Assignment 1: Enigma Machine Simulator

Instructions for Checker

1. Running the app

   The app was built in Python 3. In order to run the app, a Python 3.7 installation is needed be installed on the machine. Then, you have to run "main.py" via the Python interpreter, after you decompress the ZIP file containing the assignment's solution.

   Entering the setting is per rotor, that means that for each rotor, you'll choose the type of the rotor, its offset and its setting. The rotors that you pick will be arranged as L-M-R according to the order you picked them. So, if for example, you pick I, II and III in that order, I is the left rotor, II is the middle rotor and III is the right rotor. Later, you'll pick the reflector and the plugboard configuration, and finally the message to encrypt.

   The output is the encrypted message and the final offset of the machine (left to right).
2. Object-oriented design

   As specified in the assignment's suggestion.
3. Task 5 Message decryption using the codebook
   1) Setup the settings for the machine, according to the Enigma's codebook for the required day – in our case, it's October 29th:
      - Rotors: II – V – IV
      - Ring setting: S – I – X
      - Plugboard configuration: ZU HL CQ WM OA PY EB TR DN VI
   2) Set the rotor's offset to be the first trigram in the message, that is CON.
   3) Decrypt the second trigram to retrieve the original key: MLD → DOR.
   4) Change the rotors' offset according to the last result, DOR.
   5) Insert the actual message to the machine, that is skip the identification group RNYHP.
   6) The decrypted message is:

      ```
      GROUP SOUTH COMMAND FROM GEN PAULUS X
      SIXTH ARMY IS ENCIRCLED X
      OPERATION BLAU FAILED X
      COMMENCE RELIEF OPERATION IMMEDIATLEY
      ```
4. Task 6 Profiler report

