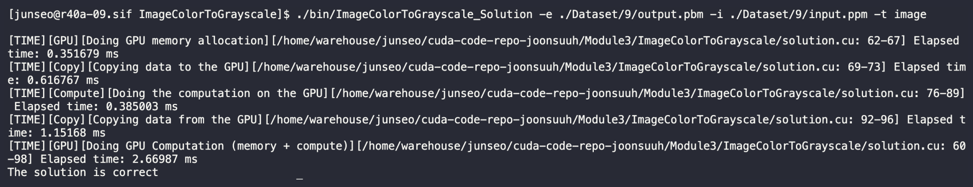
Image Color to Grayscale



```
[junseo@r40a-09.sif ImageColorToGrayscale]$ ./bin/ImageColorToGrayscale_Solution -e ./Dataset/9/output.pbm -i ./Dataset/9/input.ppm -t image

[TIME][GPU][Doing GPU memory allocation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module3/ImageColorToGrayscale/solution.cu: 62-67] Elapsed
time: 0.351679 ms
[TIME][Copy][Copying data to the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module3/ImageColorToGrayscale/solution.cu: 69-73] Elapsed tim
e: 0.616767 ms
[TIME][Compute][Doing the computation on the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module3/ImageColorToGrayscale/solution.cu: 76-89]
 Elapsed time: 0.385003 ms
[TIME][Copy][Copying data from the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module3/ImageColorToGrayscale/solution.cu: 92-96] Elapsed t
ime: 1.15168 ms
[TIME][GPU][Doing GPU Computation (memory + compute)][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module3/ImageColorToGrayscale/solution.cu: 60
-98] Elapsed time: 2.66987 ms
The solution is correct
```

Figure 1: `ImageColorToGrayscale_Solution` output

Questions

1. How many floating-point operations are being performed in your color conversion kernel? EXPLAIN.

   The kernel performs `imageWidth * imageHeight * 5` floating-point operations. Each thread (`imageWidth * imageHeight`) in the if statement performs 3 multiplications and 2 additions.

2. Which format would be more efficient for color conversion: a 2D matrix where each entry is an RGB value, or a 3D matrix where each slice in the Z axis representes a color? In other words, is it better to have color interleaved in this application? Can you name an application where the oposite is true?

   It would be more efficient to have a 2D matrix since the threads in the kernel would be able to access the RGB values in a single memory read operation.

   An application where it would be better to have color interleaved would be an algorithm that uses individual color channels separately—e.g. JPEG compression performs a discrete cosine transform on each converted color channel Y, Cb, Cr separately (source Wikipedia).

3. How many global memory reads are being performed by your kernel? EXPLAIN.

   The kernel performs `imageWidth * imageHeight * 3` global memory reads because each thread has to read the three color channels (RGB) from the input image.

4. How many global memory writes are being performed by your kernel? EXPLAIN.

   The kernel performs `imageWidth * imageHeight` global memory writes because each thread writes to the output image in the if statement.

5. Describe what possible optimizations can be implemented to your kernel to achieve a performance speedup.

   Again, we can increase performance with shared memory and using block sizes that are a multiple of the warp size (32).

6. Name three applications for color conversion

   i Device compatibility and cameras (all screens and cameras may use different color spaces and color channels)

   ii Printers (printing a color image into grayscale)

   iii Edge detection; Canny Edge Detection has to first convert the image to grayscale before applying the edge detection algorithm!