***Algorithm: Budget Check Feature******Algorithm Steps:***

*Step 1: Start the Budget Check Process*

* *Triggered when the user adds or updates an item in the grocery list.*

*Step 2: Calculate the Total Cost of the List*

* *Initialize a variable total\_cost = 0.*
* *For each item in the grocery list:  
  Multiply the quantity by the price per unit to get the item’s total cost.  
  Add the item’s cost to total\_cost.*

*Step 3: Compare Total Cost with Budget*

* *If total\_cost > budget\_limit:  
  Proceed to Step 4 (Notify User).*
* *Else:  
  Skip to Step 7 (Save List).*

*Step 4: Notify the User*

* *Display a warning message: "Budget exceeded. Please adjust your list."*

*Step 5: Provide Editing Options*

* *Allow the user to:  
  Edit item quantity  
  Remove items*
* *If the user makes changes, proceed to Step 6.*

*Step 6: Recalculate the Total Cost*

* *Recompute the total cost using the updated list.*
* *Return to Step 3 (Check if the new total is within the budget).*

*Step 7: Save the List*

* *If the total cost is within the budget, save the grocery list.*

*Step 8: End the Budget Check Process*

* *Display a success message: "List saved successfully."*

***Pseudocode for Budget Check Feature***

***Pseudocode:***

*FUNCTION budget\_check(grocery\_list, budget\_limit):  
 # Initialize total cost to 0  
 total\_cost = 0  
  
# Step 2: Calculate the total cost  
 FOR each item IN grocery\_list:  
 item\_cost = item['quantity'] \* item['price\_per\_unit']  
 total\_cost += item\_cost # Add the item cost to the total  
  
# Step 3: Check if total cost exceeds the budget  
 IF total\_cost > budget\_limit:  
 PRINT "Budget exceeded! Please adjust your list."  
  
# Step 5: Provide editing options until budget is within limit  
 WHILE total\_cost > budget\_limit:  
 PRINT "Options: Edit quantity / Remove items"  
 user\_choice = GET\_USER\_INPUT()  
  
 IF user\_choice == 'edit':  
 CALL edit\_item(grocery\_list)  
 ELSE IF user\_choice == 'remove':  
 CALL remove\_item(grocery\_list)*

*# Step 6: Recalculate the total cost  
 total\_cost = recalculate\_cost(grocery\_list)  
  
# Step 7: Save the list if within budget  
 PRINT "List saved successfully."  
 RETURN True # List saved  
  
# Helper function to recalculate total cost  
FUNCTION recalculate\_cost(grocery\_list):  
 new\_total = 0  
 FOR each item IN grocery\_list:  
 new\_total += item['quantity'] \* item['price\_per\_unit']  
 RETURN new\_total  
  
# Helper function to edit an item  
FUNCTION edit\_item(grocery\_list):  
 item\_name = GET\_USER\_INPUT("Enter item name to edit: ")  
 new\_quantity = GET\_USER\_INPUT("Enter new quantity: ")  
 FOR each item IN grocery\_list:  
 IF item['name'] == item\_name:  
 item['quantity'] = new\_quantity  
  
# Helper function to remove an item  
FUNCTION remove\_item(grocery\_list):  
 item\_name = GET\_USER\_INPUT("Enter item name to remove: ")  
 FOR each item IN grocery\_list:  
 IF item['name'] == item\_name:  
 REMOVE item FROM grocery\_list*