

쉬프트 연산자를 사용한 비트단위 ON/OFF

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

void main()
{
    unsigned short shData = 0;           // 본 데이터
    unsigned short shTemp;               // 비트 반전용 임시 데이터

    int iInputBit;
    int iInputFlag;

    int iBitSize = 0;

    int iCount;

    // Data 비트 수 계산 바이트 * 8
    iBitSize = sizeof(shData) * 8;

    while (1)
    {
        printf("비트위치 : ");
        scanf("%d", &iInputBit);

        printf("OFF/ON [0,1] : ");
        scanf("%d", &iInputFlag);

        if (iInputBit > iBitSize || iInputBit <= 0 )
        {
            printf("비트 범위를 초과하였습니다. %d\n", iInputBit);
            continue;
        }

        if ( iInputFlag != 0 )
        {
            shData = shData | (1 << iInputBit - 1);
        }
        else
        {
            shTemp = ~(1 << iInputBit - 1);
            shData = shData & shTemp;
        }

        // 비트 수 만큼 반복문

        for (iCount = 0; iCount < iBitSize; iCount++)
        {
            // 가장 왼쪽부터 출력을 해야하므로
            // 전체 비트 수 - 반복문 카운트.
            // 추가로 - 1 이 있는건 숫자 1을 왼쪽으로 쉬프트 하므로
            // 기본이 1비트 자리를 차지하고 들어가므로.

            // 확인 하고자 하는 위치의 비트에 AND 연산 후 chTemp 에 값을 넣었다.
            shTemp = shData & (1 << iBitSize - iCount - 1);

            // 몇 번째 비트인지 출력
            printf("%d 번 Bit : ", iBitSize - iCount);

            // shTemp 값이 0 이 아니면 어딘가 1비트 체크 된 것
            if (shTemp != 0) // if ( chTemp ) 로 해도 된다.
                printf("ON");
            else
                printf("OFF");

            printf("\n");
        }

        printf("\n");
    }
}
```