

```
1 def number_to_char(result):
2     # Map the result of modulo 37 to the appropriate character
3     if 0 <= result <= 25:
4         # Convert 0-25 to 'A'-'Z'
5         return chr(result + 65) # 65 is the ASCII value for 'A'
6     elif 26 <= result <= 35:
7         # Convert 26-35 to '0'-'9'
8         return str(result - 26) # 26 -> '0', 27 -> '1', ..., 35 -> '9'
9     elif result == 36:
10        # Map 36 to '_'
11        return '_'
12    else:
13        return '' # In case there's an unexpected value
14
15 def process_numbers(input_filename, output_filename):
16     try:
17         with open(input_filename, 'r') as file:
18             # Read the entire content of the file and strip leading/trailing whitespace
19             content = file.read().strip()
20
21             # Split the content into a list of numbers (assuming they are space-separated)
22             numbers = content.split()
23
24             # Open the output file in write mode
25             with open(output_filename, 'w') as output_file:
26                 # Process each number
27                 for num in numbers:
28                     try:
29                         # Convert the number to an integer and apply modulo 37
30                         result = int(num) % 37
31
32                         # Map the result to the appropriate character
33                         output = number_to_char(result)
34
35                         # Write the result to the output file
36                         output_file.write(output + "\n")
37                     except ValueError:
38                         print(f"Error: '{num}' is not a valid number. Skipping.")
39                         continue # Skip invalid numbers
40
41     print(f"Processing complete. Results written to '{output_filename}'.")
```

```
1 R
2 0
3 U
4 N
5 D
6 _
7 N
8 _
9 R
10 0
11 U
12 N
13 D
14 _
15 B
16 6
17 B
18 2
19 5
20 5
21 3
22 1
```

```
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)
```