# Report for Assignment 1

## Project chosen

Group 36

Name: PyMuPDF

URL: https://github.com/pymupdf/PyMuPDF

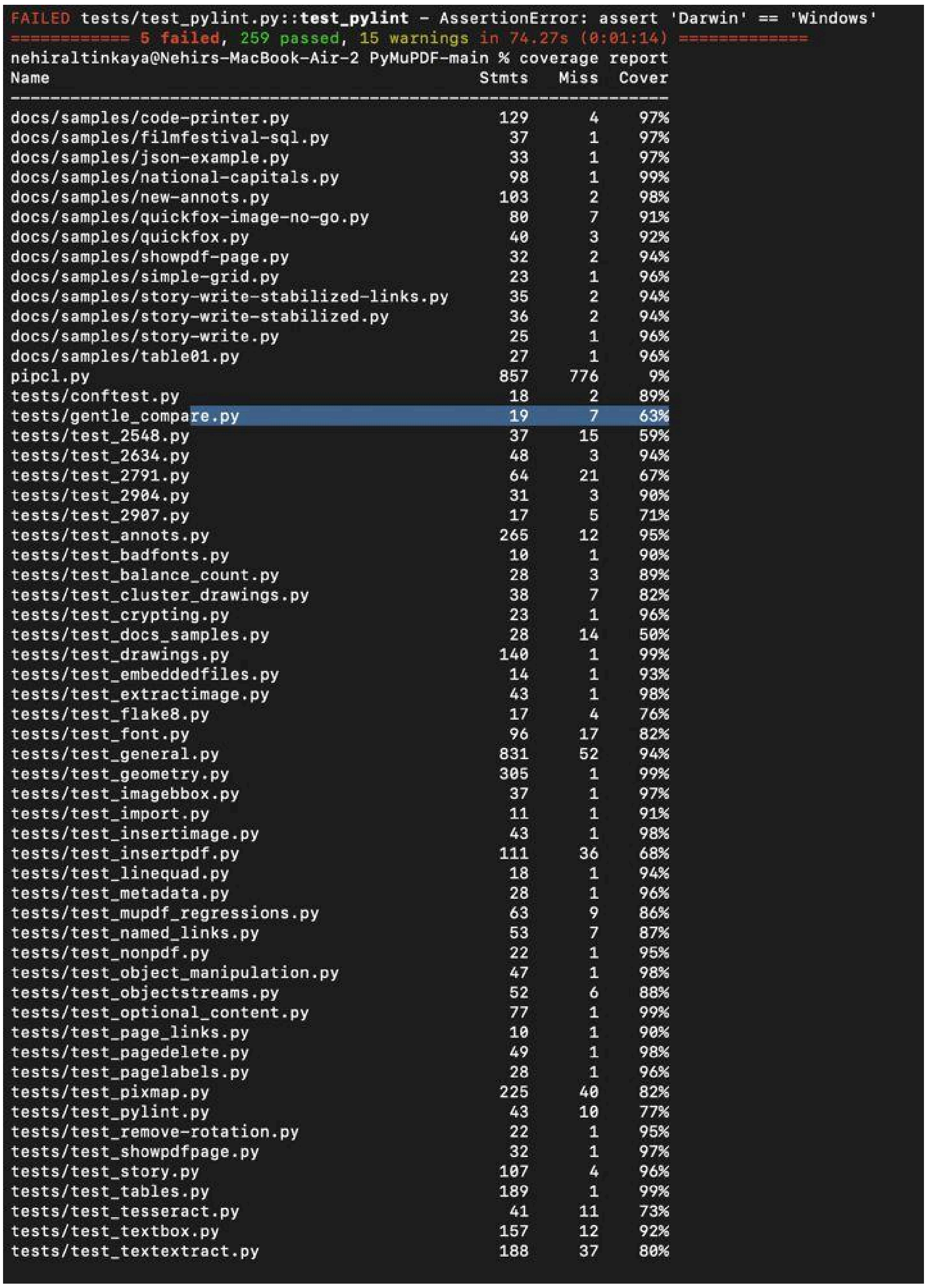
Number of lines of code and the tool used to count it: 41754, Lizard

