```python
#-------------------------------------------------------------------------------
# Name:        enigma_caller.py
#
# Notes:       Caller to scramble or unscramble messages from a WW2 M3 Enigma
#              machine.
#
# Links:       https://cryptii.com/pipes/enigma-machine
#
# TODO:
#-------------------------------------------------------------------------------

import re
from textwrap import wrap
from enigma_machine import M3EnigmaMachine

def clean_message(message):
    """Cleans a message of non-alphabetic characters and returns the result.

    Args:
        message:
            A string of text to clean.

    Returns:
        A string containing the cleaned up text.
    """
    regex = re.compile('[^a-zA-Z]')
    return regex.sub('', message)

def format_message(message, grouping=5):
    """Formats a message by spliting it into blocks for easier reading of
       encrypted text.

    Args:
        message:
            The message to be split into blocks.
        grouping:
            The block size of the split up text.

    Returns:
        A string containing the original text but grouped into blocks.
    """
    return ' '.join(wrap(message, grouping))

def create_craig_machine():
    """Let's set up the machine!!! It's been a lot of work, but I finally did it!

    Args:
        None.

    Returns:
        A M3EnigmaMachine set-up with fun settings for my 34th birthday :).
    """
    em = M3EnigmaMachine("UKW-B", ["III", "IV", "V"]) # Pythagorean triple
    em.set_rotors(["C", "J", "M"], ["I", "F", "U"]) # (9, 6, 21)
    em.set_plugboard('CR AI GM AS ON')

    return em

def main():
    em = create_craig_machine()

```

```
    print('Start Enigma Machine State...')
    print(f'\t{em}')

    print('Reflector:')
    print(f'\t{em.reflector}')

    print("Rotors:")
    for rotor in em.rotors:
        print(f'\t{rotor!r}')

    print("Plugboard:")
    print(f'\t{em.plugboard}\n')

    message_text = ('Happy thirty-fourth Birthday Craig!!!')
    clean_text = clean_message(message_text)
    encrypted_text = em.encode_message(clean_text)

    print(f'Original Text: {message_text}')
    print(f'Clean Text: {format_message(clean_text)}')
    print(f'Encrypted Text: {format_message(encrypted_text)}\n')

    print("Final Enigma Machine State...")
    print(f'\t{em}')

if __name__ == '__main__':
    main()









# Start Enigma Machine State...
#         [UKW-B, III:C:I, IV:J:F, V:M:U]
# Reflector:
#         [UKW-B, YRUHQSLDPXNGOKMIEBFZCWVJATI]
# Rotors:
#         [III, SUCAYWJLNPRTKXZBFDHVGMQEOI, notch=V, letter=C, ring=I]
#         [IV, UEOFDVZNWMCQSKYLPIHRBGJXTA, notch=J, letter=J, ring=F]
#         [V, GDKIZYWEPTVLACNSOJMXHBFRUQ, notch=Z, letter=M, ring=U]
# Plugboard:
#         [CR AI GM ON]

# Original Text: Happy thirty-fourth Birthday Craig!!!
# Clean Text: Happy thirt yfour thBir thday Craig
# Encrypted Text: VHJMV QPZYP QYUDS YZYYY FRKED UBLBN

# Final Enigma Machine State...
#         [UKW-B, III:D:I, IV:L:F, V:Q:U]
```