

# 第一章

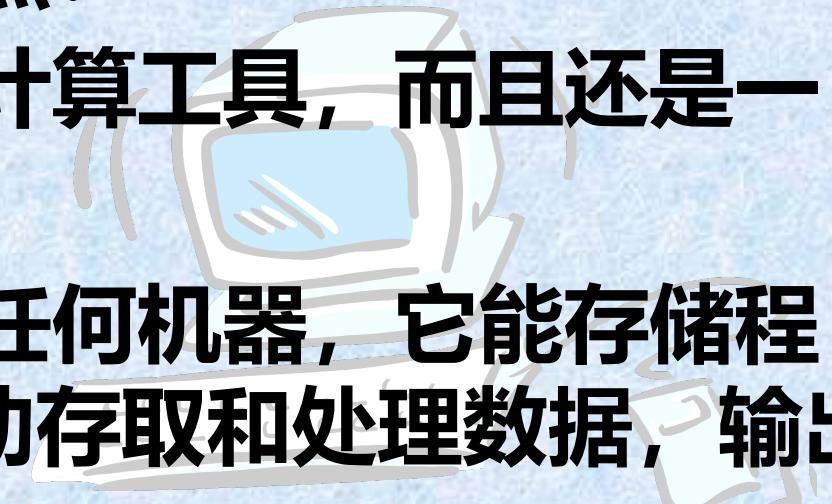
# 计算机基础知识

# 1.1 计算机概述

## 一、什么是计算机

计算机是一种能对各种信息进行存储和高速处理的电子机器(或**工具、助手**)。

对上述定义要强调两点：

- 
- ①计算机不仅是一个计算工具，而且还是一个信息处理机。
  - ②计算机不同于其它任何机器，它能存储程序，并按程序的引导自动存取和处理数据，输出人们所期望的信息。

# 1.1 计算机概述

## 二、计算机的分类

### 1.按处理对象分类

**数字计算机**: 处理非连续变化的数据, 这些数据在时间上是离散的。其基本运算部件是数字逻辑电路。

**模拟计算机**: 处理连续变化的数据, 这些数据在时间上是连续的。其基本运算部件是由运算放大器构成的通用函数运算器等组成。

**混合计算机**: 可处理数字量和模拟量。

# 1.1 计算机概述

## 2.按用途分类

**通用计算机**：为了能够解决各种问题，具有较强的通用性而设计的计算机。

它具有一定的运算速度和存储容量，带有通用的外设，配备各种系统软件和应用软件。

**专用计算机**：为了解决一个或一类特定问题而专门设计的计算机。

其软硬件的配置依据解决问题的需要而定。

# 1.1 计算机概述

## 3.按规模和处理能力分类(IEEE)

**巨型机**: 超级计算机,功能最强,价格最贵。

**小巨型机**: 与巨型机相比, 价格大幅降低。

**大型机**: 主机, 具有很强的管理和处理数据的能力, 在大企业、银行等单位使用。

**小型机**: 中小企业, VAX-II, DJS-2000。

**工作站**: 高档微机, 具有很强的图形处理能力, 应用于计算机辅助设计, Sun工作站。

**个人计算机**: IBM PC, Apple

# 1.1 计算机概述

## 三、计算机的特点

1. **运算速度快**: 每秒数万亿次, 气象预报
2. **计算精度高**: 理论上不受限制, 圆周率
3. **存储能力强**: 中等规模图书馆
4. **具有逻辑判断能力**: 算术运算

