

►Solution◄

Question 1: (15 points)

Write the assembly code to implement the following C function:

```
int selector(int array[], int i) {  
    return array[array[i]];  
}
```

Remember that the first parameter of a function is passed to the function in register `$a0` and the second parameter is passed in register `$a1`.

Solution:

```
sll    $t0, $a1, 2      # $t0 <-- i * 4  
add    $t1, $a0, $t0    # $t1 <-- Addr(array[i])  
lw     $t2, 0($t1)      # $t2 <-- array[i]  
sll    $t3, $t2, 2      # $t3 <-- 4 * array[i]  
add    $t4, $a0, $t3    # $t4 <-- Addr(array[array[i]])  
lw     $v0, 0($t4)      # $v0 <-- array[array[i]]  
jr     $ra              # return
```

Note: There is no need to save the return address because this is a leaf function (i.e., does not call other functions). If we only use temporary caller-saved registers, then we do not need to save registers on the stack, and thus do not need to do stack manipulation at all.