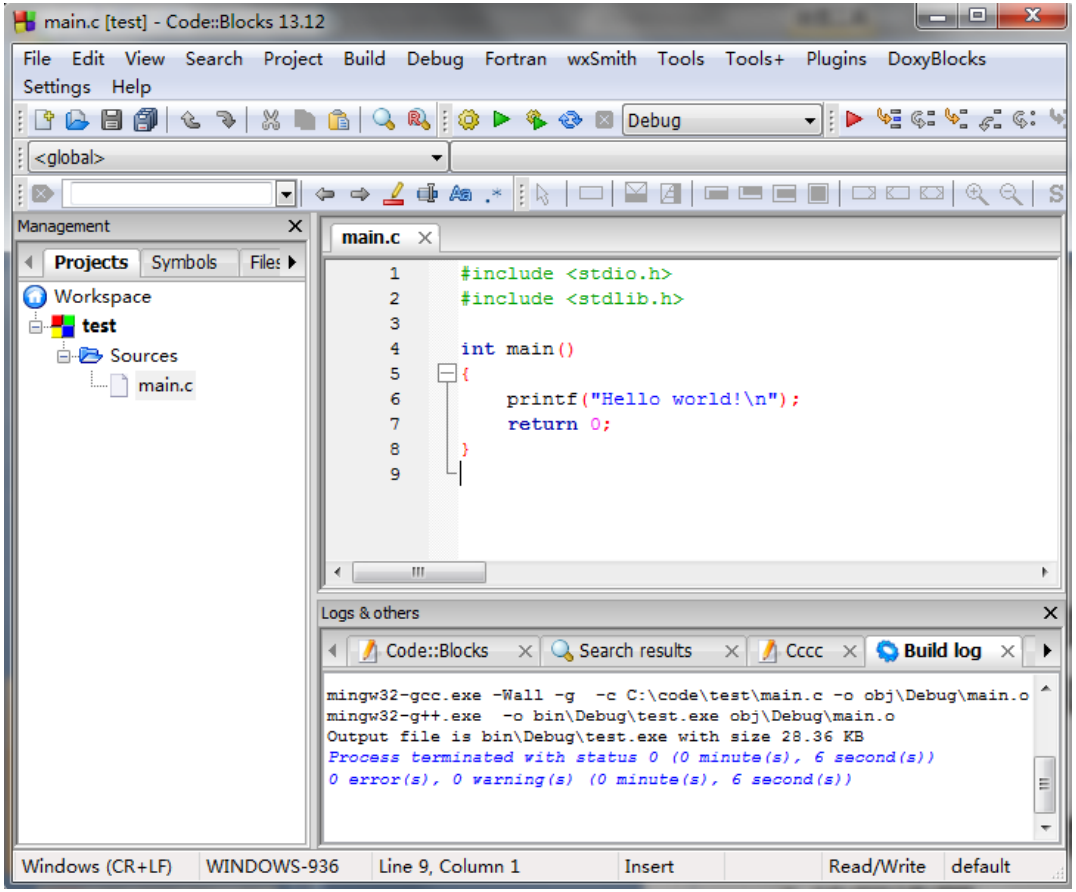
第2章 计算机中的数

请加入课程QQ群

- 群号：595851054
- 要求1：申请入群时请报真实姓名
- 要求2：入群后请把群名片改为真实姓名

复习回顾

- 上次课的内容
  - 课程简介
  - 程序和语言概览
  - C语言的发展和特点
  - 简单的C程序
  - C语言的结构特点
  - 编程流程
  - C语言编程工具



- Code::Blocks，你完成安装并开始使用了吗？

引言：一个计数游戏

今天需要弄明白的三个问题

问题1

- “数”对于计算机到底有多重要？

问题2

- 计算机是如何计数的？

问题3

- 我们应该如何理解计算机的计数？

信息在计算机内的表示

位和字节

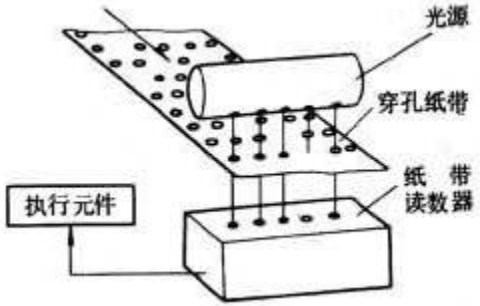
- **位**，即**bit**(Binary-digit)，又称**比特**
  - 位的取值可以是 1
  - 也可以是 0
- **字节**，即**byte**，8个bit为1个byte
  - 例如：0 0 0 0 0 0 1 0　1 0 1 0 1 1 0 1
  - 为什么是8位的？据说和Intel 8008的数据总线带宽有关，详见《The C Programming Language》

