Lab 5: Memory Mapping

Task 1.1: Endian Swap (40pts)

By default, RISC-V processors store data in little-Endian order, from the least significant byte to the most significant byte. Implement an algorithm that reads a number from memory and stores it back to the same memory location in big-endian order. You are provided with `endian_swap_starter.s` to help you get started.

Submit your assembly code and before-and-after screenshots of the data memory section for each case provided in the starter code.

Task 1.2: Stretch and Shrink Array (60pts)

Create a procedure that takes an array from RAM, first shrinks it in place to the smallest possible data type, then stores it as a double-word array without overwriting existing data. You are provided with `stretchy_array_starter.s` to help you get started.

Submit your assembly code and before-and-after screenshots of the data memory section for each case provided in the starter code.