1. In a CDA3101 assignment, students were asked to translate the **recursive** C function `foo()` shown below into MIPS assembly code. The MIPS code shown below is a solution provided by a student. You have to play the role of the TA and find and correct all the errors (logical or otherwise) as well as situations where procedure call conventions are not followed in this MIPS code.

```c
int foo(int x) {
    if (x <= 0) return 1;
    else return (x * foo(x-2));
}
```

- Strike each instruction you think is wrong and write the correct instruction next to it.
- If you need to add an instruction, write that instruction and draw an arrow to show where that instruction will go.
- Make only necessary changes

```
foo:  blez  $a0, base
      addi  $sp,$sp, -4
      sw    $ra, 0($sp)
      addi  $a0,$a0,-2
      j     foo
      lw    $ra,0($sp)
      mul   $v1,$v1,$a0
base: jr    $ra
```

**Solution:**

```
foo:  blez  $a0, base
      addi  $sp,$sp, -8
      sw    $a0,4($sp)
      sw    $ra,0($sp)
      addi  $a0,$a0,-2
      jal   foo
      lw    $a0,4($sp)
      lw    $ra,0($sp)
      addi  $sp,$sp, 8
      mul   $v0,$v0,$a0
      jr    $ra
base: addi  $v0,$zero,1
        jr    $ra
```