1-bit	0	1							二進制
2-bit	00	01	10	11					四進制
3-bit	000	001	010	011	100	101	110	111	八進制
4-bit	0000	0001	0010	0011	0100	0101	0110	0111	十六進制
$2^4=16$	1000	1001	1010	1011	1100	1101	1110	1111	F=15

A(10)=1010, B(11)=1011, C(12)=1100, D(13)=1101, E(14)=1110, F(15)=1111

5-bit	0 0000	0 0001	0 0010	0 0011	0 0100	0 0101	0 0110	0 0111	0-7
	0 1000	0 1001	0 1010	0 1011	0 1100	0 1101	0 1110	0 1111	8-15
	1 0000	1 0001	1 0010	1 0011	1 0100	1 0101	1 0110	1 0111	16-23
$2^5=32$	1 1000	1 1001	1 1010	1 1011	1 1100	1 1101	1 1110	1 1111	24-31

bit	0	1	2	3	4	5	6	7	
8-bit	0000 0000	0000 0001	0000 0010	0000 0011	0000 0100	0000 0101	0000 0110	0000 0111	0-7
	0000 1000	0000 1001	0000 1010	0000 1011	0000 1100	0000 1101	0000 1110	0000 1111	8-15
	…	…							
$2^6=256$	1111 1000	1111 1001	1111 1010	1111 1011	1111 1100	1111 1101	1111 1110	1111 1111	FF=255

bit	0	1	2	3	4	5	6	7	
16-bit	0000 0000 0000 0000	0000 0000 0000 0001	0000 0000 0000 0010	0000 0000 0000 0011	0000 0000 0000 0100	0000 0000 0000 0101	0000 0000 0000 0110	0000 0000 0000 0111	0-7
	0000 0000 0000 1000	0000 0000 0000 1001	0000 0000 0000 1010	0000 0000 0000 1011	0000 0000 0000 1100	0000 0000 0000 1101	0000 0000 0000 1110	0000 0000 0000 1111	8-15
	0000 0000 0001 0000	0000 0000 0001 0001	0000 0000 0001 0010	0000 0000 0001 0011	0000 0000 0001 0100	0000 0000 0001 0101	0000 0000 0001 0110	0000 0000 0001 0111	16-23
	0000 0000 0001 1000	0000 0000 0001 1001	0000 0000 0001 1010	0000 0000 0001 1011	0000 0000 0001 1100	0000 0000 0001 1101	0000 0000 0001 1110	0000 0000 0001 1111	24-31
	...	...							
2^{16}=65536	1111 1111 1111 1000	1111 1111 1111 1001	1111 1111 1111 1010	1111 1111 1111 1011	1111 1111 1111 1100	1111 1111 1111 1101	1111 1111 1111 1110	$\boxed{\text{1111 1111 1111 1111}}$	FF FF 65535

F = 15 = 1111 (4-bit)

bit	0	1	2	3	4	5	6	7	
32-bit	0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0001	0000 0000 0000 0000 0000 0000 0000 0010	0000 0000 0000 0000 0000 0000 0000 0011	0000 0000 0000 0000 0000 0000 0000 0100	0000 0000 0000 0000 0000 0000 0000 0101	0000 0000 0000 0000 0000 0000 0000 0110	0000 0000 0000 0000 0000 0000 0000 0111	0-7
	...	...							
$2^{32}=4\times10^9$ 4294967296	1111 1111 1111 1111 1111 1111 1111 1000	1111 1111 1111 1111 1111 1111 1111 1001	1111 1111 1111 1111 1111 1111 1111 1010	1111 1111 1111 1111 1111 1111 1111 1011	1111 1111 1111 1111 1111 1111 1111 1100	1111 1111 1111 1111 1111 1111 1111 1101	1111 1111 1111 1111 1111 1111 1111 1110	$\boxed{\text{1111 1111 1111 1111 1111 1111 1111 1111}}$	FF FF FF FF $\sim4\times10^9$

姓名: ( ) 班別: 中 分數: /10 日期: / /2013

【數字系統間的轉換】

$\times 2^n$ $\div 2$

2,8,16→10→2,8,16

$\times 16^n$ $\div 16$

◆ 十進位轉換成二、八、十六進位

$(1234)_{10} = (\_\_\_\_\_\_\_\_\_\_\_\_)_2$
$(1234)_{10} = (\_\_\_\_\_\_\_\_\_\_\_\_)_8$
$(1234)_{10} = (\_\_\_\_\_\_\_\_\_\_\_\_)_{16}$

◆ 二、八、十六進位轉換成十進位：

$(11111)_{10} = 1*10000 +1*1000 +1*100 +1*10 +1*1$
$= 1*10^4 +1*10^3 +1*10^2 +1*10^1 +1*10^0$
$(11111)_2 = 1*2^4 +1*2^3 +1*2^2 +1*2^1 +1*2^0 = (\_\_\_\_\_\_\_\_\_\_)_{10}$
$(11111)_8 = 1*8^4 +1*8^3 +1*8^2 +1*8^1 +1*8^0 = (\_\_\_\_\_\_\_\_\_\_)_{10}$
$(11111)_{16} = 1*16^4 +1*16^3 +1*16^2 +1*16^1 +1*16^0 = (\_\_\_\_\_\_\_\_\_\_)_{10}$

2→8→16

◆ 二、八(2^3)、十六(2^{16})進位間的轉換

$(10011010010)_2 \rightarrow (010\ \underline{011}\ 010\ \underline{010})_2 = (\_\_\_\_\_\_\_\_\_\_)_8$
$(10011010010)_2 \rightarrow (0100\ \underline{1101}\ 0010)_2 = (\_\_\_\_\_\_\_\_\_\_)_{16}$
$(1234)_8 = (001\ \underline{010}\ 011\ \underline{100})_2$
$= (0010\ \underline{1001}\ 1100)_2 = (\_\_\_\_\_\_\_\_\_\_)_{16}$
$(1234)_{16} = (\underline{0001}\ 0010\ \underline{0011}\ 0100)_2$
$= (001\ \underline{001}\ 000\ \underline{110}\ 100)_2 = (\_\_\_\_\_\_\_\_\_\_)_8$

位値 place value

printf("%d", n); // 十進 DEC

10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0
億	千萬	百萬	十萬	萬	千	百	十	1

二進 BIN

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

printf("%x", n); // 十六進 HEX

16^8	16^7	16^6	16^5	16^4	16^3	16^2	16^1	16^0

printf("%o", n); // 八進 OCT

8^8	8^7	8^6	8^5	8^4	8^3	8^2	8^1	8^0

十一進? 十二進? 十三進? 十四進?