Programming language: Python

## Coverage measurement

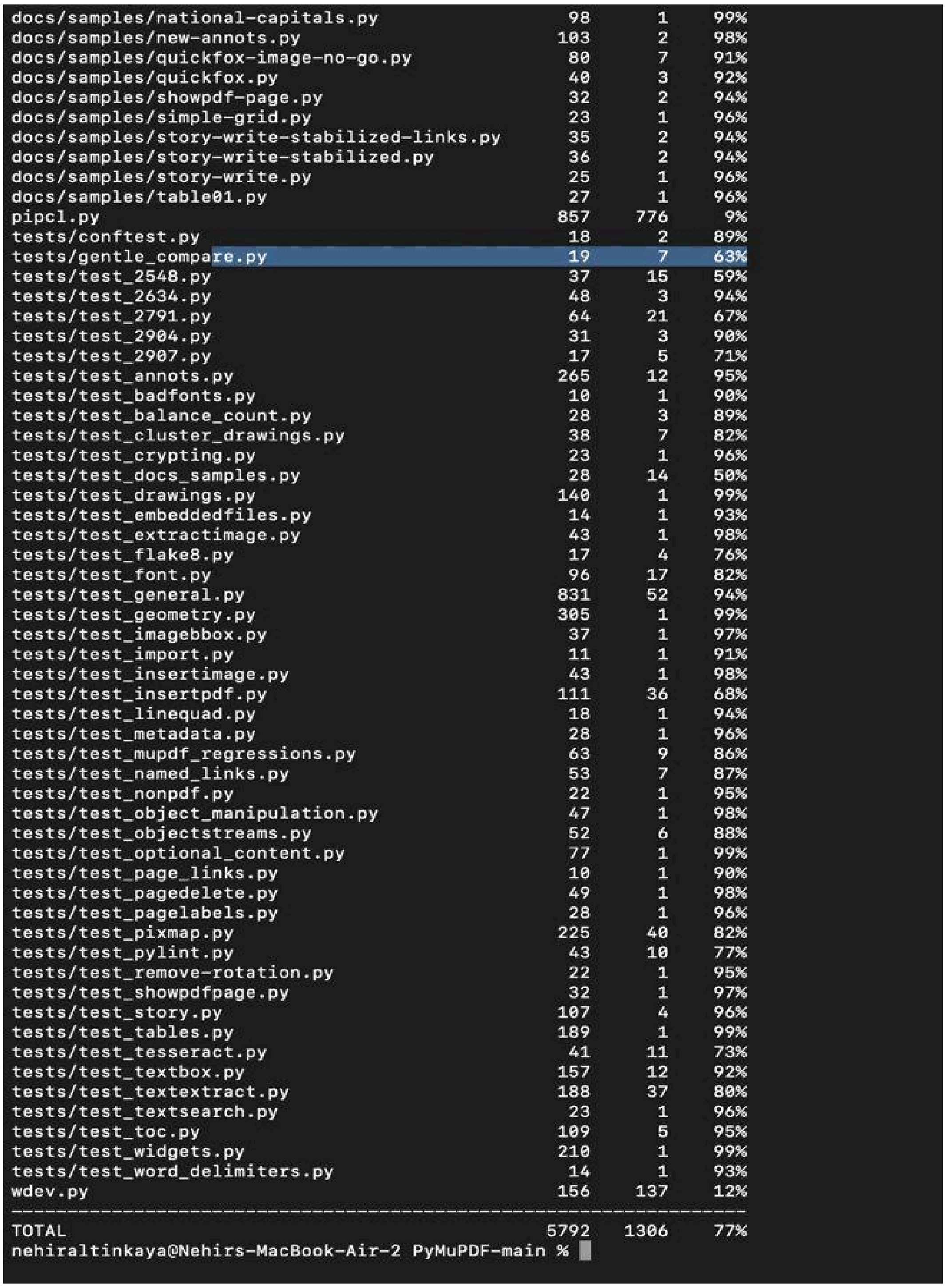
### Existing tool

We have used the tool **Coverage.py** and executed it simply with the command “coverage run -m pytest” followed by “coverage report” and “coverage html” to extract the coverage data from the tests. We have obtained the following data:



```
FAILED tests/test_pylint.py::test_pylint - AssertionError: assert 'Darwin' == 'Windows'
=========== 5 failed, 259 passed, 15 warnings in 74.27s (0:01:14) =============
nehiraltinkaya@Nehirs-MacBook-Air-2 PyMuPDF-main % coverage report
Name                                              Stmts   Miss  Cover
---------------------------------------------------------------------
docs/samples/code-printer.py                        129      4    97%
docs/samples/filmfestival-sql.py                     37      1    97%
docs/samples/json-example.py                         33      1    97%
docs/samples/national-capitals.py                    98      1    99%
docs/samples/new-annots.py                          103      2    98%
docs/samples/quickfox-image-no-go.py                 80      7    91%
docs/samples/quickfox.py                             40      3    92%
docs/samples/showpdf-page.py                         32      2    94%
docs/samples/simple-grid.py                          23      1    96%
docs/samples/story-write-stabilized-links.py         35      2    94%
docs/samples/story-write-stabilized.py               36      2    94%
docs/samples/story-write.py                          25      1    96%
docs/samples/table01.py                              27      1    96%
pipcl.py                                            857    776     9%
tests/conftest.py                                    18      2    89%
tests/gentle_compare.py                              19      7    63%
tests/test_2548.py                                   37     15    59%
tests/test_2634.py                                   48      3    94%
tests/test_2791.py                                   64     21    67%
tests/test_2904.py                                   31      3    90%
tests/test_2907.py                                   17      5    71%
tests/test_annots.py                                265     12    95%
tests/test_badfonts.py                               10      1    90%
tests/test_balance_count.py                          28      3    89%
