

進位轉換邏輯運算.cpp

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int i;
7      unsigned int Mask=0x8;
8      for(i=15 ; i>=8 ; i-=1){
9          cout<<i<<" ";
10         cout<<(i&~Mask)<<" ";
11         cout<<(i|Mask)<<endl;
12     }
13     return 0;
14 }
15
16
```

第 7 行 : unsigned int 值的範圍 0 到 4,294,967,295

第 7 行 : 宣告 Mask 為變數名稱 0x 改十進位的 8 為 16 進位

第 8 行 : for 迴圈 i 從 15 到 8 , 每次減 1

第 9 行 : 輸出 i ,

第 10 行 : &及~為二進位運算子,所以要將 i 和 Mask 改為二進位後在運算

Ex : i=15 -> 1111 , Mask=8 -> 1000

接著依照運算子順序做運算 :

~Mask = 0111

(i & 0111) = 1111 & 0111 = 0111 = 7(十進位)

第 11 行 : (i | Mask) = 1111 | 0111 = 1111 = 15(十進位)

備註 : 輸出時, 會輸出十進位的數字 (除非使用特殊語法)

運算子順序：

() [] -> .	left to right
~ ++ -- + - * & (type) sizeof	right to left
* / %	left to right
+ -	left to right
>> <<	left to right
< <= > >=	left to right
== !=	left to right
&(bitwise AND)	left to right
^(bitwise XOR)	left to right
(bitwise OR)	left to right
&&	left to right
	left to right
?:	right to left
= += -= *= /= %= &=	right to left
^= = <<= >>=	right to left
,	left to right

位元運算子	符號	範例	a	b	c
AND	&	c = a & b	1010	1001	1000
OR		c = a b	1010	1001	1011
XOR	^	c = a ^ b	1010	1001	0111
NOT	~	c = ~a	1010		0101