Documentatie PPD P2

* Info:
  + Device 1:
    - OS: Windows 10
    - Processor: 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz
    - Cores: 8
    - Logical Processors: 16
    - RAM: 16.0 GB (15.7 GB usable)
    - Cuda Cores: 2560
  + Device 2:
    - OS: Windows 10
    - Processor: AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz
    - Cores: 8
    - Logical Processors: 16
    - RAM: 16.0 GB (15.4 GB usable)
    - Cuda Cores: 1536
* Idee:
  + Programul va citi o imagine, va aplica un filtru grayscale asupra imaginii, aplicarea facandu-se utilizand CUDA, rezultatul fiind apoi salvat intr-o imagine de output
  + Formula utilizata este: x = 0.299r + 0.587g + 0.114b
* Specificatii:
  + Specifcatiile sunt puse pe codul sursa, la fiecare metoda, impreuna cu detaliile extra de implementare
* Python:
  + CUDA:
    - Grid-ul are dimensiunile (64,64,1)
    - Un block are dimensiunile (32,32,1)
    - Fiecare thread actioneaza pe un pixel (daca exista)
    - Astfel, un input poate fi o imagine de maxim 2048x2048 pixeli
    - Solutia poate fi extinsa totusi marind grid-ul si block-ul
  + Rezultate:
    - Testarea s-a facut pe o imagine de 1200x1455 pixeli
    - Pentru solutia secventiala, timpul mediu a fost de 6.2231361865997314 secunde
    - Pentru solutia CUDA, timpul mediu a fost de 1.5123622417449951 secunde
* Overall Conclusions:
  + Solutia CUDA este mult mai eficienta decat cea secventiala
  + CUDA este foarte capabil cand vine vorba de imagini/filters