



Lab 05

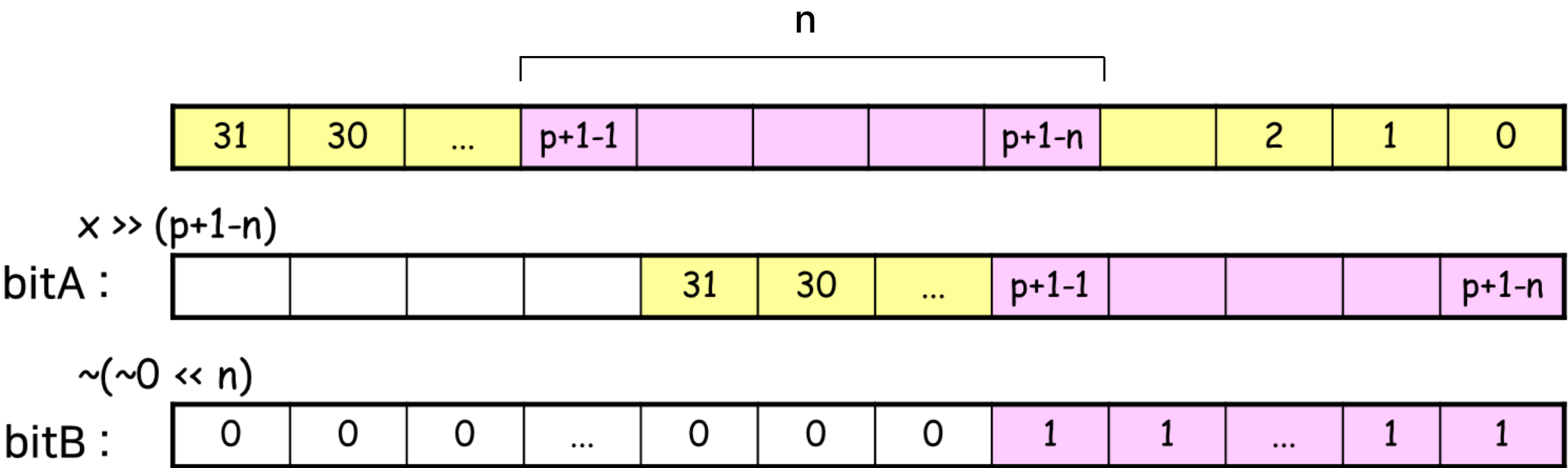
2023학년도 2학기 프로그래밍개론

01, 02, 03분반 TA 유사라
숙명여자대학교 소프트웨어학부
데이터지능연구실

{4ra}@sookmyung.ac.kr

- 실습: 연습문제 2-6의 선행 과제
- 과제: 연습문제 2-6

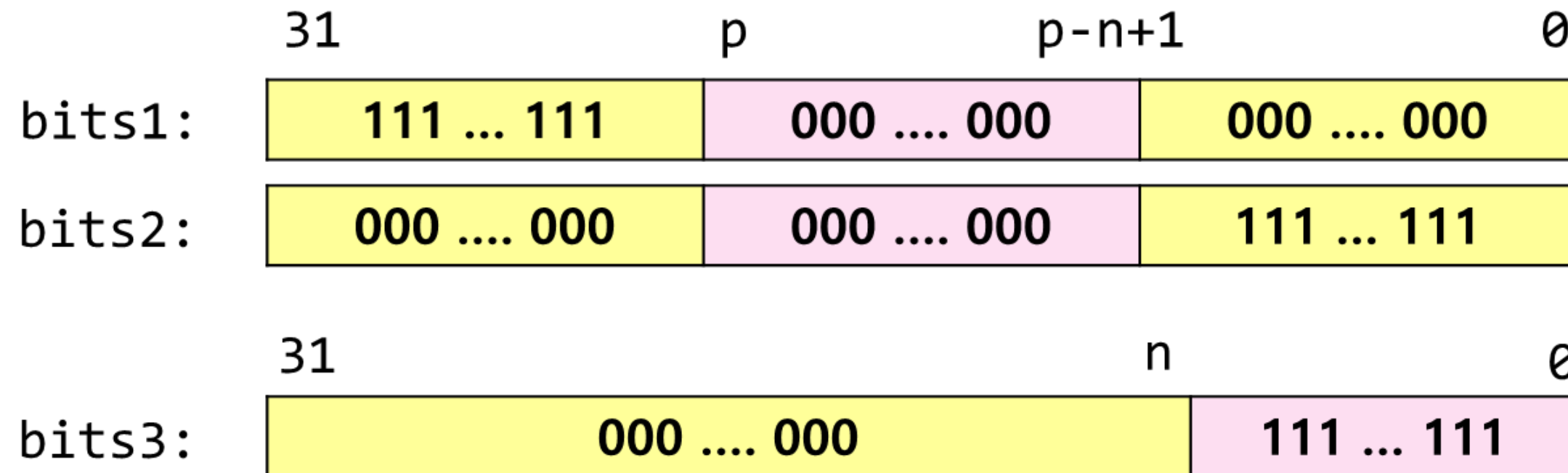
- 정수 x에서 p번 비트부터 n개의 비트를 추출하는 함수 작성



~ 0 : 11111111...111111
 $\sim 0 \ll n$: 11...1100...00
(마지막 n 비트만 0)
 $\sim(\sim 0 \ll n)$: 00...0011...11
(마지막 n 비트만 1)

