



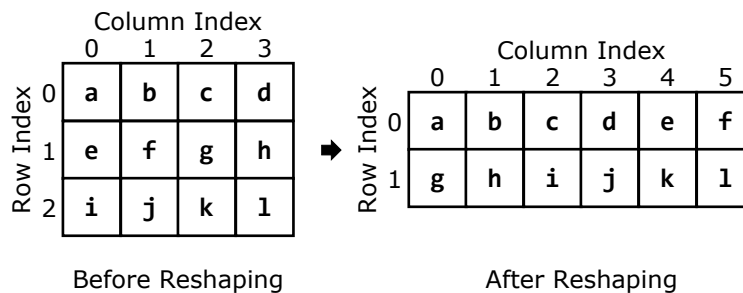
CPT-182-83/R83 (23/SP) - Programming in C++

Exam 3 (40 Points)

Deadline: Thursday, May 18, by 07:59 AM (Source Code Submitted on Canvas)

Project Overview

In MATLAB, there is a function called **reshape** which can reshape an **m-by-n** matrix into a new one with a different size **r-by-c**, keeping its original data. The example below reshapes a **3-by-4** matrix into a **2-by-6** matrix:



In this project, you are going to write a C++ program that reads in a matrix of characters (type **char**) from an input file, reshapes the matrix, and writes the reshaped matrix to an output file.

The Matrix Class

1) The class has the following private data fields:

- **data**. This is a 2-dimensional vector of characters (type **vector<vector<char>>**) that stores the elements in the matrix.
- **num_of_rows**. This variable stores the number of rows in the matrix.
- **num_of_columns**. This variable stores the number of columns in the matrix.

2) The class has a default constructor that initializes both **num_of_rows** and **num_of_columns** to **0**.

3) The class overloads the following operators:

- Stream extraction operator ("**>>**"). First of all, the operator reads in two positive integers from the input stream, the number of rows (**num_of_rows**) and number of columns (**num_of_columns**) in the matrix, respectively. Next, the operator initializes the dimension of the 2-dimensional vector **data** with the number of rows and number of columns read in. Finally, using nested for loops, the operator read in the elements in the matrix from the input file and fill the cells in **data** with the elements read in.

- Stream insertion operator ("<<"). First of all, the operator writes the number of rows and number of columns (separated by a tab character) to the output stream. Next, it writes an empty line. Finally, using nested for loops, the operator writes the elements in **data** to the output stream. Elements should be separated by a single tab character, and a newline character should be written at the end of each row.

4) The class includes only one class-member function, **reshape()**:

- The function takes two positive integers as argument, which are the **new** number of rows and **new** number of columns of the matrix **after reshaping**.
- The function does **not** return a value. It changes the current matrix object directly instead of creating a new matrix object.
- If the **new** number of rows and **new** number of columns are **illegal** (e.g., reshaping a **3-by-4** matrix into a **2-by-5** matrix, reshaping a **3-by-4** matrix into a **4-by-5** matrix), then the function should **throw an exception** with error message of "**Illegal reshaping**".
- If the **new** number of rows and **new** number of columns are **legal** (e.g., reshaping a **3-by-4** matrix into a **2-by-6** matrix, reshaping a **6-by-7** matrix into a **2-by-21** matrix), then the function should correctly reshape the matrix. It is your responsibility to **design and implement a correct algorithm** to complete the task.
- Please note that when you traverse the matrix, you need to use the **row-major order**.

The Input File

- The input file is a **plain text file** (filename: **before.txt**).
- First, there are two positive integers stored in the input file, which are the number of rows and number of columns of the matrix **before reshaping**.
- Then, elements in the matrix **before reshaping** are listed.
- Finally, there are two positive integers stored at the end of the input file, which are the number of rows and number of columns of the matrix **after reshaping**.
- Please see the picture below to better understand the input file format.

```

3 8

a b c d e f g h
i j k l m n o p
q r s t u v w x

6 4

```

➡ Before reshaping, dimension is **3-by-8**.

➡ Elements in the **3-by-8** matrix

➡ After reshaping, dimension is **6-by-4**.

- Please refer to the **sample input files** to better understand the input file format.

The Output File

- The output file is a **plain text file** (filename: **after.txt**).
- The output file stores the matrix **after reshaping**.
- Please use the overloaded **stream insertion operator** ("**<<**") in the **Matrix** class to write the reshaped matrix to the output file.
- Please refer to the **sample output files** to better understand the output file format.

The main() Program

- Create a **Matrix** object.
- Using the overloaded **stream extraction operator** ("**>>**") in the **Matrix** class, read in the matrix from the input file.
- Read in the number of rows and number of columns **after reshaping** from the input file.
- In a **try** block, call the **.reshape()** function to reshape the matrix, then using the overloaded **stream insertion operator** ("**<<**") in the **Matrix** class to write the reshaped matrix to the output file (if **no exception** thrown).
- In a **catch** block, use the **.what()** function to write the error message to the output file.

Sample Input and Output Files

Please download the sample input and output files [from Canvas](#).

Deadline: Thursday, May 18, by 07:59 AM (Source Code Submitted on Canvas)