tests/test_cluster_drawings.py                       38      7    82%
tests/test_crypting.py                               23      1    96%
tests/test_docs_samples.py                           28     14    50%
tests/test_drawings.py                              140      1    99%
tests/test_embeddedfiles.py                          14      1    93%
tests/test_extractimage.py                           43      1    98%
tests/test_flake8.py                                 17      4    76%
tests/test_font.py                                   96     17    82%
tests/test_general.py                               831     52    94%
tests/test_geometry.py                              305      1    99%
tests/test_imagebbox.py                              37      1    97%
tests/test_import.py                                 11      1    91%
tests/test_insertimage.py                            43      1    98%
tests/test_insertpdf.py                             111     36    68%
tests/test_linequad.py                               18      1    94%
tests/test_metadata.py                               28      1    96%
tests/test_mupdf_regressions.py                      63      9    86%
tests/test_named_links.py                            53      7    87%
tests/test_nonpdf.py                                 22      1    95%
tests/test_object_manipulation.py                    47      1    98%
tests/test_objectstreams.py                          52      6    88%
tests/test_optional_content.py                       77      1    99%
tests/test_page_links.py                             10      1    90%
tests/test_pagedelete.py                             49      1    98%
tests/test_pagelabels.py                             28      1    96%
tests/test_pixmap.py                                225     40    82%
tests/test_pylint.py                                 43     10    77%
tests/test_remove-rotation.py                        22      1    95%
tests/test_showpdfpage.py                            32      1    97%
tests/test_story.py                                 107      4    96%
tests/test_tables.py                                189      1    99%
tests/test_tesseract.py                              41     11    73%
tests/test_textbox.py                               157     12    92%
tests/test_textextract.py                           188     37    80%
```