YouTube www.youtube.com/user/szetoscy/videos?query=number+system

二進制 #2

姓名: ( ) 班別: 中 分數: /10 日期: / /2013

數制

我們最常用的 10 進制，其實起源於人有 10 個指頭。如果我們的祖先始終沒有擺脫手腳不分的境況，我想我們現在一定是在使用 20 進制。

其實生活中很多地方的計數方法都多少有點不同進制的影子。

二進制〔Binary System〕有兩個數符 0 和 1。進位規則是「逢二進一」,「退一還二」。如沒有襪子稱爲 0 只襪子，有一隻襪子稱爲 1 只襪子，但若有兩襪子，則我們常說的是：1 雙襪子。

生活中還有：七進制，比如星期。十二進位，比如小時或“一打”；十六進位，比如重量單位中的磅；六十進位，比如分鐘或角度... ...等。

德國著名數學家萊布尼茲(1646-1710)創立出二進制數學，在 1698 年曾預言這種數學運算雖很冗長，例如 567 的二進制形式是 1000110111_2，但對科學研究卻很重要。

1946 年，第一台電子計算機問世，二進制數被用來作為計算機的基本數，從此二進制才為人們廣泛重視，由於電子計算機是由各種電子元件組成，而電子元件只有「導通」「截止」兩種狀態，從而採用二進制容易表達、然而二進制的致命弱點是用它表示數需書寫很長，為解決這個矛盾，電子計算機除採用二進製作為它的基本數系外，還常用到八進制、十六進制等。用十六進制或八進制可以解決這個問題。因為，進制越大，數的表達長度也就越短。不過，為什麼偏偏是十六或八進制，而不其他的，諸如 9 或 20 進制呢？

2、8、16，分別是 2 的 1 次方，3 次方，4 次方。這一點使得三種進制之間可以非常直接地互相轉換。8 進制或 16 進制縮短了二進位數字，但保持了二進位數字的表達特點。這些數制的運算以及它們間的相互轉換，便成了電子計算機的數字基礎，故又有「電腦數學」之稱。

知道多些，記得多些，就會容易多些

$2^0=1$	$10^0=1$	$8^0=1$	$16^0=1$
$2^1=2$	$10^1=10$	$8^1=8$	$16^1=16$
$2^2=4$	$10^2=100$	$8^2=64$	$16^2=256$
$2^3=8$	$10^3=1000$	$8^3=512$	$16^3=4096$
$2^4=16$	$10^4=10{,}000$	$8^4=4096$	$16^4=65{,}536$
$2^5=32$	$10^5=100{,}000$	$8^5=32{,}768$	$16^5=1{,}048{,}576$
$2^6=64$		$8^6=262{,}144$	
$2^7=128$			
$2^8=256$	$2^{12}=4096$		
$2^9=512$	$2^{13}=8192$		
$2^{10}=1024$	$2^{14}=16{,}384$		
$2^{11}=2048$	$2^{15}=32{,}768$		

姓名: ( ) 班別: 中 分數: /10 日期: / /2013

數制：記數法/數數法(整數部份)

十進制 (Denary System)

進位規則是「逢十進一」，即是每逢一個數位滿 10 就進另一個數位，所以十進制有十個數符，分別是「0」、「1」、「2」、「3」、「4」、「5」、「6」、「7」、「8」和「9」。

- 十進制的數(十進數)：234，1010，23，576，701

二進制 (Binary System)

進位規則是「逢二進一」，即是每逢一個數位滿 2 就進另一個數位，所以二進制有兩個數符，分別是「0」和「1」。

- 二進制的數(二進數)：11_2, 1010_2, 111_2, 11011_2

十六進制 (Hexadecimal System)

進位規則是「逢十六進一」，即是每逢一個數位滿 16 就進另一個數位，所以十六進制有十六個數符，分別是「0」、「1」、「2」、「3」、「4」、「5」、「6」、「7」、「8」、「9」、「A」、「B」、「C」、「D」、「E」和「F」。

- 十六進制的數(十六進數)：234_{16}, 1010_{16}, 576_{16}, 701_{16}, $7AB_{16}$, $C0D_{16}$, BAD_{16}

八進制 (Octal System)

進位規則是「逢八進一」，即是每逢一個數位滿 8 就進另一個數位，所以八進制有八個數符，分別是「0」、「1」、「2」、「3」、「4」、「5」、「6」和「7」。

- 八進制的數(八進數)：234_8, 1010_8, 23_8, 576_8, 701_8

數符 digits

二進制 BIN	0	1														
八進制 OCT	0	1	2	3	4	5	6	7								
十進制 DEC	0	1	2	3	4	5	6	7	8	9						
十二進制 DoDeca	0	1	2	3	4	5	6	7	8	9	A	B				
十六進制 HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

姓名:　　　　　　　　　　　　(　　　)　班別: 中　　　分數:　　/10　　日期:　/　/2013

十進制 (Denary System)

十進數的展開式：

例一：

$$234 = 2\times10^2 + 3\times10^1 + 4\times1$$
$$= 2\times100 + 3\times10 + 4\times1$$

例二：

$$1010 = 1\times10^3 + 0\times10^2 + 1\times10^1 + 0\times1$$
$$= 1\times1000 + 0\times100 + 1\times10 + 0\times1$$

