与真值

- 机器数：在计算机中，将一个数的最高位定义为符号位，其余各位为数值位。并且规定符号位之值为0表示正，符号位之值为1表示负。用这种方法表示的数称为机器数。
- 真值：机器数的数值称为该机器数的真值。也就是正、负号后跟二进制数的绝对值就构成真值。



## 2. 原码

最高位为符号位，其余各位为该数的绝对值。



0的原码表示不惟一

### 3. 反码

- 正数：同原码
- 负数：最高位为1，其余各位对原码取反

6      0 0 0 ... 0 1 1 0

-6      1 1 1 ... 1 0 0 1

0      0 0 0 ... 0 0 0 0

-0      1 1 1 ... 1 1 1 1

## 4. 补码

正数：同原码

负数：反码 + 1

$$\begin{array}{r} 6 \quad 000 \cdots 0110 \\ -6 \quad 111 \cdots 1001 \\ \quad \quad \quad +1 \\ \hline 111 \cdots 1010 \end{array}$$

$$\begin{array}{r} 0 \quad 000 \cdots 0000 \\ -0 \quad 111 \cdots 1111 \\ \quad \quad \quad +1 \\ \hline 000 \cdots 0000 \end{array}$$

计算机是以补码形式存放数据的

(钟表原理)

## 5. 码的内存存储形式

- 码是指计算机内存中存储的数据。根据存储数据的性质，码可以分为**数值码**、**字符码**和**图象码**。
- **数值码**用于表示数的大小，常见的有整数和浮点数。
  - 整数：补码形式存放
  - 浮点数：见第2章
- **字符码**用于表示英文字母、标点符号、控制字符以及汉字和其它文字的编码。
- **图象码**是用于表示图象中像素的颜色或灰度。虽然它们都是以二进制码的表现形式存储在内存之中，但物理意义却各不相同。
- **C语言的基本处理对象是字符和数字。在16位计算机中，整型数占2个字节，在32位计算机中，整型数占4个字节。**



## 1.5.4 字符的编码表示

- 1. 字符的编码的概念
- 计算机在屏幕上能够输出英文、汉字，乃至各种各样的民族文字，其基础在于对字符进行编码以及构造相应的点阵字模。
- 用户的字符输入通过编码在内存中以二进制数码存储，机器内部对字符的处理实际上是对字符编码的处理。
- 输出的时候则是通过字符的编码去索引对应的点阵字模到内存中，然后进一步通过内存中的点阵字模去控制显象管，在屏幕的指定位置将字符显示出来。



## 2. ASCII码

---

- 西文字符的编码国际上采用的是**ASCII码**（**American Standard Code For Information Interchange**），是美国国家信息交换标准字符码的英文缩写。
- 完整的**ASCII码**字符集请参阅**附录1**。
- **ASCII码**采用单字节编码，并且规定字节的最高位留做校验位，只有低7位参与编码。



## 3. 汉字编码

- 根据标准规定和计算机的输入、存储和显示过程，汉字编码有许多方式。常用的有国标码、区位码、拼音码、五笔字型码和汉字的机内码。
- 拼音码和五笔字型码用于汉字的输入。
- 国标码：**1981年颁布的《通用汉字字符集(基本集)及其交换码标准》GB2312-80。**
- **采用两字节编码方案。**

# 1.6 算法及其表示

## ■ 1.6.1 算法的定义

**算法是指为解决某个问题所采取的方法和步骤，并且要具备下列性质。**

- (1) 有零个或多个执行算法所需要的数据作为输入。
- (2) 有一个或多个经算法处理的数据作为输出。这种数据也称为算法的执行结果。
- (3) 应只包含有限个步骤，且执行算法所需的时间也是有限时间。算法的这种性质称为算法的有限性或有穷性。
- (4) 算法每一步所规定的动作应该是唯一确定的，没有二义的。算法的这种性质称为算法的确定性。



## 1.6.2 算法的表示

- 选用一种合适的工具或方法来表达算法设计者的算法思想称为算法的表示或算法的描述。
- 算法的表示方式多种多样，如自然语言方式，流程图方式、表格方式、伪码方式,等等。
- 自然语言方式显然非常容易理解，但是在将算法转换为具体程序方面不够直观，它最适合表达算法的设计思想。
- 流程图方式则非常具体，易于将它描述的算法编写成具体的程序，但是在程序规模变大的情况下，用流程图来完整描述算法不仅多余，而且不太可能。因此，流程图方式尤其适合初学者。本书重点介绍流程图方法。