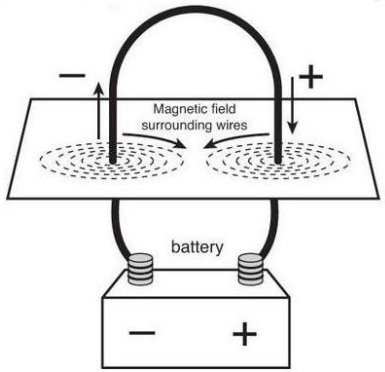
比特是计算机存储信息的基本单位

- {0,1}是区分信息所需的最少的状态
  - 单一状态能表示什么？
    - 试想当下图的曲线变成一条直线意味着什么
    - …

比特/二进制似乎是计算机存储信息的天然选择

（a）穿孔卡片





（b）磁盘磁带

（c）超大规模集成电路

字节是存储器寻址的基本单位

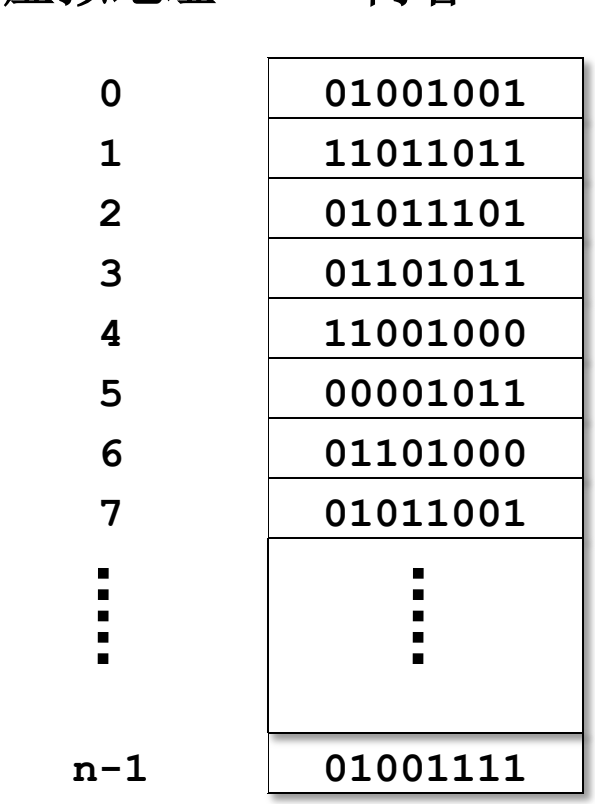
- 经常用来计算磁盘的大小
  - 1KB = 1024 Byte
  - 1MB = 1024 KB
  - 1GB = 1024 MB
  - 1TB = 1024 GB
  - 之后还有PB、EB、ZB、YB.....
- 需牢记： $1K = 2^{10} = 1024$。

并争取早日达到以下境界

【程序猿轶事】某程序猿在肉店买了1公斤肉，回家一称，他不高兴的跑回肉店说：老板不厚道，少了24克！

对程序来说，内存是这样滴

- 虚拟存储空间
  - 展示给计算机语言的一个概念性映像
  - 实际上是软硬件的复杂组合，为程序提供一个统一的字节序列。
  - 虚拟内存的原理和机制在以后的《操作系统原理》课程中会进行详细介绍



Btw，初学编程时的两大误区

- 其一：编程很简单，上一门课、学几个月就号称精通某语言。
  - 最乐观的评价叫“入门”
- 其二：看书好多地方看不懂，有些概念想不明白，我好方。
  - 编程的各部分知识是相互联系，刚开始学不明白没关系，有些概念需要对编程有更多接触后才会变得更易理解。

