TASK 3

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <sstream>
#include <iomanip>

// Define a structure to hold data
struct Data {
    int id;
    std::string name;
    double value;
};

// Function to serialize data into a string
std::string serializeData(const Data& data) {
    std::ostringstream oss;
    oss << data.id << "," << data.name << "," << std::fixed << std::setprecision(2) << data.value;
    return oss.str();
}

// Function to deserialize data from a string
Data deserializeData(const std::string& str) {
    std::istringstream iss(str);
    Data data;
    char comma;
    iss >> data.id >> comma >> data.name >> comma >> data.value;
    return data;
```

```
}

int main() {
    // Open input file
    std::ifstream inputFile("input.txt");
    if (!inputFile.is_open()) {
        std::cerr << "Error: Could not open input file." << std::endl;
        return 1;
    }

    // Read data from input file
    std::vector<Data> dataVector;
    std::string line;
    while (std::getline(inputFile, line)) {
        Data data = deserializeData(line);
        dataVector.push_back(data);
    }
    inputFile.close();

    // Sort data
    std::sort(dataVector.begin(), dataVector.end(), [](const Data& a, const Data& b) {
        return a.id < b.id;
    });

    // Open output file
    std::ofstream outputFile("output.txt");
    if (!outputFile.is_open()) {
        std::cerr << "Error: Could not open output file." << std::endl;
        return 1;
    }
```

```
    // Write sorted data to output file
    for (const auto& data : dataVector) {
        outputFile << serializeData(data) << std::endl;
    }
    outputFile.close();

    std::cout << "Data has been sorted and written to output file successfully." << std::endl;

    return 0;
}
```