2a Give one test case that behaves differently under dynamic scoping versus static scoping. Explain the test case and how they behave differently in your write-up.

```
function parent(){
  function func1(){
    var x = 1;
    f2();
  }
  function func2(){
    console.log(x);
  }
  var x = 2;
  func1();
}
container();
```

- Dynamic scoping: `parent()` would print out 1 because `func2()` would get its scope from `func1()` in which `x` is redefined as 1.
- Static scoping: `parent()` would print out 2 because `func2()` would get its scope from its parent function in which `x` is defined as 3.

3d Explain whether the evaluation order is deterministic as specified by the judgment form $e \to e'$.

Yes, the evaluation order is determined by the judgment form $e_1 \to e_1{}'$. This is evident in a simple step such as $SearchBinary_1$, where $e_1 \to e_1{}'$ for a binary operation between e1 and e2. In this case, the evaluation order is left associative.

4 What is the evaluation order for $e_1 + e_2$? Explain. How do we change the rules obtain the opposite evaluation order?

The evaluation order is left associative, which means that within the inner most expression, the left side is evaluated first. In order to change this we would have to change the Search rules to step e_2 instead of stepping e_1; likewise, in the Do rules, we could check for values on the right side of the expression as opposed to the left.

5a Give an example illustrating the usefulness of short-circuit evaluation.

With an AND it evaluates the e_1 and checks it before evaluating e_2. This is useful because if the evaluation of e_1 doesn't pass the necessary conditions, then the program can save time by not attempting to evaluate e_2.