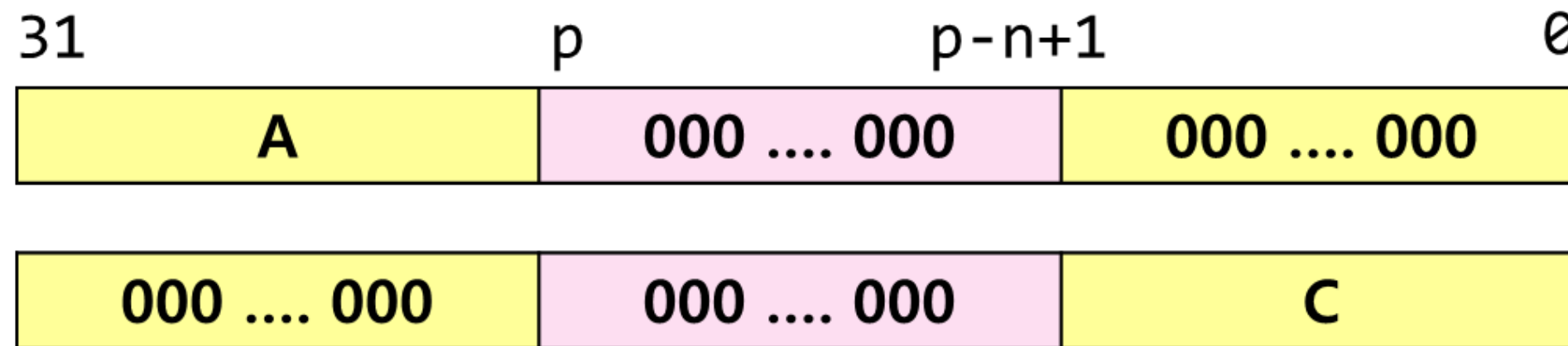
```
/* get n bits from position p */
unsigned getbits(unsigned x, int p, int n)
{
    return (x >> (p+1-n)) & ~(\sim 0 << n); // bitA & bitB 와 동일
}
```

- How can you get each of the following :

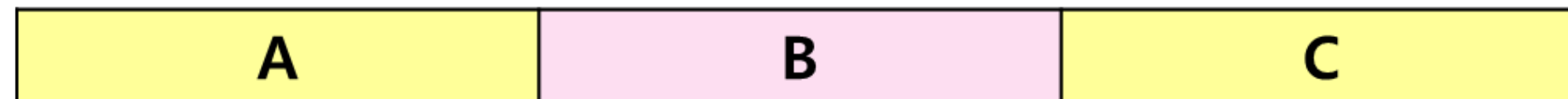


실습) 연습문제 2-6의 선행과제

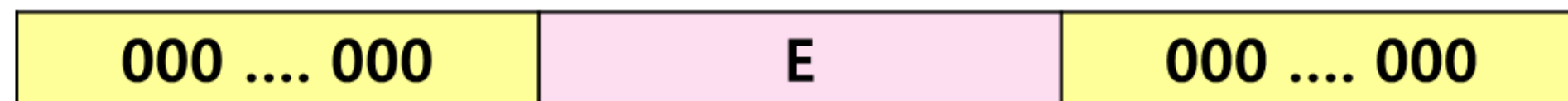
- How can you get each of the following :