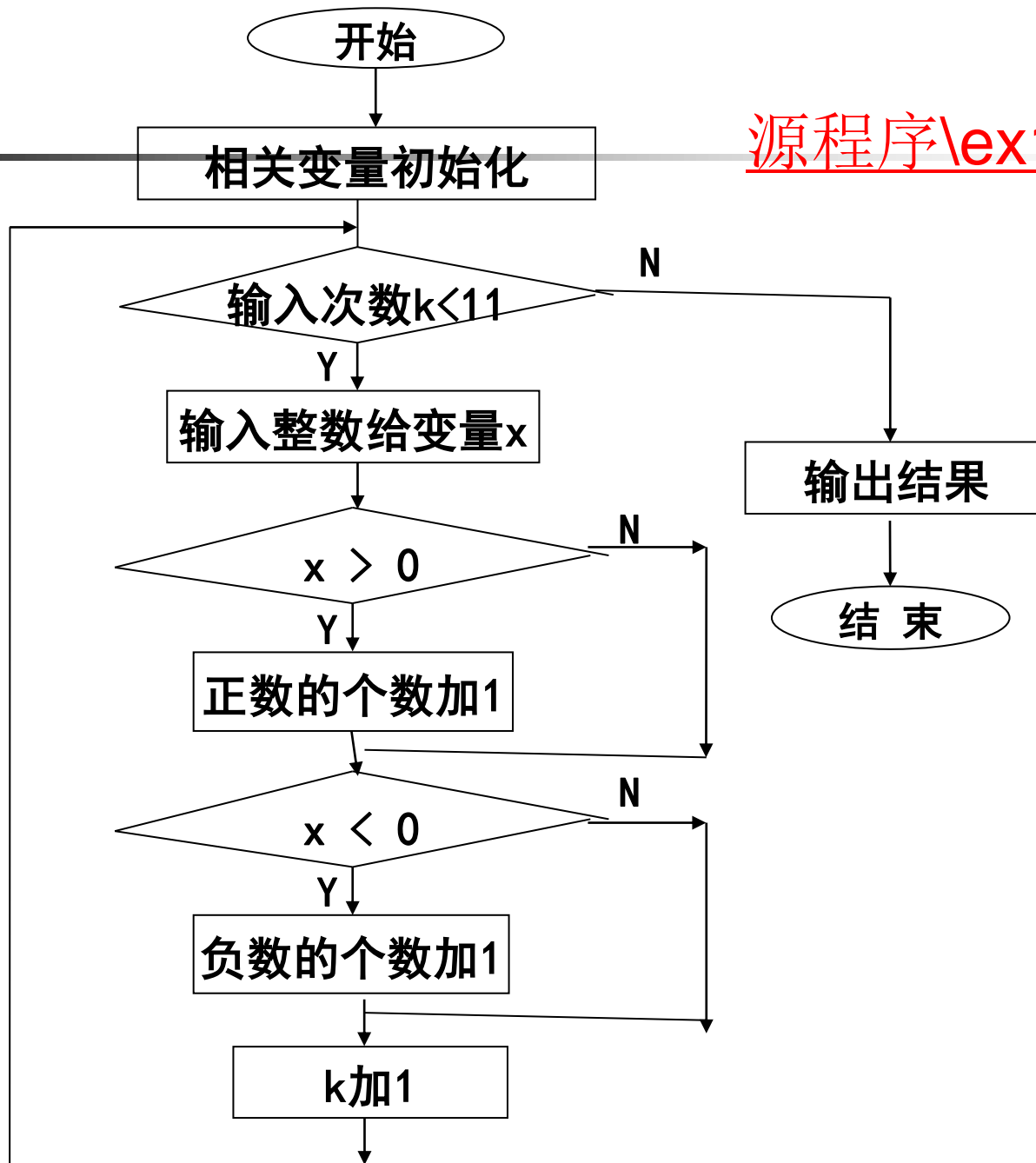
# 1. 流程图的图符

- 流程图又称为框图。它是用它的图符来表示算法的每一步骤以及这些步骤之间的相互联系。学习用流程图来表达自己的算法思想首先要从认识组成流程图的图符的功能入手。流程图图符及其功能由表1.4 描述。