Figure 1: Coverage Result by Existing Tool (first part)



```
docs/samples/national-capitals.py                 98      1    99%
docs/samples/new-annots.py                       103      2    98%
docs/samples/quickfox-image-no-go.py              80      7    91%
docs/samples/quickfox.py                          40      3    92%
docs/samples/showpdf-page.py                      32      2    94%
docs/samples/simple-grid.py                       23      1    96%
docs/samples/story-write-stabilized-links.py      35      2    94%
docs/samples/story-write-stabilized.py            36      2    94%
docs/samples/story-write.py                       25      1    96%
docs/samples/table01.py                           27      1    96%
pipcl.py                                         857    776     9%
tests/conftest.py                                 18      2    89%
tests/gentle_compare.py                           19      7    63%
tests/test_2548.py                                37     15    59%
tests/test_2634.py                                48      3    94%
tests/test_2791.py                                64     21    67%
tests/test_2904.py                                31      3    90%
tests/test_2907.py                                17      5    71%
tests/test_annots.py                             265     12    95%
tests/test_badfonts.py                            10      1    90%
tests/test_balance_count.py                       28      3    89%
tests/test_cluster_drawings.py                    38      7    82%
tests/test_crypting.py                            23      1    96%
tests/test_docs_samples.py                        28     14    50%
tests/test_drawings.py                           140      1    99%
tests/test_embeddedfiles.py                       14      1    93%
tests/test_extractimage.py                        43      1    98%
tests/test_flake8.py                              17      4    76%
tests/test_font.py                                96     17    82%
tests/test_general.py                            831     52    94%
tests/test_geometry.py                           305      1    99%
tests/test_imagebbox.py                           37      1    97%
tests/test_import.py                              11      1    91%
tests/test_insertimage.py                         43      1    98%
tests/test_insertpdf.py                          111     36    68%
tests/test_linequad.py                            18      1    94%
tests/test_metadata.py                            28      1    96%
tests/test_mupdf_regressions.py                   63      9    86%
tests/test_named_links.py                         53      7    87%
tests/test_nonpdf.py                              22      1    95%
tests/test_object_manipulation.py                 47      1    98%
tests/test_objectstreams.py                       52      6    88%
tests/test_optional_content.py                    77      1    99%
tests/test_page_links.py                          10      1    90%
tests/test_pagedelete.py                          49      1    98%
tests/test_pagelabels.py                          28      1    96%
tests/test_pixmap.py                             225     40    82%
tests/test_pylint.py                              43     10    77%
tests/test_remove-rotation.py                     22      1    95%
tests/test_showpdfpage.py                         32      1    97%
tests/test_story.py                              107      4    96%
tests/test_tables.py                             189      1    99%
tests/test_tesseract.py                           41     11    73%
tests/test_textbox.py                            157     12    92%
tests/test_textextract.py                        188     37    80%
tests/test_textsearch.py                          23      1    96%
tests/test_toc.py                                109      5    95%
tests/test_widgets.py                            210      1    99%
tests/test_word_delimiters.py                     14      1    93%
wdev.py                                          156    137    12%
------------------------------------------------------------------
TOTAL                                           5792   1306    77%
nehiraltinkaya@Nehirs-MacBook-Air-2 PyMuPDF-main % 
```

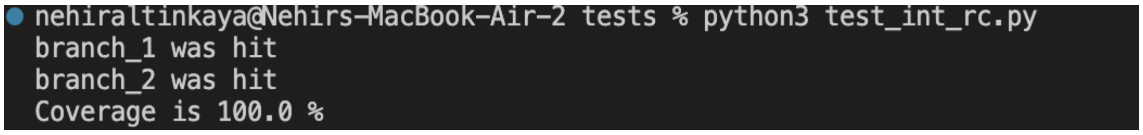
Figure 2: Coverage Result by Existing Tool (second part)

### Your own coverage tool
Nehir Altinkaya:

int_rc:

Link:
https://github.com/misraozoktas/sep_group36_pymupdf/blob/bbe7db8881b4684f3206fbf6afa95852c5f56061/tests/test_int_rc.py

Screenshot of the Coverage Results Output:

