Avila, Sean Rendel S.

Grocery Inventory Simulation

```
#include <iostream>
#include <vector>
#include <algorithm>

void printInventory(const std::vector<std::string>& inventory) {
   std::cout << "[";
   for (size_t i = 0; i < inventory.size(); ++i) {
      std::cout << "\"" << inventory[i] << "\"";
      if (i < inventory.size() - 1) {
         std::cout << ", ";
      }
   }
   std::cout << "]" << std::endl;
}

std::vector<std::string> updateInventory(const std::vector<std::string>& currentInventory,
                                         const std::vector<std::string>& newShipment,
                                         const std::vector<std::string>& soldOutItems) {
```

```
    std::vector<std::string> updatedInventory = currentInventory;
    updatedInventory.insert(updatedInventory.end(), newShipment.begin(),
newShipment.end());
    std::sort(updatedInventory.begin(), updatedInventory.end());

updatedInventory.erase(std::unique(updatedInventory.begin(),updatedInventory.end()),upd
atedInventory.end());
    for (const auto& item : soldOutItems) {
        updatedInventory.erase(std::remove(updatedInventory.begin(),
updatedInventory.end(), item), updatedInventory.end());


    }
    return updatedInventory;
}

int main() {
    std::vector<std::string> currentInventory = {"apples", "bananas", "pears"};
    std::vector<std::string> newShipment = {"kiwis", "bananas", "grapes"};
    std::vector<std::string> soldOutItems = {"apples", "oranges"};
    std::cout << "Current Inventory: ";
    printInventory(currentInventory);
    std::cout << "Shipment: ";
    printInventory(newShipment);
    std::cout << "Soul-out items: ";
    printInventory(soldOutItems);
    std::vector<std::string> updatedInventory = updateInventory(currentInventory,
newShipment, soldOutItems);
```

```
    std::cout << "Updated Inventory: ";
    printInventory(updatedInventory);

    return 0;

}
```