1. Câu hỏi về khái niệm và lý thuyết Steganography

* Steganography là gì? Giải thích sự khác biệt giữa steganography và cryptography.
* Tại sao lại sử dụng kỹ thuật LSB (Least Significant Bit) trong steganography? Lợi ích và nhược điểm của LSB là gì?
* Các phương pháp khác để thực hiện steganography là gì? Ngoài LSB, còn có các kỹ thuật nào?
* Ứng dụng của steganography trong thực tế là gì?
* Tại sao chúng ta cần một chuỗi kết thúc (1111111111111110) trong thông điệp? Điều này có vai trò gì trong việc giải mã?
* Steganography khác với watermarking như thế nào?
* Khi nào nên sử dụng steganography thay vì mã hóa (encryption)?
* Bảo mật của steganography có thể bị phá vỡ không? Nếu có, bằng cách nào?

2. Câu hỏi về mã hóa và giải mã trong đoạn mã

* Hàm text\_to\_bits có tác dụng gì? Tại sao phải chuyển đổi thông điệp thành chuỗi nhị phân?
* Vì sao chỉ thay đổi các bit ít quan trọng của mỗi kênh màu mà không phải là bit quan trọng nhất?
* Kết thúc của thông điệp được xác định như thế nào?
* Tại sao chúng ta cần kiểm tra chuỗi 1111111111111110 để dừng giải mã?
* Điều gì sẽ xảy ra nếu ảnh không chứa thông điệp đã mã hóa mà bạn chạy decode\_image?
* Giá trị của r, g, b là gì? Tại sao cần đọc cả ba kênh màu (RGB) khi mã hóa và giải mã thông điệp?
* Tại sao ta không sử dụng các kênh màu alpha hoặc độ trong suốt trong ảnh (nếu có)?

3. Câu hỏi về mã nguồn cụ thể

* Giải thích ý nghĩa của từng dòng mã trong hàm encode\_image.
* Cách binary\_message hoạt động như thế nào trong hàm encode\_image?
* Trong hàm decode\_image, tại sao lại cần đọc cả ba kênh màu (RGB) trong từng pixel?
* Đoạn mã int(format(r, '08b')[:-1] + binary\_message[data\_index], 2) thực hiện điều gì?
* Lý do sử dụng img.convert("RGB") là gì?
* Nếu ảnh được lưu ở định dạng PNG và sau đó được chuyển đổi thành JPEG, liệu thông điệp có bị mất không? Tại sao?
* Nếu ta thay đổi output\_path thành cùng tên với ảnh gốc, điều gì sẽ xảy ra?
* Tại sao lại cần binary\_message = binary\_message[:-16] trong hàm decode\_image?
* Hàm decode\_image sẽ hoạt động ra sao nếu không có chuỗi kết thúc trong binary\_message?
* Làm sao để nhận biết khi nào ảnh đã đạt tới giới hạn nhúng?

4. Câu hỏi nâng cao về bảo mật và cải tiến

* Có cách nào để phát hiện một ảnh đã bị nhúng dữ liệu không?
* Làm cách nào để cải tiến đoạn mã trên để tăng dung lượng của thông điệp nhúng?
* Bạn sẽ làm gì để đảm bảo an toàn hơn cho thông điệp nhúng, chẳng hạn bằng cách mã hóa (encryption) trước khi nhúng?
* Có thể mở rộng kỹ thuật này sang các định dạng file khác (âm thanh, video) như thế nào?
* Kỹ thuật LSB có thể áp dụng cho ảnh grayscale (ảnh đen trắng) không? Cách thức sẽ khác thế nào?
* Có cách nào để làm cho thông điệp khó bị phát hiện hơn nữa không?

5. Câu hỏi về xử lý hình ảnh và Python

* Pillow là gì và tại sao lại cần sử dụng thư viện này?
* Tại sao ảnh cần chuyển sang chế độ RGB mà không phải các chế độ khác?
* Cách xử lý pixel ảnh bằng thư viện Pillow là gì?
* Làm sao để hiển thị ảnh mã hóa lên màn hình trong Python?
* Nếu muốn kiểm tra dung lượng của thông điệp có thể nhúng tối đa trong ảnh, ta nên làm gì?