寫出下列各十進數的展開式

Q1. 23：__________　　Q2. 586：__________　　Q3. 701：__________

用十進數表示下列數式的值

例：　$7\times10^4 + 0\times10^3 + 2\times10^2 + 8\times10^1 + 3\times1$

$= 70283$

用十進數表示下列各數式的值

Q1.　$5\times10^3 + 0\times10^2 + 0\times10^1 + 2\times1$

Q2.　$5\times10^4 + 7\times10^3 + 6\times10^2 + 0\times10^1 + 0\times1$

二進制 (Binary System)

二進數的展開式及其轉換成十進制的值：

例一：

$$11_2 = 1\times2^1 + 1\times1$$
$$= 1\times2 + 1\times1$$
$$= 2 + 1$$
$$= 3$$

例二：

$$1010_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times1$$
$$= 1\times8 + 0\times4 + 1\times2 + 0\times1$$
$$= 8 + 0 + 2 + 0$$
$$= 10$$

寫出下列各二進數的展開式及其轉換成十進制的值

Q1. 111_2：__________　　Q2. 11011_2：__________

用二進數表示下列數式的值

例：　$1\times2^4 + 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times1$

$= 11001_2$

用二進數表示下列各數式的值

Q1.　$1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times1$

Q2.　$1\times2^4 + 0\times2^3 + 2\times2^2 + 0\times2^1 + 0\times1$

姓名: ( ) 班別: 中 分數: /10 日期: / /2013

十六進制 (Hexadecimal System)

十六進數的展開式及其轉換成十進制的值：

例一：

$$\begin{aligned}234_{16} &= 2\times16^2+3\times16^1+4\times1\\ &= 2\times256+3\times16+4\times1\\ &= 512+48+4\\ &= 564\end{aligned}$$

例二：

$$\begin{aligned}7AB_{16} &= 7\times16^2+A\times16^1+B\times1\\ &= 7\times256+10\times16+11\times1\\ &= 1792+160+11\\ &= 1963\end{aligned}$$

寫出下列各十六進數的展開式及其轉換成十進制的值

Q1. 1010_{16} Q2. 576_{16} Q3. 701_{16} Q4. $C0D_{16}$

用十六進數表示下列各數式的值

例： $15\times16^4+0\times16^3+7\times16^2+5\times16^1+0\times1$

$=F0750_{16}$

用十六進數表示下列各數式的值

Q1. $6\times16^3+C\times16^2+0\times16^1+2\times1$

Q2. $3\times16^4+6\times16^3+9\times16^2+F\times16^1+0\times1$

八進制 (Octal System)

八進數的展開式及其轉換成十進制的值：

例一：

$$\begin{aligned}234_8 &= 2\times8^2+3\times8^1+4\times1\\ &= 2\times64+3\times8+4\times1\\ &= 128+24+4\\ &= 156\end{aligned}$$

例二：

$$\begin{aligned}1010_8 &= 1\times8^3+0\times8^2+1\times8^1+0\times1\\ &= 1\times512+0\times64+1\times8+0\times1\\ &= 512+8\\ &= 520\end{aligned}$$