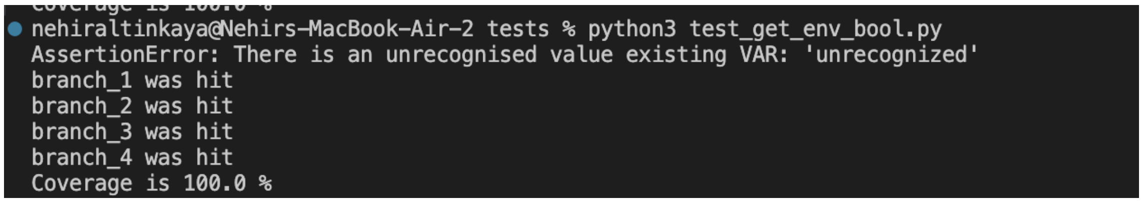


```
nehiraltinkaya@Nehirs-MacBook-Air-2 tests % python3 test_int_rc.py
branch_1 was hit
branch_2 was hit
Coverage is 100.0 %
```

get_env_bool:

Link:
https://github.com/misraozoktas/sep_group36_pymupdf/blob/bbe7db8881b4684f3206fbf6afa95852c5f56061/tests/test_get_env_bool.py

Screenshot of the Coverage Results Output:



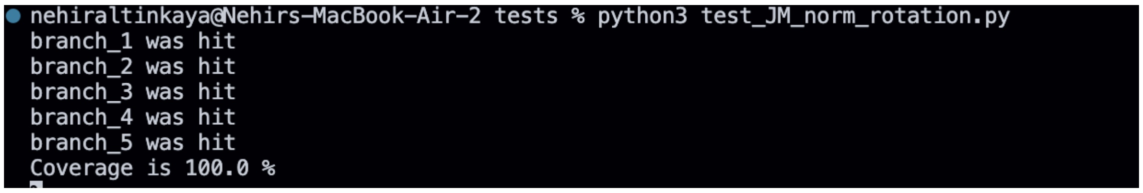
```
nehiraltinkaya@Nehirs-MacBook-Air-2 tests % python3 test_get_env_bool.py
AssertionError: There is an unrecognised value existing VAR: 'unrecognized'
branch_1 was hit
branch_2 was hit
branch_3 was hit
branch_4 was hit
Coverage is 100.0 %
```

Turkan Badalzade:

JM_norm_rotation:

Link:
https://github.com/misraozoktas/sep_group36_pymupdf/blob/1a52e6554580bda141eb44f34f6dc49aa2937e5e/tests/test_JM_norm_rotation.py

Screenshot of the Coverage Results Output:

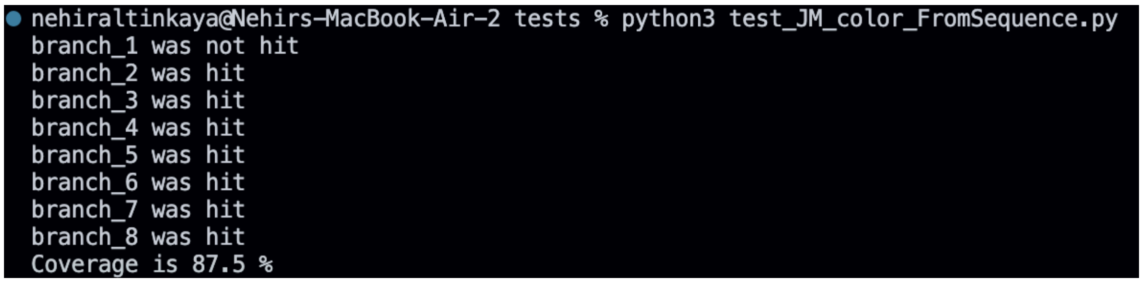


```
nehiraltinkaya@Nehirs-MacBook-Air-2 tests % python3 test_JM_norm_rotation.py
branch_1 was hit
branch_2 was hit
branch_3 was hit
branch_4 was hit
branch_5 was hit
Coverage is 100.0 %
```

JM_color_FromSequence:

Link:
https://github.com/misraozoktas/sep_group36_pymupdf/blob/1a52e6554580bda141eb44f34f6dc49aa2937e5e/tests/test_JM_color_FromSequence.py

Screenshot of the Coverage Results Output:



