ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет программной инженерии и компьютерной техники

Кафедра «ВТ»

ЛАБОРАТОРНАЯ РАБОТА № _5_

ПО ДИСЦИПЛИНЕ «Система языкового программирования»

Выполнил(а): Чан Чунг Дык

Группа: P3202

Санкт-Петербург

2017/2018

Assignment: Image Rotation

You have to create a program to rotate a BMP image of any resolution to 90 degrees clockwise.

BMP File Format

BMP (BitMaP) format is a raster graphics format, which means that it stores an image as a table of colored dots (pixels). In this format the color is encoded with numbers of a fixed size (can be 1, 4, 8, 16, or 24 bits). If 1 bit is used per pixel, the image is black and white. If 24 bits are used, the number of different colors possible is roughly 16 million. We only implement the rotation of 24-bit images.

Architecture

We want to think about program architecture that is extensible and modular.

1. Describe the pixel structure struct pixel to not work with the raster table directly (as with completely structureless data). This should always be avoided.

2. Separate the inner image representation from the input format. The rotation is performed on the inner image format, which is then serialized back to BMP. There can be changes in BMP format, you might want to support other formats, and you do not want to couple the rotation algorithm tightly to BMP.

To achieve that, define a structure structure image to store the pixel array (continuous, now without padding) and some information that should really be kept. For example, there is absolutely no need to store BMP signature here, or any of the never-used header fields. We can get away with the image width and height in pixels. You will need to create functions to read an image from BMP file and to write it to BMP file (probably also to generate a BMP header from the inner representation).

3. Separate file opening from its reading.

4. Make error handling unified and handle errors in exactly one place (for this very program it is enough). To achieve that, define the from_bmp function, which will read a file from the stream and will return one of the codes that show whether the operation completed successfully or not. Remember the flexibility concerns. Your code should be easy to use in applications with graphical user interface (GUI) as well as in those without GUI at all, so throwing prints into stderr all over the place is not a good option: restrict them to the error handling piece of code. Your code should be easily adaptable for different input formats as well.