

Unit Testing

Table of Contents

1	GitHub Url.....	2
2	Functions.....	2
3	Test Cases	2
4	Test Coverage	3
5	Mutation Testing.....	4
6	Appendix	5
6.1	MutPy results.....	5

Table of Figures

Figure 1 - Passed unit tests.....	3
Figure 2 - 100% Statement Coverage.....	3
Figure 3 - 99% Branch Coverage	3
Figure 4 - Python version	4
Figure 5 - Mutant #48.....	4
Figure 6 - Manual mutation	5

1 GitHub Url

The code can be found in the following repository:

https://github.com/fluencyk/SSW-567_Final-Project

2 Functions

- `scan()`: This is an empty function serving as a placeholder for a hardware implementation
- `load_data_from_database()`: This is an empty function serving as a placeholder for a database connection
- `decode(encoded_document_string)`: This function takes the encoded document string and identifies the fields of the document. It returns all fields with their corresponding values in a json format.
- `encode(document)`: This function takes a document object containing all fields and values. It then encodes the fields and adds check sums. It returns the first and second lines in the machine-readable format as a compound string.
- `control_check_digits(encoded_document_string)`: This function takes a machine-readable string, identifies the check digits, and checks them against the field values. It returns a string indicating whether the tests have passed or if an error occurred in natural language. In case of an error, it specifies the effected fieldname.
- `get_check_digit(encoded_string)`: This is a helper function that was created with the intention of keeping the code readable. It takes individual strings and calculates the corresponding check sum. It is used in the `control_check_digits` and the `encode` function to get the checksums of individual fields.

3 Test Cases

For the test cases, the two stand-in functions for the scanning device and the database were implemented as mocks. Both empty functions were also included in the tests for complete code coverage.

The following test cases were written for the main functions:

- `test_decode()`: This function tests if a machine-readable string is correctly transformed into a human-readable document. It compares the input string with the expected document output.
- `test_encode()`: This function tests if a document is correctly translated into a machine-readable string. It compares a document input with the expected string output.
- `test_control_digits_valid()`: This function tests if checksum digits are correctly identified and calculated. It compares a valid input with the expected passed output.

- test_control_digits_invalid_*(): Four of these functions test invalid checksums. One function was implemented for each document field that has a checksum. It compares the expected Error message and checks if the right field is detected.
- test_get_check_digit*(): Three functions were implemented to test the helper function with different input types: uppercase + numbers, lowercase + numbers, uppercase + numbers + symbols

All test cases passed without any failures:

```

carmencouzyn@Carmens-iMac SSW_567 % python3 -m unittest MTTDtest.py
.....
-----
Ran 12 tests in 0.004s

OK

```

Figure 1 - Passed unit tests

4 Test Coverage

The coverage test shows a statement coverage of 100% and a branch coverage of 99%.

```

carmencouzyn@Carmens-iMac SSW_567 % coverage report -m
Name          Stmts  Miss  Cover   Missing
-----
MRTD.py        58      0   100%
MTTDtest.py    76      0   100%
-----
TOTAL          134      0   100%

```

Figure 2 - 100% Statement Coverage

```

carmencouzyn@Carmens-iMac SSW_567 % coverage report -m
Name          Stmts  Miss Branch BrPart  Cover   Missing
-----
MRTD.py        58      0     16      1    99%   159->163
MTTDtest.py    76      0      2      1    99%   211->exit
-----
TOTAL          134      0     18      2    99%

```

Figure 3 - 99% Branch Coverage

5 Mutation Testing

Once almost full test coverage was achieved and all unit tests passed, mutation testing was implemented using MutPy. According to the documentation, MutPy supports Python 3.3 to 3.7. Figure 4 shows the command version query showing the active python version 3.7.15.

```
● (base) carmencouzyn@Carmens-iMac SSW_567 % python3 --version
Python 3.7.15
```

Figure 4 - Python version

Running the mutation tests with mut.py on version 3.7.15 generates 48 mutants. However, the results show that all mutations survive. However, making the changes manually, results in the test cases failing. The full console output can be viewed in the appendix.

