BRITISH COLUMBIA
INSTITUTE OF TECHNOLOGY

COMP 8551

Assembly Language Lab I Exercises

1. Review the code samples provided. Make sure you are able to see disassembled code in Visual Studio, and understand basic x86 instructions.

2. Make sure you understand and can reproduce the example shown in class of functions with arguments passed by value and reference.

3. Write a windows application that reads in a colour image and a "kernel" image. The kernel image should be much smaller (e.g., 1/10th the size of the original image). The user should be provided with a simple user interface (in a window) to specify the two image files and a blending factor. The blending factor is a value between 0 and 1 (or 0% and 100%) that determines how the kernel image is blended with the original image, according to the following:

   ```
   pBlendImg[i][j] = pOrigImg[i][j] * blendFac + pKernelImg[i][j] * (1 - blendFac);
   ```

   where **`pOrigImg`** is the original image, **`pKernelImg`** is the kernel image, **`pBlendImg`** is the resulting image, and **`blendFac`** is the blending factor. The code should use a simple for loop to iterate over each pixel in **`pOrigImg`**. The program should provide the user with a button to start the blending, then display the original, kernel and blended images, each in its own window. The total time taken to blend the images should be reported to the user where they specified the input parameters (filenames, blending factor, etc.).

4. Write two additional functions to perform the blending, providing the user with the option to select either of these methods over the two you have already provided above. The first function should use MMX intrinsics (see http://concatenative.org/wiki/view/SSE, and the tutorial and sample code at http://www.softkam.ro/index.php?pid=alphablend). The second function should use SSE intrinsics. The user should be given the ability to choose from either of these methods, or the two above.