   The profiler which was used is cProfile, a built-in profiler for Python.
   The result of the profiler on "tester.py", the test script for 1000 creations of an Enigma machine and an encryption of a short message using them, is below:

```
         1529438 function calls (1529418 primitive calls) in 1.948 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    1.948    1.948 <string>:1(<module>)
     1000    0.001    0.000    0.001    0.000 Enigma.py:13(__init__)
     3000    0.043    0.000    0.102    0.000 Enigma.py:18(rotorStep)
     1000    0.013    0.000    1.157    0.001 Enigma.py:25(translate)
     3000    0.128    0.000    1.042    0.000 Enigma.py:32(forward)
    11000    0.008    0.000    0.008    0.000 Plugboard.py:10(<genexpr>)
     1000    0.176    0.000    0.764    0.001 Plugboard.py:6(__init__)
        1    0.000    0.000    0.000    0.000 Reflector.py:6(__init__)
        1    0.000    0.000    0.000    0.000 Reflector.py:9(generateReflector)
        3    0.000    0.000    0.001    0.000 Rotor.py:10(__init__)
     6005    0.005    0.000    0.005    0.000 Rotor.py:16(notch)
     3125    0.009    0.000    0.038    0.000 Rotor.py:20(advance)
    18000    0.247    0.000    0.748    0.000 Rotor.py:23(forward)
        3    0.000    0.000    0.001    0.000 Rotor.py:26(generateRotor)
     3125    0.009    0.000    0.029    0.000 Substitutor.py:13(shift)
    12000    0.018    0.000    0.425    0.000 Substitutor.py:20(reverse)
   258333    0.511    0.000    0.681    0.000 Substitutor.py:5(letterToIndex)
    65229    0.129    0.000    0.208    0.000 Substitutor.py:9(indexToLetter)
    27000    0.054    0.000    0.133    0.000 Translator.py:19(forward)
     1004    0.181    0.000    0.381    0.000 Translator.py:7(__init__)
        4    0.000    0.000    0.000    0.000 enum.py:265(__call__)
     6000    0.013    0.000    0.013    0.000 enum.py:329(__iter__)
    24000    0.034    0.000    0.034    0.000 enum.py:330(<genexpr>)
     3000    0.004    0.000    0.005    0.000 enum.py:332(__len__)
        4    0.000    0.000    0.000    0.000 enum.py:515(__new__)
    27130    0.011    0.000    0.011    0.000 enum.py:597(value)
        2    0.000    0.000    0.000    0.000 enum.py:801(__and__)
     4000    0.006    0.000    0.008    0.000 memoize.py:3(helper)
     1000    0.003    0.000    0.014    0.000 re.py:179(search)
     1000    0.003    0.000    0.003    0.000 re.py:286(_compile)
        2    0.000    0.000    0.000    0.000
sre_compile.py:223(_compile_charset)
        2    0.000    0.000    0.000    0.000
sre_compile.py:250(_optimize_charset)
        4    0.000    0.000    0.000    0.000 sre_compile.py:388(_simple)
        1    0.000    0.000    0.000    0.000 sre_compile.py:482(_compile_info)
        2    0.000    0.000    0.000    0.000 sre_compile.py:539(isstring)
        1    0.000    0.000    0.000    0.000 sre_compile.py:542(_code)
        1    0.000    0.000    0.000    0.000 sre_compile.py:557(compile)
      7/1    0.000    0.000    0.000    0.000 sre_compile.py:64(_compile)
        7    0.000    0.000    0.000    0.000 sre_parse.py:111(__init__)
       12    0.000    0.000    0.000    0.000 sre_parse.py:159(__len__)
       28    0.000    0.000    0.000    0.000 sre_parse.py:163(__getitem__)
        4    0.000    0.000    0.000    0.000 sre_parse.py:167(__setitem__)
        7    0.000    0.000    0.000    0.000 sre_parse.py:171(append)
     13/7    0.000    0.000    0.000    0.000 sre_parse.py:173(getwidth)
        1    0.000    0.000    0.000    0.000 sre_parse.py:223(__init__)
       26    0.000    0.000    0.000    0.000 sre_parse.py:232(__next)
       19    0.000    0.000    0.000    0.000 sre_parse.py:248(match)
       19    0.000    0.000    0.000    0.000 sre_parse.py:253(get)
       11    0.000    0.000    0.000    0.000 sre_parse.py:285(tell)
      3/1    0.000    0.000    0.000    0.000 sre_parse.py:407(_parse_sub)
      3/1    0.000    0.000    0.000    0.000 sre_parse.py:470(_parse)
        1    0.000    0.000    0.000    0.000 sre_parse.py:76(__init__)
        6    0.000    0.000    0.000    0.000 sre_parse.py:81(groups)
        1    0.000    0.000    0.000    0.000 sre_parse.py:828(fix_flags)
        2    0.000    0.000    0.000    0.000 sre_parse.py:84(opengroup)
        1    0.000    0.000    0.000    0.000 sre_parse.py:844(parse)
        2    0.000    0.000    0.000    0.000 sre_parse.py:96(closegroup)
        1    0.017    0.017    1.948    1.948 tester.py:8(test)
    27130    0.049    0.000    0.060    0.000 types.py:135(__get__)
        1    0.000    0.000    0.000    0.000 {built-in method _sre.compile}
    65229    0.039    0.000    0.039    0.000 {built-in method builtins.chr}
        1    0.000    0.000    1.948    1.948 {built-in method builtins.exec}
       35    0.000    0.000    0.000    0.000 {built-in method builtins.isinstance}
341895/341891    0.084    0.000    0.084    0.000 {built-in method
builtins.len}
       14    0.000    0.000    0.000    0.000 {built-in method builtins.min}
   581900    0.131    0.000    0.131    0.000 {built-in method builtins.ord}
       65    0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler'
objects}
        1    0.000    0.000    0.000    0.000 {method 'extend' of 'list' objects}
        6    0.000    0.000    0.000    0.000 {method 'find' of 'bytearray' objects}
        2    0.000    0.000    0.000    0.000 {method 'get' of 'dict' objects}
        1    0.000    0.000    0.000    0.000 {method 'items' of 'dict' objects}
     1000    0.002    0.000    0.002    0.000 {method 'join' of 'str' objects}
    30000    0.009    0.000    0.009    0.000 {method 'keys' of 'dict' objects}
     1000    0.008    0.000    0.008    0.000 {method 'search' of
'_sre.SRE_Pattern' objects}
     1000    0.003    0.000    0.003    0.000 {method 'split' of 'str' objects}
```