**Assignment # 04:  
question#02:**

**Part a**

**Code:**

#include <iostream>

#include <vector>

using namespace std;

struct Product

{

int id;

string name;

};

vector<Product> inventory;

void add\_product(int id, const string& name)

{

Product newProduct = { id, name };

inventory.push\_back(newProduct);

}

void removeProduct(int id)

{

for (auto it = inventory.begin(); it != inventory.end(); ++it)

{

if (it->id == id)

{

inventory.erase(it);

cout << "Product with ID " << id << " removed from inventory." << endl;

return;

}

}

cout << "Product with ID " << id << " not found in inventory." << endl;

}

int main()

{

add\_product(1, "camera"); //instant insertion

add\_product(2, "headphones");

add\_product(3, "watch");

add\_product(4, "mouse");

removeProduct(1); // instant removal

removeProduct(4);

cout << "Current Inventory:" << endl;

cout << "Names: " << endl;

for (const auto& product : inventory)

{

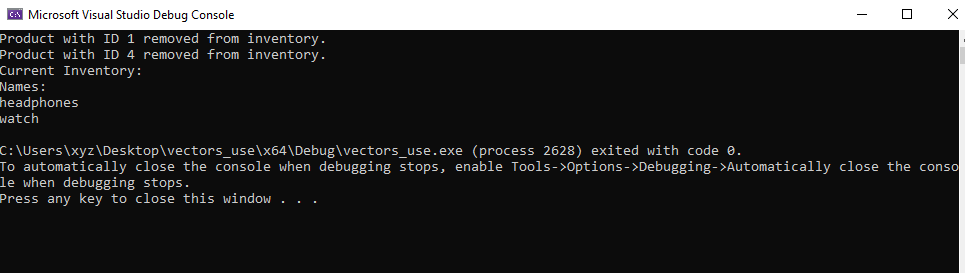
cout << product.name << endl;

}

return 0;

}

**output:**

****

**Part b**

**Code:**

#include <iostream>

#include <vector>

#include <algorithm>

#include <chrono>

using namespace std;

using namespace chrono;

void bubbleSort(vector<int>& arr)

{

int n = arr.size();

for (int i = 0; i < n - 1; ++i)

{

for (int j = 0; j < n - i - 1; ++j)

{

if (arr[j] > arr[j + 1])

{

swap(arr[j], arr[j + 1]);

}

}

}

}

int main() {

const int size = 100000;

vector<int> data(size);

// Initialize the vector in descending order

for (int i = size; i > 0; --i) {

data.push\_back(i);

}

// Measure execution time for Bubble Sort

auto startBubbleSort = high\_resolution\_clock::now();

bubbleSort(data);

auto stopBubbleSort = high\_resolution\_clock::now();

auto durationBubbleSort = duration\_cast<milliseconds>(stopBubbleSort - startBubbleSort);

// Print execution time for Bubble Sort

cout << "Bubble Sort Execution Time: " << durationBubbleSort.count() << " milliseconds" << endl;

// Verify that the vector is sorted

cout << "First 10 integers: ";

for (int i = 0; i < 10; ++i)

{

cout << data[i] << " ";

}

cout << endl;

cout << "Last 10 integers: ";

for (int i = size - 10; i < size; ++i) {

cout << data[i] << " ";

}

cout << endl;

// Reset the vector to descending order

reverse(data.begin(), data.end());

// Measure execution time for STL sort

auto startSTLSort = high\_resolution\_clock::now();

sort(data.begin(), data.end());

auto stopSTLSort = high\_resolution\_clock::now();

auto durationSTLSort = duration\_cast<milliseconds>(stopSTLSort - startSTLSort);

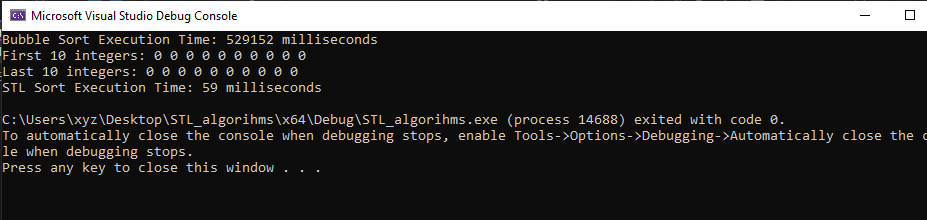
// Print execution time for STL sort

cout << "STL Sort Execution Time: " << durationSTLSort.count() << " milliseconds" << endl;

return 0;

}

**output:**

****