CPE 212 - Fundamentals of Software Engineering
Project01 - C++ Review

5 points

Important Note

All CPE 212 projects are automatically graded in order to provide you timely feedback. So, it is critical that you follow all directions for the preparation and submission of your projects for grading. Failure to follow the directions may result in zero credit (0 points).

Due Dates

Description	Points	Due Date
Project Code	5	January 17, 2020 by 11:59pm

Project Goals

A fundamental software engineering skill is the design, implementation, and testing of a software component that may be integrated into a larger software product. In order to do this, the software component must conform to a previously agreed upon interface format. As part of this assignment, you will practice this skill by writing several functions that will be integrated into a larger software product provided by the instructor. Along the way you will review basic C++ programming skills required for successful completion of CPE 212.

Project Description

For this project, you will complete the provided partial C++ program by implementing six functions that perform simple image processing operations on images stored as 16x16, two dimensional arrays of integers. The image processing operations are:

1. Load Image
2. Flip
3. Rotate
4. Transpose

You must implement each of these operations as a C++ function. Your code for the six functions above must appear in the file named project01.cpp in order to compile correctly. Remember, spelling and capitalization count in Linux/C++.

The function main() has already been implemented for you to provide a common way for everyone to test their code. The function main() will scan a sample input file and perform the requested series of image processing operations by invoking your functions as needed.

Prototypes for the six functions are already provided for you in main.cpp (DO NOT modify main.cpp !!!). All program output is performed by the code in the main.cpp file – do not include any output statements in the file project01.cpp

Project01 Compilation Instructions

This project consists of two C++ files – one named main.cpp provided by the instructor and one named project01.cpp written by you, along with a file named makefile to help you compile your program. A makefile contains the sequence of commands required to compile and assemble (link) your executable program. The provided file works on blackhawk. If you examine the structure of main.cpp, you will see an extra #include directive near the end of the file – this statement will cause the preprocessor to import the function definitions you have written and saved in the project01.cpp file. So, for this

assignment, all you must do to compile the program is to use the following command at the Linux command line

```
make
```

which will create an executable named project01 from the provided main.cpp and the project01.cpp file you have written. While this is not a standard use of or location for a #include statement, it will help facilitate automatic grading of your submission.

If your program compiled successfully, you may then type

```
./project01  NameOfInputFile
```

to execute your program assuming that the input file is located in the same directory with the executable. (For example, ./project01 p01input1.txt)

Remember, submit ONLY the project01.cpp file for grading.

Project01 Submission Instructions

You will be provided with a C++ file named main.cpp which contains the main() function of the program. The function main() contains code that invokes several supporting functions that you must implement. The support functions you will write must be placed in a file named project01.cpp [lowercase letters, no spaces] in order for your program to compile and execute.

Important directions you must follow:

1. Do not modify the file named main.cpp!!! Failure to satisfy this requirement will result in zero credit (0 points) on this assignment.
2. All of your work (the support functions) must be placed in a file named project01.cpp. All lowercase letters in the filename with no spaces. Failure to satisfy this requirement will result in zero credit (0 points) on this assignment.

3. All output to the monitor (stdout) will be performed by the code provided in main.cpp

Your program must be fully commented (including variable and parameter declarations, functions, logical blocks) in order to receive debugging assistance from the instructor and teaching assistants.

NOTE

Your project submission will be automatically graded so it is **EXTREMELY IMPORTANT** that you READ and FOLLOW the project directions.

Submit project01.cpp via the Project01 Canvas dropbox.

Submissions by email receive ZERO CREDIT (0 points)!!!

Submissions that do not compile receive ZERO CREDIT (0 points)!!!

Project01 Include File Constraints

You may only include the following include files:

```
#include "project01.hpp"

#include <iostream>

#include <fstream>

#include <string>
```

Hint

Use the function prototypes appearing in **main.cpp** to create a file named **project01.cpp** that contains empty function definitions. Be sure that you can successfully use ***make*** to compile this skeleton solution before adding any additional code.

The **LoadImage** function is critical – **if this function does not work, then your project will fail every test**. Make sure that **LoadImage** works before continuing with any other functions.

After **LoadImage**, implement the simplest operations **Flip** and **Transpose** ***one at a time for each direction***, testing each with the appropriate input file. Rotations may be implemented by combinations of the three simplest operations.

Project01 Support Function Specifications

```
// Enumerations are used to define the direction
//  and rotation of the image. These values are
//  passed into the Flip and Rotate functions
enum Direction
{
    D_Horizontal = -1,
    D_Vertical = 1
};

enum Rotation
{
    R_Clock = -1,
    R_CounterClock = 1
};

// Defines the image matrix
typedef int Image[MAX_ROWS][MAX_COLS];
// load the image matrix from the file described.
// if an error is ever encountered, this should return false
// otherwise this should return true and modify the Image matrix
// according to whats in the file.
//
// This probably should be the first function you work on.
// if this function doesn't function correctly then your
// project is dead in the water.
bool LoadImage(const std::string & imageFile, Image image);

// This will flip the matrix according to the direction provided.
bool Flip(Image image, Direction d);

// this will rotate the matrix according to the direction provided.
bool Rotate(Image image, Rotation r);

// this will transpose the image, which rotates it 180 degrees.
void Transpose(Image image);
```

Project01 Sample Image File

```
# p01image1.txt -- Sample Image #1

0 0 3 1 1 7 0 0 0 1 1 0 1 5 1
1 3 0 4 4 0 1 0 1 9 1 0 2 1 0
0 6 1 1 0 1 8 0 1 0 1 5 0 1 1
0 1 1 0 0 0 0 3 0 3 1 0 0 3 0
3 0 7 3 2 0 0 0 0 0 0 1 1 7 8
1 0 0 1 1 1 1 0 2 0 5 7 1 0 1
0 0 1 1 0 1 1 1 0 5 1 1 8 0 0
0 1 1 1 0 1 1 6 0 0 2 1 6 0 3
6 0 0 1 0 1 1 0 2 1 1 0 0 1 0
1 1 0 3 1 0 0 0 0 1 1 6 1 1 9
1 1 0 1 1 0 0 0 1 0 1 1 0 0 1
1 3 0 0 2 4 1 6 0 1 1 0 5 1 2
6 0 1 0 0 0 9 1 1 1 7 4 0 1 8
5 7 0 0 0 1 1 0 1 0 5 1 3 0 1
5 1 1 1 3 7 0 0 2 7 0 8 1 6 0
```