

Question 1: What is the output of the following function:

```
function foo(f, n) {
  if (n === 1) {
    return function(x) { return x * 10; }
  } else if(n > 100) {
    return function(x) { return foo(f, n / 10)(f(x)+1); }
  } else {
    function bar(x) {
      return 1 + foo(f, n / 10)(f(x));
    }
    return bar;
  }
}

function bar(x) {
  return x;
}

console.log(foo(bar, 1000)(5));
```

Question 2: Consider the following function:

```
function foo(x, y) {
  if (y > 0) {
    console.log('A');
  }
  function bar() {
    if (x + y < 10) {
      console.log('B');
    }
    if (y < 5) {
      console.log('C');
    }
  }
  if (x / 10 <= 1) {
    bar();
  }
}
```

Part a: Give three values for x and y that will make foo(x, y) display:

A
C

x =	x =	x =
y =	y =	y =

Question 3: Consider the following function:

```
function foo(f) {  
  f(function() {  
    console.log("A");  
  },  
  function(x, y) {  
    if (x !== y || x + y > 0) {  
      return function() { console.log("B"); } ;  
    } else {  
      if (x % 2 === 0) {  
        console.log("C");  
      }  
      return function(z) { console.log(z); };  
    }  
  });  
}
```

Part a: Give a value for `f`, such that `foo(f)` will display the following output. **Your answer must not use `console.log`.**

A
C
D

f =

Question 4: The following function, `reduceF`, combines `reduce` with a filtering function:

```
// reduceF(a: T[], pred: (x: T) => boolean, f: (x: S, y: T) => S, init: S): S
function reduceF(a, pred, f, init) {
  let result = init;
  for (let i = 0; i < a.length; ++i) {
    if (pred(a[i])) {
      result = f(result, a[i]);
    }
  }
  return result;
}
```

Part a. Write a function that consumes an array `a`, and returns the sum of squares of all the even numbers in the array.

```
sumSquareEven([2, 5, 6, 8, 9]) // produces 104 (=2*2+6*6+8*8)
sumSquareEven([1, 3, 7]) // produces 0
```

```
// sumSquareEven(a: number[]): number
function sumSquareEven(a) {
  let pred = function(x) {

  };

  let f = function(x, y) {

  };

  let init =          ;

  return reduceF(a, pred, f, init);
}
```