6. Câu hỏi thực hành và tình huống mở rộng

* Nếu giáo viên yêu cầu sửa đoạn mã để nhúng thông điệp vào chỉ một kênh màu (ví dụ, chỉ màu đỏ), bạn sẽ làm thế nào?
* Nếu có nhu cầu ẩn một thông điệp lớn hơn mà ảnh hiện tại không thể chứa được, bạn sẽ giải quyết thế nào?
* Có cách nào để tạo một hàm encode\_image nhận vào hai thông điệp khác nhau và lưu ở hai ảnh khác nhau?
* Giáo viên yêu cầu kiểm tra độ khác biệt giữa ảnh gốc và ảnh mã hóa, bạn sẽ thực hiện thế nào?
* Giải thích tại sao kết quả giải mã có thể bị sai hoặc có chuỗi ký tự ngẫu nhiên. Bạn có thể đề xuất cách nào để giảm thiểu lỗi này không?
* Nếu giáo viên yêu cầu thêm thông điệp vào các vị trí pixel ngẫu nhiên thay vì theo thứ tự, làm sao bạn sẽ làm điều này?
* Bạn có thể mở rộng mã để tạo một chương trình GUI đơn giản cho mã hóa và giải mã không?

**1. Khái niệm và lý thuyết Steganography**

**Steganography là gì?**  
Steganography là kỹ thuật giấu thông tin trong một phương tiện khác, ví dụ như hình ảnh, âm thanh hoặc video, để thông điệp không bị phát hiện. Mục đích là để bảo vệ thông tin bằng cách ẩn nó trong một dữ liệu khác.

**Sự khác biệt giữa steganography và cryptography:**

* **Steganography**: Ẩn thông điệp trong một dữ liệu khác để không ai biết có thông điệp tồn tại.
* **Cryptography**: Mã hóa thông điệp để chỉ những người có khóa mới có thể đọc được. Mặc dù thông điệp có thể bị phát hiện, nhưng nội dung của nó vẫn an toàn.

**2. Kỹ thuật LSB và các phương pháp khác**

**Tại sao sử dụng kỹ thuật LSB (Least Significant Bit) trong steganography?**  
Kỹ thuật LSB thay đổi bit ít quan trọng nhất của pixel trong ảnh, giúp ẩn thông điệp mà không làm thay đổi đáng kể chất lượng hình ảnh.

**Lợi ích và nhược điểm của LSB:**

* **Lợi ích**:
  + Dễ thực hiện
  + Giữ chất lượng hình ảnh cao
* **Nhược điểm**:
  + Dễ bị phát hiện nếu không sử dụng thêm biện pháp bảo mật
  + Nhạy cảm với các thao tác xử lý ảnh như nén.

**Các phương pháp khác để thực hiện steganography:**

* **DCT (Discrete Cosine Transform)**: Sử dụng trong ảnh JPEG.
* **DWT (Discrete Wavelet Transform)**: Ẩn thông tin trong miền sóng.
* **LSB Matching**: Thay đổi nhiều bit cùng lúc.

**3. Ứng dụng và chuỗi kết thúc**

**Ứng dụng của steganography trong thực tế:**

* Bảo mật thông tin nhạy cảm
* Nhận diện bản quyền
* Giao tiếp bí mật.

**Tại sao cần chuỗi kết thúc (1111111111111110)?**  
Chuỗi này đánh dấu kết thúc thông điệp, giúp quá trình giải mã biết khi nào dừng lại.

**4. Sự khác biệt giữa steganography và watermarking**

**Steganography khác với watermarking:**

* **Steganography**: Ẩn thông tin để không ai phát hiện.
* **Watermarking**: Thêm thông tin nhận diện vào dữ liệu để bảo vệ bản quyền, thông thường vẫn có thể nhìn thấy.

**5. Khi nào sử dụng steganography thay vì mã hóa**

**Khi nào nên sử dụng steganography thay vì mã hóa?**  
Sử dụng steganography khi cần ẩn thông tin mà không muốn người khác biết rằng có thông tin tồn tại, trong khi mã hóa sẽ chỉ bảo vệ nội dung.

**6. Bảo mật của steganography**

**Bảo mật của steganography có thể bị phá vỡ không?**  
Có, thông qua các phương pháp như phân tích tần số, phát hiện bất thường trong dữ liệu.