逻辑运算  
判断或比较

5. **具有自动执行能力**: 无需人工干预

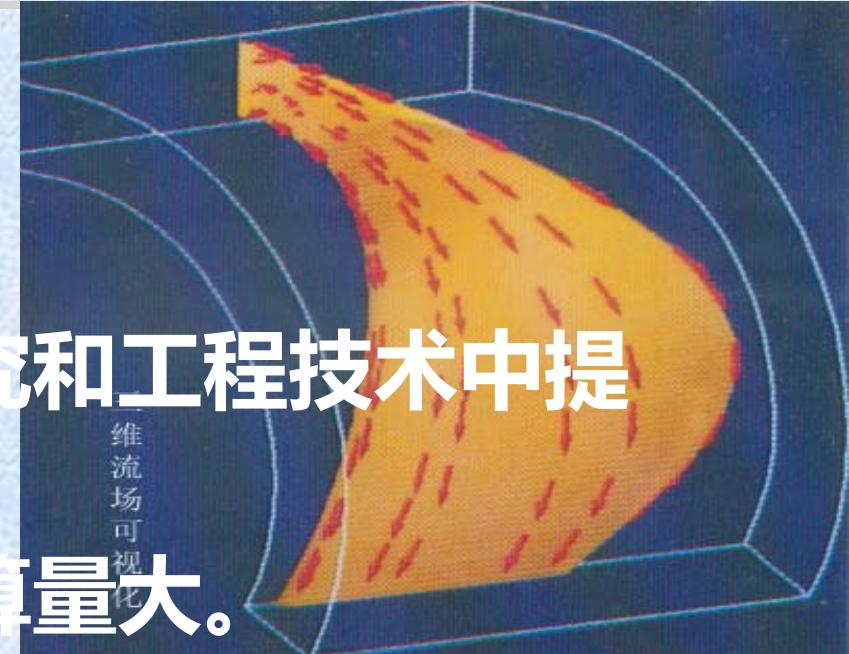
# 1.1 计算机概述

## 四、计算机的应用领域

### 1.科学计算或数值计算

利用计算机来完成科学的研究和工程技术中提出的数学问题的计算。

实际问题→数学模型→计算量大。



### 2.数据处理或信息处理

指对数据进行收集、存储、整理、分类、统计、加工、检索和传播等一系列活动的统称。

信息时代海量数据的管理和有效利用。

# 1.1 计算机概述

## 3. 过程控制或实时控制

利用计算机及时采集检测数据，按最优值迅速地对控制对象进行自动调节或自动控制。

无人自动化工厂。

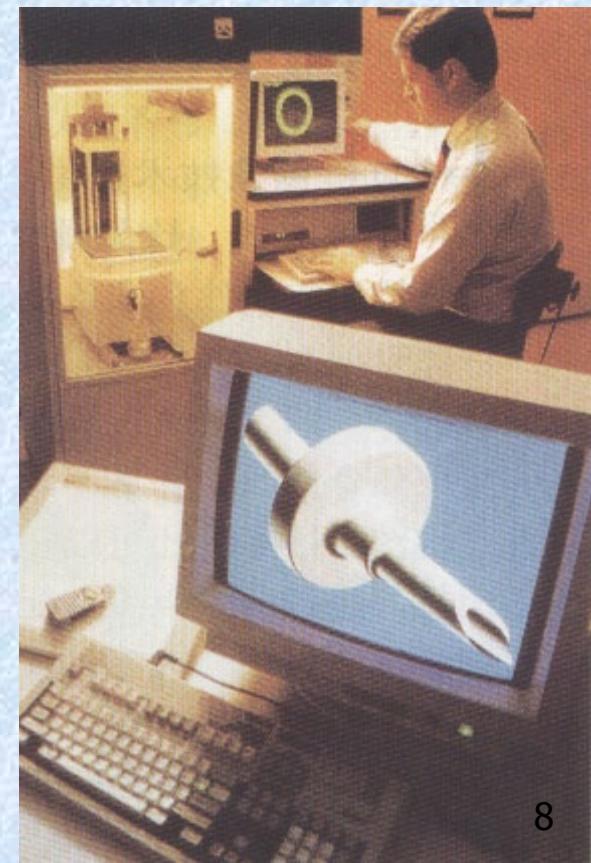
## 4. 计算机辅助技术

计算机辅助设计：CAD

计算机辅助制造：CAM

计算机集成制造系统--CIMS

计算机辅助教学：CAI



# 1.1 计算机概述

## 5. 人工智能

利用计算机模拟或部分模拟人的智能活动，  
如感知、判断、理解、学习、图像识别等。

实用技术：智能机器人、专家系统

## 6. 通信网络

Internet → 网上银行、网上订票

网上教学、网上医疗  
网上税收、网上出版



# 1.2 计算机运算基础

$$1 + 1 = 10$$

二进制数

$$6 + 3 = 11$$

八进制数

$$9 + 9 = 12$$

十六进制

用来收集、传送、处理信息的计算机，  
由于实现技术的原因，不能采用十进制  
进行操作，因此人们采用其他进制数。

# 1.2 计算机运算基础

## 一、数制及其转换

### 1.数制的概念

数制是用一组固定的数码和一套统一的规则来表示数目的方法。

十进制数码: 0 1 2 3 4 5 6 7 8 9

十进制规则: 逢十进一

进位记数制	非进位记数制
表示数值大小的数码与它在数中的位置有关。 如十进制数: 123. 45	表示数值大小的数码与它在数中的位置无关。 如罗马数字: I , II , III, IV, ---, X

## 1.2 计算机运算基础

### 进位记数制的要素：

① **基数**：指各种进位记数制中允许选用基本数码的个数。例如十进制的数码有：

0,1,2,3,4,5,6,7,8,9→基数是10

② **位权**：每个数码所表示的数值等于该数码乘以一个与数码所在位置相关的常数，这个常数叫做权值。例如：

$$123.4 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1}$$

# 1.2 计算机运算基础

## 2. 常用的数制

数制	十进制	二进制	八进制	十六进制
数码个数	0, 1, ..., 9	0, 1	0, 1, ..., 7	0, 1, ..., 9, A, B, C, D, E, F
基数	10	2	8	16
规则	逢十进一 借一当十	逢二进一 借一当二	逢八进一 借一当八	逢十六进一 借一当十六
权	$10^i$	$2^i$	$8^i$	$16^i$
形式表示	Decimal	Binary	Octal	Hexadecimal

注: ①  $i$  为整数

$$\textcircled{2} (N)_R = a_{n-1}R^{n-1} + a_{n-2}R^{n-2} + \cdots + a_1R^1 + a_0R^0 + a_{-1}R^{-1} + \cdots + a_{-m}R^{-m}$$

其中:  $R$  表示基数,  $a$  表示某进制的数码

③几种进位计数制的对应关系

# 1.2 计算机运算基础

## 几种进制数之间的对应关系

十进制	二进制	八进制	十六进制
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

# 1.2 计算机运算基础

## 3.不同进制数的相互转换

### (1)二进制数与十进制数的互换

计算机→二进制，人→十进制

#### 二进制数转换成十进制数

按权展开，然后求和，就可把二进制数转换成十进制数。例如：

$$\begin{aligned}(101.1)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} \\ &= (?)_{10}\end{aligned}$$

# 1.2 计算机运算基础

## 十进制数转换成二进制数

十进制数有整数和小数两部分。

在转换时，整数部分采用**除2取余法**

小数部分采用**乘2取整法**

然后通过小数点将转换后的二进制数连接起来即可。例如：

$$(105.625)_{10} = (?)_2$$

# 1.2 计算机运算基础

## (2)二进制数与八进制数的互换

### 二进制数转换成八进制数

**三位并一位**：以小数点为基准，整数部分从右到左，小数部分从左到右，每三位一组，不足三位添0补足，然后把每组的三位二进制数按权展开后相加，得到相应的一位八进制数码，再按权的顺序连接即得相应的八进制数。

例如： $(1011100.00101011)_2 = (?)_8$

$(001, 011, 100. 001, 010, 110)_2 = (134. 126)_8$

1      3      4 . 1      2      6

# 1.2 计算机运算基础

## 八进制数转换成二进制数

**一位拆三位**：将每一位八进制数写成对应的三位二进制数，然后按权连接即可。

例如：  $(123.67)_8 = (?)_2$

1    2    3 . 6   7    (八进制)

001, 010, 011. 110111   (二进制)