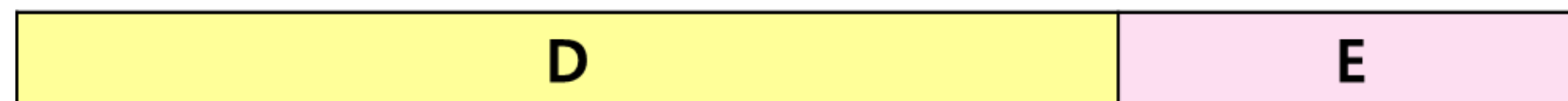
- from x :



- and the following :



- from y:



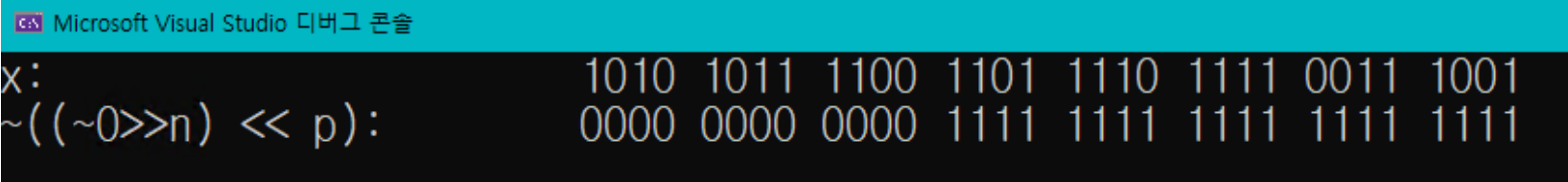
```
#include <stdio.h>
#include <string.h>

void printbinary(unsigned x) {
    int i;
    for (i = 0; i < 32; i++) {
        putchar(x & 0x80000000 ? '1' : '0');
        if (i%4 == 3) {
            putchar(' ');
        }
        x <<= 1;
    }
    putchar('\n');
}
```

```
main() {
    int p = 20, n = 6;
    int x = 0xABCDEF39;
    printf("x: \t\t\t"); printbinary(x);
    printf("~((~0>>n) << p): \t"); printbinary(~((~0>>n) << p));
}
```

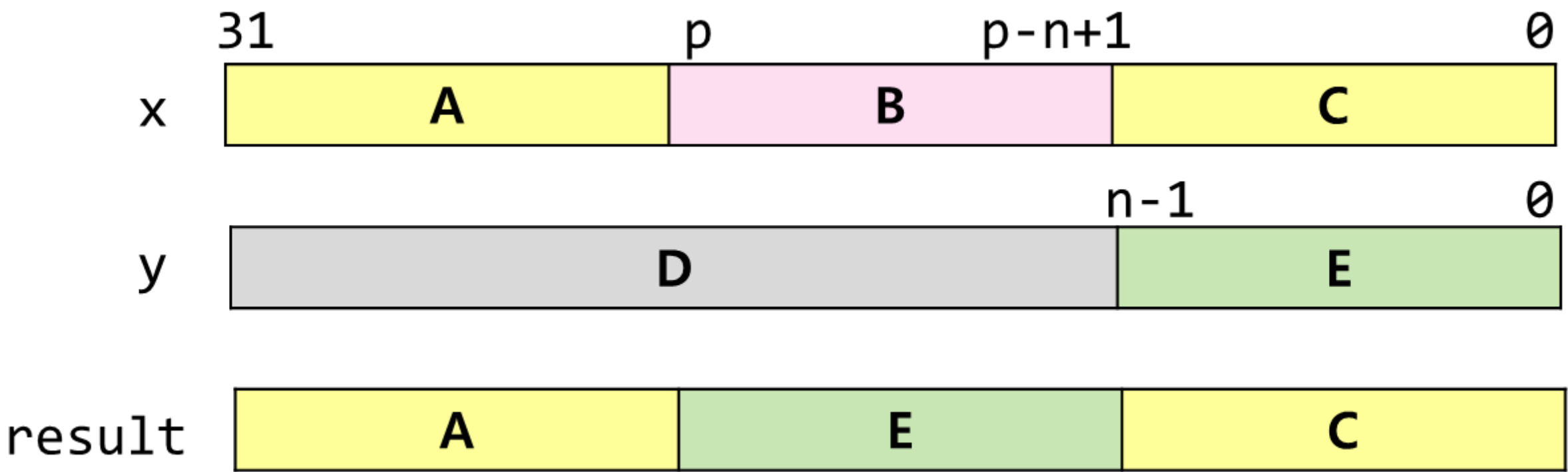
- **printbinary()** 함수를 사용하여 앞에서 구한 식들이 올바른지 확인할 수 있다.

