

Note this post-mortem from an earlier year might help you to get the assignment correct.

### Fall 2001 Assembly Language Post-mortem on Assignment #3

- ① I am looking for a good division of labor between the main program and the subroutine. It's not enough to simply put a couple of lines of the code into a correctly called subroutine. The name of the subroutine is something like `findPrime`. Therefore it should either find all of the primes (which I didn't expect, but at least I gave credit for it), or it should find whether `n` is a prime. That is, I expected that you would mimic the function which in Java would have the header

```
private boolean findPrime (int n) { ...}
```

which would return true (1) in `$v0` if `$a0` were a prime number, and return false (0) in `$v0` if `$a0` were not prime. It receives its arguments in the `$a` registers; it returns its answer in the `$v` registers. Never leave by any other way but `jal`. Don't use global variables (i.e. don't reach for those registers that you did not pass in). (-10%)
- ② Pseudo-code should refer to meaningful variable names. [cf. (9) below]
- ③ The test case 6 .. 23 worked fine.
- ④ The test case 2 .. 5 worked fine.
- ⑤ Don't allow the lines of your program file to fold. Use shorter lines or (within reason; I'm getting old) smaller type.
- ⑥ Late (-5% per half day) [in 2005, not accepted late]
- ⑦ If you know that there's an error and say as much as you know about the error, it is better than pretending there isn't one and letting me find it. E.g. "It loops infinitely. My best guess as to why is ..."
- ⑧ A procedure is best named by what it does, not where it is: not `InnerLoop` or `L2`.
- ⑨ Make your pseudocode self-contained, in English or Java-like. No mention of registers or MIPS instructions. [cf. (2) above]
- ⑩ Use English for comments. Don't make comments repeat the code.
- (11) Your function needs pseudocode and variable usage table. See sample on handout for the assignment, last page. (-5%)
- (12) Comment does not agree with code.
- (13) Sample runs included? (-10%)
- (14) Initialize all variables!
- (15) Do not get help from students who have previously taken this course, or their materials. (-100%)
- (16) Main saves `$t`; subroutine saves `$s` (-5% for each half)
- (17) My eye can read code better if operands are lined up in one column and operators lined up in another column, with labels in column 1, and monospaced font.
- (18) Let nothing intervene between stacking registers and calling a subroutine, for style's sake.
- (19) [not applicable in 2005]
- (20) [not applicable in 2005]