

Testing Schedule

Input	Expected/Boundary/Invalid	Expected output	Test result
Shopping list name: English supermarket Item name: milk bottles Amount: 3 Cost: 4 Item name: egg cartons Amount: 2 Cost: 10 Item name: end	Expected	English supermarket 3 milk bottles: \$12.00 2 egg cartons: \$20.00 Total cost is \$32.00 Number of items is 5 Number of unique items is 2	English supermarket 3 milk bottles: \$12.00 2 egg cartons: \$20.00 Total cost is \$32.00 Number of items is 5 Number of unique items is 2
Shopping list name: Japanese supermarket Item name: ミルク Amount: 33 Amount too large. Try again. Amount: 3 Cost: 4 Item name: 卵 Amount: 2 Cost: 10 Item name: Cease	Difficult edge case	Japanese supermarket 3 ミルク: \$12.00 2 卵: \$20.00 Total cost is \$32.00 Number of items is 5 Number of unique items is 2	Japanese supermarket 3 ミルク: \$12.00 2 卵: \$20.00 Total cost is \$32.00 Number of items is 5 Number of unique items is 2
Shopping list name: Bottle of water Item name: water bottle Amount: 1 Cost: \$2	Expected	Next line should be: "Item name: "	Next line was: "Use a numerical value. Try again."
Shopping list name: Bottle of water Item name: water bottle Amount: 1 Cost: \$2 Item name: done	Expected	Bottle of water 1 water bottle: \$2.00 Total cost is \$2.00 Number of items is 1 Number of unique items is 1	Bottle of water 1 water bottle: \$2.00 Total cost is \$2.00 Number of items is 1 Number of unique items is 1
Shopping list name: sNACKs Item name: expensive snack Amount: 1 Cost: Use a numerical value. Try again. Cost: 10 dollars Item name: cheap snacks Amount: 3 Cost: 10 cents Item name: eND	Expected	sNACKs 1 expensive snack: \$10.00 3 cheap snacks: \$0.30 Total cost is \$10.30 Number of items is 4 Number of unique items is 2	sNACKs 1 expensive snack: \$10.00 3 cheap snacks: \$30.00 Total cost is \$40.00 Number of items is 4 Number of unique items is 2
Shopping list name: sNACKs Item name: expensive snack Amount: 1 Cost: 10 dollars Item name: cheap snacks Amount: 3 Cost: 10 cents Item name: end	Expected	sNACKs 1 expensive snack: \$10.00 3 cheap snacks: \$0.30 Total cost is \$10.30 Number of items is 4 Number of unique items is 2	sNACKs 1 expensive snack: \$10.00 3 cheap snacks: \$0.30 Total cost is \$10.30 Number of items is 4 Number of unique items is 2
Shopping list name: Sorting test Item name: Cat Amount: 1 Cost: 100 Item name: Banana Amount: 2 Cost: 1 Item name: apple Amount: 2 Cost: 1 Item name: Zebra Amount: 1 Cost: 200 Item name: Fish Amount: 1 Cost: 20 Item name: end	Expected	Sorting test 2 apple: \$2.00 2 Banana: \$2.00 1 Cat: \$100.00 1 Fish: \$20.00 1 Zebra: \$200.00 Total cost is \$324.00 Number of items is 7 Number of unique items is 5	Sorting test 2 apple: \$2.00 2 Banana: \$2.00 1 Cat: \$100.00 1 Fish: \$20.00 1 Zebra: \$200.00 Total cost is \$324.00 Number of items is 7 Number of unique items is 5
Shopping list name: Cost boundary testing Item name: One Amount: 1 Cost: -0.01	Boundary	Cost boundary testing 5 Five: \$5.00 4 Four: \$40.00	Cost boundary testing 5 Five: \$5.00 4 Four: \$40.00

(given filename unusable). Shopping list name: / \\ \ Invalid name. Name must also be acceptable file name (filename cannot contain "/"). Shopping list name: \0 Item name: e Amount: e Use an integer value. Try again. Amount: 1 Cost: 1 Item name: end			
Removed shopping lists subfolder and ran program.	Invalid	Program should detect issue and create folder automatically	When name was entered program detected that it could not write files (due to missing folder) and declined any name given. Somewhat detected issue and didn't crash completely but didn't help either.
{see above} after modifications	Invalid	Program should detect issue and create folder automatically	Program created folder then continued with ordinary function
Shopping list name: Item name length boundary testing Item name: Name too short. Try again. Item name: 1 Amount: 1 Cost: 1 Item name: 10 charact Amount: 1 Cost: 1 Item name: 50 characters ee Amount: 1 Cost: 1 Item name: 99 characters ee ee Amount: 1 Cost: 1 Item name: 100 characters ee ee Amount: 1 Cost: 1 Item name: 101 characters ee ee Name too long. Try again. Item name: testing successful Amount: 1 Cost: 1 Item name: end	Boundary	Program should allow 1 to 100 length character names (inclusive) and request user reentry for anything else.	As expected
A.D. tested, needed clarification "amount" and "cost", modified to "quantity" and "individual cost"	Human		
A.D. tested again, found inconvenient that there was no way to modify entered data if changed mind/mistake. Also found no weight-based option e.g. produce inconvenient. Liked how capitalisation was kept consistent with list	Human		
Shopping list name: cancel Item name: cancel Quantity: cancel Entry cancelled, enter new/re-enter item. Item name: cancel Quantity: 3 Individual cost: cancel Entry cancelled, enter new/re-enter item. Item name: 2 Quantity: 2 Individual cost: 2 Confirm: 2 2: \$4.00? cancel	Error-prevention/rec overy	cancel 1 confirm: \$1.00 Total cost is \$1.00 Number of items is 1 Number of unique items is 1	As expected; program now requests user confirmation and allows user to redo at any stage during entry.

<p>Entry cancelled, enter new/re-enter item.</p> <p>Item name: confirm Quantity: 1 Individual cost: 1 Confirm: 1 confirm: \$1.00? confirm Item saved.</p> <p>Item name: done</p>			
<p>Shopping list name: eggs Item name: eggs Quantity: 2 Individual cost: 2 Confirm: 2 eggs: \$4.00? 1 Item saved.</p> <p>Item name: milk Quantity: 3 Individual cost: 3 Confirm: 3 milk: \$9.00? 1 Item saved.</p> <p>Item name: eggs "eggs" is already on list; overwrite? 0 Overwrite canceled.</p> <p>Item name: milk "milk" is already on list; overwrite? 1 Continue new entry:</p> <p>Quantity: 4 Individual cost: 4 Confirm: 4 milk: \$16.00? 1 Item saved.</p> <p>Item name: end</p>	<p>Expected/Error recovery</p>	<p>eggs</p> <p>2 eggs: \$4.00 4 milk: \$16.00</p> <p>Total cost is \$20.00 Number of items is 6 Number of unique items is 2</p>	<p>As expected; overwrite came with proper warning, allowed user to cancel and successfully replaced identical item.</p>