**STEGANOGRAPHY**

**Abstract:**

Steganography is a method of concealing the presence of communication by embedding information within other, seemingly ordinary data. While various file types can serve as carriers for hidden messages, digital images—particularly in formats like .bmp—are among the most widely used, due to their prevalence online and ability to hold data discreetly. There are many steganographic methods, each with distinct advantages and limitations, to embed secret information within an image. Some methods prioritize the invisibility of the hidden content, while others allow for the inclusion of larger messages. Steganography differs from encryption by disguising the fact that any sensitive information is present; this can make it a useful tool for covert communication. When done well, steganography embeds data within files in such a way that casual viewers are unlikely to notice anything unusual. Specialized software is typically required to detect and extract such hidden information. In practice, steganography is commonly applied to embed one file, like a text document, into another, such as an image file.

**Requirements**

1. **Input**: The program should take a .bmp image file as the carrier and a text file containing the message to embed.
2. **Size Verification**: Ensure that the size of the .bmp image is sufficient to hold the entire message from the text file.
3. **Signature Option**: Include an option to embed a recognizable "signature" or "magic string" to mark an image as containing hidden data.
4. **Extraction Option**: Enable the ability to decode and retrieve the hidden message from the steganographic image.
5. **Command-Line Interface**: This should be a command-line application, with all options and inputs specified via command-line arguments.

The project files include:

* **Code files** (e.g., encode.c, decode.c, project\_main.c) that appear to handle encoding and decoding functions, likely for embedding information within a bitmap image (steganography).
* **Header files** (common.h, encode.h, decode.h) to manage shared functions and constants across the project.
* **Bitmap images** (e.g., beautiful.bmp, output.bmp) that are probably used as cover images for encoding messages.
* **Text files** (secret.txt, output.txt) which could serve as input/output files for encoded and decoded messages.