$(123.67)_8 = (1010011.110111)_2$

# 1.2 计算机运算基础

## (3)二进制数与十六进制数的互换

### 二进制数转换成十六进制数

**四位并一位：**以小数点为基准，整数部分从右到左，小数部分从左到右，每四位一组，不足四位添0补足，然后把每组的四位二进制数按权展开后相加，得到相应的一位十六进制数码，再按权的顺序连接即得相应的十六进制数。

例如：  $(1011110.00011)_2 = (?)_{16}$

$(0101, 1110.0001, 1000)_2 = (?)_{16}$

5      E .    1      8

# 1.2 计算机运算基础

## 十六进制数转换成二进制数

**一位拆四位**：把一位十六进制数写成对应的四位二进制数，然后按权连接即可。

例如： $(123.EF)_{16} = (?)_2$

1      2      3 .    E      F      (十六进制)

0001, 0010, 0011. 1110, 1111    (二进制)

$(123.EF)_{16} = (100100011.11101111)_2$

十进制数：512D或512

八进制数：127Q

二进制数：1011B

十六进制：A8H

# 1.2 计算机运算基础

## 4. 计算机为什么采用二进制

**电路简单**：计算机是由逻辑电路组成，而逻辑电路通常只有两个状态。

**可靠性高**：两个状态表示的二进制两个数码，数字传输和处理不容易出错。

**运算简单**：二进制运算法则简单。

**逻辑性强**：计算机工作原理是建立在逻辑运算基础上的，逻辑代数是逻辑运算的理论依据。

# 1.2 计算机运算基础

## 二、存储单位及地址

### 1.位(bit, b)

位是计算机存储数据的最小单位，一个二进制位只能表示两种状态，如0、1。

### 2.字节(Byte, B)

字节是数据处理的基本单位，一个字节是由八位二进制数组成。

**1Byte=8bit** → 01000001

## 1.2 计算机运算基础

存储器容量大小的单位： KB、 MB、 GB、 TB

$$1\text{KB} = 2^{10} = 1024\text{B}$$

$$1\text{MB} = 1024 \times 1024 = 1048576\text{B}$$

$$1\text{GB} = 1024 \times 1024 \times 1024\text{B}$$

$$1\text{TG} = 1024 \times 1024 \times 1024 \times 1024\text{B}$$

存储体结构：



# 1.2 计算机运算基础

## 3.字(Word)

**字是CPU通过数据总线一次存取、加工和传送数据的长度。一个字通常由一个或若干个字节组成。字长越长，计算机性能越强。**

**常用的字长：8位、16位、32位、64位等。**

计算机数据

{ 数值型：整数、小数且有正负  
非数值型：数字、字母、汉字等

# 1.2 计算机运算基础

## 三、数值型数据表示

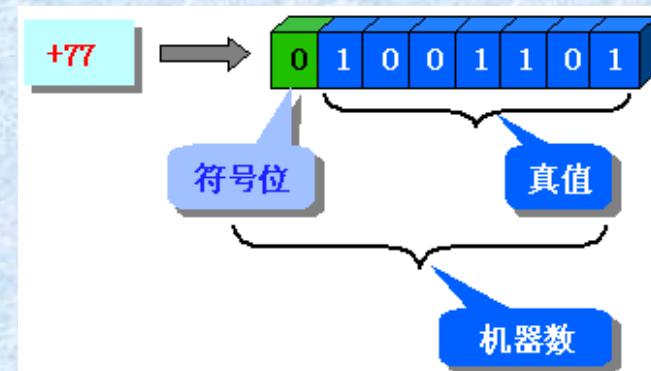
### 1. 机器数与真值

数值型数据(符号 + 数字) → 数码化

**规定**: + → 0, - → 1

**例如**:  $(+68)_{10} = (01000100)_2$

$(-68)_{10} = (11000100)_2$



**机器数**: 将符号和数字组合的二进制数

**真值**: 由机器数所表示的实际值大小

# 1.2 计算机运算基础

## 2. 原码、反码和补码

### (1) 原码

**规定：**用符号位和数值位表示一个带符号数

正数符号→0，负数符号→1

**例如：**求二进制数+10011, -10011的原码。

$$[+10011]_{\text{原}} = 00010011$$

$$[-10011]_{\text{原}} = 10010011$$

**又如：**求十进制数+65, -66的原码。

## 1.2 计算机运算基础

零的原码形式有两种：

$$[+0]_{\text{原}} = 00000000$$

$$[-0]_{\text{原}} = 10000000$$

原码表示数的范围：

8位： -127 ~ +127

16位： -32767 ~ +32767

用原码表示一个数，与真值之间转换方便。  
对乘除法比较合适，但对加减法容易出错。

## 1.2 计算机运算基础

### (2) 反码

**规定：**正数的反码与原码相同，负数的反码是对该数的原码除符号位外各位取反。

**例如：**求二进制数+10011, -10011的反码。

$$[+10011]_{\text{反}} = 00010011$$

$$[-10011]_{\text{反}} = 11101100$$

**零的反码形式有两种：**

$$[+0]_{\text{反}} = 00000000$$

$$[-0]_{\text{反}} = 11111111$$

任意数的反码的反码即是原码本身

# 1.2 计算机运算基础

## (3) 补码

**规定：**正数的补码与原码相同，负数的补码是对该数的原码除符号位外各位取反，末位加1。

**例如：**求二进制数+10011, -10011的补码。

$$[+10011]_{\text{补}} = 00010011$$

$$[-10011]_{\text{补}} = 11101101$$

**零的原码形式有两种：**

$$[+0]_{\text{补}} = 00000000$$

$$[-0]_{\text{补}} = 00000000$$

任意数的补码的补码即是原码本身

# 1.2 计算机运算基础

**补码表示数的范围：**

**8位： -128 ~ +127**

**16位： -32768 ~ +32767**

**引入补码后，减法运算可转换为加法运算。**

$$[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$$

$$[X - Y]_{\text{补}} = [X + (-Y)]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$$

