Scope Example
What does the function `test` print if the language uses static scoping? What does it print with dynamic scoping? (otherwise assume C++ syntax and semantics, e.g. call by value).

```cpp
int n = 1;    // global

print_plus_n(int x) {
    cout << x + n;
}
increment_n() {
    n = n + 2;
    print_plus_n(n);
}

test() {
    int n;
    n = 200;
    print_plus_n(7);

    n = 50;
    increment_n();
    cout << n;
}
```

With Static Scoping:

With Dynamic Scoping:
Functional programming Questions:
1. a) What is a first class citizen in a programming language?
   
b) Give an example of a first class citizen in scheme.

2. What is programming in a “purely functional style”?

3. What is the result of the following in Scheme:
   
   \( \text{(map (lambda (x) (+ x 50)) '(1 2 3 4))} \)

4. Assuming that the following definitions are executed in this order:
   
   \( \text{(define x `(3 28 400))} \)
   \( \text{(define y (cons (cdr x) `(6 15 77)))} \)
   
   What is the result of typing the following into the Scheme interpreter:
   
   \( y \) => ???
   \( (\text{cons} \ 'x \ (\text{cdr} \ (\text{cdr} \ x))) \) => ???

5. Write a recursive Scheme function, \textit{merge\_sorted} that takes two sorted lists as parameters and returns a single list that contains all of the elements of both lists in sorted order. You can assume that the two lists: both contain only integer values > 0, and are sorted from smallest to largest. The two lists may not be of the same length.

   Example:
   
   \( \text{(merge\_sorted `(4 8 26) `(6 200))} \) => (4 6 8 26 200)