To prove that the extreme negative results of the mutation tests are false and do not have any implications on the validity and effectiveness of the test cases, the following mutant will be manually created and killed. Figure 5 shows mutant number 48, which changes line 57 of the decode function.

```
- [# 48] SIR MRTD:
-----
53:         'country_code': string2[10:13], \
54:         'birth_date': string2[13:19], \
55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
- 57:         'personal_number': string2[28:37]}}
+ 57:         'personal_number': string2[28:]}}
58:
59:
60:         return json.dumps(return_str)
61:
-----
[0.16780 s] survived
```

Figure 5 - Mutant #48

Figure 6 displays the results of applying these changes manually. It shows that three test cases failed. Namely, two test cases for the check sum control function as well as the test case of the decode function. This manual mutant proves the written test case to be valid.

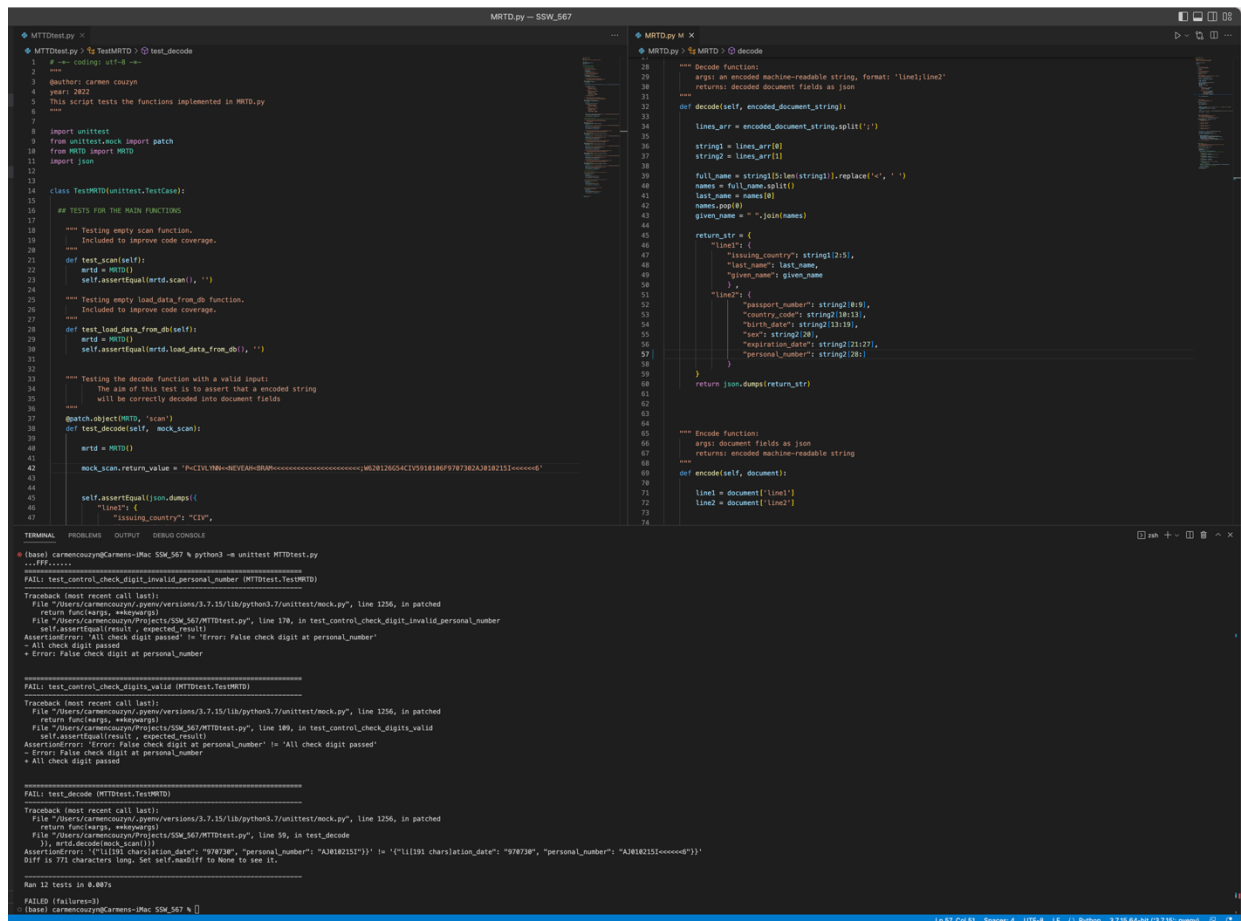


Figure 6 - Manual mutation

6 Appendix

6.1 MutPy results

```
carmencouzyn@Carmens-iMac SSW_567 % mut.py --target MRTD.py --unit-test MTTDtest.py
```

```
-m --color
[*] Start mutation process:
  - targets: MRTD.py
  - tests: MTTDtest.py
[*] 12 tests passed:
  - MTTDtest [0.00424 s]
[*] Start mutants generation and execution:
  - [# 1] AOR MRTD:

72:         line2 = document['line2']
73:
74:
75:
- 76:         string1 = (((('P<' + line2['country_code']) + line1['last_name']) +
+ '<<') + line1['given_name'].replace(' ', '<'))
+ 76:         string1 = (((('P<' - line2['country_code']) + line1['last_name']) +
+ '<<') + line1['given_name'].replace(' ', '<'))
```

```

77:         total_string_length = 44
78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
-----
[0.17279 s] survived
- [# 2] AOR MRTD:
-----

72:         line2 = document['line2']
73:
74:
75:
- 76:         string1 = (((('P<' + line2['country_code']) + line1['last_name']) +
+ 76:         string1 = (((('P<' + line2['country_code']) - line1['last_name']) +
'<<') + line1['given_name'].replace(' ', '<'))
'<<') + line1['given_name'].replace(' ', '<'))
77:         total_string_length = 44
78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
-----

[0.16964 s] survived
- [# 3] AOR MRTD:
-----

72:         line2 = document['line2']
73:
74:
75:
- 76:         string1 = (((('P<' + line2['country_code']) + line1['last_name']) +
+ 76:         string1 = (((('P<' + line2['country_code']) + line1['last_name']) -