**例如：用补码计算十进制数  $35 - 65 = ?$**

**目前计算机中加减法基本采用补码运算。**

# 1.2 计算机运算基础

## 总结

①一个正数的原码、反码和补码的表示形式相同，符号位置0，其它位是数的真值。

负数的原码	符号位→1	其余位是该数的绝对值
负数的反码	符号位→1	其余各位逐位取反
负数的补码	符号位→1	其余各位逐位取反，末位加1

②真值零的表示：

$[+0]_{原} = 00000000$	$[-0]_{原} = 10000000$	不唯一
$[+0]_{反} = 00000000$	$[-0]_{反} = 11111111$	不唯一
$[+0]_{补} = 00000000$	$[-0]_{补} = 00000000$	唯一

# 1.2 计算机运算基础

## 四、字符型数据编码

### 1.ASCII码

**American Standard Code for Information Interchange (ASCII, 美国标准信息交换码)。**

**国际通用的信息交换标准代码(ISO 646)。**

**ASCII码是对数字、字母、通用符号和控制符号等字符进行编码。128种**

**0000000→1111111**

# 1.2 计算机运算基础

高三位  
**b<sub>6</sub>b<sub>5</sub>b<sub>4</sub>**

## ASCII码表(7位)

低四位  
**b<sub>3</sub>b<sub>2</sub>b<sub>1</sub>b<sub>0</sub>**

	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	、	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

# 1.2 计算机运算基础

## ASCII码的字符集：

①10个数字： 0,1,2,3,4,5,6,7,8,9

②52个大、小写字母

③25个特殊字符

比较大小： 0<...<9<A<...<Z<a<...<z

[例]将China五个字符的ASCII码查出并存放在内存中。

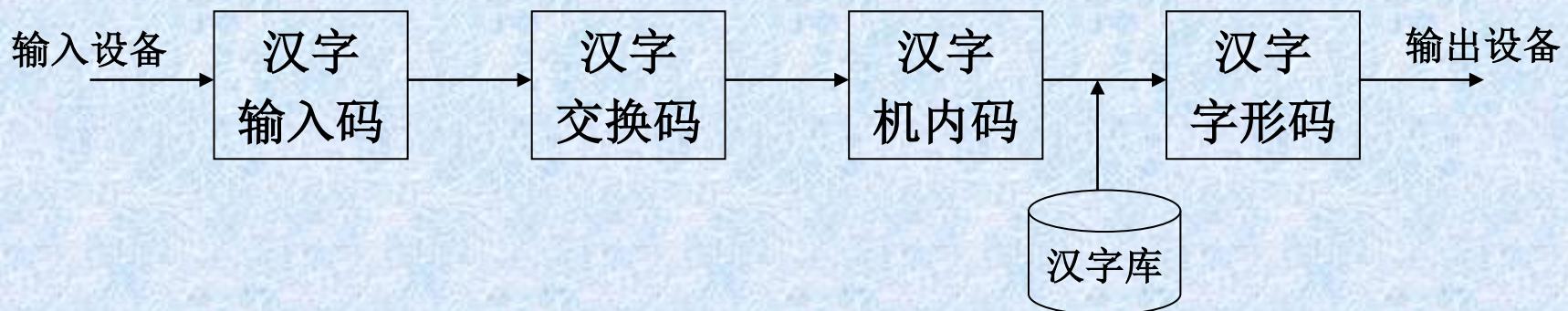
1000011	字符 C
1101000	字符 h
1101001	字符 i
1101110	字符 n
1100001	字符 a

