Due February 10, 5:00 p.m.

Problem 1

Write a procedure called safe-fn that takes two procedures, fn and test as arguments and returns a procedure. When the procedure returned by safe-fn is applied to an argument, arg, it returns the symbol error whenever the value of (test arg) is #f. Otherwise, it returns the value of (fn arg). E.g.,

```
> (define safe-sqrt (safe-fn sqrt (lambda (n) (>= n 0))))
> (safe-sqrt 4)
2
> (safe-sqrt -4)
error
```

Problem 2

Write a procedure called get-example that takes a predicate p? as an argument and returns a procedure. When the procedure returned by get-example is applied to a list, lst, it returns a member of lst that satisfies the predicate p? if one exists. Otherwise, it returns the symbol none.

```
> (define get-even (get-example even?))
> (get-even '(2 4 6))
2
> ((get-example even?) '(2 4 6))
2
> ((get-example odd?) '(2 4 6))
none
```

Submitting

Don't forget to submit your work using the submit101 command!

```
submit101 g-asmt06 asmt06
```

(If the name of your directory is different from asmt06, change asmt06 to whatever the name of your directory is.)