数的进制

- 计算机选择了二进制
- 人经常使用的是十进制
- 当然还有其他进制的存在
- 要编程，最好还是能熟悉二进制的表示方式

◆自检一下，试读这句话

世界上只有10种人：懂二进制的和不懂二进制的。

N进制数的特征

二进制	0，1，10，11，100，101，110，111，1000，1001…
八进制	0，1，2，3，4，5，6，7，10，11，12，13，14，15…
十进制	0，1，2，3，4，5，6，7，8，9，10，11，12，13，14…
十六进制	0，1，2，3，4，5，6，7，8，9，a，b，c，d，e，f，10…

- N是基数，每个数位可以是N个数字的一个：0、1、…、N-1
- 每个数字在不同的数位上，由低向高按N的次幂递增。
- 运算时满N进1，借1当N。
- 数字不够，字母来凑

【程序猿语录】今天是9号，恩，那明天就是a号……

N进制数的区分表示

- 任何数制的数都可以写成数字序列，为了便于区分，我们可以将数字序列**括起来加上基数的下标**，比如：
  - 512是十进制数（十进制数可以省略下标）
  - $(1001)_2$是二进制数
  - $(715)_8$是八进制数
  - $(57af)_{16}$是十六进制数

数字的解析式表示

- 设某个m位N进制的整数是 $A_{m-1}A_{m-2}\cdots A_1A_0$ 则它的解析式是：

$$A_{m-1}A_{m-2}\cdots A_1A_0 = A_{m-1}\times N^{m-1} + A_{m-2}\times N^{m-2} + \cdots + A_1\times N^1 + A_0\times N^0$$

  - 例如 $5237=5\times10^3+2\times10^2+3\times10^1+7\times10^0$
  - 又如 $(101)_2=1\times2^2+0\times2^1+1\times2^0$
  - 再如 $(a4)_{16}=10\times16^1+4\times16^0$

N进制整数往十进制的转换

- 方法：写出N进制数的解析式表示，然后计算结果。例如：
  - $(101)_2=1x2^2+0x2^1+1x2^0=5$
  - $(a4)_{16}=10x16^1+4x16^0=164$

十进制整数往二进制的转换

- 方法：将十进制数不断除以2并取其余数，直到商为0，就可以得到从低到高的二进制数的数字序列。比如20=(10100)$_2$

除以二	商	余数	数字序列B_i
第一次	20	0	$B_0=0$
第二次	10	0	$B_1=0$
第三次	5	1	$B_2=1$
第四次	2	0	$B_3=0$
第五次	1	1	$B_4=1$
	0		

反序获得最终结果

二进制整数往8或16进制转换

- 要想把一个二进制数转换为十六进制数，只需要：
  - 将一个二进制数从低向高每4位划为一组（位数不足时补0），
  - 再将每组4位二进制数转换为1位十六进制数即可。
  - 如$(101011)_2 \rightarrow (0010\ \ 1011)_2 \rightarrow (2b)_{16}$
- 转换为八进制数就是3位一组，其余类似

十进制整数往8或16进制的转换

- 方法1：除留余数法，类似于十进制转换为二进制的方法，只是除数换成8或者16

- 方法2：先将十进制数转换为二进制数，再将二进制数转换为八进制数或者十六进制数。

推荐！

十进制小数如何转换为二进制？

- 可以像整数一样转换吗？不行！请思考原因。
- 方法：把小数部分不断乘以基数，取整数部分，直到积的小数部分为零。比如$0.25=(0.01)_2$

乘以二	积	整数部分	小数部分
第一次	0.5	0	0.5
第二次	1.0	1	0.0

顺序收集获得二进制结果的小数部分

◆精度问题：十进制有限小数并不总能精确存放在计算机

●试把0.3转换为二进制？

万圣节/圣诞节，傻傻分不清楚

- 为什么“走火入魔的程序猿”会分不清万圣节和圣诞节？
  - 【友情提示1】：万圣节是每年10月31日，圣诞节是每年12月25日。
  - 【友情提示2】：八进制与十进制之惑
  - 【答案】：$(10)_{10}=(12)_8$，$(31)_8=(25)_{10}$