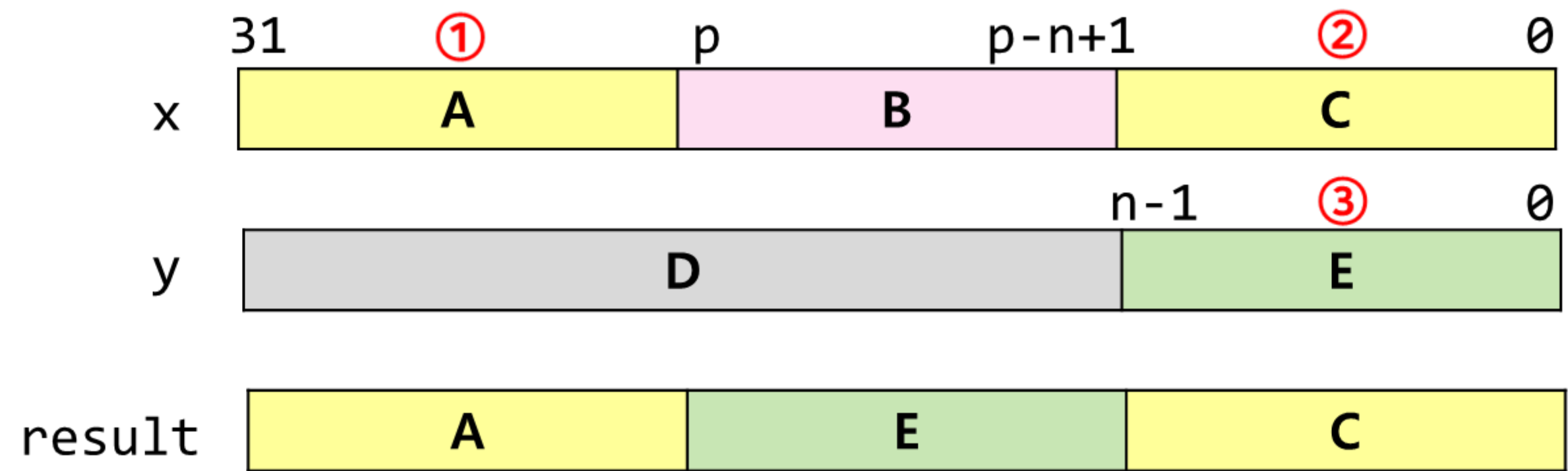
- 실행 결과 :



```
Microsoft Visual Studio 디버그 콘솔
x: 1010 1011 1100 1101 1110 1111 0011 1001
~((~0>>n) << p): 0000 0000 0000 1111 1111 1111 1111 1111
```

- x에서 p번 비트 이후 n개의 비트는 y 의 가장 오른쪽 n 비트로 변경하고, 나머지 비트들은 그대로 둔 결과를 리턴하는 함수 `setbits(x,p,n,y)`를 작성하시오.