## 2. 用流程图描述算法

- 问题：输入**10**个整数，统计正数和负数的个数。
- 算法设计  
变量**x**：存储输入的整数，  
**k**：表示输入的次数，  
**positive**：表示正数的个数，  
**negtive**：表示负数的个数。



# 思考及练习

(1) 扩充例1.15的功能

(如求正数和及平均值)

(2) 例1.15固定输入10个数，怎样实现输入任意个数？

(3) 仿写：求最值

输入10个整数，求最大值和最小值

(3) 仿写：求  $n!$ ， $n$ 键盘输入

将求阶乘的功能定义为函数



## 1.7 学习C语言与程序设计的方法

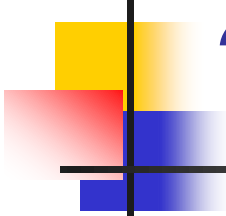
作为初学者，**如何学会、掌握、以及熟练使用C语言？**  
如何用C语言去解决实际工作中的应用问题呢？这主要涉及到**四个方面**的学习和训练。

- 一、学习并理解C语言的语法和语义
- 二、学习并掌握一些基本数据结构和常用算法的设计
- 三、学习并熟悉C语言的集成开发环境
- 四、熟悉C语言相关的库函数

# 1) 学习并理解C语言的语法和语义

- 语义指的是构成C语言的各种符号的含义。
- 由于C程序是由C语言的符号所组成，因此只有准确理解各种符号的语义，才有可能合适的使用各种符号来表达自己的编程思想。
- 另一方面，C的语法规定了C语言符号集中各种符号的结合方式，规定了各种表达式、语句、函数、乃至程序结构等的构造规则。
- 只有理解和掌握C语言的语法，才有可能正确的用各种符号去构造程序的各种成分，并进而构成满足算法要求、能解决实际应用问题的程序。





## 2) 学习并掌握一些基本数据结构和常用算法的设计

- C语言对机器硬件有很强的操纵能力，在大多数情况下可以代替汇编语言。因此，要学好用好C，需要对字符、整型变量、数组元素在内存中的存储形式，对原码、补码，对字节中的位（bit）的操作做到熟练掌握。
- 深刻理解各类指针及所指对象的声明、使用以及物理含义。对结构，结构数组、链表都要熟练掌握。
- 由于文本是一个基本研究对象，对文本操作的各种算法，如：统计字符个数、单词个数、行数，字符串的各种操作所涉及的各种算法，以及教材例子中给出的各种算法（如排序、查找等）都应该熟练掌握。这样才能奠定程序设计的扎实基础。



### 3) 熟悉集成开发环境

- 几乎所有的程序设计语言都有自己的集成开发环境。集成开发环境将源程序的创建、编辑，以及其后的编译、链接，乃至程序的调试与跟踪全部都集成于一体。为编程者提供了方便的程序开发环境。因此，编程者要能够进行程序的设计和开发，调试与跟踪，就必须学会使用相应的集成开发环境。编程者只有学会使用集成开发环境提供的这些功能，才能进行程序的设计与开发。



## 4) 熟悉库函数

- 在C语言中，编译系统提供了一些标准的库函数。如输入/输出的标准库函数、字符串处理的标准库函数、数学计算的标准库函数等。编程者只要知道这些标准库函数的功能、调用方式、返回何种结果，并且能够在自己的程序中调用这些标准库函数就行了。完全没有必要自己去编写这些函数。



# 本章小结

- 本章首先通过介绍学习C语言程序设计的第一个例子，说明了如何创建并运行第一个C程序同时，对第一个C程序进行了解释分析。
- 从C语言的产生与发展，C语言的标准化，以及C语言的特征三个方面对C语言进行了概述。
- 简单介绍了计算机硬件系统的概念。
- 从进位计数制、数的机器码表示，以及字符的编码表示三个方面介绍了数和字符的机器码表示。
- 尤其是ASCII码，要熟练掌握。
- 在算法方面，介绍了算法的定义、算法的表示、算法的程序实现。
- 最后，从学习并理解C语言的语法和语义、学习并掌握一些基本数据结构和常用算法的设计、熟悉集成开发环境，以及熟悉库函数四个方面介绍了C语言的学习方法。