circularity

What is the result of the following program?

```
(defvar x (mvec 2 3))
(set-right! x x)
(set-left! x x)
x
```

○ '#(#(2 #(2 3)) #(2 3))

○ x='#(x x) or something similar. Both (left x) and (right x) are x itself.

○ Error

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

eval order

What is the result of the following program?

```
(defvar x 0)
(ivec x (begin (set! x 1) x) x)
```

○ '#(0 1 1)

○ '#(0 1 0)

○ '#(1 1 1)

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

mvec as arg

What is the result of the following program?

```
(defvar x (mvec 1 2))
(deffun (f x)
  (vset! x 0 0))
(f x)
x
```

○ '#(0 2)

○ '#(1 2)

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

var as arg

What is the result of the following program?

```
(defvar x 12)
(deffun (f x)
  (set! x 0))
(f x)
x
```

○ 0

○ 12

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

seemly aliasing a var

What is the result of the following program?

```
(defvar x 5)
(deffun (set1 x y)
  (set! x y))
(deffun (set2 a y)
  (set! x y))
(set1 x 6)
x
(set2 x 7)
x
```

○ 5; 5

○ 6; 7

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

mutable var in vec

What is the result of the following program?

```
(defvar x 3)
(defvar v (mvec 1 2 x))
(set! x 4)
v
x
```

○ '#(1 2 3); 4

○ '#(1 2 4); 4

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

aliasing mvec in mvec

What is the result of the following program?

```
(defvar v (mvec 1 2 3 4))
(defvar vv (mvec v v))
(vset! (vref vv 1) 0 100)
vv
```

○ '#(#(100 2 3 4) #(100 2 3 4))

○ '#(#(1 2 3 4) #(1 2 3 4))

○ '#(#(1 2 3 4) #(100 2 3 4))

○ Error

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

vset! in let

What is the result of the following program?

```
(defvar x (mvec 123))
(let ([y x])
  (vset! y 0 10))
x
```

○ '#(123)

○ '#(10)

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

set! in let

What is the result of the following program?

```
(defvar x 123)
(let ([y x])
  (set! y 10))
x
```

○ 10

○ 123

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?

seemingly aliasing a var again

What is the result of the following program?

```
(defvar x 10)
(deffun (f y z)
  (set! x z)
  y)
(f x 20)
x
```

○ 20; 20

○ 10; 20

○ Other

Please specify

Why do you think this program was included? What do you think it's checking?