

## 6 Bonus: What is the problem?

看到題目中的程式，我首先想到浮點數誤差。也就是當十進位小數轉成二進位時，可能產生二進位無窮循環小數，但電腦可以儲存的空間是有限的，所以就會對無窮循環小數做截斷，因而產生微小的誤差！比對時可能因為微小誤差導致判斷結果與常識不合！

我就寫了一支程式來驗證，以下是原始碼

```
#include <stdio.h>
#include <stdint.h>
union ud{
    uint64_t u;
    double d;
};

int main(){
    union ud a, b, c;
    a.d = 0.1; b.d = 0.2; c.d = 0.3;
    printf("a = %llx\nb = %llx\nc = %llx\n", a.u, b.u, c.u);
    union ud tmp; tmp.d = a.d+b.d;
    printf("a+b = %llx\n", tmp.u);

    union ud x; x.d = 0;
    for(int i = 0 ; i < 3 ; ++i) x.d += 0.1;
    printf("0.1+0.1+0.1=\t%llx\n0.3=\t\t%llx\n", x.u, c.u);
    return 0;
}
```

輸出結果為

```
a = 3fb9999999999999a
b = 3fc9999999999999a
c = 3fd3333333333333
a+b = 3fd3333333333334
0.1+0.1+0.1= 3fd3333333333334
0.3= 3fd3333333333333
```

可以看到0.3的double literal與0.1+0.1+0.1的最後一個16進位digit不同，也就是上述的小數誤差(當然也可以看到a與b的hex最後一位為a，就是循環9到最後面時四捨五入變成a的)

## 小結

因為小數誤差的關係，導致0.1+0.1+0.1會比0.3還要大一點點，但這一點點就會導致原題敘中的程式的for迴圈退出！

## 參考資料

[0.30000000000000004](#)以及此網站內所列的連結參考資料

## 附註

因為本文件的原始檔為Markdown(md原始碼也付在資料夾內了)，所以不方便截圖，只以複製實驗結果的方式呈獻。