

Kết quả:

- Tạo file `binary_message.txt` chứa dữ liệu nhị phân
- Kiểm tra file đầu ra: Đọc file đầu ra thấy chuỗi nhị phân

Task 2: Xáo trộn file nhị phân và giấu vị trí xáo trộn bằng thuật toán LSB trên ảnh

Mục tiêu: xáo trộn file nhị phân và giấu vị trí xáo trộn bằng thuật toán LSB trên ảnh.

- Xáo trộn file nhị phân của thông điệp.

`./generate_random_bit_swap.py {file nhị phân ban đầu}`

```
ubuntu@stego-dsv-lsbimage:~$ ./generate_random_bit_swap.py binary_message.txt
Đã tạo file thông tin hoán đổi bit tại bit_swap_info.txt
Đã cập nhật file binary_file với chuỗi bit mới: 0100010100...
```

- File nhị phân mới được tạo cùng với file vị trí xáo trộn

```
ubuntu@stego-dsv-lsbimage:~$ ls
binary_message.txt  convert_message.py  extract_positions.py  output_original.pdf
binary_to_text.py   decode.py            generate_random_bit_swap.py  reverse_bit_shuffle.py
bit_swap_info.txt   encode.py            hide_in_image.py
checkwork           extract_from_image.py original.png
ubuntu@stego-dsv-lsbimage:~$ cat bit_swap_info.txt
Swap Order: [18, 14, 6, 16, 4, 15, 20, 23, 31, 13, 30, 7, 8, 25, 12, 3, 10, 5, 9, 17, 1, 28, 22,
24, 21, 26, 11, 27, 0, 29, 19, 2]
```

- Hoàn thành code của file giấu tin bằng thuật toán LSB trên ảnh

```
bit_idx = 0
```

```
for y in range(height):
```

```
    for x in range(width):
```

```
        if bit_idx >= len(binary_data):
```

```
            break
```

```
        r, g, b = pixels[y, x]
```

```
        if bit_idx < len(binary_data):
```

```
            r = (r & 0xFE) | int(binary_data[bit_idx])
```

```
            bit_idx += 1
```

```
        if bit_idx < len(binary_data):
```

```
            g = (g & 0xFE) | int(binary_data[bit_idx])
```