# 1.2 计算机运算基础

## 2. 汉字编码

**汉字处理技术**：汉字输入、汉字输出、计算机内部的编码问题。

根据汉字处理过程中的不同要求，有多种编码形式。



# 1.2 计算机运算基础

## (1) 汉字输入码

**作用**：让用户直接使用标准键盘输入汉字。

**特点**：规则简单，重码率低，击键次数少。

**分类**：数字编码→电报码、区位码等

字音编码→全拼、双拼等

字形编码→五笔字型、郑码等

混合编码→自然码、智能ABC等

## 1.2 计算机运算基础

### (2)汉字交换码

在汉字信息处理系统与通信处理系统之间进行汉字信息交换时所使用的编码。

**设计汉字交换码编码体系要考虑：**

- ☆被编码的汉字个数尽量多；
- ☆编码的长度尽可能短；
- ☆编码具有唯一性；
- ☆码制的转换要方便。

按照国家标准GB/T-2312-1980编码的汉字交换码→国标码。

# 1.2 计算机运算基础

**国家标准GB/T 2312-1980：**

**信息交换用汉字编码字符集--基本集**

**图形字符 一级汉字3755个(按拼音排序)**

**(7445个) 二级汉字3008个(按部首排序)**

**字母、数字和特殊图形记号等**

**国标码规定：**一个汉字采用两个字节来表示

0XXX XXXX

第一字节

0XXX XXXX

第二字节

**例如：**啊→区位码→1601

**国标码 = 区位码 + 3232→4833**

# 1.2 计算机运算基础

## (3) 汉字机内码

汉字机内码是在设备和信息处理系统内部存储、处理、传输汉字用的代码。

目前我国使用的内码是国标码高位1。

国标码	—	0	1	xxx	xxxx	0	1	xxx	xxxx
机内码	—	1	0	xxx	xxxx	1	0	xxx	xxxx

汉字机内码 = 汉字国标码 + 8080H

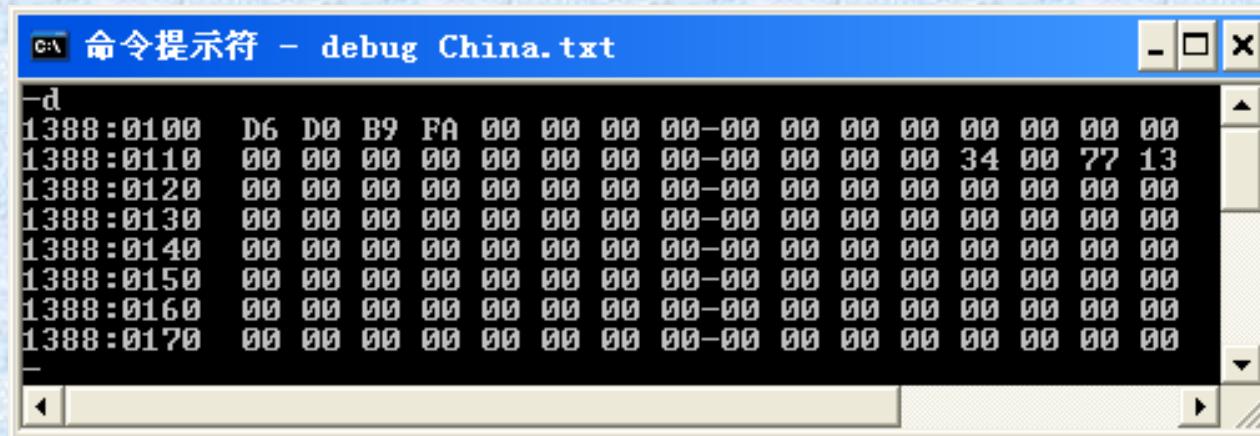
例如：啊 → 机内码 = 3021H + 8080H  
= B0A1H

# 1.2 计算机运算基础

又如：“中国” → 汉字机内码 = ?

汉字	区位码	汉字国标码	汉字机内码
中	5448	8680=5650H	D6D0H
国	2590	57122=397AH	B9FAH

通过Debug查看汉字机内码：



## 1.2 计算机运算基础

### Debug的由来：

1937年，美国青年霍德华·艾肯找到IBM公司为其投资200万美圆研制计算机，即马克1号，从这时起IBM公司正式跨进计算机领地。

为马克1号编制程序的是一位女数学家雷斯·霍波。有一天她在调试程序时出现故障，拆开继电器后，发现有只飞蛾被夹扁在触点中间，从而卡住了机器的运行。于是霍波把程序故障统称为臭虫(BUG)，把排除程序故障叫DEBUG。

## 1.2 计算机运算基础

### (4)汉字字形码

**字形码**是一种用点阵表示汉字字形的编码，它主要用于汉字输出(打印、显示等)时产生的汉字字形。

**点阵大小类型**： $16 \times 16$ 、 $24 \times 24$

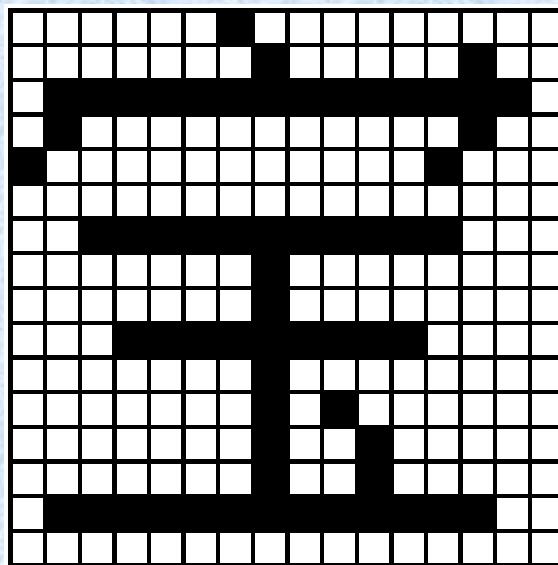
$32 \times 32$ 、 $48 \times 48$ 以上

**汉字库**：一个汉字系统所允许使用的全部汉字的汉字字形编码的集合。

## 1.2 计算机运算基础

**例如：**把一个方块横向和纵向都分为16格。  
若用1表示黑点，用0表示白点，则 $16 \times 16$ 的点阵汉字可用256位二进制数来表示，占用32B。

**汉字“宝”的 $16 \times 16$ 点阵数字化信息：**



02H 00H 01H 04H  
7FH FEH 40H 04H  
80H 08H 00H 00H  
3FH F8H 01H 00H  
01H 00H 1FH F0H  
01H 00H 01H 40H  
01H 20H 01H 20H  
7FH FCH 00H 00H

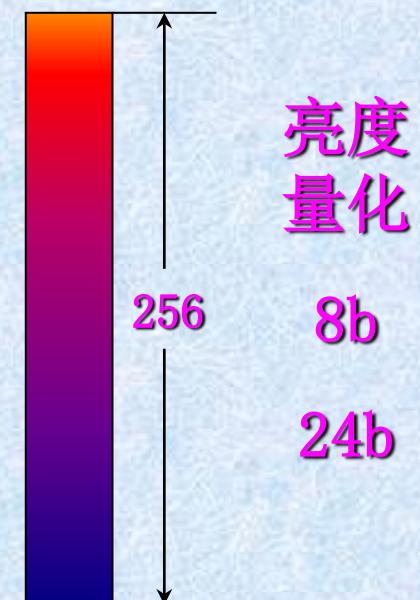
# 1.2 计算机运算基础

## 五、多媒体信息编码

### 1. 图像的数字化

连续空间位置的离散和数字化  
亮度值的离散和数字化

空间位  
置采样  
 $m \times n$   
 $20 \times 13$



# 1.2 计算机运算基础

## 图像的主要参数：

**图像分辨率：**指数字图像的实际像素数目，它反映图像在屏幕中显示的大小。

**颜色深度：**指记录每个像素所使用的二进制位数。

位数	颜色数	说明
4位	16种	Windows 3.x中画笔支持16种颜色
8位	256种	多媒体应用中的最低颜色深度
16位	32768种	RGB5:5:5, 剩余1位表示其它属性(透明度)
24位	16M种	真彩色, 超出人眼所能识别的颜色范围
32位	16M种	RGB8:8:8, 剩余8位表示其它属性(透明度)