'<<') + line1['given_name'].replace(' ', '<'))
'<<') + line1['given_name'].replace(' ', '<'))
77:         total_string_length = 44
78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
-----

[0.16968 s] survived
- [# 4] AOR MRTD:
-----

72:         line2 = document['line2']
73:
74:
75:
- 76:         string1 = (((('P<' + line2['country_code']) + line1['last_name']) +
+ 76:         string1 = (((('P<' + line2['country_code']) + line1['last_name']) +
'<<') + line1['given_name'].replace(' ', '<'))
'<<') - line1['given_name'].replace(' ', '<'))
77:         total_string_length = 44
78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'

```

```

80:
-----
[0.17075 s] survived
- [# 5] AOR MRTD:
-----

74:
75:
76:         string1 = (((('P<' + line2['country_code']) + line1['last_name']) +
'<<') + line1['given_name'].replace(' ', '<'))
77:         total_string_length = 44
- 78:         for x in range(total_string_length - len(string1)):
+ 78:         for x in range(total_string_length + len(string1)):
79:             string1 += '<'
80:
81:
82:         string2 = (((((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number'])))
-----

[0.17429 s] survived
- [# 6] AOR MRTD:
-----

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:
- 82:         string2 = (((((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number'])))
+ 82:         string2 = (((((((((line2['passport_number'] -
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number'])))
83:
84:         return (string1 + ';' + string2
85:
86:
-----

[0.17116 s] survived
- [# 7] AOR MRTD:
-----

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'

```



```

80:
81:
- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number'])))
+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) - line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number'])))
83:
84:         return (string1 + ';') + string2
85:
86:

```

```

[0.16722 s] survived

```

```

- [# 8] AOR MRTD:

```

```

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:
- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number'])))
+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) -
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number'])))
83:
84:         return (string1 + ';') + string2
85:
86:

```

```

[0.16950 s] survived

```

```

- [# 9] AOR MRTD:

```

```

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:

