C++ programming task

The goal of this task is to create an matrix class library that supports simple matrix manipulation and analysis. This library must support a variety of matrix types and the following operations:

- Creation: Given a width, height and element type (float or uint8), create the matrix.
- Width: This method should provide the width of the matrix.
- Height: This method should provide the height of the matrix.
- Copy: This method should return an identical copy of the matrix.
- Accessor function(s): This method should provide element access to the matrix, i.e.: get or set the value of the element at row 4, column 3
- Histogram binning: This method should compute the histogram of the matrix. It takes the number of bins to use as an argument and returns a vector where the ith entry contains the number of elements that lie in the ith range of possible element values. For example, if we have an uint8 matrix, there are 256 possible element values. If you specify 8 bins, then the first bin will count the number of elements in the matrix that are in the range [0,31], the second bin will count the number of elements in the range [32,63], etc. In general the class function should deal with the fact that different matrix element types have different ranges (e.g. assume that floats have the range [0,1], uint8 are [0,255], uint16 are [0,65535], etc.), but for this exercise only implement this function for uint8 matrices.

The matrix element depth (e.g. float, uint8, etc.) is specified when the matrix object is created. Optionally, it is desirable that after matrix creation the user code can perform simple operations such as copy using references to generic matrix objects, without having to know the type of the matrix.

For instance, if the user code creates a float matrix and then passes a reference to this object to some other part of the software, the software should be able to do things like copy, reshape, etc. without ever knowing (or caring about) the actual type of the matrix. This library should support at the least two different element depths: uint8 and float.

For your convenience, a CSV file containing example matrix data as well as a function to read that data into a vector has been provided for testing. In the CSV file the first element is the number of rows (H), the second element is the number of columns (W) and the remaining H*W elements are the elements values in row major order (p_00, p_01, …, p10, p11, …, p_rc, ...), where r is the row and c is the column).

Define and implement the above class or classes, and give a simple usage program that loads the matrix provided and prints the histogram bin values to the terminal. Code should compile and focus should be placed on class **structure** and **organization**. Add

documentation where you feel necessary. **Supply the command to compile your code**. Describe your implementation and others you considered. Describe the structure of each class and why it is most appropriate.

DO NOT worry if your first design isn't ideal. If you run out of time, describe what you would have liked to do and/or what you'd change about your design. It is better to finish the task with a bad design and talk about how to improve it in your comments than not to finish the task at all.

Summary of tasks:

- Create a matrix class for simple manipulation and analysis
- Support uint8 and float element depths
- Creation of matrix
- Height/Width functions
- Accessor functions
- Copy function
- Histogram binning function (only for uint8 matrices)
- Simple usage program that reads matrix data from a csv file using the provided function, tests the class, and prints the matrix histogram to the terminal.
- Compilation instructions