寫出下列各八進數的展開式及其轉換成十進制的值：

Q1. 23_8 Q2. 576_8 Q3. 701_8 Q4. 777_8

用八進數表示下列各數式的值

例： $1\times8^4+0\times8^3+2\times8^2+5\times8^1+7\times1$

$=10257_8$

Q1. $7\times8^3+0\times8^2+0\times8^1+4\times1$

Q2. $3\times8^4+0\times8^3+5\times8^2+6\times8^1+0\times1$

檢查數位及二進制 #6

姓名: ( ) 班別: 中 分數: /10 日期: / /2013

十進制數轉換為二進制數

10 進制數轉換成二進位數字，這是一個連續除 2 的過程：

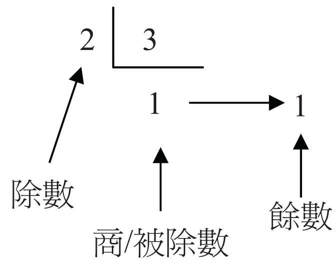
把要轉換的數，除以 2，得到商和餘數，將商繼續除以 2，直到商為 0。

最後將所有餘數倒序排列，得到數就是轉換結果。

把 3 轉換成二進數：

被除數	計算過程	商	餘數
3	3 ÷ 2	1	1

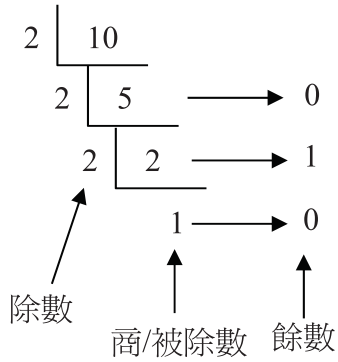
3 轉換成二進數，結果是 11_2。



把 10 轉換成二進數：

被除數	計算過程	商	餘數
10	10 ÷ 2	5	0
5	5 ÷ 2	2	1
2	2 ÷ 2	1	0

10 轉換成二進數，結果是 1010_2。

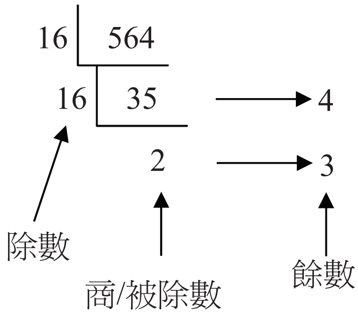


十進制數轉換為十六進制數

10 進制數轉換成十六進位數字，這是一個連續除 16 的過程：

把要轉換的數，除以 16，得到商和餘數，將商繼續除以 16，直到商為 0。

最後將所有餘數倒序排列，得到數就是轉換結果。



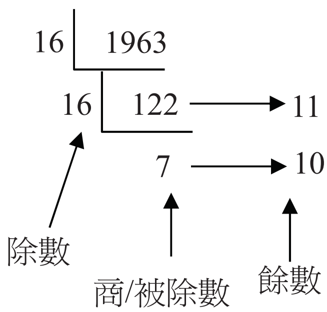
把 564 轉換成十六進數：

被除數	計算過程	商	餘數
564	564 ÷ 16	35	4
35	35 ÷ 16	2	3

564 轉換成十六進數，結果是 234_{16}。

把 1963 轉換成十六進數：

1963 轉換成十六進數，結果是 $7AB_{16}$。



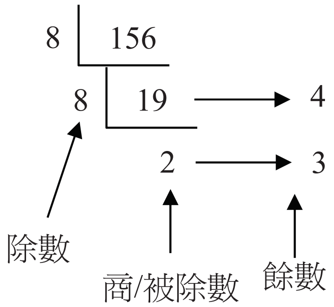
姓名: ( ) 班別: 中 分數: /10 日期: / /2013

十進制數轉換為八進制數

10 進制數轉換成八進位數字，這是一個連續除 8 的過程：

把要轉換的數，除以 8，得到商和餘數，將商繼續除以 8，直到商為 0。

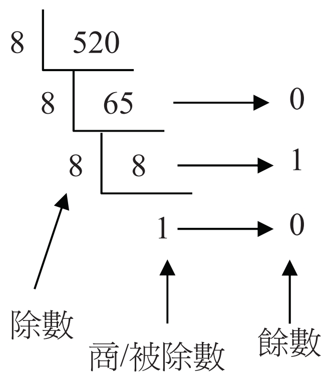
最後將所有餘數倒序排列，得到數就是轉換結果。



把 156 轉換成八進數：

被除數	計算過程	商	餘數
156	156 ÷ 8	19	4
19	19 ÷ 8	2	3

156 轉換成八進數，結果是 234_8。



把 520 轉換成八進數：

被除數	計算過程	商	餘數
520	520 ÷ 8	65	0
65	65 ÷ 8	8	1
8	8 ÷ 8	1	0

520 轉換成八進數，結果是 1010_8。

把下列各十進數轉換成二進數。

(以 $a_{n+1}2^{n+1}+a_n2^n+\cdots+a_1 2^1+a_0 2^0$ 形式表示，當中 n 和 $a_i, i=1,2,3,\ldots$ 為整數)

Q1.a. $2^{11}+2^7+2^5+5$

Q1.b. 將在(a)部份所得的二進數轉換成八進數及十六進數

Q2. $2^{12}+2^9+2^6+2^4+6$

把下列各十進數轉換成八進數。

(以 $a_{n+1}8^{n+1}+a_n8^n+\cdots+a_1 8^1+a_0 8^0$ 形式表示，當中 n 和 $a_i, i=1,2,3,\ldots$ 為整數)

Q3.a. $7\times 8^5+5\times 8^4+2\times 8^2+27$

Q3.b. 將在(a)部份所得的八進數轉換成二進數然後轉換為十六進數

Q4. $8^9+3\times 8^7+7\times 8^6+4\times 8^3+40$

把下列各十進數轉換成十六進數。

(以 $a_{n+1}16^{n+1}+a_n16^n+\cdots+a_1 16^1+a_0 16^0$ 形式表示，當中 n 和 $a_i, i=1,2,3,\ldots$ 為整數)

Q5.a. $13\times 16^4+8\times 16^3+14\times 16^2+59$

Q5.b. 將在(a)部份所得的十六進數轉換成二進數然後轉換為八進數

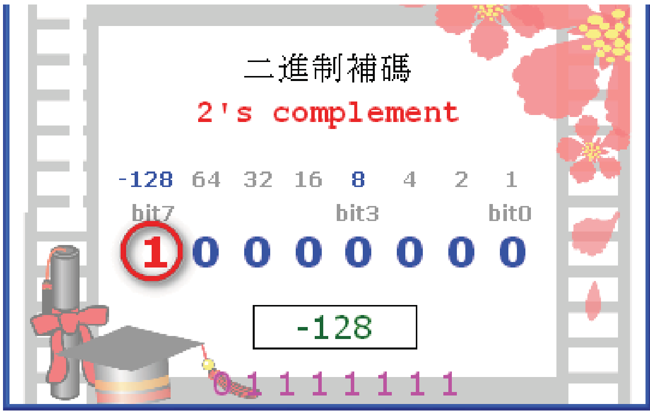
Q6. $6\times 16^8+10\times 16^7+16^5+11\times 16^2+48$

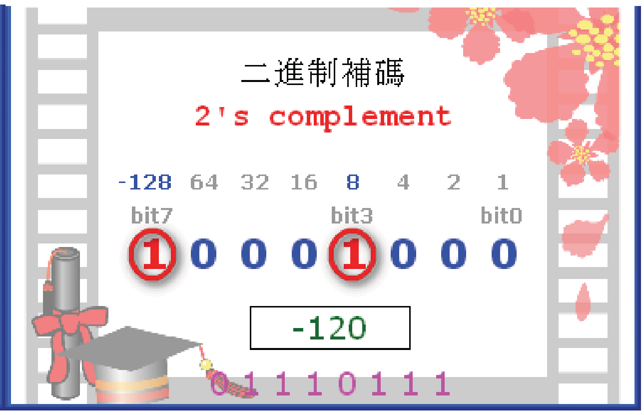
Q7. 下列各數的數值是否相等？ 101111001010_2、5712_8、3018、BCA_{16}