```

```

- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) - str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
83:
84:         return (string1 + ';' ) + string2
85:
86:

```

[0.17122 s] survived

- [# 10] AOR MRTD:

```

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:
- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) -
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
83:
84:         return (string1 + ';' ) + string2
85:
86:

```

[0.17253 s] survived

- [# 11] AOR MRTD:

```

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:
- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +

```

```

line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date'])) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number'])) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date'])) +
line2['sex']) - line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date'])) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
83:
84:         return (string1 + ';') + string2
85:
86:

```

```

[0.16834 s] survived
- [# 12] AOR MRTD:

```

```

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:
- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number'])) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date'])) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date'])) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number'])) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date'])) +
line2['sex']) + line2['expiration_date']) -
str(self.get_check_digit(line2['expiration_date'])) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
83:
84:         return (string1 + ';') + string2
85:
86:

```

```

[0.16947 s] survived
- [# 13] AOR MRTD:

```

```

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:
- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number'])) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date'])) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date'])) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))

```

```

+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) - line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
83:
84:         return (string1 + ';') + string2
85:
86:

```

```

[0.16605 s] survived
- [# 14] AOR MRTD:

```

```

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:
- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) -
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
83:
84:         return (string1 + ';') + string2
85:
86:

```

```

[0.17212 s] survived
- [# 15] AOR MRTD:

```

```

78:         for x in range(total_string_length - len(string1)):
79:             string1 += '<'
80:
81:
- 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
+ 82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +

```

```

line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date'])) + line2['personal_number']) +
'<<<<<<') - str(self.get_check_digit(line2['personal_number']))
83:
84:         return (string1 + ';' ) + string2
85:
86:
-----

[0.18166 s] survived
- [# 16] AOR MRTD:
-----

80:
81:
82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
83:
- 84:         return (string1 + ';' ) + string2
+ 84:         return (string1 - ';' ) + string2
85:
86:
87:
88:         """ Control check digits function:
-----

[0.16874 s] survived
- [# 17] AOR MRTD:
-----

80:
81:
82:         string2 = (((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number']))
83:
- 84:         return (string1 + ';' ) + string2
+ 84:         return (string1 + ';' ) - string2
85:
86:
87:
88:         """ Control check digits function:
-----

[0.17029 s] survived
- [# 18] AOR MRTD:
-----

144:
145:

```

```

146:
147:         if re.search('[a-z]', character):
- 148:             encoded_character = (ord(character) - ord('a')) + 10
+ 148:             encoded_character = (ord(character) + ord('a')) + 10
149:
150:
151:         elif re.search('[A-Z]', character):
152:             encoded_character = (ord(character) - ord('A')) + 10

```

```

[0.17082 s] survived
- [# 19] AOR MRTD:

```

```

144:
145:
146:
147:         if re.search('[a-z]', character):
- 148:             encoded_character = (ord(character) - ord('a')) + 10
+ 148:             encoded_character = (ord(character) - ord('a')) - 10
149:
150:
151:         elif re.search('[A-Z]', character):
152:             encoded_character = (ord(character) - ord('A')) + 10

```

```

[0.16648 s] survived
- [# 20] AOR MRTD:

```

```

148:             encoded_character = (ord(character) - ord('a')) + 10
149:
150:
151:         elif re.search('[A-Z]', character):
- 152:             encoded_character = (ord(character) - ord('A')) + 10
+ 152:             encoded_character = (ord(character) + ord('A')) + 10
153:
154:
155:         elif re.search('[0-9]', character):
156:             encoded_character = int(character)

```

```

[0.16878 s] survived
- [# 21] AOR MRTD:

```

```

148:             encoded_character = (ord(character) - ord('a')) + 10
149:
150:
151:         elif re.search('[A-Z]', character):
- 152:             encoded_character = (ord(character) - ord('A')) + 10
+ 152:             encoded_character = (ord(character) - ord('A')) - 10
153:
154:
155:         elif re.search('[0-9]', character):
156:             encoded_character = int(character)