数的两种不同编码方式

- **无符号数**：只能表示非负数
  - ◆通常表示不具有数字意义的值，比如身份证号，内存地址等等

- **有符号数**：能表示负数、零和正数
  - ◆有数字意义的值，各种计算必备

计算机能表示的整数是有限的

- 有限的位数只能表示有限范围内的整数！
  - 比如，2个比特能表示的无符号整数：
    - $(00)_2 = 0$
    - $(01)_2 = 1$
    - $(10)_2 = 2$
    - $(11)_2 = 3$

- 注意：只有意识到这一点，才能正确理解以后编程中遇到的“数值精度”、“溢出”等问题

有符号数如何表示？

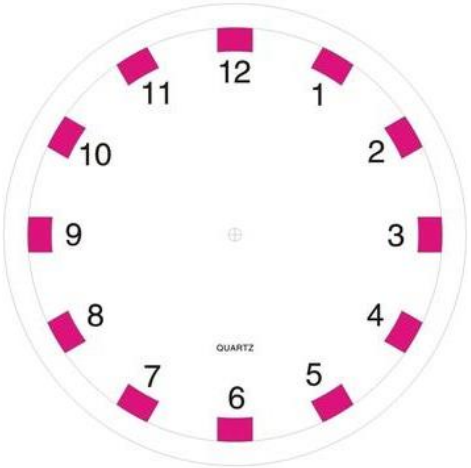
- 最直观的方案：**原码**，把最高位当成符号位（0代表正，1代表负），其余位表示数值
  - 如3个比特能表示的有符号数：
    - $(000)_2=+0$, $(100)_2=-0$
    - $(001)_2=+1$, $(101)_2=-1$
    - $(010)_2=+2$, $(110)_2=-2$
    - $(011)_2=+3$, $(111)_2=-3$

曾经出现过的数的编码方案

- 原码
  - ◆优点：人类直观易懂，喜闻乐见
  - ◆缺点：运算时需要更多电路处理符号位
- 移码（淘汰，不提）
- 反码（淘汰，不提）
- 补码
  - ◆一统天下，几乎所有的现代计算机都采用补码！

补码的由来：模和补数

- 模：一个计量系统的计数范围
  - 比如时钟，模=12
    - 计数总是不会超过12
    - 12点过后，就是1点



- 补数：两个小于模的正数，和为模则互为补数
  - 当模=12，1和11，2和10，...，6和6 互为补数

补码的由来：带模的计算

- 带模的计量系统
  - ◆运算结果超出计数范围则表示为模的余数
    - ●现假设模为12：
      - ✪6+7=13 →（对12求余数）1
      - ✪8+10=18 →（对12求余数）6
  - ◆可以化减法为加法：
    - ●减一个数等价于加上它的补数（设模为12）
      - ✪9-4 相当于 9+8=17 →（对12求余数） 5
      - ✪6-1 相当于 6+11=17 →（对12求余数） 5
  - ◆好处：做加减法不需要判断符号位！

二进制补码

- 补码：把补数应用于计算机
  - 十进制数n的二进制补码表示为$[n]_{补}$
  - m位二进制数的模为2^m，即$(100\ldots0)_2$

m+1位，m个0

  - 非负数的补码即为原码！
    - 设m=4，$[6]_{补}=[6]_{原}=(0110)_2$
  - 那么负数的补码如何表示？

如何由原码求补码（一）

- 方法一：**根据补码定义计算，负数的补码就是绝对值的补数的原码。**先获得模，然后获得该负数的绝对值，最后用模的原码减去该绝对值的原码，所得结果即该负数的补码。

◆假设8位二进制数，求-7的补码（模=2^8）

$$[-7]_{补}=(\underbrace{100000000}_{\text{模的原码，9位}})_2-(\underbrace{00000111}_{\text{绝对值的原码，8位}})_2$$

$$=(\underbrace{11111001}_{\text{-7的补码，8位}})_2$$

如何由原码求补码（二）

- 方法二：**取反加一。** 去掉符号位，其余数位按位求反，结果加1，然后添上符号位。
  - 假设8位二进制数，求-7的补码
    - 第一步：-7的原码去符号位 $(0000111)_2$

      7位
    - 第二步：按位去反$(1111000)_2$
    - 第三步：加1 $(1111001)_2$
    - 第四步：添上符号位 $(11111001)_2$

