add\_product

The function add\_product is responsible for adding a new product to the inventory.

The function takes two parameters:

* products: an array of product\_t structs representing the current inventory.
* num\_products: a pointer to an integer representing the current number of products in the inventory.

The function first checks if the inventory is full. If it is, the function prints an error message and returns.

If the inventory is not full, the function prompts the user to enter the details of the new product, including its name, description, quantity, and price.

The function then creates a new product\_t struct with the input values and adds it to the products array at the next available index. The function increments the num\_products pointer to reflect the new number of products in the inventory.

Finally, the function prints a message indicating that the product has been successfully added to the inventory.

modify\_product

This function first checks if the inventory is empty. If it is, the function returns. Then, it asks the user for the name of the product to modify and searches for it in the inventory. If the product is not found, the function returns. Otherwise, it asks the user for the new details for the product and updates the product in the inventory. Finally, it prints a message confirming that the product was modified in the inventory.

delete\_product

Assuming you have a product\_t struct defined somewhere in your code, this function should do the following:

* Ask the user to enter the ID of the product they want to delete.
* Search through the products array for a product with that ID.
* If the product is not found, print an error message and return.
* Otherwise, shift all the elements after the product to delete down by one index, effectively "deleting" the product.
* Decrement num\_products by 1 to reflect the fact that there is one fewer product in the array.
* Print a success message.

search\_products

This function takes an array of product\_t structs as input, along with the number of products in the array. It prompts the user to enter a search term, then loops through each product in the array to see if the product name matches the search term. If a matching product is found, it prints the name and price of the product. If no matching products are found, it prints a message indicating that no products were found.

display\_product

This function first prints out the headers for the table of products, and then loops through the array of products and prints out the details of each one using printf with appropriate formatting. The %.2f format specifier is used to print out the price field with two decimal places.

generate\_sales\_report

This function calculates the total sales of each product by multiplying the price of the product with the quantity sold, and then adds up the sales of all products to get the total sales. It then prints a formatted sales report, including the name, price, quantity sold, and sales of each product, as well as the total sales.