Calling convention rules

All programs and functions must follow all calling convention rules:

1. **Save registers appropriately:** Caller- and callee-saved registers should be saved as needed at the appropriate times.
    - All `$s` registers that get modified in a function should be saved at the top/bottom of that function.
    - Any `$t` registers whose values must survive a function call should be saved before/after that call
    - `$ra` should be saved/restored in any function that calls another as if it were an `$s` register.
2. **Keep functions independent:** There should be no data sharing via registers between functions other than `$a` for arguments and `$v` for return values.
3. **Stack discipline:** Each function, if it modifies `$sp`, should restore it before returning.
4. **Mind the floats:** The `$f` floating point registers also have roles like the above; see here for more information. This will matter in BuildEff.
5. **Contiguous, well-organized functions:** Functions should be contiguous with a single entry-point and clear return point(s).
6. **The `main` function isn't special:** The calling conventions must be followed in every function, *including* ***`main`****!*
7. **No exit syscall:** While it is not illegal to use the exit system call (syscall 10), I am going to prohibit it here, as students often use it to avoid having to learn to return properly from `main`. Your main function should return (`jr $ra`) when finished rather than using the exit syscall.