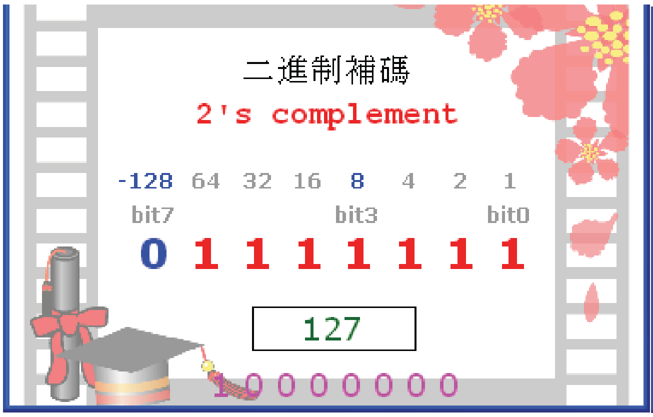
檢查數位及二進制 #8

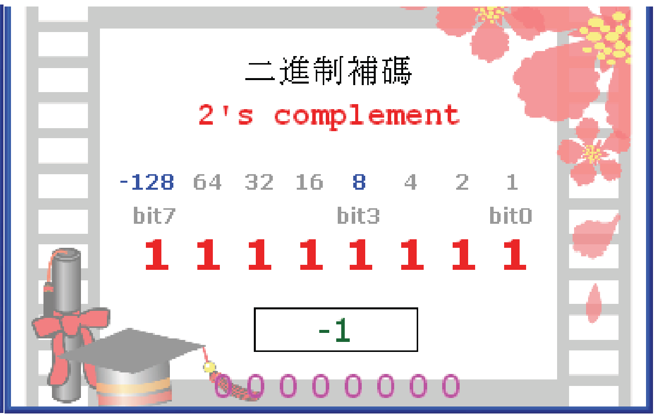
姓名: ( ) 班別: 中 分數: /10 日期: / /2013

二進制補碼 Two's complement http://www.ablmcc.edu.hk/~scy/home/javascript/number-system.htm

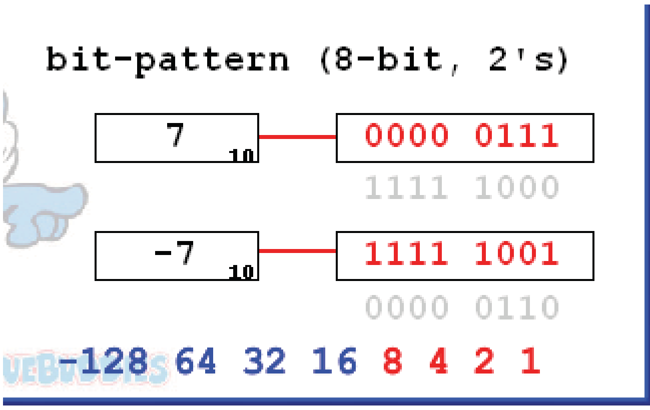


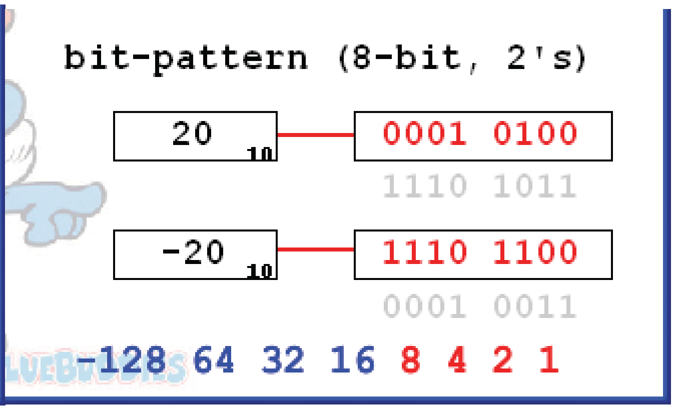


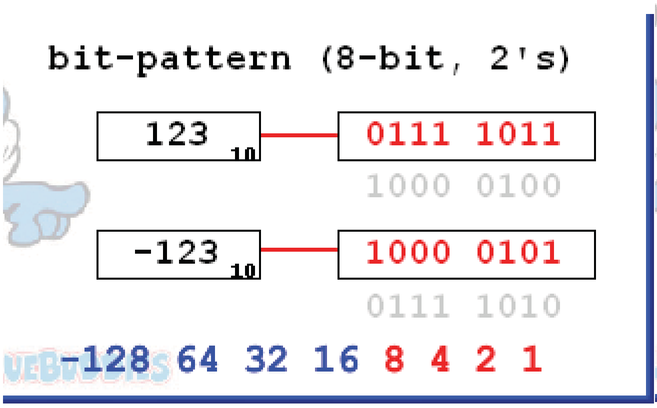


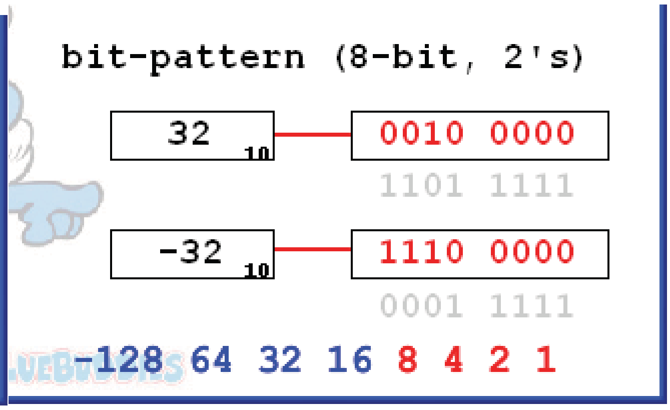


Negation 相反: 0 ↔ 1









32–20 = 0010 0000 + 1110 1100
= 1 0000 1100 (carry=1)

Q4. 1993 PRIM 是一特別類的電腦，只處理下列 8 個字符：

A、B、C、D、E、F、G、H

(a) (i) 約翰是 PRIM 電腦設計員之一。他認為 PRIM 雖只處理 8 個字符，
但仍應使用 ASCII 碼作為內部字符表示。試提出一個理由來支持這個觀點。

(ii) 最後決定是 PRIM 以 3 個二進制位來表示字符。✓
舉出一個原因，解釋為什麼採用此表示法，而不採用 ASCII 碼。✗

(iii) 約翰其後又提議下列的 JOHN 編碼法：✗

A 000	B 101	C 111	D 011
E 100	F 001	G 110	H 010

但最後決定採用下列的 PRIM 編碼法：✓

A 000	B 001	C 010	D 011
E 100	F 101	G 110	H 111

試舉出採用 PRIM 編碼法而不採用 JOHN 編碼法的一個原因。

(b) 電腦 X 及電腦 Y 皆為 PRIM 電腦，彼此以 4 條數據線互相通信，
其中 3 條傳送表示碼，1 條傳送偶「奇偶檢驗位」。假設用的是 PRIM 編碼法。