## 1.2 计算机运算基础

### 图像数据量的计算：

图像文件的大小是指在磁盘上存储整幅图像所需的字节数。

**数据量 = 图像分辨率 × 颜色深度 / 8(B)**

**[例题]**一幅  $640 \times 480$  的真彩色图像，未压缩的图像数据量是多少？

$$640 \times 480 \times 24 / 8 = 921600 \text{B} = 900 \text{KB}$$

# 1.2 计算机运算基础

## 图像的文件格式

**BMP**: Windows标准图像文件格式

**JPG**: 一种高效率压缩格式(1:10 ~ 20)

**GIF**: 用于交换图片的，对灰度  
图像表现佳，但不超过256色的图像。



**PNG**: 流式网络图形格式，它使用LZ77  
派生的无损数据压缩算法。

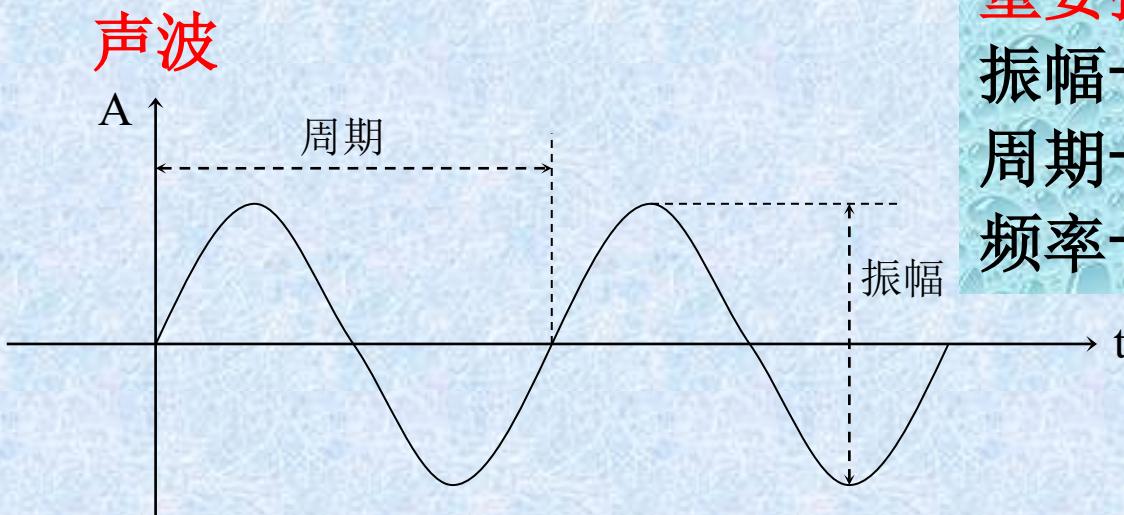
PNG存储灰度图像时图像深度达16位

PNG存储彩色图像时图像深度达48位

# 1.2 计算机运算基础

## 2. 声音数字化

**声音是通过一定介质传播的连续的波。**



**重要指标：**

振幅→音量的大小

周期→重复出现的时间间隔

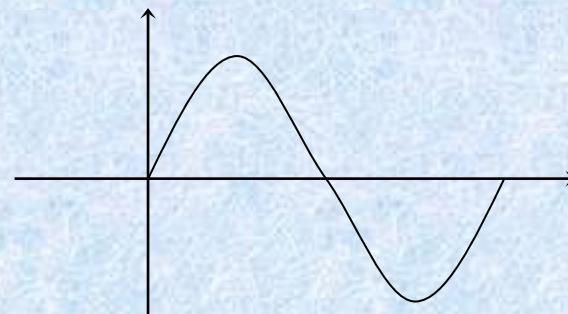
频率→信号每秒钟变化次数

# 1.2 计算机运算基础

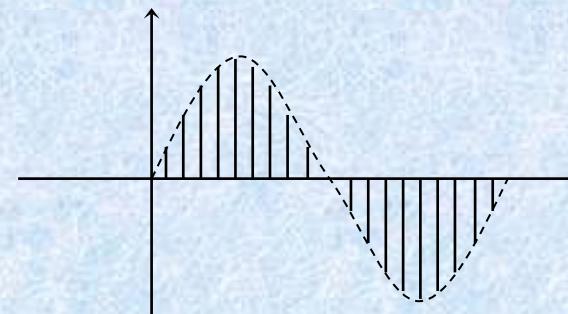
## 声音数字化过程：



连续的模拟声音信号



声音信号的采样



示意



离散的音频信号

# 1.2 计算机运算基础

## 声音数字化三要素

采样频率	量化位数	声道数
每秒钟抽取声波幅度样本的次数	每个采样点用多少二进制位表示数据范围	使用声音通道的个数
采样频率越高 声音质量越好 数据量也越大	量化位数越多 音质越好 数据量也越大	立体声比单声道的表现力丰富， 但数据量翻倍
11. 025kHz 22. 05 kHz 44. 1 kHz	8位=256 个值 16位=65536个值	单声道 立体声

# 1.2 计算机运算基础

**数字音频数据的计算公式：**

**数据量 = 采样频率×量化位数  
×声道数/8(字节/秒)**

采样频率 (kHz)	量化位数 (bit)	数据量(KB/s)	
		单声道	立体声
11. 025	8	10. 77	21. 53
	16	21. 53	43. 07
22. 05	8	21. 53	43. 07
	16	43. 07	86. 13
44. 1	8	43. 07	86. 13
	16	86. 13	172. 27

# 1.2 计算机运算基础

## 音频的文件格式：

**WAV文件**：WAV是Microsoft/IBM共同开发的PC波形文件。因未经压缩，文件数据量很大。声音层次丰富，还原音质好。

**MP3文件**：MP3(MPEG Audio layer3)是一种按MPEG标准的音频压缩技术制作的音频文件。高压缩比(11:1)，优美音质。

**WMA文件**：WMA(Windows Media Audio)是Windows Media格式中的一个子集(音频格式)。压缩到MP3一半。

# 1.3 计算机工作原理

## 一、指令和指令系统

### 1. 指令及其格式

**指令**：能被计算机识别的命令，它是硬件可执行的、完成一个基本操作所发出的命令。

**基本操作**：加、减、乘、除、存数、取数等

**指令格式**：

操作码	地址码或数据
-----	--------

# 1.3 计算机工作原理

## 2. 指令分类与功能

**指令系统：**计算机能识别所有指令的集合。

指令	功能
数据传送指令	将数据在存储器之间进行传送
数据处理指令	对数据进行运算和变换
控制转移指令	控制程序中指令的执行顺序
输入输出指令	实现主机与外设间传输信息的指令
其他指令	执行停机、空操作和等待的指令

