

A SIMPLE TUTORIAL ON EASY BMP LIBRARY:

First, listen to the presentation of the code that converts a color image to black and white thus compressing the image significantly.

To run a program that uses the EasyBMP library, you need the following files:

[EasyBMP.cpp](#), [EasyBMP.h](#), [EasyBMP_BMP.h](#), [EasyBMP_DataStructures.h](#), and [EasyBMP_VariousBMPutilites.h](#)

Once you download these files, you can now test the program we presented in the lab for converting a color image to black-and-white. The source code for this program is listed below:

```
#include "EasyBMP.h"
using namespace std;

int main( int argc, char* argv[] )
{

    BMP Background;
    Background.ReadFromFile(argv[1]);

    BMP Output;
    int picWidth = Background.TellWidth();
    int picHeight = Background.TellHeight();
    Output.SetSize( Background.TellWidth() , Background.TellHeight() );
    Output.SetBitDepth(1);

    for (int i = 1; i < picWidth-1; ++i)
        for (int j = 1; j < picHeight-1; ++j) {
            int col = (Background(i, j)->Blue + Background(i,j)->Green + 10* Background(i,j)-
>Red)/12;
            if (col > 127) {
                Output(i,j)->Red = 255;
                Output(i,j)->Blue = 255;
                Output(i,j)->Green = 255; }
            else
            {
                Output(i,j)->Red = 0;
                Output(i,j)->Blue = 0;
                Output(i,j)->Green = 0;
            }
        }
    Output.WriteToFile( argv[2]);
    return 0;
}
```

You can download this code [here](#). Next you need an image file to convert to black and white. Here is one. Now, you are ready to run the program.

Let the source program be `bw.cpp`. Let the image be [in.bmp](#)

The following commands will produce the black and white image `out.bmp`:

```
% g++ -o bw.out bw.cpp EasyBMP.cpp  
%bw.out in.bmp out.bmp
```

Finally, make sure that `out.bmp` has been created in the current directory. Open it and check that it is the black and white version of `in.bmp`.

Now you are ready to work on the problems of project # 2 and beyond!