```
### Logan Mayfield
### Interpreter 2 top-level semantic tests

## *************************** Semantic Tests *****************************

check "expression semantics":
  run("((not 0))") is 1
  run("((not 1))") is 0

  run("((xor))") is 0
  run("((xor 1))") is 1
  run("((xor 0 1 0 1))") is 0
  run("((xnor))") is 1
  run("((xnor 1))") is 0
  run("((xnor 0 1 0 1))") is 1

  run("((and))") is 1
  run("((and 0))") is 0
  run("((and 1 1))") is 1
  run("((and 1 0 1))") is 0
  run("((nand))") is 0
  run("((nand 0))") is 1
  run("((nand 1 1))") is 0
  run("((nand 1 0 1))") is 1

  run("((or))") is 0
  run("((or 1))") is 1
  run("((or 0 0))") is 0
  run("((or 0 1 0))") is 1
  run("((nor))") is 1
  run("((nor 1))") is 0
  run("((nor 0 0))") is 1
  run("((nor 0 1 0))") is 0

  run("((if 1 0 1))") is 0
  run("((if 0 0 1))") is 1

  run("((cond (1 0) (else 1)))") is 0
  run("((cond (0 0) (else 1)))") is 1
  run("((cond (0 0) (0 0) (else 1)))") is 1
  run("((cond (0 1) (1 0) (else 1)))") is 0

end

check "function semeantics":
  ## for "run" the last element of the list is the 'code' we wish to execute
  ## all the items preceding it are function definitions
  run("((fun (f) 1) (f))") is 1
  run("((fun (f x) (not x)) (f 0))") is 1
  run("((fun (f x) (not x)) (f 1))") is 0

  ## run-main is all function definitions, the second argument to run-main
  ## are the 'user' supplied arguments to main.
  run-main("((fun (main x y z) (xor x y z 1)))",[list: 1,0,1]) is 1


  ## Free Identifiers/Static Scope .. invoking g should not
  ## bind y in the body of f
  run("((fun (f x) (and (g x) y)) (fun (g y) (not y)) (f 0))") raises "Free"
  ## no duplicate parameter names
  run("((fun (f x x) (xor x x)) 0)") raises "Duplicate"

end
```

```
check "short circuits":
  ## executing f should raise a Free Identifier error, if short circuiting works
  ##  the these should work. if not, the free id error will pop up
  run("((fun (f x) y) (and 0 (f 1)))") is 0
  run("((fun (f x) y) (nand 0 (f 1)))") is 1
  run("((fun (f x) y) (or 1 (f 1)))") is 1
  run("((fun (f x) y) (nor 1 (f 1)))") is 0
end
```