例如，字符「E」以下列的位格式傳送：

1	0	0	1
內部代碼			奇偶檢驗位

(i) 假設沒有傳送錯誤，
電腦 $\boxed{X}$ 傳送以下字符(char)時，電腦 $\boxed{Y}$ 會接收什麼位格式(bit-pattern)？
(1) "F" (2) "G"

(ii) 就下列各情況，指出是否可檢測到傳送錯誤 Error。
若「否」，寫出電腦 Y 接收到的字符。

情況	電腦 X 傳送	電腦 Y 接收到位格式	結果
1	"C"	0111	
2	"C"	0011	

(iii) 為加強$\boxed{\text{檢測錯誤}}$的能力，
電腦 X 每傳送 3 個位格式後，再傳送一個附加的位格式，如下圖所示：

若	第一個位格式	=	a_1	a_2	a_3	a_4
	第二個位格式	=	b_1	b_2	b_3	b_4
	第三個位格式	=	c_1	c_2	c_3	c_4
則	附加位格式	=	e_1	e_2	e_3	e_4

其中 e_i = 位格式 | a_i | b_i | c_i | 的偶奇偶檢驗位，i = 1,2,3。

就下列各電腦 Y 所接收的碼序列，指出是否檢測到傳送錯誤。
若「否」，寫出電腦 Y 接收到的字符。

情況 1：　{ 0110, 1111, 0011, 1010 }，

情況 2：　{ 0101, 0000, 1100, 1100 }。

Q2. 1994 某電腦使用 6 位元(bits)來表示整數及字符。

第 6 位　　　　　　　　　　　　　　　　第 1 位

0	0	1	1	1	1
1	0	1	1	0	0

所表示的整數或字符，可根據下列偽代碼/流程圖求出：

若 bit 6 = 0 則 　　刪除 bit 6 　　求餘下 5 位元以「二進補碼」所表示的整數 J 　　輸出整數 J
否則 　　求 6 位元以「二進制無符號數」所表示的整數 K 　　H = K+33　　　32+33 　　以 H 為 ASCII 值，找出相應的字符 CHAR 　　輸出字符 CHAR

(a) 下列是以十進制顯示的 6 位字。寫出所表示的字符或整數：

　(i) 15　　　　(ii) 26　　　　(iii) 44

(b) 寫出表示下列________的 6 位字的組合形式 (bit pattern)。

　(i) 整數 8　　　(ii) 整數–15　　　(iii) 字符「W」

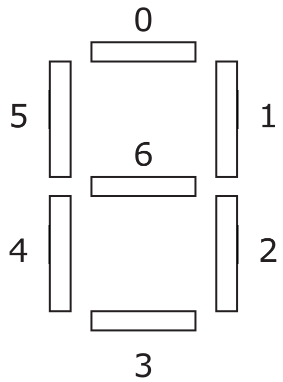
(c) (i) 寫出可表示出的最小整數。

　(ii) 寫出可表示出的最小 ASCII 值的字符。

(d) 修改流程圖中方程式 H = K+33，使所表示出的字符包括所有小寫字母，而不包括大寫字母。

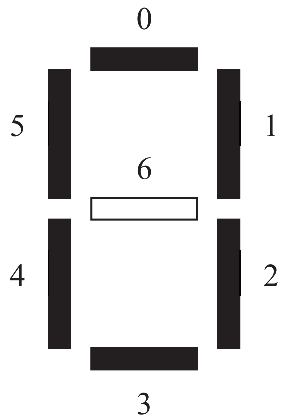
Q3. 1996

一顯示板由七條光棒（編號 0–6）組成，如下圖所示：



每條光棒均可開啟或關掉。

藉開啟適當光棒，可將一個十進制數字（0, 1, … , 9）顯示在板上。例如開啟編號號 0、1、2、3、4 及 5 光棒，便顯示出數字 0，如下所示：



現使用一專用電腦來控制顯示板各光棒的開關。

該電腦依下列規則執行一字節 byte(=8 bits)長的指令：

規則 1：	位(bit) 0 至 6 用來控制圖樣的顯示。 若位 i 是 1，則開啟光棒 i； 若位 i 是 0，則關掉光棒 i。
規則 2：	位 7 用來控制顯示的持續時間。 若位 7 是 0，則圖樣的顯示持續 2 秒。 若位 7 是 1，則圖樣的顯示持續 3 秒。

位 7　　　　　　　　　　　　　　　　　　位 0

	1	0	0	1	1	1	1

(a) 閱讀下列程序

0000 0110 (06)
1111 1101 (FD)
1111 1111 (FF)

(i) 依先後次序寫出程序所顯示圖樣。

(ii) 程序總共用了多少時間來顯示圖樣？

(b) (i) 編寫一程序，依次顯示數字 2 和 3。每一數字須顯示兩秒。

(ii) 編寫一程序，依次顯示數字 2、3、4 和 5。
在顯示兩個數字間，程序須將所有光棒關掉三秒。
程序的總執行時間須為 17 秒。
1010 A, 1011 B, 1100 C, 1101 D, 1110 E, 1111 F

