Common mistakes in programming exam 2 (ordered by number of occurrences):

1. Using odd numbered floating point registers for double precision floats:

   Note: for single precision floating numbers we can use both odd and even numbered floating point registers
   but for double precision, we have to use even numbered registers

   Wrong:

```
l.d $f3, 0($t0)
     ^
    odd
```

   Correct:

```
l.d $f2, 0($t0)
     ^
    even
```

2. In floating point comparison, there is no 'gt'. To achieve the same result, we need to check complement of less than.

   Comparison:

```
if $f0 > $f2
    then branch to 'label'
```

   Essentially, these two comparisons are the same: if (! ($f0 <= $f2)) - and - if ($f0 > $f2)

   Wrong:

```
c.gt.d $f0, $f2
bc1t label
  ^
if result of comparison is true, then branch to 'label'
```

   Correct:

```
c.le.d $f0, $f2
bc1f label
  ^
if result of comparison is false, then branch to 'label'.
```

3. To convert a integer to floating point:
   I) move the integer to co-processor 1
   II) convert the integer than is in co-processor 1 to floating point

   Example:
   $t0 (integer) --> $f0 (single precision floating point)

```
mtc1 $t0, $f0
 ^
move integer to co-processor 1
```

```
        cvt.s.w $f0, $f0
            ^
        convet to single precision from word (integer)

    Example:
        $t0 (integer) --> $f0 (double precision floating point)

        mtc1 $t0, $f0
         ^
    move integer to co-processor 1
        cvt.d.w $f0, $f0
            ^
        convet to double precision from word (integer)

4. Saving register to static variables:

    Example:
        store value in register $t0 to static variable 'array_pointer'

        .data
    array_pointer:  .word   0

        .text
    la $t9, array_pointer
     ^
load the address of variable 'array_pointer' into register $t9

    sw $t0, 0($t9)
     ^
store the content of register $t0 into memory at the address of $t9

    memory[array_pointer] = $t0


    Example:
        load value from static variable 'array_pointer' into register $t0

        .data
    array_pointer:  .word   0

        .text
    la $t9, array_pointer
     ^
load the address of variable into register $t9

    lw $t0, 0($t9)
     ^
```

load the content in memory at address of $t9 into $t0

```
$t0 = memory[array_pointer]
```