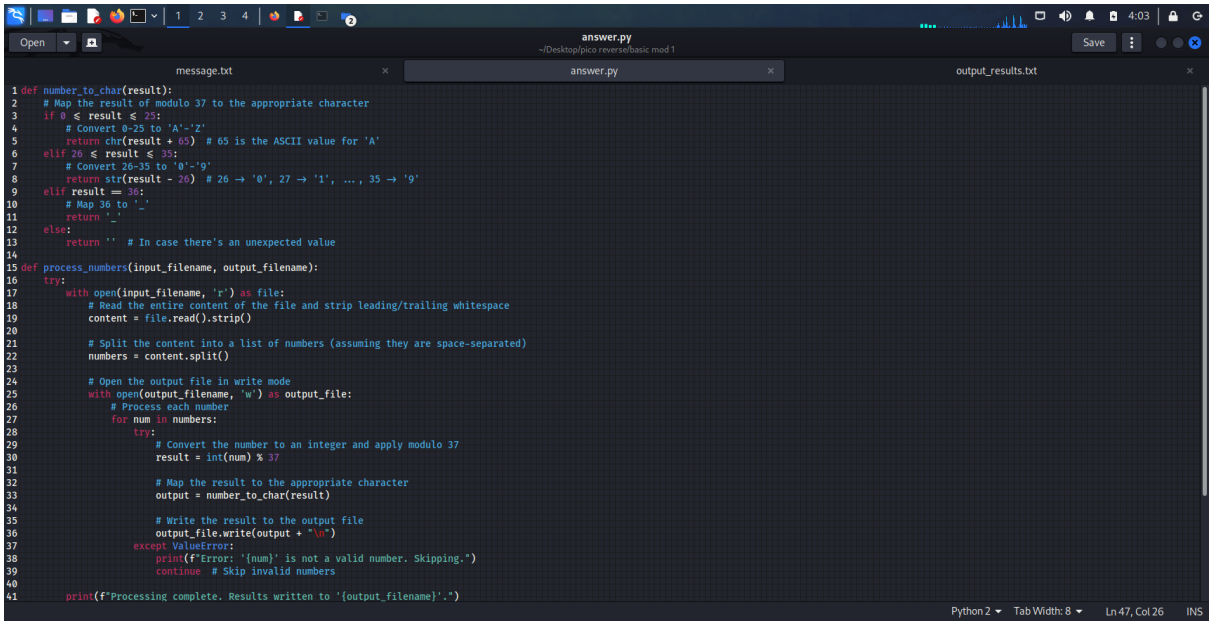

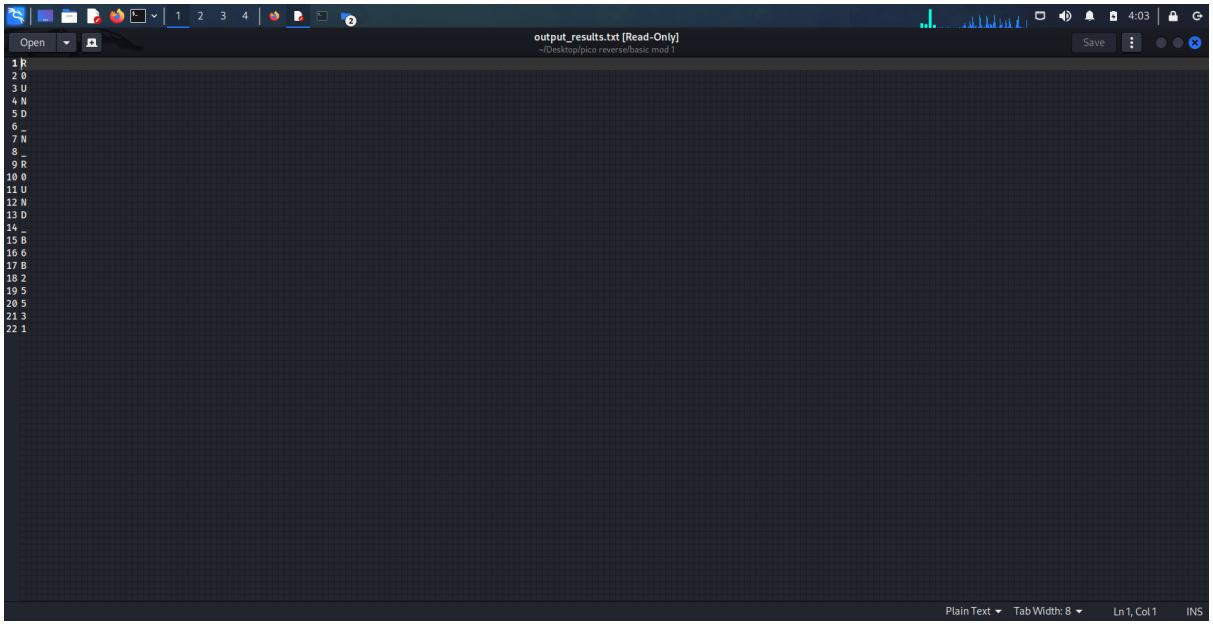


```
def number_to_char(result):
    # Map the result of modulo 37 to the appropriate character
    if 0 <= result <= 25:
        # Convert 0-25 to 'A'-'Z'
        return chr(result + 65)  # 65 is the ASCII value for 'A'
    elif 26 <= result <= 35:
        # Convert 26-35 to '0'-'9'
        return str(result - 26)  # 26 -> '0', 27 -> '1', ..., 35 -> '9'
    elif result == 36:
        # Map 36 to '_'
```

```
        return '_'
    else:
        return ''  # In case there's an unexpected value

def process_numbers(input_filename, output_filename):
    try:
        with open(input_filename, 'r') as file:
            # Read the entire content of the file and strip leading/trailing whitespace
            content = file.read().strip()

            # Split the content into a list of numbers (assuming they are space-separated)
            numbers = content.split()

            # Open the output file in write mode
            with open(output_filename, 'w') as output_file:
                # Process each number
                for num in numbers:
                    try:
                        # Convert the number to an integer and apply modulo 37
                        result = int(num) % 37

                        # Map the result to the appropriate character
                        output = number_to_char(result)

                        # Write the result to the output file
                        output_file.write(output + "\n")
                    except ValueError:
                        print(f"Error: '{num}' is not a valid number. Skipping.")
                        continue  # Skip invalid numbers

        print(f"Processing complete. Results written to '{output_filename}'.")
```

```
    except FileNotFoundError:
        print(f"Error: The file '{input_filename}' was not found.")

# Example usage:
input_filename = "message.txt"  # Replace with your actual input file path
output_filename = "output_results.txt"  # Replace with your desired output file path

process_numbers(input_filename, output_filename)
```