**7. Câu hỏi về mã hóa và giải mã**

**Hàm text\_to\_bits có tác dụng gì?**  
Chuyển đổi thông điệp văn bản thành chuỗi nhị phân để có thể nhúng vào ảnh.

**Vì sao chỉ thay đổi các bit ít quan trọng?**  
Để không làm giảm chất lượng hình ảnh.

**Kết thúc của thông điệp được xác định như thế nào?**  
Bằng chuỗi kết thúc đã định nghĩa.

**Tại sao cần kiểm tra chuỗi 1111111111111110?**  
Để xác định điểm dừng trong quá trình giải mã.

**Điều gì sẽ xảy ra nếu ảnh không chứa thông điệp đã mã hóa?**  
Chương trình có thể trả về thông điệp rỗng hoặc lỗi.

**Giá trị của r, g, b là gì?**  
Chúng là các giá trị màu sắc trong không gian màu RGB. Cần đọc cả ba kênh để đảm bảo thông điệp được nhúng đồng đều.

**8. Câu hỏi về mã nguồn cụ thể**

**Giải thích ý nghĩa của từng dòng mã trong hàm encode\_image.**  
Cần xem xét từng dòng mã để hiểu cách thức hoạt động, nhưng thường thì nó liên quan đến việc đọc pixel, thay đổi LSB và ghi lại kết quả.

**Cách binary\_message hoạt động như thế nào trong hàm encode\_image?**  
Chuyển đổi thông điệp thành chuỗi nhị phân và nhúng vào các bit LSB của ảnh.

\*\*Tại sao cần đọc cả ba

**Tại sao cần đọc cả ba kênh màu (RGB) trong từng pixel?**  
Để có thể nhúng thông điệp một cách đồng đều và tối ưu hóa không gian nhúng. Mỗi kênh màu có thể chứa thông tin, do đó việc đọc tất cả giúp tăng dung lượng thông điệp có thể nhúng.

**Đoạn mã int(format(r, '08b')[:-1] + binary\_message[data\_index], 2) thực hiện điều gì?**  
Đoạn mã này chuyển đổi giá trị màu đỏ (r) thành chuỗi nhị phân 8 bit, bỏ đi bit ít quan trọng nhất, sau đó ghép với bit nhị phân của thông điệp đang được nhúng, và cuối cùng chuyển đổi lại thành giá trị số nguyên.

**Lý do sử dụng img.convert("RGB") là gì?**  
Để đảm bảo rằng ảnh đang ở định dạng RGB, giúp quá trình xử lý pixel diễn ra chính xác.

**Nếu ảnh được lưu ở định dạng PNG và sau đó được chuyển đổi thành JPEG, liệu thông điệp có bị mất không? Tại sao?**  
Có thể, vì JPEG là định dạng nén có mất dữ liệu, có thể làm thay đổi các pixel và làm mất thông điệp nhúng.

**Nếu ta thay đổi output\_path thành cùng tên với ảnh gốc, điều gì sẽ xảy ra?**  
Ảnh gốc sẽ bị ghi đè và thông điệp nhúng có thể bị mất nếu không có bản sao lưu.

**Tại sao lại cần binary\_message = binary\_message[:-16] trong hàm decode\_image?**  
Để loại bỏ chuỗi kết thúc, chỉ giữ lại phần thông điệp thực tế đã nhúng.

**Hàm decode\_image sẽ hoạt động ra sao nếu không có chuỗi kết thúc trong binary\_message?**  
Nếu không có chuỗi kết thúc, hàm có thể không biết khi nào dừng lại, dẫn đến việc giải mã không chính xác hoặc lỗi.

**Làm sao để nhận biết khi nào ảnh đã đạt tới giới hạn nhúng?**  
Có thể kiểm tra số lượng bit có sẵn trong ảnh (số pixel x 3 cho RGB) và so sánh với độ dài của thông điệp nhúng.

**9. Câu hỏi nâng cao về bảo mật và cải tiến**

**Có cách nào để phát hiện một ảnh đã bị nhúng dữ liệu không?**  
Có, thông qua phân tích tần số hoặc sử dụng các thuật toán phát hiện bất thường.

**Làm cách nào để cải tiến đoạn mã trên để tăng dung lượng của thông điệp nhúng?**  
Có thể sử dụng các kênh màu khác nhau hoặc áp dụng các kỹ thuật như DCT hoặc DWT để nhúng nhiều thông tin hơn.

**Bạn sẽ làm gì để đảm bảo an toàn hơn cho thông điệp nhúng?**  
Mã hóa thông điệp trước khi nhúng để tăng cường bảo mật.

