Assignment 4 – Part 2 (Bitmap Option)
Structures & Functions
COSC 051

For this assignment you are requested to refactor the software you wrote for assignment 4. You will use a structure to encapsulate all relevant data items that represent a Bitmap Image. The structure you create may contain other structures. As an example the header section of a bitmap image contained three structures, your BITMAPIMAGE_STRUCT should contain as a minimum:

- `bmpfile_magic`, `bmpfile_header` and `BITMAPINFOHEADER` structures
- `InputFile stream`
- `string to hold the filename`
- `an array or dynamic array`

The creation of a structure for your `imageTool` will allow you to open and manipulate several bitmap images in the same program.

The second part of the assignment requires you to write functions to operate on the bitmap image. ALL operations on a bitmap will happened by calling one of your functions. You will have to pass a BITMAPIMAGE_STRUCT variable to your function. Examples include the following functions:

```c
// Returns true if the bitmap file exists and was opened successfully, false otherwise
bool openBitMap(BITMAPIMAGE_STRUCT & b);

// Returns number of bytes read if success, -1 otherwise
int readBitMapHeader(BITMAPIMAGE_STRUCT & b);

// Returns number bytes of written if success, -1 otherwise
int writeBitMapHeader(BITMAPIMAGE_STRUCT & b);

// Returns number of bytes read on success, -1 otherwise
int readBitMapData(BITMAPIMAGE_STRUCT & b);

// Returns number of bytes written on success, -1 otherwise
int writeBitMapData(BITMAPIMAGE_STRUCT & b);

// copies the header and data from bitmap image b to bitmap image c
bool copyBitMap(const BITMAPIMAGE_STRUCT & b, BITMAPIMAGE_STRUCT & c);

// Converts the current bitmap to gray scale, returns true on success, false otherwise
bool convertToGray(BITMAPIMAGE_STRUCT & b);

// closes the bitmap image file and deletes any necessary memory
void closeBitMap(BITMAPIMAGE_STRUCT & b);

//Return size of bytes need to hold color data for image
int sizeofColorData(BITMAPIMAGE_STRUCT b);
```
Treat the bitmap image as 4 logical quadrant image. Rearrange any of the quadrants in the image.

```c
void jumbleImage(BITMAPIMAGE_STRUCT& b);
```

**Example Structures**

```c
typedef struct BITMAPIMAGE_STRUCT{
    bmpfile_image bimage;
    bmfile_header;
    .
    .
    .
    unsigned char *pixelArray;
    .
    .
    etc;
} BitmapImageStruct;
```

To receive any points your program must use a structure for a bitmap image and implement at least 9 of the above functions. Write a program using your structure and functions. Note the user can supply multiple (up to 2) bitmap images on the command line. The first argument is the 1\textsuperscript{st} bitmap image, the 2\textsuperscript{nd} argument is the output file for the first bitmap. The 3\textsuperscript{rd} argument is the name of the second input bitmap image and the 4\textsuperscript{th} argument is the output file name for the second input bitmap image.

```
./a.out  inputImage_1  outputImage_1  inputImage_2  outputImage_2
```

Program due date is Thursday November 17, 2011 at 11:59 pm. Extension to the program due date will not be granted for this assignment. Please plan appropriately.

This graded assignment is worth 100 points and will be counted as part of your programming grade for the course.

The product that you submit must be your own work. Collaboration is allowed as specified within the syllabus for this course. For this assignment, you are not required to submit an acknowledgement statement.