```

```

[0.17251 s] survived
- [# 22] AOR MRTD:
-----
159:         elif re.search("[$&+,:;=?@#|'<>.-^*()%!]", character):
160:             encoded_character = 0
161:
162:
- 163:         weighted_c = int(encoded_character) * weighting_sequence[index %
len(weighting_sequence)]
+ 163:         weighted_c = int(encoded_character) / weighting_sequence[index %
len(weighting_sequence)]
164:
165:
166:         check_digit += weighted_c
167:
-----

[0.17040 s] survived
- [# 23] AOR MRTD:
-----
159:         elif re.search("[$&+,:;=?@#|'<>.-^*()%!]", character):
160:             encoded_character = 0
161:
162:
- 163:         weighted_c = int(encoded_character) * weighting_sequence[index %
len(weighting_sequence)]
+ 163:         weighted_c = int(encoded_character) // weighting_sequence[index
% len(weighting_sequence)]
164:
165:
166:         check_digit += weighted_c
167:
-----

[0.17244 s] survived
- [# 24] AOR MRTD:
-----
159:         elif re.search("[$&+,:;=?@#|'<>.-^*()%!]", character):
160:             encoded_character = 0
161:
162:
- 163:         weighted_c = int(encoded_character) * weighting_sequence[index %
len(weighting_sequence)]
+ 163:         weighted_c = int(encoded_character) ** weighting_sequence[index
% len(weighting_sequence)]
164:
165:
166:         check_digit += weighted_c
167:
-----

[0.16840 s] survived
- [# 25] AOR MRTD:
-----

```

```

159:         elif re.search("[$&+,;=?@#|'<>.-^*()%!]", character):
160:             encoded_character = 0
161:
162:
- 163:             weighted_c = int(encoded_character) * weighting_sequence[index %
len(weighting_sequence)]
+ 163:             weighted_c = int(encoded_character) * weighting_sequence[index *
len(weighting_sequence)]
164:
165:
166:             check_digit += weighted_c
167:
-----

[0.16595 s] survived
- [# 26] AOR MRTD:
-----

164:
165:
166:             check_digit += weighted_c
167:
- 168:             check_digit = check_digit % 10
+ 168:             check_digit = check_digit * 10
169:             return check_digit
-----

[0.16576 s] survived
- [# 27] ASR MRTD:
-----

75:
76:             string1 = (((('P<' + line2['country_code']) + line1['last_name']) +
'<<') + line1['given_name'].replace(' ', '<'))
77:             total_string_length = 44
78:             for x in range(total_string_length - len(string1)):
- 79:                 string1 += '<'
+ 79:                 string1 -= '<'
80:
81:
82:             string2 = ((((((((((line2['passport_number'] +
str(self.get_check_digit(line2['passport_number']))) + line2['country_code']) +
line2['birth_date']) + str(self.get_check_digit(line2['birth_date']))) +
line2['sex']) + line2['expiration_date']) +
str(self.get_check_digit(line2['expiration_date']))) + line2['personal_number']) +
'<<<<<<') + str(self.get_check_digit(line2['personal_number'])))
83:
-----

[0.16649 s] survived
- [# 28] ASR MRTD:
-----

162:
163:             weighted_c = int(encoded_character) * weighting_sequence[index %
len(weighting_sequence)]
164:

```



```

165:
- 166:         check_digit += weighted_c
+ 166:         check_digit -= weighted_c
167:
168:         check_digit = check_digit % 10
169:         return check_digit
-----

[0.16733 s] survived
- [# 29] COI MRTD:
-----

120:         document_field_value = line2_decoded[field['field_name']]
121:         calculated_check_digit =
self.get_check_digit(document_field_value)
122:         index = field['check_digit_index']
123:         document_check_digit = line2_encoded[index]
- 124:         if document_check_digit != str(calculated_check_digit):
+ 124:         if not (document_check_digit != str(calculated_check_digit)):
125:             return 'Error: False check digit at
{}'.format(field['field_name'])
126:
127:         return 'All check digit passed'
128:
-----

[0.17148 s] survived
- [# 30] COI MRTD:
-----

143:         for (index, character) in enumerate(encoded_string):
144:
145:
146:
- 147:         if re.search('[a-z]', character):
+ 147:         if not (re.search('[a-z]', character)):
148:             encoded_character = (ord(character) - ord('a')) + 10
149:
150:
151:         elif re.search('[A-Z]', character):
-----

[0.16851 s] survived
- [# 31] COI MRTD:
-----

147:         if re.search('[a-z]', character):
148:             encoded_character = (ord(character) - ord('a')) + 10
149:
150:
- 151:         elif re.search('[A-Z]', character):
+ 151:         elif not (re.search('[A-Z]', character)):
152:             encoded_character = (ord(character) - ord('A')) + 10
153:
154:
155:         elif re.search('[0-9]', character):
-----

