



# 시스템프로그래밍기초 실습

---

## Ch7. Bitwise Operators and Enumeration Types

## 실습 예제 1) bitwise.c – bit\_print()

```
void bit_print(int a)
{
    int i;
    int n = sizeof(int) * CHAR_BIT;
    int mask = 1 << (n - 1);

    for (i = 1; i <= n; ++i) {
        putchar(((a & mask) == 0) ? '0' : '1');
        a <<= 1;
        if (i % CHAR_BIT == 0 && i < n)
            putchar(' ');
    }
    putchar('\n');
}
```

## 실습 예제 1) bitwise.c – pack()

```
int pack(char a, char b, char c, char d)
{
    int p;

    p = a;
    p = (p << CHAR_BIT) | b;
    p = (p << CHAR_BIT) | c;
    p = (p << CHAR_BIT) | d;

    return p;
}
```

## 실습 예제 1) bitwise.c – unpack()

```
char unpack(int p, int k)
{
    int n = k * CHAR_BIT;
    unsigned mask = 255;

    mask <<= n;
    return ((p & mask) >> n);
}
```

## 실습 예제 1) bitwise.c – main() (1)

```
#include <stdio.h>
#include <stdlib.h> /* for exit(), atoi() */
#include <limits.h> /* for CHAR_BIT */

void bit_print(int);
int pack(char, char, char, char);
char unpack(int, int);

int main(int argc, char *argv[])
{
    if (argc < 3) {
        printf("usage : bitwise num1, num2\n");
        exit(1);
    }

    int num1 = atoi(argv[1]);
    int num2 = atoi(argv[2]);

    /* bit operators */
    printf("%d, %d : \n", num1, num2);
    bit_print(num1);
    bit_print(num2);
}
```

## 실습 예제 1) bitwise.c – main() (2)

```
/* bit operators */
printf("%d, %d :\n", num1, num2);
bit_print(num1);
bit_print(num2);

printf("~%d, ~%d :\n", num1, num2);
bit_print(~num1);
bit_print(~num2);

printf("%d & %d :\n", num1, num2);
bit_print(num1 & num2);

printf("%d | %d :\n", num1, num2);
bit_print(num1 | num2);

printf("%d ^ %d :\n", num1, num2);
bit_print(num1 ^ num2);

/* bit shift */
printf("%d << 2, %d >> 5 :\n", num1, num2);
bit_print(num1 << 2);
bit_print(num2 >> 5);
```

## 참고자료) 비트 연산자(&, |, ^)

연산자	비트1	비트2	결과
&	0	0	0
	0	1	0
	1	0	0
	1	1	1
	0	0	0
	0	1	1
	1	0	1
	1	1	1
^	0	0	0
	0	1	1
	1	0	1
	1	1	0

두 비트가 모두 1이면 1

두 비트 중 하나라도 1이면 1

두 비트가 서로 다르면 1

## 실습 예제 1) bitwise.c – main() (3)

```
/* packing */
char c1 = 'c';
char c2 = 'h';
char c3 = 'a';
char c4 = 'r';

printf("%c, %c, %c, %c :\n", c1, c2, c3, c4);
bit_print(c1);
bit_print(c2);
bit_print(c3);
bit_print(c4);

int pdata = pack(c1, c2, c3, c4);
printf("packed data :\n");
bit_print(pdata);

printf("unpack(pdata, 0) : %c\n", unpack(pdata, 0));
printf("unpack(pdata, 2) : %c\n", unpack(pdata, 2));

return 0;
```

}



## 과제 1) sign.c

```
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <unistd.h>

void moving_sign(char, char, char, char);
void print_sign(int);

int main(void)
{
    moving_sign('H', 'E', 'Y', ' ');
    return 0;
}
```

# 과제 1) sign.c

```
void print_sign(int a)
{
    printf(" %c %c %c %c \n",
        a/(256*256*256), a/(256*256), a/256, a);
}

void moving_sign(char _a, char _b, char _c, char _d)
{
    char a = _a, b = _b, c = _c, d = _d, tmp;

    /* 정수 p를 최상위 문자부터 차례대로 a b c d로 값을 할당해주세요. */
    int p;

    while(1) {
        system("clear");
        print_sign(p);
        sleep(1);
        /* 최상위 문자를 tmp에 저장했다가
         * 정수 p를 문자만큼 이동시키고
         * 저장해두었던 tmp를 최하위 문자에 할당하세요.
         */

    }
}
```

# 과제 검사방법

## 예제 1) bitwise.c

1. 실행 결과 확인

## 과제 1) sign.c

1. 실행 결과 확인
2. 작성한 코드 검사