**Có thể mở rộng kỹ thuật này sang các định dạng file khác (âm thanh, video) như thế nào?**  
Có thể sử dụng các phương pháp tương tự để nhúng thông tin vào các mẫu sóng âm thanh hoặc khung hình video.

**Kỹ thuật LSB có thể áp dụng cho ảnh grayscale (ảnh đen trắng) không? Cách thức sẽ khác thế nào?**  
Có, nhưng chỉ cần thay đổi bit ít quan trọng nhất trong kênh màu duy nhất. Quy trình tương tự nhưng chỉ cần làm việc với một kênh.

**Có cách nào để làm cho thông điệp khó bị phát hiện hơn nữa không?**  
Sử dụng các thuật toán phức tạp hơn hoặc thay đổi vị trí nhúng ngẫu nhiên trong ảnh.

**10. Câu hỏi về xử lý hình ảnh và Python**

**Pillow là gì và tại sao lại cần sử dụng thư viện này?**  
Pillow là thư viện Python cho phép xử lý ảnh dễ dàng. Nó cung cấp các chức năng để mở, chỉnh sửa và lưu ảnh.

**Tại sao ảnh cần chuyển sang chế độ RGB mà không phải các chế độ khác?**  
RGB là chế độ màu phổ biến cho ảnh, giúp dễ dàng thao tác và nhúng thông tin.

**Cách xử lý pixel ảnh bằng thư viện Pillow là gì?**  
Có thể truy cập và thay đổi giá trị của mỗi pixel thông qua các phương thức của Pillow, như **load()** để lấy dữ liệu pixel.

**Làm sao để hiển thị ảnh mã hóa lên màn hình trong Python?**  
Sử dụng **show()** từ Pillow để mở ảnh trong trình xem ảnh mặc định.

\*\*Nếu muốn kiểm tra dung lượng của thông điệp có thể nhúng tối đa trong ảnh, ta nên làm gì

**Nếu muốn kiểm tra dung lượng của thông điệp có thể nhúng tối đa trong ảnh, ta nên làm gì?**  
Để kiểm tra dung lượng tối đa có thể nhúng, bạn có thể thực hiện các bước sau:

1. **Tính toán số pixel trong ảnh:** Sử dụng kích thước của ảnh (chiều rộng x chiều cao).
2. **Xác định số bit có thể nhúng:** Mỗi pixel trong ảnh RGB có 3 kênh màu (đỏ, xanh lá cây, xanh dương), và mỗi kênh có thể chứa 1 bit thông tin nếu bạn sử dụng phương pháp LSB. Do đó, số bit có thể nhúng = số pixel x 3.
3. **Chuyển đổi bit sang byte:** Chia tổng số bit cho 8 để có số byte có thể nhúng.

Ví dụ:

Pythoncopy

1width, height = img.size

2num\_pixels = width \* height

3max\_bits = num\_pixels \* 3 # 3 bits cho mỗi pixel

4max\_bytes = max\_bits // 8 # Chuyển đổi sang bytes

**11. Câu hỏi về ứng dụng thực tiễn**

**Những ứng dụng nào thực tế cho steganography?**

* **Bảo mật thông tin:** Giấu thông điệp nhạy cảm trong ảnh hoặc âm thanh để truyền tải an toàn.
* **Quản lý bản quyền:** Nhúng thông tin bản quyền vào ảnh hoặc video để xác minh quyền sở hữu.
* **Ghi chú cá nhân:** Giấu thông điệp trong ảnh gia đình hoặc kỷ niệm.
* **Chuyển giao thông tin bí mật:** Trong các lĩnh vực quân sự hoặc tình báo.

**Liệu steganography có thể bị lạm dụng không?**  
Có, như bất kỳ công nghệ nào, steganography có thể bị sử dụng cho mục đích xấu, chẳng hạn như giấu thông tin bất hợp pháp hoặc độc hại.

**Có những quy định pháp lý nào liên quan đến steganography không?**  
Các quy định pháp lý có thể khác nhau tùy theo quốc gia và có thể bao gồm luật về quyền riêng tư, bảo mật thông tin và bản quyền.

**Có cách nào để phát hiện steganography trong các file không?**  
Có, thông qua các công cụ phân tích tần số, phân tích cấu trúc file, hoặc các thuật toán phát hiện bất thường để xác định xem có sự hiện diện của thông tin nhúng hay không.