**程序：**指用户根据某一问题的解决步骤，选用一组指令进行有序排列的集合。

# 1.3 计算机工作原理

## 二、计算机程序设计

**举例说明：计算  $7+2=?$**

### 文字描述的计算程序

计算步骤	解题命令
1	从存储器中取出7到运算器的0号寄存器中
2	从存储器中取出2到运算器的1号寄存器中
3	将0号和1号寄存器中的数据相加，得和9
4	将计算结果9存入存储器中
5	在输出设备中打印计算结果9
6	停机

# 1.3 计算机工作原理

## 计算程序的简写形式

指令顺序	操作码	操作数
1	取数	7
2	取数	2
3	加法	7, 2
4	存数	9
5	打印	9
6	停机	

操作名称	操作码
取数	0100
加法	0101
存数	1010
打印	1000
停机	1111

## 操作数存放单元

数的操作地址	存放的数
0001	0111(7)
0010	0010(2)
0011	计算结果

# 1.3 计算机工作原理

## 用二进制表示的计算程序

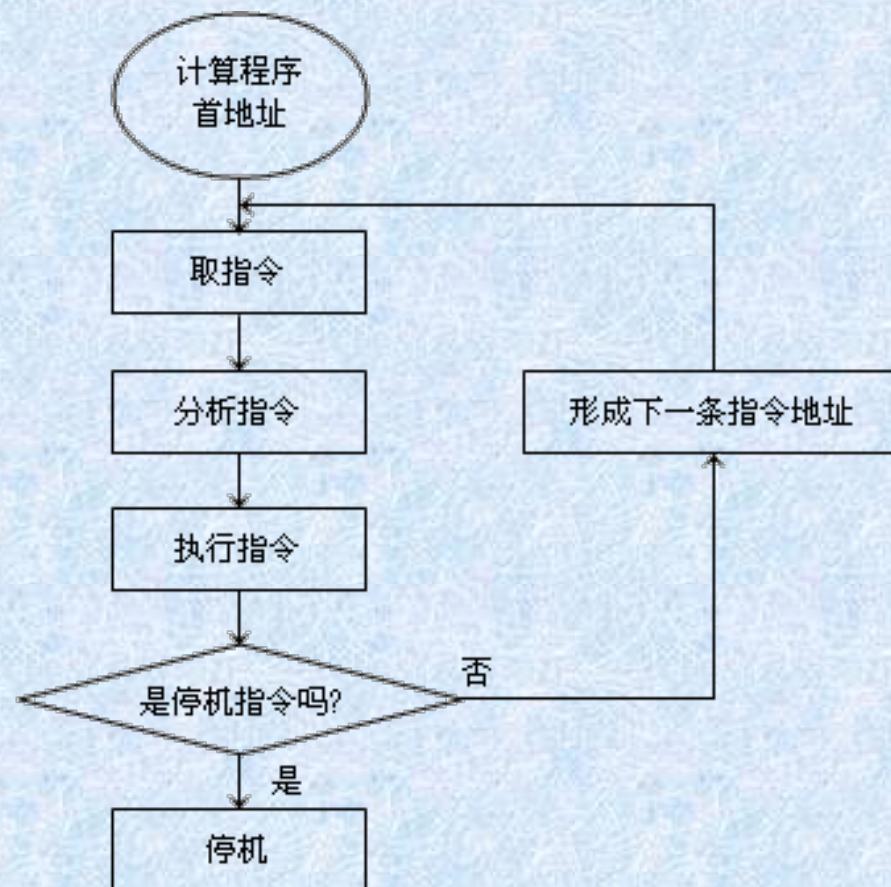
指令地址	操作码	地址码	所完成的操作
0101	0100	0001	$R_0 \leftarrow (D_1)$
0110	0100	0010	$R_1 \leftarrow (D_2)$
0111	0101	0001	$R_0 \leftarrow (R_0) + (R_1)$
1000	1010	0011	$D_3 \leftarrow (R_0)$
1001	1000	0011	打印机 $\leftarrow (D_3)$
1010	1111		停机

## 存储器布局

单元地址	存储单元内容	
0001	00000111	7
0010	00000010	2
0011		计算结果
0100		
0101	01000001	取数指令
0110	01000010	取数指令
0111	01010001	加法指令
1000	10100011	存数指令
1001	10000011	打印指令
1010	1111	停机指令
1011	⋮	

# 1.3 计算机工作原理

## 三、计算机程序执行



# 1.4 计算学科的典型问题

## 一、排序问题

**排序是把给定数据集合中的元素按照一定的标准来安排先后次序的过程。**

**选择排序算法：**对给定的一个数据表，算法从第一个元素开始扫描整个列表，找到最小或最大的元素，并将其与第一个位置的元素交换。然后算法从第二个位置的元素开始扫描剩下的列表，找到次小或次大的元素，并将其与第二个位置的元素交换。如此循环，直到所有的元素都被排好序为止。

初始排序数据	[49	78	65	97	36	13]
第一轮排序后	13	[78	65	97	36	49]
第二轮排序后	13	36	[65	97	78	49]
第三轮排序后	13	36	49	[97	78	65]
第四轮排序后	13	36	49	65	[78	97]
第五轮排序后	13	36	49	65	78	[97]
最后排序结果	13	36	49	65	78	97

选择排序算法是由一个双层循环控制，算法时间复杂度是 $O(n^2)$

# 1.4 计算学科的典型问题

部分排序算法的时间效率比较 (单位: 毫秒)