Q3. 1997 某公司用機械人將貨物從一地點移到另一地點。

機械人接受有 7 個位的指令如下：

位 6	位 5	位 4	位 3	位 2	位 1	位 0
0	0	1	1	0	0	1
方向		步數		動作		

每一指令使機械人選擇一個方向（向北、向東、向南或向西），朝該方向移動數步（步數為 0、1、2 或 3），然後完成一個動作（提起貨物、放下貨物或無動作）。

機械人的方向、步數及動作分別根據 6 – 5 位、4 – 3 位及 2 – 1 位按下表決定：

6 – 5 位	00	01	10	11
方向	向北	向東	向南	向西

4 – 3 位	00	01	10	11
步數	0	1	2	3

2 – 1 位	00	01	10
動作	無動作	提起貨物 P	放下貨物 R

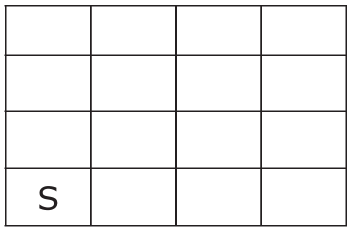
位 0 是偶數奇偶校驗位。

注意：向頁頂的方向代表向北，向右邊的方向代代表向東，每一方格代表一步。

S 點是機械人的起步點。

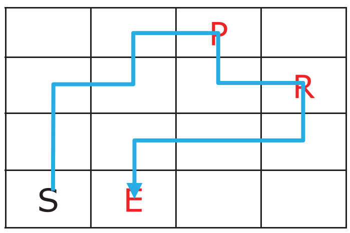
(a) 將上圖抄到你的答題簿。在你答題簿的圖上，畫出機械人執行以下指令序列時所經的路線：（在圖上使用字符「P」來表示機械人提起貨物的位置，使用「R」來表示機械人放下貨物的位置，使用「E」來表示機械人最後的位置。）

```
00 10 00 1
01 11 01 0
10 01 10 1
11 10 00 1
10 01 00 0
01 01 00 0
```



(b) 上圖顯示機械人的某一路線：

機械人在 S 點起步，在 P 點提起貨物，在 R 點放下貨物及在 E 點終止它的行程。寫出上述路線及動作的指令序列。



(c) 寫出一含有兩個指令的序列。使機械人在起步位置提起貨物，然後向北面移動一步後將貨物下。

Q3. 1998 一種自動洗衣機使用以下的 5 位指令來指定其操作：

位 4	位 3	位 2	位 1	bit 0
0	0	0	1	1

最左 3 位	操作	最右 2 位 00	最右 2 位 01	最右 2 位 10
000	排水	排水	--	--
001	注水	低水位	正常水位	高水位
010	將水加熱	25°C	40°C	60°C
011	旋轉洗衣鼓	5 分鐘	8 分鐘	15 分鐘
100	旋乾	2 分鐘	5 分鐘	10 分鐘

(a) 試以文字描述下列指令序列所指定的洗衣機操作步驟：

001 01
011 10
000 00
100 00

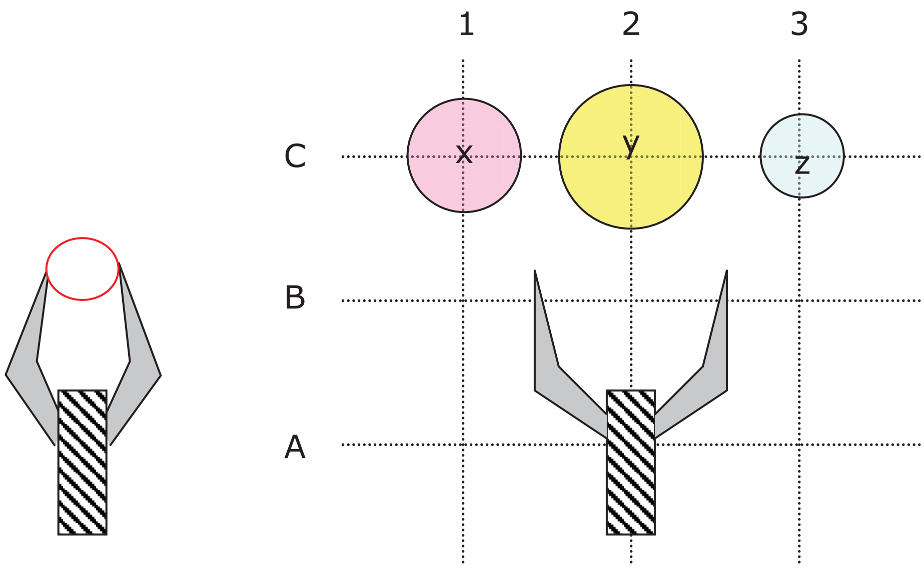
(b) 試寫出與下列算法相應的指令序列：

步驟 1: 注水至高水位
步驟 2: 將水加熱至 40°C
步驟 3: 旋轉洗衣鼓 13 分鐘
步驟 4: 排水
步驟 5: 注水至正常水位
步驟 6: 旋轉洗衣鼓 15 分鐘
步驟 7: 排水
步驟 8: 旋乾 9 分鐘

(c) 試舉出以下指令序列不合理的一個理由。

001 10
011 10
000 00
010 00

Q2. 1999



一機械臂可拾起和疊高一些等高的實心圓柱體。圖一為顯示機械臂與圓柱體的平面圖。此機械臂可伸展和縮回，也可上升和下降，但只可沿水平綫 A 移動。此機械臂由以下各對操作控制：

操作	描述		操作	描述
「握」	以爪握	⇔	「放」	從爪中放下物件
「升」	將物件升上一級（升幅等同一圓柱體之高度）	⇔	「降」	將物件降下一級（降幅等同一圓柱體之高度）
「伸」	臂伸展一格	⇔	「縮」	臂縮回一格
「左」	臂左行一格	⇔	「右」	臂右行一格

升降、左右、伸縮、握放

Operation	Description		Operation	Description
HOLD	Grips object with the jaws	⇔	**RELEASE**	Releases objects from the jaws
UP	Raises object one level up	⇔	**DOWN**	Sinks object one level down
EXTEND	Extends the arm one grid	⇔	**RETRACT**	Retracts the arm one grid
LEFT	Moves arm one grid to the left	⇔	**RIGHT**	Moves arm one grid to the right

每項操作以 3 位表示。每對操作中，其中項操作之位模式為另一項之一的反碼。「握」**HOLD**、「降」**DOWN**、「伸」**EXTEND**、「右」**RIGHT** 之位模式 bit-pattern 分別為 001、101、011、111。試求以下各操作之位模式：放升縮左