如何由补码求原码

- 方法：补码的补码，即为原码
  - 已知$[-7]_{补}$，求-7的原码
    - 第一步：-7的补码去符号位 $(1111001)_2$（7位）
    - 第二步：按位取反$(0000110)_2$
    - 第三步：加1 $(0000111)_2$
    - 第四步：添上符号位 $(10000111)_2$

补码原码互换的死角

- 有一个特殊的负数，它的补码没有对应的原码
  - 对于4位的二进制数，试求-8的原码？
  - 原码表示的数的范围是对称的，而补码不是。
  - 对于N位的二进制数
    - 原码表示范围： $-(2^{N-1}-1)\sim(2^{N-1}-1)$
    - 补码表示范围： $-2^{N-1}\sim(2^{N-1}-1)$
    - -2^{N-1}是孤独的，$[-2^{N-1}]_{补}=(\underbrace{100\ldots0}_{N位})_2$

诡异的加法结果，正正得负？

- 假定用补码表示4位的二进制有符号数，计算6+3的结果，即

  $(0110)_2+(0011)_2=(1001)_2=-7$

- 有限计数范围内的加法运算会导致“溢出”
  - “溢出”：运算结果超过计数最大值（模），只好按模的余数处理
  - 【思考1】减法、乘法和除法会不会导致溢出？
  - 【思考2】溢出是补码的错吗？

ASCII码

ASCII

abbr.（缩写）

1.=American Standard Code for Information Interchange 美国信息交换标准码

ASCII码的产生

- 字符如何在计算机中以二进制表示？
  - 编码：建立起字符和二进制整数的一一映射
  - 26个大写字符+26个小写字符+10个数字+常用符号+控制符号，单字节足矣
- 相互通信的标准问题，特定符号用哪种二进制数表示？
  - ANSI制定，ISO定为国际标准

ASCII码表

高四位 / 低四位		ASCII非打印控制字符										ASCII 打印字符												
		0000					0001					0010		0011		0100		0101		0110		0111		
		0					1					2		3		4		5		6		7		
		+进制	字符	ctrl	代码	字符解释	+进制	字符	ctrl	代码	字符解释	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	ctrl
0000	0	0	BLANK NULL	^@	NUL	空	16	▶	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH	头标开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX	正文结束	19	‼	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	◆	^D	EOT	传输结束	20	¶	^T	DC4	设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ	查询	21	§	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK	确认	22	■	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	震铃	23	↨	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	换行/新行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	竖直制表符	27	←	^[	ESC	转意	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	换页/新页	28	∟	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	回车	29	↔	^]	GS	组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO	移出	30	▲	^6	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI	移入	31	▼	^-	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	Δ	^Back space

C源代码文件的ASCII码表示

```
#include <stdio.h>
int main()
{
    printf("hello, world\n");
}
```

#	i	n	c	l	u	d	e	<sp>	<	s	t	d	i	o	.
35	105	110	99	108	117	100	101	32	60	115	116	100	105	111	46

h	>	\n	\n	i	n	t	<sp>	m	a	i	n	(	)	\n	{
104	62	10	10	105	110	116	32	109	97	105	110	40	41	10	123

\n	<sp>	<sp>	<sp>	<sp>	p	r	i	n	t	f	(	"	h	e	l
10	32	32	32	32	112	114	105	110	116	102	40	34	104	101	108

l	o	,	<sp>	w	o	r	l	d	\	n	"	)	;	\n	}
108	111	44	32	119	111	114	108	100	92	110	34	41	59	10	125

‘0’等于0吗？

1. 形式上，’0’带一对单引号，0没有
2. 性质上，’0’是字符，0是数值
3. ASCII码规定，’0’对应的整数是48，’1’对应的是49，‘2’对应的是50…以此类推
4. 所以，’0’不等于0！

作业 2017/10/13

1. （1）把你的学号的后四位乘以你出生的月份（请先注明自己是几月出生的），得到一个十进制数；
（2）把这个十进制数转换成16位的二进制数，写出转换过程和最终结果，并判断是否溢出；
（3）截取步骤2得到的二进制数的低8位，将其视为有符号数的补码编码，并转化为10进制数，写出转换过程和最终结果。

注意事项：

（1）作业写在纸上；
（2）作业纸抬头写上学号和姓名；
（3）周三（10月18日课间）交给助教