#### Q1.1 1 Point

Consider the following program, written in the language of lecture6c.rkt from class:

where e is some arbitrary expression in the language.

Suppose that executing the above snippet does not trigger an error. Which of the following are possible values that could be returned (just the value part of the result, not the store part)? Select all that apply.



## Q1.2 1 Point

Consider the following program, written in the language of lecture6c.rkt from class:

where e is some arbitrary expression in the language.

Suppose that executing the above snippet does not trigger an error. Which of the following are possible values that could be returned? Select all that apply.



In lecture6c.rkt we implemented a language with mutable state (through boxes) and begin statements (represented with the seqC constructor in our AST) for sequencing commands. Consider the following incorrect variant of our implementation of seqC in that interpreter:

Suppose b is a variable bound to a box that initially contains the value 0. On which of the following examples does the above implementation of seqc give incorrect behavior, in the sense that calling eval will yield a different result than calling eval in a correct implementation. (Assume that the rest of the implementation of eval-env is the same as in lecture6c.rkt, aside frrom seqc.)

```
(begin (set-box! b 1) (+ (set-box! b 2) (unbox b)))
(begin (unbox b) (unbox b))
(begin (unbox b) (+ (set-box! b 2) (unbox b)))
(begin (set-box! b 1) (+ (unbox b) (unbox b)))
```

Consider the following program:

Let v1 be the result of evaluating this program in a language where let and lambda have lexical scope, and let v2 be the result of evaluating this program in a lanuage where let and lambda have dynamic scope.

Which of the following is true?

- $\boxed{ v1 = 9, v2 = 9 }$
- v1 = 4, v2 = 9
- $\sqrt{v_1 = 4}, v_2 = 4$
- $\boxed{ v1 = 9, v2 = 4 }$

Consider the following expression written in the language used in the programming part of problem set 3:

Evaluating this expression should raise an unbound identifier error for the recursive occurrence of fact in the definition of fact. We saw how this happens by considering the behavior of this example with a substitution-based interpreter with our environment-based interpreter that had lexical scope.

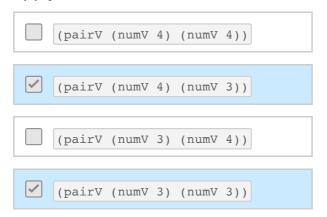
What happens if we change the semantics of the language to use dynamic scope instead and then execute this example?

- It returns 120.
- O It returns 5.
- O It returns 1.
- It returns an unbound identifier error.

Consider the following two lambda expressions written in the language of problem set 3:

call the first lambda expression [e1], and call the second [e2].

Let e be a closed expression in the same language, such that evaluating the expression (pair (e1 e) (e2 e)) does not return an error and returns the value v. Which of the following could v possibly be (select all that apply)?



#### Q6.1 1 Point

In the programming assignment, you are asked to implement an interpreter for a language with if statements. Here is an incorrect implementation of ifc statements for that language using a store-passing style like we saw in class:

where <code>env</code> and <code>sto</code> and <code>Result</code> are as in <code>lecture6c.rkt</code> from class.

On which of the following examples does the above implementation of <code>ifc</code> give incorrect behavior, in the sense that calling <code>eval</code> will yield a different result than calling <code>eval</code> in a correct implementation.

```
(if #t (+ 1 1) (+ 1 (lambda x 1))

(if #t (+ 1 1) (+ 2 2)

(if #f (+ 1 1) (+ 2 2)

(if #f (+ 1 1) (+ 1 (lambda x 1))
```

### Q6.2 1 Point

Here is another incorrect implementation of ifc.

Suppose the variable b is bound to a box initially containing the value 0. On which of the following examples does the above implementation of ifc give incorrect behavior?

```
(if #f (set-box! b 10) (unbox b))

(if #t (set-box! b 10) (unbox b))

(if (equal? (unbox b) 0) (unbox b) (+ 1 (unbox b)))

(if (begin (set-box! b 1) #t) (unbox b) 2)
```

# Pset 3 - Comprehension

Ungraded

2 Hours, 18 Minutes Late

Student

Kal Sastra

**Total Points** 

- / 10 pts

Question 1	
(no title)	2 pts
1.1 (no title)	1 pt
(no title)	ίρι
1.2 (no title)	1 pt
Question 2	
(no title)	1 pt
Question 3	
(no title)	1 pt
Question 4	
(no title)	2 pts
Question 5	
	2 4
(no title)	2 pts
	2 pts
Question 6	
Question 6 (no title)	2 pts
Question 6	
Question 6 (no title) 6.1 (no title)	2 pts 1 pt
Question 6 (no title)	2 pts