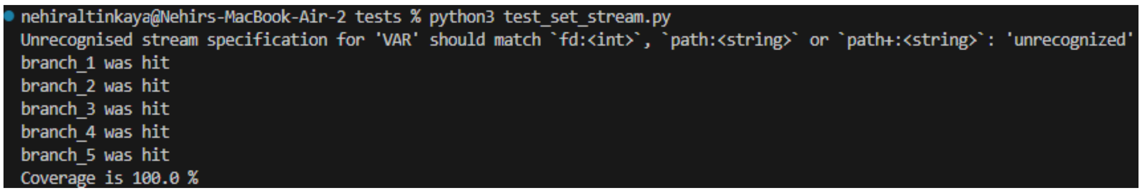
```
nehiraltinkaya@Nehirs-MacBook-Air-2 tests % python3 test_JM_color_FromSequence.py
branch_1 was not hit
branch_2 was hit
branch_3 was hit
branch_4 was hit
branch_5 was hit
branch_6 was hit
branch_7 was hit
branch_8 was hit
Coverage is 87.5 %
```

Erin Elagoz:

set_stream:

Link:
https://github.com/misraozoktas/sep_group36_pymupdf/blob/8476e20fffc47eff0a310b06ac5540e0f0141cd2/tests/test_set_stream.py

Screenshot of the Coverage Results Output:

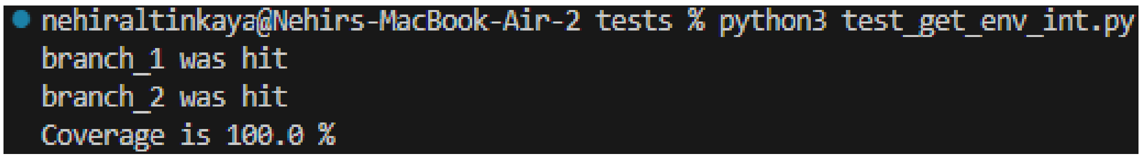


```
nehiraltinkaya@Nehirs-MacBook-Air-2 tests % python3 test_set_stream.py
Unrecognised stream specification for 'VAR' should match `fd:<int>`, `path:<string>` or `path+:<string>`: 'unrecognized'
branch_1 was hit
branch_2 was hit
branch_3 was hit
branch_4 was hit
branch_5 was hit
Coverage is 100.0 %
```

get_env_int:

Link:
https://github.com/misraozoktas/sep_group36_pymupdf/blob/8476e20fffc47eff0a310b06ac5540e0f0141cd2/tests/test_get_env_int.py

Screenshot of the Coverage Results Output:



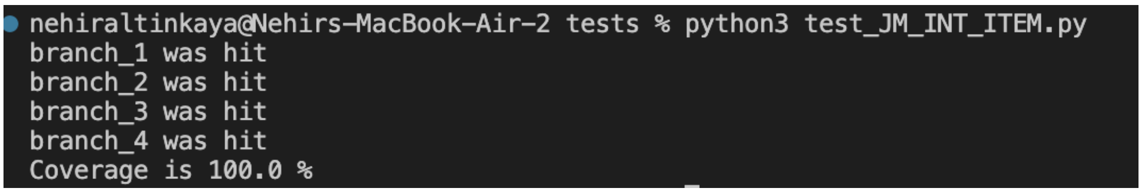
```
nehiraltinkaya@Nehirs-MacBook-Air-2 tests % python3 test_get_env_int.py
branch_1 was hit
branch_2 was hit
Coverage is 100.0 %
```

Misra Ozoktash:

JM_INT_ITEM:

Link:
https://github.com/misraozoktas/sep_group36_pymupdf/blob/30971c7254454080a7a691ff2133209d9edc7d79/tests/test_JM_INT_ITEM.py

Screenshot of the Coverage Results Output:

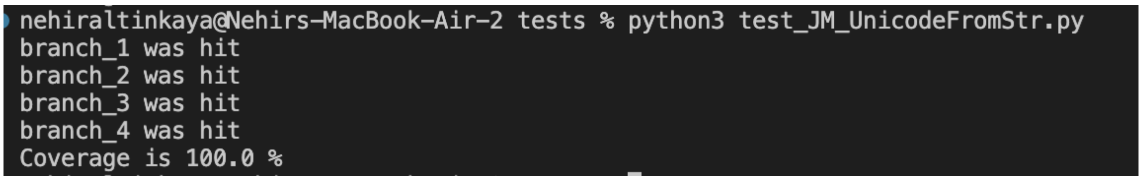


```
nehiraltinkaya@Nehirs-MacBook-Air-2 tests % python3 test_JM_INT_ITEM.py
branch_1 was hit
branch_2 was hit
branch_3 was hit
branch_4 was hit
Coverage is 100.0 %
```

