

Assignment #1: Restaurant Finder Rubric

PART 1 (50%)	Mark
<u>Cursor Movement</u> <ul style="list-style-type: none">• In Mode 0, moving the joystick moves the cursor around the screen. Moving to the edge of the screen shifts the map view, displaying a new section of the map and recentering the cursor (10 marks).• In Mode 1, moving the joystick up and down moves the highlight block in the appropriate direction, changing the highlighted restaurant (5 marks).• In either mode, pressing the joystick button switches to the other mode (5 marks).	/20
<u>Displaying Restaurant Names</u> <ul style="list-style-type: none">• When switching to Mode 1, 30 restaurant names are displayed on the LCD screen using the standard text multiplier of 1. The highlight block starts on the top restaurant (5 marks).• Pressing the joystick button on a highlighted restaurant consistently and correctly centers the cursor on that restaurant's location when transitioning back to Mode 0 (5 marks).	/10
<u>Displaying Restaurants as Dots</u> <ul style="list-style-type: none">• Tapping on the map display visualizes all nearby restaurants (those within the map screen) as small dots (10 marks).• Tapping on the black space of the screen (off the map) does not visualize restaurants (5 marks).• Note that we will not test button presses on the boundary between the map and the black space.	/15
<u>Insertion Sort Implementation</u> <ul style="list-style-type: none">• Insertion sort properly maintains the sorting invariant and runs in $O(n^2)$ time in the worst case (15 marks).• In Mode 1, restaurant names are displayed in order from nearest to farthest by calling the insertion sort implementation (10 marks).• In Mode 0, the restaurant dots displayed after tapping the screen correspond correctly to the nearest restaurants (10 marks).	/35
<u>Code Style and README</u> <ul style="list-style-type: none">• Code documentation (comments, complete file headers), code readability, and general style (5 marks).• Code is modular and well-designed. No unnecessary repeated code, reasonable structure. Functions are used where appropriate (5 marks).• Submission has correct function/file names (README has no extension, submission file is either tar.gz or .zip, Makefile is included, build-mega directory is not included) (5 marks).• README has the required components: complete header, running instructions are thorough and correct, assumptions are stated where needed (5 marks).	/20
PART 1 GRADE	/100

PART 2 (50%)	Mark
<u>Quicksort Implementation</u> <ul style="list-style-type: none"> The quicksort implementation properly maintains the sorting invariant and is implemented correctly (15 marks). 	/15
<u>Scrollable Restaurant List</u> <ul style="list-style-type: none"> In Mode 1, scrolling below the bottom restaurant in the list displays the next 30 restaurants (5 marks). In Mode 1, scrolling above the top restaurant in the list displays the next 30 restaurants (5 marks). Marks may be deducted if: Attempting to scroll off either end of the list does not either wrap around or do nothing. 	/10
<u>Button Column</u> <ul style="list-style-type: none"> There are two buttons, one for selecting the rating and one for selecting the sort method. Tapping each button cycles through the modes intuitively (10 marks). Marks may be deducted if: button design or cycling of modes is unintuitive, buttons are drawn over the map / out of the designated area. 	/10
<u>Rating Selection</u> <ul style="list-style-type: none"> Restaurants are only included if they have a rating above or equal to the rating threshold displayed on the rating selector button (10 marks). Restaurants with ratings below the threshold are filtered out before sorting (ie, they are not unnecessarily included in the sort) (5 marks). The filtering matches the threshold displayed on the rating selector button (5 marks). Restaurants display as dots only if they are above the rating threshold (5 marks). 	/25
<u>Sort Selection</u> <ul style="list-style-type: none"> In QSORT mode, quicksort is called to sort the restaurants and in ISORT mode, insertion sort is called to sort the restaurants (5 marks). In BOTH mode, both insertion sort and quicksort are called to sort the restaurants. Care is taken to ensure that both sorts start from the same restaurant array (ie, by rereading from the SD card in-between sorts) (10 marks). In all three modes, the time taken in ms for each sort is displayed to the Serial monitor in the appropriate format (5 marks). 	/20
<u>Code Style and README</u> <ul style="list-style-type: none"> Code documentation (comments, complete file headers), code readability, and general style (5 marks). Code is modular and well-designed. No unnecessary repeated code, reasonable structure. Functions are used where appropriate (5 marks). Submission has correct function/file names (README has no extension, submission file is either tar.gz or .zip, Makefile is included, build-mega directory is not included) (5 marks). README has the required components: complete header, running instructions are thorough and correct, assumptions are stated where needed (5 marks). 	/20
PART 2 GRADE	/100

TOTAL ASSIGNMENT GRADE = (PART 1 GRADE)*0.5 + (PART 2 GRADE)*0.5	/100
--	------