Task 2: Coding 1

Write an application in C++ that reads a CSV file, processes it and writes the output as CSV file back to disk.

To read the input: The CSV file basically contains a 2D matrix of numbers, where each line holds a single row: “2`<delim>`4`<delim>`99`<delim>\n`”. The delimiter can be either space or a single comma (‘,’). Write the output in the same format as the input.

Once the input data is read, your application should perform filtering of “bad” values. Any entry of the matrix is “bad” when it has a value of 0. The application should now replace these bad values and compute a valid value by interpolating it from its surrounding, i.e., from the spatial neighbors in the matrix. You may assume that no two bad values are adjacent to each other.

Write the matrix with the replaced values to disk.

Your program should accept the input and output path as `program <input> <output>`.

Remarks:

- You may not use any 3rd party code or libraries. You may use the standard headers* and STL. Standard headers are listed here: http://en.cppreference.com/w/

Task 3: Coding 2

Write a function that sorts a container like *std::vector* or *std::list* containing objects according to the ordering of objects given in a second container. That is, the comparison to determine the sorted ordering must be done based on the second container; the first container must be reordered based on these comparisons.

Example:

Container 1: [“a”, “b”, “c”]
Container 2: [4, 7, 1]

Result after applying the function:

Container 1: [“c”, “a”, “b”]
Container 2: [1, 4, 7]

Constraints:

- You may not assume that the representation and internal structure of the objects within container 2 can be modified. In particular, the object in container 2 might not have a comparison operator, but your sort function can take more than two parameters. You may assume that the objects are copy-constructible. It should work for any class.

- You may not use any 3rd party code or libraries. You may use the standard headers* and STL. Standard headers are listed here: http://en.cppreference.com/w/