

## 220: Discussion 2

Round 2! This set of problems require you to demonstrate understanding of important concepts in JavaScript like types and closures. For these exercises, recall that `typeof x` is an operator that returns a string based on the type of `x`. This will usually be one of the following values: “number”, “string”, “boolean”, “function”, “object”, or “undefined” - though the last two we will not worry about.

### Problem 1

```
function red(x) {
  function blue(y) {
    return x + '!';
  }
  return blue;
}
let purple = 'green';
let orange = red(purple);
let gray = orange('black');
console.log(typeof purple.length);
console.log(typeof purple.length === 3);
console.log(typeof orange);
console.log(typeof gray);
console.log(typeof gray.length);
```

What does this code output?

### Problem 2

```
function apply(f, a) {
  return f(a);
}

function compose(f, g) {
  return function(a) {
    return f(g(a));
  }
}

function p(x) { return x + 1 };
function q(x) { return x * 2 };
function r(x) { return x * x };
console.log(compose(compose(p, q), compose(r, q))(3));
console.log(compose(compose(r, q), compose(p, q))(2));
```

What does this code output?

### Problem 3

```
function hamp() {
  function berk(x) {
    console.log(x);
    return hamp();
  }
  return berk;
}
hamp()('second')('last')('first');
```

What does this code output?

## Problem 4

A *confirming function* for a sequence of digits, called a *code*, takes a single digit as its only argument. If the digit does not match the first (left-most) digit of the code to be confirmed, it returns `false`. If the digit does match, then the confirming function returns `true` if the code has only one digit, or another confirming function for the rest of the code if there are more digits to confirm.

Complete the implementation of `confirmer` so that when `confirmer` takes a positive integer `code`, it returns a confirming function for the digits of that code. Hint: `confirmer` is a higher order function! What about the others?

---

```
// Return a confirming function for the argument CODE
// >>> confirmer(204)(2)(0)(4) // The digits of 204 are 2, then 0, then 4.
// true
// >>> confirmer(204)(2)(0)(0) // The third digit of 204 is not 0.
// false
// >>> confirmer(204)(2)(1) // The second digit of 204 is not 1.
// false
// >>> confirmer(204)(20) // The first digit of 204 is not 20.
// false
function confirmer(code) {

  function confirm1(d, t) {
    function result(digit) {
      if (d === digit) {
        return t;
      } else {
        return false;
      }
    }
    return result;
  }

  // Return a confirming function that returns REST when given the digits of PREFIX.
  // For example, if c = extend(12, confirmer(34)), then c(1)(2) returns confirmer(34),
  // so that c is a confirming function for 1234.
  function extend(prefix, rest) {
    let left = Math.floor(prefix / 10);
    let last = prefix % 10;
    if (prefix < 10) {
      // [TODO] Insert an expression here!

      return -----;
    } else {
      // [TODO] Insert an expression here!

      return -----;
    }
  }

  return extend(code, true);
}
```

---