操作	描述		操作	描述
「握」	001	⇔	「放」	
「升」		⇔	「降」	101
「伸」	011	⇔	「縮」	
「左」		⇔	「右」	111

(a)(i) 設機械臂及三個圓柱體之放置如圖一所示。
試將下表抄寫到答題簿中，並寫出與以下算法相應之指令序列：

步驟 1：將臂由 A2 移至 A3。「伸」臂一格。「握」圓柱體 z。 （3 個指令）
步驟 2：「縮」臂，將圓柱體 z 從 B3 移至 B1 ，
將之疊高並「放」在位於 C1 之圓柱體 x 之上。 （6 個指令）
步驟 3：「縮」臂並「降」下。將臂移至 A2。「伸」臂一格。
「握」位於 C2 之圓柱體 y。 （5 個指令）
步驟 4：移動臂，把圓柱體 y「放」在 C3，然後「縮」臂。 （5 個指令）
步驟 5：移動臂，把圓柱體 z 移至 C2 及放下。 （10 個指令）
步驟 6：移動臂，把圓柱體 x 疊高放在位於 C3 之圓柱體 y 之上。 （10 個指令）

步驟	指令									
	1	2	3	4	5	6	7	8	9	10
1	*111*	*011*	*001*							
2										
3	*100*	*101*	*111*	*011*	*001*					
4										
5	*000*	*000*	*010*	*011*	*001*	*100*	*101*	*111*	*011*	*110*
6										

(ii) 步驟 7 由以下指令組成。試述完成步驟 7 後之結果。

指令														
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
100	*101*	*000*	*000*	*011*	*001*	*100*	*111*	*111*	*010*	*010*	*011*	*110*	*100*	*100*

(b) 為什麼使用低階 low-level 控制碼，控制機械臂的操作，較為可取？

Q6. 1999 一個專用監視器 monitor 可以顯示 10 行的字符，每行可顯示 14 個字符。每個顯示字符由一個字節代表，重複出現的字符，數據由兩個字節代表，且以下列形式表示：

第一個字節		第二個字節
7 ... 0	位	7 ... 0
字符的重複次數		字符

第一字節的最左位為 1，代表重複字符表示式。

第一字節最右 7 位為字重複的次數。第二字節為字符的二進制的位模式。

例如，

位模式	描述
10010100 00111111	20 個重複的？ (63)
01100001	字符 a (97)

(a) 用最短的位模式寫出 20 個重複的 A。

(b) 以下的位模式分別代表什麼？

(i) 0100 0000

(ii) 1111 1111 0011 1111

(c) 監視器顯示以下十行字符：

A	A	A	A	A	A	A	A	A	A	A	A	A	A
D	O	S	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?

用$\boxed{\text{最短}}$的位模式，寫出表示以上監視器所顯示的字符。

行	位模式 bit-pattern
第 1 行	
第 2 至 10 行	

(d) 寫出$\boxed{\text{最短}}$的位模式，以表示一整屏幕共 140 個的"？"。

Q4. 2000

假設電腦的字長為6位(bit)，而所有電腦連接成一電腦網絡系統，各電腦有一個唯一的網絡識別編號 (Net ID)。 各Net ID由2字 (word)組成，舉例來說，Net ID 23-41 (以十進制表示) 和 56-16 (以十進制表示) 的位模式bit-pattern如下所示：

	十進制表示：							-						
十進制表示：	23							-	41					
位模式：	0	1	0	1	1	1		-	1	0	1	0	0	1

							-					
十進制表示：	56						-	16					
位模式：	1	1	1	0	0	0	-	0	1	0	0	0	0

(a) 寫出 Net ID 5-12 的位模式。

(b) 寫出最大的 Net ID （以十進制表示）。

系統的網絡有三個類別，各有不同格式的Net ID。

各Net ID分為三部分：(i)網絡類別class、(ii)LAN編號 和(iii)電腦編號。

A類 Net ID格式：位模式的首位是 "0"

0	N	N	N	C	C	-	C	C	C	C	C	C

B類 Net ID格式：位模式的首兩位是 "10"

1	0	N	N	N	N	-	N	N	C	C	C	C

C類 Net ID格式：位模式的首兩位是 "11"

1	1	N	N	N	N	-	N	N	N	N	C	C

（貯存LAN編號的位，用N表示。 貯存電腦編號的位，用C表示）例如：

Net ID	位模式	網絡類別	LAN編號		電腦編號	
			二進制	十進制	二進制	十進制
23-41	010111-101001	A	101	5	11101001	233
56-16	111000-010000	C	10000100	132	00	0

(c) 完成下表，並將該表抄寫至答題簿內。

Net ID	位模式	網絡類別	LAN編號		電腦編號	
			二進制	十進制	二進制	十進制
31-1	011111-000001					

(d) 假設何先生欲建立一B類別網絡，其LAN編號為"110101"，
該網絡的最小Net ID為45−16。

(i) 以二進制位和十進制位表示法，寫出此區域網絡最大有效的Net ID。

(ii) 假設各電腦都有一個唯一的電腦編號，此區域網絡最多可連接多少部電腦？

2004

5.(a) 試根據 ASCII 表，找出下列字符以 8 位表示的位模式(bit pattern):

(i) A

(ii) B

p 和 q　是兩組位模式 bit-pattern(二進制位)，分別是 8 位長。

p#q　這個運算定義如下所示: #是運算符號

p	q	p#q
0	0	0
0	1	0
1	0	0
1	1	1

例如:

p	00110000
q	01010000
p#q	00010000

(b)根據 ASCII 表，假設 p 和 q 分別是字符'A'和'B'的位模式開端加上一個零。

(i)試找出 p#q 的位模式和其代表的 ASCII 字符。

p#q 的位模式:

ASCII 字符:

(ii)試找出 11110000#p 的位模式。

(iii) 若 11110011 # e = 11110011

和 11011110 # e = 11011110，試找出 e 的位模式。

(c)假設 p 和 q 分別是十進制整數 12 和-12 的位模式，以 8 位二進制補碼表示。

試找出 p#q 的位模式和其代表的十進制整數。

	位模式	十進制整數
p		12
q		-12
p#q		