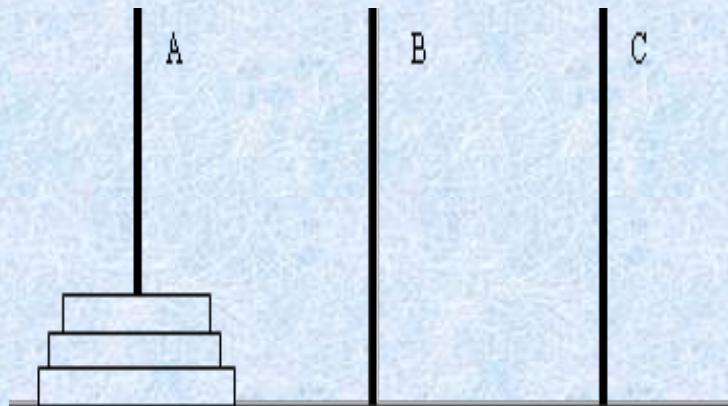
排序算法	10	100	1K	10K	100K	1M
插入排序	0.000258	0.008619	0.764	56	5145	515621
冒泡排序	0.000276	0.005643	0.545	61	8174	549432
选择排序	0.000237	0.006438	0.488	47	4717	478694
快速排序	0.000291	0.003051	0.030	0.311	3.634	39
归并排序	0.000723	0.006225	0.066	0.561	5.48	70
基数排序	0.005181	0.021	0.165	1.65	11.428	117
哈希排序	0.000522	0.003372	0.036	0.518	4.152	61

每一种排序算法对时间的效率和空间的要求不尽相同，没有哪一种是绝对最优的，在实用时需要根据不同情况适当选用，也可多种方法结合使用。

# 1.4 计算学科的典型问题

## 二、汉诺塔问题

**印度古老传说：**在世界中心贝拿勒斯的圣庙里，一块黄铜板上插着三根宝石针A、B和C。印度教的主神梵天在创造世界时，在其中一根针上从下到上地穿好了由大到小的64片金片，这就是所谓的汉诺塔问题。



不论白天黑夜，总有一个僧侣在按下面的法则移动这些金片：一次只移动一片，不管在哪根针上，小片必须在大片上面。僧侣们预言，当所有金片移到另外一根针上时，世界将在一声霹雳中消灭，而梵塔、庙宇和众生也都将同归于尽。

# 1.4 计算学科的典型问题

不管这个传说的可信度有多大，如果仅考虑把64片金片，由一根针上移到另一根针上，并且始终保持上小下大的顺序。这需要多少次移动呢？这里需要使用**递归算法**。

假设有n片，移动次数是 $f(n)$

显然 $f(1)=1$ ， $f(2)=3$ ， $f(3)=7$ ，且 $f(k+1)=2*f(k)+1$

不难证明 $f(n)=2^n-1$

当 $n=64$ 时， $f(64)=2^{64}-1=18446744073709551615$ 次

如果每秒钟移动一次，共需多长时间呢？

一年有31536000秒，则

$18446744073709551615/31536000 \approx 584942417355$ 年

# 1.4 计算学科的典型问题

## 三、国王的婚姻

国王：艾述（喜爱数学）

宰相：孔唤石（数学家）

公主：秋碧贞楠（邻国）

公主：求出 $48770428644836899$ 的一个真因子

国王：2, 3, 4, ..., 30000多数据（一天）

公主：验证一下，223092871

宰相：将全国百姓按自然数的顺序编号，百姓用自己的编号去除公主的数，谁除尽来领赏。

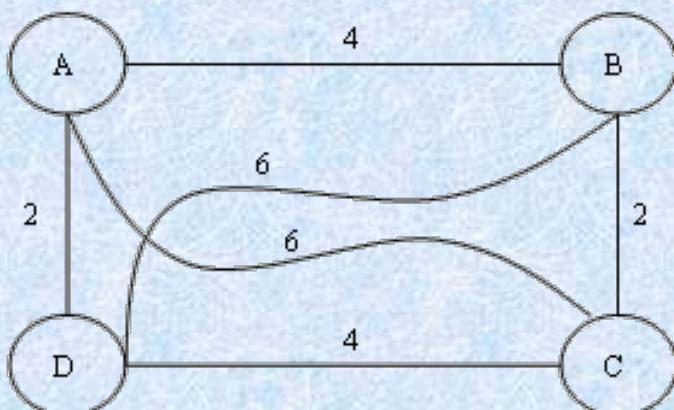
**童话说明：**①国王本人计算（串行算法，时间复杂性）

②全国百姓计算（并行算法，空间复杂性）

# 1.4 计算学科的典型问题

## 四、旅行商问题

**旅行商问题(TSP)的描述：**一位商人去n个城市推销货物，所有城市走一遍后，再回到起点，问如何事先确定好一条最短的路线，使其旅行的费用最少。



路径ABCDA的总距离是:  $4+2+4+2=12$   
路径ABDCA的总距离是:  $4+6+4+6=20$   
路径ACBDA的总距离是:  $6+2+6+2=16$   
路径ACDBA的总距离是:  $6+4+6+4=20$   
路径ADCBA的总距离是:  $2+4+2+4=12$   
路径ADBKA的总距离是:  $2+6+2+6=16$

# 1.4 计算学科的典型问题

城市数目为4时，组合路径数为6

城市数目为n时，组合路径数为 $(n-1)!$

当城市数目不多时要找到最短距离的路线并不难，但随着城市数目的不断增大，组合路线数将呈指数级数规律急剧增长，以至到达无法计算的地步，这就是所谓的组合爆炸问题。

假如城市的数目增为20个，组合路径数则为

$$(20-1)! \approx 1.216 \times 10^{17}$$

若计算机以每秒检索1000万条路线的速度计算，也需要花上386年的时间。

# 本章小结

- 计算机的基本概念（定义、分类、特点、应用）
- 数制之间的相互转换（重点）
- 原码、反码和补码（重点）
- ASCII码、汉字编码（重点）
- 多媒体信息编码
- 计算机工作原理
- 计算学科的典型问题

通过排序问题、汉诺塔问题、国王的婚姻、旅行商问题来说明人所固有的能力与局限性、计算机的计算能力与局限性，以及问题到底有多复杂。