2a). One test case that behaves differently under dynamic scoping versus static scoping is:

```
const x = 10
const plus = function(x) {return function(y) {return x + y}};
jsy.print(plus(1)(2))
```

With static scoping, the result would be 3 because the previously declared const x is out of scope. However, in dynamic scoping, the result is 12 because it would use the value declared for x on the first line, which is 10 instead of 1.

3d). The evaluation order is deterministic as specified in the judgement form e->e' because it always evaluates from left to right. Step must be called on the left first and must be evaluated before the right side can be.

4). In the expression e1 + e2, e1 would be evaluated first, then e2, and then the + operator. To change the order in which it is evaluated, we could reverse the eval function so that e2 is evaluated first, then e1, and the + operator is still the last to be evaluated.

5a). The && operator is useful in short circuit evaluation. For example, a && b will first check if a is valid, if not, it will short circuit and results in better performance since it then knows that there is no need to waste time checking the right side of the expression.

5b). e1 && e2 does short circuit because it first checks if e1 is valid, if it isn't, it will short circuit because an AND statement cannot be true when one element is already evaluated as false.