# Third Year B. Tech (EL & CE)

**Semester: V Subject:** Object-Oriented Programming Lab

**Name: Shreerang Mhatre Class: TY**

**Roll No: 52 Batch: A3**

# Experiment No: 08

**Name of the Experiment**: **List and Arrays**

**Performed on: 01/11/2023**

**Submitted on: 01/11/2023**

**Problem Statement:**

Write a program in C++ to manage a shopping list. Each shopping list item is represented by a string stored in a container. Your design requires a print function that prints out the contents of the shopping list.

* Create an empty list.
* Append the items, "eggs", "milk”, “sugar", "chocolate" and "flour" to the list. Print the list.
* Remove the first element from the list. Print the list.
* Insert the item, "coffee" at the beginning of the list. Print the list.
* Find the item, "sugar" and replace it with "honey." Print the list.
* Insert the item, "baking powder" before "milk" in the list. Print the list.
* Sort and search the item in the list.

**Output:**

// C:\Users\SHREERANG\Desktop\sbsns>cd "c:\Users\SHREERANG\Desktop\sbsns\" && g++

Shopping List:

eggs

milk

sugar

chocolate

flour

After removing the first item:

milk

sugar

chocolate

flour

After inserting 'coffee' at the beginning:

coffee

milk

sugar

chocolate

flour

After replacing 'sugar' with 'honey':

coffee

milk

honey

chocolate

flour

After inserting 'baking powder' before 'milk':

coffee

baking powder

milk

honey

chocolate

flour

Sorted Shopping List:

baking powder

chocolate

coffee

flour

honey

milk

'chocolate' found in the list.

**Code:**

#include <iostream>

#include <vector>

#include <algorithm>

int main() {

    std::vector<std::string> shoppingList;

    // Append items to the list

    shoppingList.push\_back("eggs");

    shoppingList.push\_back("milk");

    shoppingList.push\_back("sugar");

    shoppingList.push\_back("chocolate");

    shoppingList.push\_back("flour");

    // Print the list

    std::cout << "Shopping List:" << std::endl;

    for (const std::string& item : shoppingList) {

        std::cout << item << std::endl;

    }

    // Remove the first element

    shoppingList.erase(shoppingList.begin());

    // Print the modified list

    std::cout << "\nAfter removing the first item:" << std::endl;

    for (const std::string& item : shoppingList) {

        std::cout << item << std::endl;

    }

    // Insert "coffee" at the beginning

    shoppingList.insert(shoppingList.begin(), "coffee");

    // Print the modified list

    std::cout << "\nAfter inserting 'coffee' at the beginning:" << std::endl;

    for (const std::string& item : shoppingList) {

        std::cout << item << std::endl;

    }

    // Find and replace "sugar" with "honey"

    for (std::string& item : shoppingList) {

if (item == "sugar") {

            item = "honey";

        }

    }

    // Print the modified list

    std::cout << "\nAfter replacing 'sugar' with 'honey':" << std::endl;

    for (const std::string& item : shoppingList) {

        std::cout << item << std::endl;

    }

    // Insert "baking powder" before "milk"

    auto it = std::find(shoppingList.begin(), shoppingList.end(), "milk");

    if (it != shoppingList.end()) {

        shoppingList.insert(it, "baking powder");

    }

    // Print the modified list

    std::cout << "\nAfter inserting 'baking powder' before 'milk':" << std::endl;

    for (const std::string& item : shoppingList) {

        std::cout << item << std::endl;

    }

    // Sort the list

    std::sort(shoppingList.begin(), shoppingList.end());

    // Print the sorted list

    std::cout << "\nSorted Shopping List:" << std::endl;

    for (const std::string& item : shoppingList) {

        std::cout << item << std::endl;

    }

    // Search for an item in the list

    std::string searchItem = "chocolate";

    auto searchResult = std::find(shoppingList.begin(), shoppingList.end(), searchItem);

    if (searchResult != shoppingList.end()) {

        std::cout << "\n'" << searchItem << "' found in the list." << std::endl;

    } else {

        std::cout << "\n'" << searchItem << "' not found in the list." << std::endl;

    }

    return 0;

}







