Assignment E.318-C

Implement from scratch an algorithm for adding steganographic text to a digital image based on the least significant bit (LSB). You can use whatever programming language, including computer algebra systems, and public libraries (as long as not offering as a primitive steganographic/watermarking functions). In any case write your source code by respecting the standard requirements of readability and documentation (indent the code and add comments! Use meaningful names for variables and functions. Avoid or limit global variables. Structure and factorize the code)

The cover images (containers) can be freely obtained by public web resources and can be of four types: N (open nature, landscapes), S (still lifes), P (portraits), T (text).

Your algorithm should take as input the bitmap (type N, S, P, or T), a positive int k, and the hidden text t. Parameter k denotes the maximum number of least significant bits that can be used for injecting the hidden text t: values of k > 1 are admissible.

Implement a second algorithm for extracting the hidden text from a given stego image.

For any text t, you are expected to correctly manage things so that the recipient of the stego file can determine the length of the text to be extracted. Notice that not all possible lengths of t are compatible with k (and image size) and you may need to use more than k bits. Thus t should be not intended as an absolute max but rather as a recommended max.

Proper padding of t should be also taken into consideration in those cases where the combination of k and image size allows for hidden messages longer than |t|: in this case random padding can be useful, still caring to correctly manage the exact length of t.

You should also implement an algorithm that, given as input the original bitmap and the stego-bitmap, outputs the values of MSE (mean squared error, https://en.wikipedia.org/wiki/Mean_squared_error), PSNR (pick signal-to-noise ratio, https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio), and SSIM (structural similarity, https://en.wikipedia.org/wiki/Structural_similarity).

Use the implemented algorithms for testing several images (at least N, where $N \geq 6 \cdot$ team_size), for different values of k and t. Ensure that, for each image, you run multiple tests, measuring: MSE, PSNR, SSIM, running time, percentage of success in the extraction of the hidden text, and other possibly relevant numbers.
Provide suitable plotting of MSE, PSNR and SSIM for each image, as k or |t| vary and assess whether there is some threshold in |t| making the measures significantly varying. (You can obtain quick plots from gnuplot - http://www.gnuplot.info/ - or standard spreadsheets)

Last but not least, compare the results obtained for different types of cover images (N, S, P, and T) and discuss the differences.