

Sample Midterm

Question #1. Convert the following function from an implicit lambda into an explicit one:

```
(define (do-nothing)
  '(I am a function with no arguments))
```

Question #2. Suppose that we've started Scheme up, and we begin to slowly type the following definition at the prompt:

```
> (define (problem-two arg)
    (se muhahaha (/ 0 0)))
```

What will happen when we press Enter, and why? (Consider this a trick question.)

Question #3. Write a function called `its-all-good?` whose two arguments are a predicate and a sentence. `its-all-good?` should return `#t` only if each element applied with the predicate returns `#t`. For example:

```
> (its-all-good? number? '(3 1 4 1 5 9 2 6))
#t
```

```
> (its-all-good? even? '(2 4 6 8 10))
#t
```

```
> (its-all-good? odd? '(2 3 5 7 9))
#f
```

Question #4. In your lab assignments, you might have noticed that `vowel?` needed to be defined; it's not part of standard Scheme. To get some practice playing with conditionals: write three versions of the `vowel?` predicate. For each version, follow one of the following constraints:

1. Use only `if` and `cond`.
2. Use `member?`.
3. Use only `and` and `or`.

Question #5. I've heard that in ancient texts, scribes removed vowels from their sentences to save paper. Write a `de-voweler` function that takes in a sentence, and returns a sentence with all the vowels removed. For example:

```
> (de-voweler '(one ring to bring them all))
(n rng t brng thm ll)
```