2.5 Bit Operation

```c
#include <stdint.h>
#include <stdio.h>
int main()
{
    int32_t number = 0;
    scanf("%d", &number);
    int32_t bit = 1;
    bit = bit << 31;
    for (int i = 0; i < 32; i++)
    {
        if (bit & number)
            printf("1");
        else
            printf("0");
        bit = bit >> 1;
    }
    return 0;
}
```

原程式碼在應該進位轉零時只有進位沒有將1消掉
例如2應該轉成10，但輸出變成11
問題來自於bit型態為有號數
導致迴圈中右移(ARSH) 保留最高位

$(10000000000000000000000000000000)_2$
$(11000000000000000000000000000000)_2$
$(11100000000000000000000000000000)_2$

所以將bit改為無號數(uint32_t)就可正常運作

```
947016S_hw02$ ./hw0205
2
00000000000000000000000000000010lapor@LaporBrokeHisBrain:
/vscode_c/CSIE-Programming-II/40947016S_hw02$ ./hw0205
13
00000000000000000000000000001101lapor@LaporBrokeHisBrain:
/vscode_c/CSIE-Programming-II/40947016S_hw02$ ./hw0205
8
00000000000000000000000000001000lapor@LaporBrokeHisBrain:
/vscode_c/CSIE-Programming-II/40947016S_hw02$ ./hw0205
10
00000000000000000000000000001010lapor@LaporBrokeHisBrain:
/vscode_c/CSIE-Programming-II/40947016S_hw02$
```