```

```

[0.17351 s] survived
- [# 32] COI MRTD:
-----
151:         elif re.search('[A-Z]', character):
152:             encoded_character = (ord(character) - ord('A')) + 10
153:
154:
- 155:         elif re.search('[0-9]', character):
+ 155:         elif not (re.search('[0-9]', character)):
156:             encoded_character = int(character)
157:
158:
159:         elif re.search("[$&+,;=?@#|'<>.-^*()%!]", character):
-----

[0.16873 s] survived
- [# 33] COI MRTD:
-----
155:         elif re.search('[0-9]', character):
156:             encoded_character = int(character)
157:
158:
- 159:         elif re.search("[$&+,;=?@#|'<>.-^*()%!]", character):
+ 159:         elif not (re.search("[$&+,;=?@#|'<>.-^*()%!]", character)):
160:             encoded_character = 0
161:
162:
163:         weighted_c = int(encoded_character) * weighting_sequence[index %
len(weighting_sequence)]
-----

[0.16835 s] survived
- [# 34] ROR MRTD:
-----
120:         document_field_value = line2_decoded[field['field_name']]
121:         calculated_check_digit =
self.get_check_digit(document_field_value)
122:         index = field['check_digit_index']
123:         document_check_digit = line2_encoded[index]
- 124:         if document_check_digit != str(calculated_check_digit):
+ 124:         if document_check_digit == str(calculated_check_digit):
125:             return 'Error: False check digit at
{}'.format(field['field_name'])
126:
127:         return 'All check digit passed'
128:
-----

[0.18243 s] survived
- [# 35] SIR MRTD:
-----
35:
36:         string1 = lines_arr[0]
37:         string2 = lines_arr[1]

```

```

38:
- 39:     full_name = string1[5:len(string1)].replace('<', ' ')
+ 39:     full_name = string1[:len(string1)].replace('<', ' ')
40:     names = full_name.split()
41:     last_name = names[0]
42:     names.pop(0)
43:     given_name = ' '.join(names)

```

```

[0.17050 s] survived
- [# 36] SIR MRTD:

```

```

35:
36:     string1 = lines_arr[0]
37:     string2 = lines_arr[1]
38:
- 39:     full_name = string1[5:len(string1)].replace('<', ' ')
+ 39:     full_name = string1[5:].replace('<', ' ')
40:     names = full_name.split()
41:     last_name = names[0]
42:     names.pop(0)
43:     given_name = ' '.join(names)

```

```

[0.16748 s] survived
- [# 37] SIR MRTD:

```

```

43:     given_name = ' '.join(names)
44:
45:     return_str = {\
46:         'line1': {\
- 47:         'issuing_country': string1[2:5], \
+ 47:         'issuing_country': string1[:5], \
48:         'last_name': last_name, \
49:         'given_name': given_name}, \
50:     \
51:     'line2': {\

```

```

[0.16820 s] survived
- [# 38] SIR MRTD:

```

```

43:     given_name = ' '.join(names)
44:
45:     return_str = {\
46:         'line1': {\
- 47:         'issuing_country': string1[2:5], \
+ 47:         'issuing_country': string1[2:], \
48:         'last_name': last_name, \
49:         'given_name': given_name}, \
50:     \
51:     'line2': {\

