a)

$$\frac{\langle e, \rho, \sigma \rangle \Downarrow \langle v, \sigma' \rangle}{\langle \text{VAL}(x, e), \rho, \sigma \rangle \rightarrow \langle \rho, \sigma\{\rho(x) \mapsto v\} \rangle} \text{(DefineGlobal)}$$

whether $x \in \text{dom } \rho$ or $x \notin \text{dom } \rho$ doesn't matter, defines a new x in ρ

b)
```
(Val x 10)
(Val f (lambda () x))
(Val x 5)
(if (= (f) 5)
    'Uscheme-semantics
    'new-semantics)
```

This first binds x to 10. Then creates a lambda which depends on the "most relevant" value of x. It then (re) binds x to a different value. If uses Uscheme semantics, the binding of x has changed, and the function will no longer access the original x value of 10 If using new semantics, the original x value is only covered up, but is still the "most relevant" value of the time of function declaration, so it will return 10

c) I prefer the Uscheme semantics, because the idea of having multiple values all bound to the name x, and which value is chosen depends on where you are in the program. I would much prefer to declare different variable names and have each one with a concrete value.