**Question no. 1**

**#include <iostream>**

**#include <vector>**

**using namespace std;**

**vector<int> Reverse(vector<int> vect) {**

**vector<int> result;**

**for (int i = vect.size() - 1; i >= 0; i--) {**

**result.push\_back(vect[i]);**

**}**

**return result;**

**}**

**int main() {**

**vector<int> nums = {1, 2, 3, 4, 5};**

**nums = Reverse(nums);**

**for (int i = 0; i < nums.size(); i++) {**

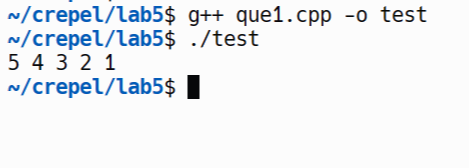
**cout << nums.at(i) << " ";**

**}**

**cout << endl;**

**return 0;**

**}**



**Question no. 2**

→ **#include <iostream>**

**#include <vector>**

**void printDiagonal(std::vector<std::vector<int>> &vals, int row, int col) {**

**while (row >= 0 && col < vals[0].size()) {**

**std::cout << vals[row][col] << " ";**

**row--;**

**col++;**

**}**

**std::cout << std::endl;**

**}**

**int main() {**

**std::vector<std::vector<int>> vals = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};**

**int row = 2;**

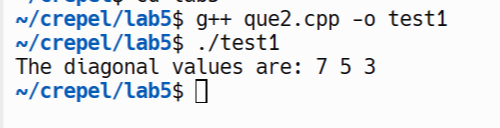
**int col = 0;**

**std::cout << "The diagonal values are: ";**

**printDiagonal(vals, row, col);**

**return 0;**

**}**



**Question no. 3**

**#include <algorithm>**

**#include <iostream>**

**#include <vector>**

**class Tensor {**

**public:**

**void sort(std::vector<int> &v) {**

**std::sort(v.begin(), v.end());**

**for (const auto &i : v) {**

**std::cout << i << " ";**

**}**

**std::cout << std::endl;**

**}**

**};**

**int main() {**

**Tensor tensor;**

**std::vector<int> v = {5, 3, 1, 4, 2};**

**std::cout << "Original vector: ";**

**for (const auto &i : v) {**

**std::cout << i << " ";**

**}**

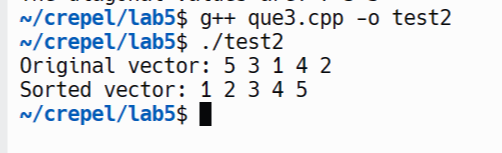
**std::cout << std::endl;**

**std::cout << "Sorted vector: ";**

**tensor.sort(v);**

**return 0;**

**}**

****

**Question no.4**

**→ There are several issues with the provided C++ class:**

**Post-increment in a const function: The function getIncrementedData() is marked as const which means it should not change any member variables of the class. Yet, it tries to increase the data which is forbidden. This can be solved by returning data + 1 rather than incrementing data.**

**Accessing non-static member in static function: The function getCount() is a static function, but it tries to access data, which is a non-static member variable. Static member functions can only directly access other static members (variables or functions). In order to do this, simply delete the line that displays the data.**

**Uninitialized static variable: The static variable count is defined but not defined anywhere. You should declare it outside the class, in the implementation file (.cpp).**

**#include <iostream>**

**class Example {**

**public:**

**Example(int y = 10) : data(y) {} // end Example constructor**

**int getIncrementedData() const {**

**return data + 1;**

**} // end function getIncrementedData**

**static int getCount() { return count; } // end function getCount**

**private:**

**int data;**

**static int count;**

**};**

**// Define static member outside the class**

**int Example::count = 0;**

**int main() {**

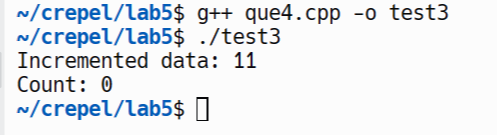
**Example ex;**

**std::cout << "Incremented data: " << ex.getIncrementedData() << std::endl;**

**std::cout << "Count: " << Example::getCount() << std::endl;**

**return 0;**

**}**

**→ **