DO NOT OPEN EXAM BOOKLET UNTIL INSTRUCTED TO DO SO.
Question 1 What do the following programs display? *Note that none of these programs produce errors.*

Part a (5 points) *Difficulty: ⋆*

```javascript
function F1(x) {
    if (x === 4) {
        return;
    }
    console.log(x);
    F1(x + 1);
}
F1(2);
```

Part b (5 points) *Difficulty: ⋆*

```javascript
function F2() {
    console.log('A');
    function inner() {
        console.log('B');
    }
    console.log('C');
    inner();
}
F2();
```

Part c (5 points) *Difficulty: ⋆*

```javascript
function F3(x) {
    return {
        get: function() { return x; },
        set: function(y) { x = y; }
    };
}
let A = F3(100);
let B = F3(10);
B.set(200);
console.log(A.get);
```

```javascript
let A = F3(100);
let B = F3(10);
B.set(200);
console.log(A.get());
```
Part d (5 points) Difficulty: **

```javascript
function F4(x) {
    return {
        x: x,
        get: function() { return x; },
        set: function(y) { x = y; }
    };
}
let o = F4(5);
o.set(10);
console.log(o.x);
console.log(o.get());
```

Part e (5 points) Difficulty: **

```javascript
function F5(arr) {
    return {
        get: function(index) { return arr[index]; },
        set: function(newArr) { arr = newArr; }
    };
}
let a = [10, 20, 30];
let o = F5(a);
a[0] = 100;
console.log(o.get(0));
o.set([5000]);
console.log(a[0]);
```
Question 2 Consider the following function.

```javascript
function F7(f, g, h) {
    f(function(x) {
        if (x > 0) {
            g(function() { console.log('A'); });
        } else {
            h(function() { console.log('B'); });
        }
    });
}
```

Part a (5 points) Difficulty: **
Give values of f, g, and h such that F7(f, g, h) displays the following output. **Your answer must not use console.log or raise errors.**

A

B

Part b (5 points) Difficulty: **
Give values of f, g, and h such that F7(f, g, h) displays the following output. **Your answer must not use console.log or raise errors.**

B
**Question 3** (5 points) *Difficulty: *  
Consider the following function.

```javascript
function F9(f) {
  f(function(x) { console.log(x); });
}
```

Give a value for `f` such that `F9(f)` displays the output:

220

**Your answer must not use `console.log` or raise errors.**

---

**Question 4** (5 points) *Difficulty: **  
Consider the following function.

```javascript
function F8(f, r, n) {
  if (n === 0) {
    return r;
  } else {
    return F8(f, f(n, r), n - 1);
  }
}
```

Give a value for `f`, such that `F8(f, 0, n)` returns the value $0 + 1 + 2 + \cdots + n$. (The function must not display anything, i.e., do not use `console.log`.)
Question 5  This question concerns streams, using the implementation in the exam notes. In your answers, you may assume that the stream is unbounded (i.e., you can ignore $\text{sempty}$).

Part a  (5 points)  Difficulty: ⋆

Describe the following stream, or just list the first five values that it produces.

```
nats.filter(n => n % 2 === 0).filter(n => n % 5 === 0)
```

Note that $\text{nats}$ is the stream of natural numbers (definition is in the exam notes.)

Part b  (5 points)  Difficulty: ⋆

Write a stream of the multiples of three, i.e. 0, 3, 6, 9, ⋯. Your code can call the functions and definitions in the exam notes. You do not have to to reproduce them in your solution.
Part c (5 points) Difficulty: ★★★
Write a function called `alt` with the following type:

```haskell
alt<A>(stream: Stream<A>): Stream<A>
```

The result of `alt` should be a new stream that has every other element of the input stream, including the first element.

This is an example of `alt`’s behavior:

```haskell
let naturals = snode(0, memo0(() => naturals.map(x => x + 1)));
alt(naturals) // produces 0, 2, 4, 6, 8, ...
```
Part d (10 points) Difficulty: ★★
Write a function called zipWith with the following type:

```typescript
zipWith<A, B, C>(f: (x: A, y: B) => C,
                   s1: Stream<A>,
                   s2: Stream<B>): Stream<C>
```

The zipWith function is a variant of map, that applies a single function to pairs of corresponding elements from two streams. Specifically, if \( s1 \) produces the values \( a_0, a_1, a_2, \ldots \) and \( s2 \) produces the values \( b_0, b_1, b_2, \ldots \), then \( \text{zipWith}(f, s1, s2) \) will produce the values \( f(a_0, b_0), f(a_1, b_1), f(a_2, b_2), \ldots \).

This is an example of zipWith’s behavior:

```javascript
let naturals = snode(0, memo0(() => naturals.map(x => x + 1)));
zipWith((x, y) => x + x, naturals, naturals) // produces 0, 2, 4, 6, 8 ...
```

Part e (5 points) Difficulty: ★★★
Describe the following stream, or just list the first five values that it produces.

```javascript
let mystery = snode(1, memo0(() =>
                         snode(1, memo0() =>
                             zipWith((x, y) => x + y, mystery, mystery.tail())))))
```

Page 8 of 9 Version A
Question 6  This question concerns error handling with results, using the implementation in the exam notes.

Part a (5 points)  Difficulty: ⋄
What does the following program display?

```javascript
function E0(n) {
  if (n === 0) {
    return new Success(1); }
  else {
    return new Failure(0); }
}
E0(1).then(function(v) { console.log(v); });
console.log("Done");
```

Part b (5 points)  Difficulty: ⋄
Consider the following function:

```javascript
function E2(r) {
  r.then(function(n) {
    console.log("A");
  });
}
```
Give a value for r such that E2(r) displays the following output:

A

Your answer must not use console.log or raise errors.

Part c (5 points)  Difficulty: ⋆⋆
What does the following program display?

```javascript
function E1(n) {
  if (n === 0) {
    return new Success(1);
  } else {
    return E1(n - 1).then(r => {
      console.log(n);
      return new Success(r * n)
    });
  }
}
E1(5);
```

End of exam.