

ISE 108 Practice Midterm 2

This exam consists of four questions, and is worth up to 100 points. Partial credit may be awarded at the instructor's discretion.

You may not use cell phones, pagers, or any other electronic devices during the exam, except for the computers that are installed in the exam room. Please turn all prohibited devices off for the duration of the exam. You may not communicate with other students during the exam.

At the start of the exam, your instructor will tell you where to find the starting code for the exam problems, as well as how to submit your work when you are finished.

1. Text Manipulation (25 points)

The “text_splitter” sketch prompts the user to enter an input string. Complete this sketch so that it breaks the input String in half and displays the two halves of the String, one per line. (**Hint:** use the String methods `length()` and `substring()`)

For example, given the input “abcdef”, your program should produce output similar to the following:



2. Generating Class Numbers (25 points)

The Computer Science department at a particular university assigns class numbers to courses according to the following pattern: the letters "CSE", followed by a random 3-digit integer between 100 and 499 (inclusive), followed by the section number: the letter "L", the digit "0", and a random integer between 1 and 4 (inclusive). For example, CSE203L01 and CSE462L03 are course numbers that follow this pattern.

The `CourseNumber` sketch implements a program that generates random course numbers according to this pattern. Complete the `getCourseNumber()` function, which returns a randomly-generated course number according to the pattern given above. This function does not take any arguments, but it returns the new course number as a `String`.

3. Arrays (25 points)

The "Odds" program generates an randomly-sized array containing a random sequence of 0 and 1 values. Complete the `isOdd()` function, which takes an array of integers containing only 1s and 0s, and returns `true` if all the 1s in the array add up to an odd value (and `false` if the sum is even). For example, given the array `{0, 1, 1, 0, 1}`, `isOdd()` would return `true`, because the 1s add up to 3, which is odd. Do not modify the `setup()` function while solving this problem!

4. Image Manipulation (25 points)

The "manipulation" sketch loads and displays an image of a cat. When the user clicks the mouse button, the `mangle()` function changes the image, pixel by pixel, as follows:

- If the pixel's red and green components add up to a value greater than 150, replace the pixel with a new, all-blue pixel (`color(0, 0, 255)`).
- Otherwise, if the pixel's blue and green components add up to a value greater than 150, replace the pixel with a new, all-red pixel (`color(255, 0, 0)`).
- If neither condition is met, don't change the pixel at all.

Complete the `mangle()` function in the sketch so that it meets the criteria above. **Hint:** use Processing's `red()`, `green()`, and `blue()` functions to measure the color levels of a specific pixel!