Jerry Cain
CS 106AX

Handout #4
October 2, 2023

Section 1 Handout

Portions of this handout by Eric Roberts, Patrick Young, and Ryan Eberhardt.

This week in section, your first priority is to meet your section leader and discover what sections in CS106AX are all about. Your section leader and fellow students will spend the first part of this week's session on introductions and telling you the things you need to know, such as where to get help and how to sign up for interactive grading. Post introductions, the section will move on to cover some of the important material from class in a setting that is small enough for you to ask questions and find out what you need to know. You may not finish all the problems in this handout; the remaining problems can be used as exam practice problems.

Problem 1: Narcissistic Numbers

Problem focus: writing simple, console-only JavaScript functions.

A n-digit number is considered narcissistic if and only if it's equal to the sum of each of its digits raised to the n^{th} power. For example, 153, 370, 371, and 407 are all three-digit narcissistic numbers, because all of the following are true:

$$1^3 + 5^3 + 3^3 \text{ equals } 153$$
$$3^3 + 7^3 + 0^3 \text{ equals } 370$$
$$3^3 + 7^3 + 1^3 \text{ equals } 371$$
$$4^3 + 0^3 + 7^3 \text{ equals } 407$$

To be clear, numbers are very rarely narcissistic, since it's unusual for an n-digit number to incidentally equal to the sum of its digits raised to the n^{th} power. In fact, the four numbers listed above are the only three-digit narcissistic numbers.

There are a several more, though. The numbers 1 through 9, inclusive, are all (trivially) narcissistic. More interesting examples include 9474, 54748, and 88593477. They're each narcissistic because each of the following statements is true:

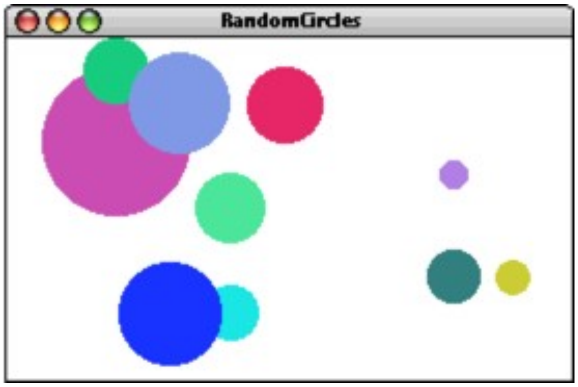
$$9^4 + 4^4 + 7^4 + 4^4 \text{ equals } 9474$$
$$5^5 + 4^5 + 7^5 + 4^5 + 8^5 \text{ equals } 54748$$
$$8^8 + 8^8 + 5^8 + 9^8 + 3^8 + 4^8 + 7^8 + 7^8 \text{ equals } 88593477$$

Write a program called `Narcissistic` that accepts a positive integer and returns `true` if the provided number is narcissistic according to the above description, and `false` otherwise. You're encouraged to decompose your solution and write a helper function that counts the number of digits in a number.

Problem 2: Random circles

Problem focus: writing small interactive graphics programs

Write a graphics program that draws a set of ten circles, each with a different size, position, and color. Each circle should have a randomly chosen color, a randomly chosen radius between 5 and 50 pixels, and a randomly chosen position on the canvas, subject to the condition that the entire circle must fit inside the canvas without extending past the edge. The following sample run shows one possible outcome:



Assuming the existence of some **`randomInteger(min, max)`** and **`randomColor()`** functions which work as you'd expect.

On some runs of this program, you might not *see* ten circles. Why?

Once you have done this, try extending the program so that a user can interactively create more circles. This will behave similar to the line-drawing program demonstrated in lecture. On `"mousedown"`, create a circle with a 0px diameter and arbitrary fill color centered at the mouse. On `"drag"`, keep the circle centered at the original click point, but enlarge it such that the edge of the circle follows the mouse.

Problem 3: Tracing function execution

Problem focus: understanding JavaScript function and scope mechanics.

For the program below, trace through its execution by hand to show what output is produced when it runs.

```
/*
 * File: CalculateBill.js
 * ------------------·
 * This program is designed to test your understanding of parameters,
 * return values, and closures. As such, some lines are intentionally
 * tricky, but we believe in you!
 */
const TAX_RATE = 0.10;
const TIP_RATE = 0.50;
const STANFORD_DISCOUNT = 44;

function main() {       /* program execution starts here */
   let numSalads = 4;
   let numPizzas = 6;
   let finalPrice = calculateBill(numSalads, numPizzas);
   console.log("Your final price is: $" + finalPrice + ".");
}

function calculateBill(numPizzas, numSalads) {
   let total = 0;
   function addSaladCosts(n) {
      total += n * 5;
   }
   function addPizzaCosts(n) {
      if (n < 4) {
         total += 20;
      } else {
         total += 70;
      }
   }
   addSaladCosts(numSalads);
   addPizzaCosts(numPizzas);
   addTax(total);
   console.log("Your total before tip is: $" + total + ".");
   total = addTip(total);
   return total – STANFORD_DISCOUNT;
}

function addTax(total) {
   total *= 1 + TAX_RATE;
}

function addTip(total) {
   total *= 1 + TIP_RATE;
   return total;
}
```