

```
; single operation arithmetic tests for edge conditions
;; March 29, 1988 by Masinter
```

```
(do-test-group ("One Arg arithmetic opcodes"
               :before (test-defun and-eq-frob (frob test)
               (and (eq frob 'frob) test)))

(do-test
 "Single operand arithmetic test"
 (macrolet
  ((check (op)
    ' (or (and-eq-frob 'frob (eql ,op (setq val (pop arg))))
      (warn "~S (arg1 = ~S) not ~S" ',op arg1 val))))
  (every #'(lambda (arg)
    (let ((arg1 (pop arg))
        (val nil))
      (and (check (il:llsh arg1 1))
            (check (il:lrsh arg1 1))
            (check (il:llsh arg1 8))
            (check (il:lrsh arg1 8))))))
    (progl '( (0 0 0 0 0)
              (1 2 0 256 0)
              (-1 -2 2147483647 -256 16777215)
              (1.0 2 0 256 0)
              (65535 131070 32767 16776960 255)
              (-65536 -131072 2147450880 -16777216 16776960)
              (65536 131072 32768 16777216 256)
              (-65537 -131074 2147450879 -16777472 16776959)
              (1073741824 -2147483648 536870912 0 4194304))
      ' (let ((values '(0 1 -1 1.0 65535 -65536 65536 -65537
                        1073741824)))
        (mapcar #'(lambda (arg1)
          (list arg1 (il:llsh arg1 1)
                  (il:lrsh arg1 1)
                  (il:llsh arg1 8)
                  (il:lrsh arg1 8))) values)
        ))))
  )
STOP
```