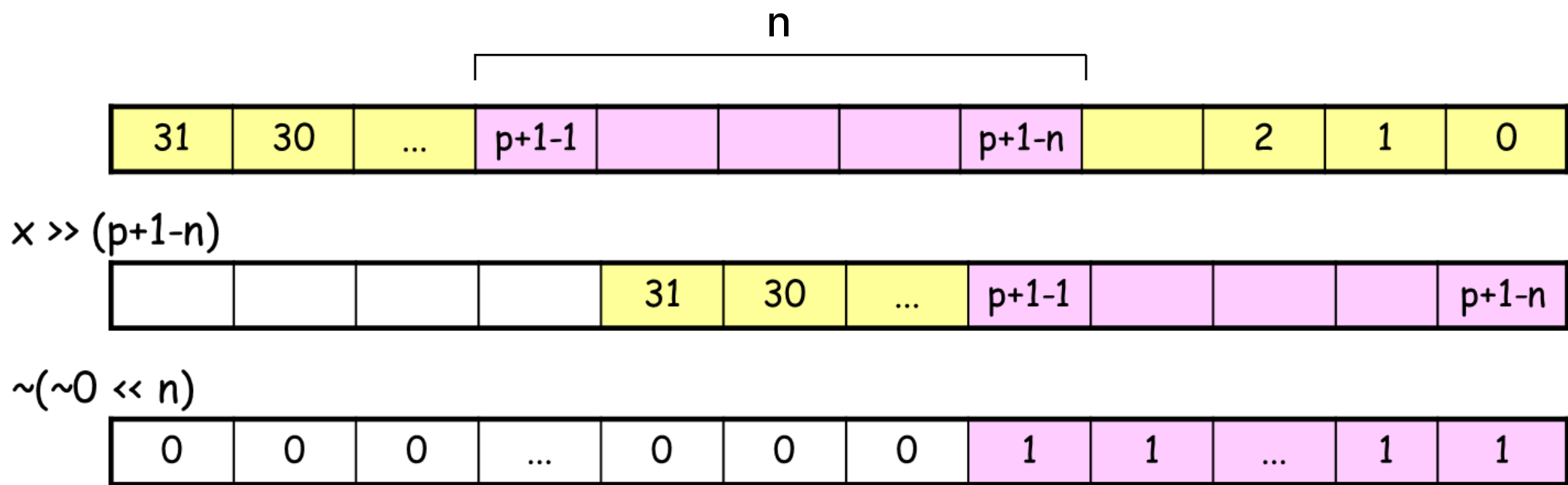


• A 구간 , B 구간, C 구간 bit들을 추출하기 위한 변수들 필요

- ① : A 구간 추출하는 변수 → 예) `int bitsA;`
- ② : C 구간을 추출하는 변수 → 예) `int bitsC;`
- ③ : E 구간을 추출하는 변수 → 예) `int bitsE;`

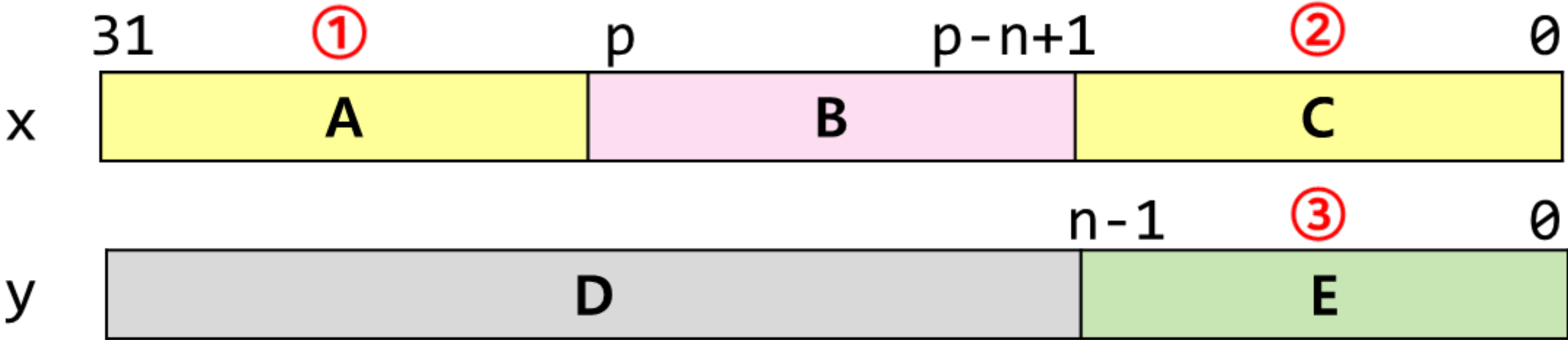
HINT !) 이전 실습의 `printbinary()` 함수를 활용하여 올바르게 진행하고 있는지 확인할 수 있음 (제출 시에는 주석 처리)

- 정수 x에서 p번 비트부터 n개의 비트를 추출하는 함수 작성



~ 0 : 11111111...111111
 $\sim 0 \ll n$: 11...1100...00
(마지막 n 비트만 0)
 $\sim(\sim 0 \ll n)$: 00...0011...11
(마지막 n 비트만 1)

```
/* get n bits from position p */
unsigned getbits(unsigned x, int p, int n)
{
    return (x >> (p+1-n)) & ~(\sim 0 << n);
}
```



