**★★★☆☆**

**題組：**Volume 101 (10100-10199)

**題號：**10144 : Expression

**解題者：王劭陽**

**解題日期：**2014**年**3**月**19**日**

**題意：**

給定兩個二進制數A和B的位數n，使用A0~An-1和B0~Bn-1代表A、B的各個位，求A＋B的溢出位的布爾表達式（只允許使用NAND（與非）運算，用`|`表示）。

所有標題字型，中文為「標楷體」14號粗體，英文或數字為「Times New Roman」14號粗體。其餘內文，中文為「細明體」12號不粗體，英文或數字為「Times New Roman」12號不粗體。

如果有分段，段首內縮兩個字。行高調整為「固定行高」，行距調整為「18pt」。上述格式的樣式存於本檔案的「軟體比賽內文」樣式。紙張橫寬為標準的14.5cm，高度亦為標準的21.3cm。

**題意範例：**

1 ➔ ((A0|B0)|(A0|B0))   
2 ➔ ((A1|B1)|(((A0|B0)|(A0|B0))|((A1|A1)|(B1|B1))))

**解法：**

設 A = An-1 An-2 … A0, B = Bn-1 Bn-2 … B0

設 C是一個array，保存A＋B過程中每一位的進位情況，

即C[0] = 0, C[i]代表第i - 1位是否有進位。

顯然若Ai、Bi、C[i]中至少有兩個1時，C[i+1]等於1，否則等於0。

寫成布爾表達式如下：

C[i+1] = Ai．Bi + Ai．C[i] + Bi．C[i]

於是題目要求我們輸出的答案就是C[n]了。

因為題目只允許是用NAND運算，所以我們現在將公式變形。

已知 X．Y = ¬(X|Y)

X + Y = ¬¬(X + Y) = ¬(¬X．¬Y) = ¬X|¬Y

¬X = ¬X + ¬X = X|Y 則有：

C[i+1] = Ai．Bi + Ai．C[i] + Bi．C[i]

= Ai．Bi + (Ai + Bi)．C[i]

= ¬(Ai|Bi) + (¬Ai|¬Bi)．C[i]

= ¬(Ai|Bi) + ¬((¬Ai|¬Bi)|C[i])

= (Ai|Bi)|((¬Ai|¬Bi)|C[i])

= (Ai|Bi)|(((Ai|Ai)|(Bi|Bi))|C[i])

公式已得，A、B均已知，C[0]為0，直接通過遞推的手段得出C[n]，輸出即可。

**解法範例：**

無。

**討論：**

無。

**程式：**

#include <stdio.h>

#include <stdlib.h>

char \*\*s;

void workOut(int n) {

s = (char \*\*)malloc(101 \* sizeof(char \*));

s[1] = (char \*)malloc(50 \* 101 \* sizeof(char));

sprintf(s[1], "((A0|B0)|(A0|B0))");

for (int i = 1; i < n; i++) {

s[i + 1] = (char \*)malloc(50 \* 101 \* sizeof(char));

sprintf(s[i + 1], "((A%d|B%d)|(%s|((A%d|A%d)|(B%d|B%d))))", i, i, s[i], i, i, i, i);

}

}

int main() {

workOut(100);

int ttt;

scanf("%d", &ttt);

for (int tt = 0; tt < ttt; tt++) {

int n;

scanf("%d", &n);

printf("%s\n", s[n]);

if (tt != ttt - 1) {

printf("\n");

}

}

}