```

```

[0.17370 s] survived

```

```

- [# 39] SIR MRTD:
-----
48:         'last_name': last_name, \
49:         'given_name': given_name}, \
50:         \
51:         'line2': {\
- 52:         'passport_number': string2[0:9], \
+ 52:         'passport_number': string2[:9], \
53:         'country_code': string2[10:13], \
54:         'birth_date': string2[13:19], \
55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
-----

[0.16806 s] survived
- [# 40] SIR MRTD:
-----
48:         'last_name': last_name, \
49:         'given_name': given_name}, \
50:         \
51:         'line2': {\
- 52:         'passport_number': string2[0:9], \
+ 52:         'passport_number': string2[0:], \
53:         'country_code': string2[10:13], \
54:         'birth_date': string2[13:19], \
55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
-----

[0.17015 s] survived
- [# 41] SIR MRTD:
-----
49:         'given_name': given_name}, \
50:         \
51:         'line2': {\
52:         'passport_number': string2[0:9], \
- 53:         'country_code': string2[10:13], \
+ 53:         'country_code': string2[:13], \
54:         'birth_date': string2[13:19], \
55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
57:         'personal_number': string2[28:37]}}
-----

[0.16667 s] survived
- [# 42] SIR MRTD:
-----
49:         'given_name': given_name}, \
50:         \
51:         'line2': {\
52:         'passport_number': string2[0:9], \
- 53:         'country_code': string2[10:13], \
+ 53:         'country_code': string2[10:], \
54:         'birth_date': string2[13:19], \

```

```

55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
57:         'personal_number': string2[28:37]}}

```

```

[0.16517 s] survived
- [# 43] SIR MRTD:

```

```

50:         \
51:         'line2': {\
52:         'passport_number': string2[0:9], \
53:         'country_code': string2[10:13], \
- 54:         'birth_date': string2[13:19], \
+ 54:         'birth_date': string2[:19], \
55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
57:         'personal_number': string2[28:37]}}
58:

```

```

[0.17197 s] survived
- [# 44] SIR MRTD:

```

```

50:         \
51:         'line2': {\
52:         'passport_number': string2[0:9], \
53:         'country_code': string2[10:13], \
- 54:         'birth_date': string2[13:19], \
+ 54:         'birth_date': string2[13:], \
55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
57:         'personal_number': string2[28:37]}}
58:

```

```

[0.16236 s] survived
- [# 45] SIR MRTD:

```

```

52:         'passport_number': string2[0:9], \
53:         'country_code': string2[10:13], \
54:         'birth_date': string2[13:19], \
55:         'sex': string2[20], \
- 56:         'expiration_date': string2[21:27], \
+ 56:         'expiration_date': string2[:27], \
57:         'personal_number': string2[28:37]}}
58:
59:
60:         return json.dumps(return_str)

```

```

[0.16250 s] survived
- [# 46] SIR MRTD:

```

```

52:         'passport_number': string2[0:9], \
53:         'country_code': string2[10:13], \

```

```

54:         'birth_date': string2[13:19], \
55:         'sex': string2[20], \
- 56:         'expiration_date': string2[21:27], \
+ 56:         'expiration_date': string2[21:], \
57:         'personal_number': string2[28:37]}}
58:
59:
60:     return json.dumps(return_str)

```

```

[0.16596 s] survived
- [# 47] SIR MRTD:

```

```

53:         'country_code': string2[10:13], \
54:         'birth_date': string2[13:19], \
55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
- 57:         'personal_number': string2[28:37]}}
+ 57:         'personal_number': string2[:37]}}
58:
59:
60:     return json.dumps(return_str)
61:

```

```

[0.17158 s] survived
- [# 48] SIR MRTD:

```

```

53:         'country_code': string2[10:13], \
54:         'birth_date': string2[13:19], \
55:         'sex': string2[20], \
56:         'expiration_date': string2[21:27], \
- 57:         'personal_number': string2[28:37]}}
+ 57:         'personal_number': string2[28:]}}
58:
59:
60:     return json.dumps(return_str)
61:

```

```

[0.16780 s] survived
[*] Mutation score [11.46690 s]: 0.0%
- all: 48
- killed: 0 (0.0%)
- survived: 48 (100.0%)
- incompetent: 0 (0.0%)
- timeout: 0 (0.0%)

```