HPC Lab Week 2

Here are some questions for applications of HPC in numerical computations, searching, and computer vision. Implement the following using multithreading/multiprocessing in Python or C++.

1. **Prime in a Power Set:** Array X is with numbers [1,2,3,...N] with N = 25. Create a power set Y (https://www.mathsisfun.com/sets/power-set.html). Find out list Z of the subsets whose sum is a prime number. Z should be in the original order of the Y. Implement this a) using serial operations with nested for loops, b) using multiprocessing functionality to independently perform operations. Create threads or multiple processes based on number of available CPU cores (use built-in functions). Analyze different time taken for each of these implementations by changing a) value of N, b) number of threads used (2,3…no_of_available_cores).
2. **Word Search in English Dictionary:** Here is a text file containing over 466k English words https://raw.githubusercontent.com/dwyl/english-words/master/words.txt. Write a program to find list L of palindrome words. Elements of L should be in the original order of the dictionary. Implement the palindrome finder a) using serial linear search, b) using approach that divides given array and perform simultaneous search using threading/multiprocessing. Create threads or multiple processes based on number of available CPU cores (use built-in functions). Analyze different time taken for each of these implementations by changing number of threads used (2,3…no_of_available_cores).
3. **Image Convolution:** Here is a picture of handsome hunk Mr. Donald Trump https://ocdn.eu/images/pulscms/NGY7MDA_/f8d05506e9250de59fa645f0fb7020e4.jpg. Write a program to perform convolution on the image matrix. Since the images have 3 channels the convolution filter should have size f*f*3, where f is the size of filter. For demo you can create vertical filters matrix like [[1,0,-1],[1,0,-1],[1,0,-1]] repeating for 3 channels to make it 3*3*3. Create random convolution filters of size 10*10*3. a) using serial computation using nested for loops, b) using a parallel processing approach that convolves different parts of image at the same time. Results can be further aggregated as the final convolution output. c) using built-in convolution functions in numpy or scipy. Analyze different time taken for each of these implementations by a) changing size of filter f, b) scaling up the original into bigger ones, c) number of threads used (2,3…no_of_available_cores).