

- 다음은 goto 문장을 이용한 프로그래밍 방법에 대한 문제들이다. C 언어로 물음에 답하시오.
 - 슬라이드 2-34에서의 if-then-else 문장을 if-then 문장과 goto 문장으로 다시 쓰시오. 여기에서 if-then 문장은 else가 배제되는 형태를 가리키며, 복합문은 허용되지 않는다(then 절에 복수 개의 문장이 올 수 없다).
 - 슬라이드 2-35에서의 while 문장을 (a)에서와 같은 요령으로 다시 작성하시오.
- 0x12345678이 메모리 주소 8번지에 저장되어 있다. 물음에 답하시오.
 - Big Endian을 전제로 각 주소에 저장되는 값을 결정하시오.
 - Little Endian에 대하여 (a)를 반복하시오.
- 다음은 특정 32 비트 주소에 저장된 값을 \$s0에 적재하기 위한 MIPS 코드이다. 물음에 답하시오. 단, 32비트 주소의 상위 16비트와 하위 16비트가 각각 A_upper와 A_lower라고 가정한다.
 - 괄호 안을 채우시오.


```
lui    $t0, A_upper
(    ) $t0, $t0, A_lower
lw     $s0, 0($t0)
```
 - 다음 코드가 위의 코드와 동일한 기능이 되도록 A_upper_adjusted를 결정하시오.


```
lui    $t0, A_upper_adjusted
lw     $s0, A_lower($t0)
```
- 다음 C 문장을 MIPS instruction sequence로 바꾸어 표현하시오. 단, 모든 변수는 레지스터에 할당되어 있으며 상수는 십진수로 표시되었다고 가정한다.
 - $a = b + 10000;$
 - $a = b + 100000;$
- 다음 C 문장을 MIPS instruction sequence로 바꾸어 표현하시오. 단, 배열을 제외한 모든 변수는 레지스터에 할당되어 있다고 가정한다.
 - $B[4] = A[i];$
 - $B[8] = A[i-j];$
- 현재 PC에 저장된 주소가 0x20000000이다. 물음에 답하시오.
 - 0x20001000 주소로 제어를 이동하려고 한다. 0x20000000에 branch instruction이 들어 있다고 가정하고 이 branch instruction을 완성하시오.
 - Jump instruction으로 (a)를 반복하시오.
 - 0x30000000 주소로 제어를 이동하려고 한다. 방법을 제시하시오.

7. 다음은 MIPS instruction들이다. 레지스터 PC와 \$s1에 저장된 값을 각각 0x20000000과 0x80000000이라고 가정하고 제시된 instruction 이후에 접근하는 메모리 주소들을 각각 쓰시오.
- (a) lw \$t0, 0x100(\$s1)
 - (b) beq \$zero, \$zero, 0x100
 - (c) j 0x1000000
 - (d) addi \$t0, \$s1, 100
8. 다음은 instruction format의 설계에 대한 문제들이다. 레지스터의 개수를 32개에서 128개로 증가시키고 instruction의 수도 네 배로 증가시키려고 한다. 물음에 답하시오.
- (a) R format에 미치는 영향에 대하여 쓰시오.
 - (b) I format에 미치는 영향에 대하여 쓰시오.
 - (c) J format에 미치는 영향에 대하여 쓰시오.
 - (d) 프로그램의 길이에 미치는 영향에 대하여 쓰시오.
9. 슬라이드 2-47을 참조하여 다음과 같이 정의되는 fib 함수를 MIPS instruction sequence로 표현하시오.

```
int fib (int n)
{
    if (n == 0) return 0;
    if (n == 1) return 1;
    return fib(n-1) + fib(n-2);
}
```