

►Solution◄

Question 1: (20 points)

In a database, a date of birth is stored into a single 32-bits word in the following format: MDYY, where M is a single byte used to store the month as an unsigned 8-bit integer, D is a single byte used to store the day as an unsigned 8-bit integer, and YY are two bytes used to store the year as an unsigned 16-bit integer. Assume that this word has been loaded into register `$s0`. Write a sequence of MIPS assembly instructions that, after its execution, will result in register `$v0` containing a 32-bit unsigned integer representing the day of birth extract from the word described above.

Solution: Create a mask, AND with the mask, and then shift:

```
li      $t0, 0x00FF          # only can load a 16-bit constant
sll     $t0, $t0, 16         # $t0 <-- 0x00FF0000
and     $t1, $s0, $t0        # $t1 <-- 0x00dd0000
srl     $v0, $t1, 16         # $v0 <-- 0x000000dd
```

Solution: Shift first, and then AND with a different mask:

```
li      $t0, 0x00FF          # $t0 <-- 0x000000FF
srl     $t1, $s0, 16         # $t1 <-- 0x0000mmdd
and     $v0, $t1, $t0        # $v0 <-- 0x000000dd
```

Solution: No masks, use only shifts:

```
sll     $t1, $s0, 8          # $t1 <-- 0xddyyyy00
srl     $v0, $t1, 24         # $v0 <-- 0x000000dd
```