① : A 구간 추출하는 변수 → 예) `int bitsA;`

EX) A 구간 추출 방법

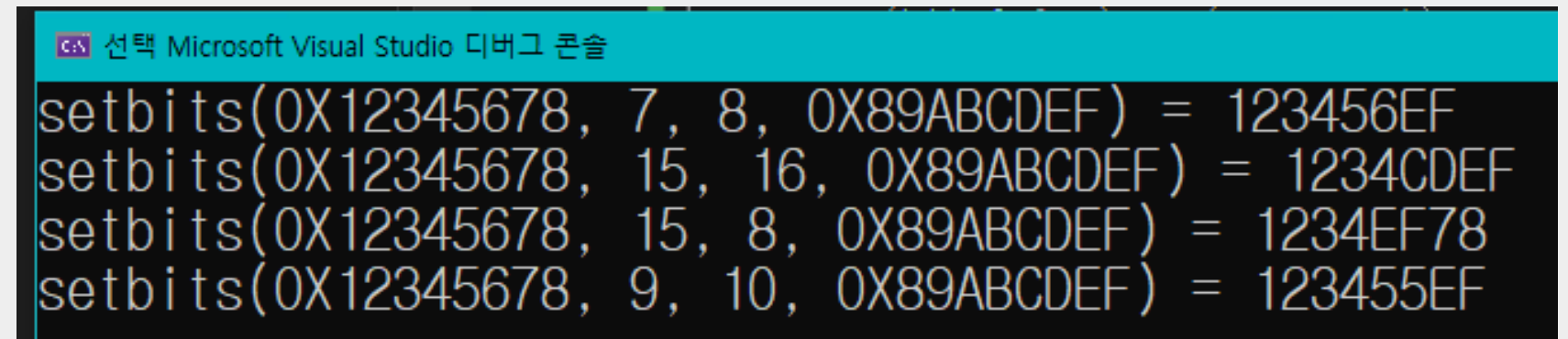
- 1. A 구간을 추출하는 변수 `int bitsA` 생성
- 2. `bitsA`와 `x`를 논리연산자를 통해 A구간 추출

→ 추출한 A 구간 E구간 C구간을 OR 연산하여 `result` 완성



```
unsigned setbits(unsigned x, int p, int n, unsigned y)
{
    /* 구현 */
}
```

```
main()
{
    /* main 함수는 과제 중 수정하지 않고 제출 */
    printf("setbits(0X12345678, 7, 8, 0X89ABCDEF) = %X\n", setbits(0X12345678, 7, 8, 0X89ABCDEF));
    printf("setbits(0X12345678, 15, 16, 0X89ABCDEF) = %X\n", setbits(0X12345678, 15, 16, 0X89ABCDEF));
    printf("setbits(0X12345678, 15, 8, 0X89ABCDEF) = %X\n", setbits(0X12345678, 15, 8, 0X89ABCDEF));
    printf("setbits(0X12345678, 9, 10, 0X89ABCDEF) = %X\n", setbits(0X12345678, 9, 10, 0X89ABCDEF));
}
```



선택 Microsoft Visual Studio 디버그 콘솔

```
setbits(0X12345678, 7, 8, 0X89ABCDEF) = 123456EF
setbits(0X12345678, 15, 16, 0X89ABCDEF) = 1234CDEF
setbits(0X12345678, 15, 8, 0X89ABCDEF) = 1234EF78
setbits(0X12345678, 9, 10, 0X89ABCDEF) = 123455EF
```

<실행 결과>

- 과제 제출 기한
 - 01, 02분반 : 10월 17일 (화) 23:59 까지
 - 03분반 : 10월 18일 (수) 23:59 까지
 - 제출 기한 이후 24시간 이내 지각 제출할 경우 : 2점 감점
 - 그 이후는 0점 처리 (추가 제출 받지 않음)
- 제출 장소
 - 스노우보드 과제 제출 페이지에 업로드

- 소스파일(.c)과 과제보고서(.docx)가 담긴 **압축파일(.zip) 제출**
 - 압축 파일 이름: Lab04_학번_이름.zip
'Lab04_학번_이름'으로 된 c파일 + 'Lab04_학번_이름'으로 된 .docx파일
- 과제보고서(.docx) 양식
 - 스노우보드에서 다운로드
 1. 실행 결과 화면 캡처한 이미지 첨부
 2. 소스 코드
 3. 소스 코드에 대한 설명 (간략하게 3-4줄)

- 조교 메일로 질문 보내기

- 4ra@sookmyung.ac.kr

- 질문시 주의사항

- 메일에 반드시 과목, 분반, 전공, 이름, 학번 명시
- 몇 번 과제에서 어떤 부분이 막혔는지, 어떤 부분이 문제인지 코드와 함께 자세한 설명 첨부
(그냥 코드만 보내면 (or 코드 없이 실행 결과만 보내면) 어디가 문제인지 알 수 없어요)
- 답장이 늦을수도 있으니 이 점 고려하여 미리 질문 (특히 과제 제출 마지막날 유의!)
- 그 외 출석 등 다른 질문들도 메일로