JM_UnicodeFromStr:

Link:
https://github.com/misraozoktas/sep_group36_pymupdf/blob/30971c7254454080a7a691ff2133209d9edc7d79/tests/test_JM_UnicodeFromStr.py

Screenshot of the Coverage Results Output:



```
nehiraltinkaya@Nehirs-MacBook-Air-2 tests % python3 test_JM_UnicodeFromStr.py
branch_1 was hit
branch_2 was hit
branch_3 was hit
branch_4 was hit
Coverage is 100.0 %
```

## Coverage improvement

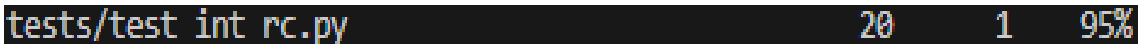
### Individual tests

Nehir Altinkaya:

int_rc:

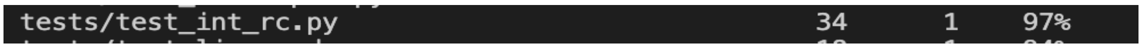
Link: https://github.com/misraozoktas/sep_group36_pymupdf/commit/32b4a9c23be4f3435234e569a1a8208ec1d66fd0

Screenshot of the Old Coverage Results:



```
tests/test int rc.py                                20       1     95%
```

Screenshot of the New Coverage Results:



```
tests/test_int_rc.py                                34       1     97%
```

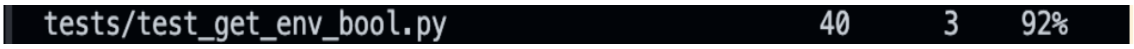
Reason for Coverage Improvement:

The coverage has improved by 2% as can also be seen from the images above. The coverage has improved when functions were made for test cases and new test cases were added as functions for covering different branches. For instance, when the case “is not positive or zero”, an improvement has been observed.

get_env_bool:

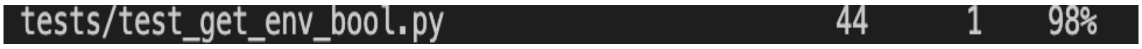
Link: https://github.com/misraozoktas/sep_group36_pymupdf/commit/32b4a9c23be4f3435234e569a1a8208ec1d66fd0

Screenshot of the Old Coverage Results:



```
tests/test_get_env_bool.py                          40       3     92%
```

Screenshot of the New Coverage Results:



```
tests/test_get_env_bool.py                          44       1     98%
```

Reason for Coverage Improvement:

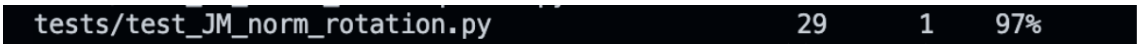
The coverage has improved by 6% which can also be seen from the images above. An improvement was observed in the coverage when functions were made for test cases and new test cases were added to the test file, as functions, for covering different branches. For example, there were improvements in the coverage when the test cases for the branches returning true, returning false, returning default and returning assertion were added individually .

Turkan Badalzade:

test_JM_norm_rotation:

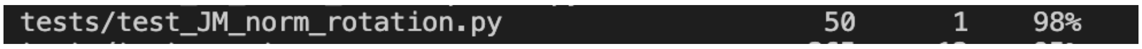
Link: https://github.com/misraozoktas/sep_group36_pymupdf/commit/d089b154783fa50ce27508faf8d5260c8508b9cb

Screenshot of the Old Coverage Results:



```
tests/test_JM_norm_rotation.py                      29       1     97%
```

Screenshot of the New Coverage Results:



```
tests/test_JM_norm_rotation.py                      50       1     98%
```

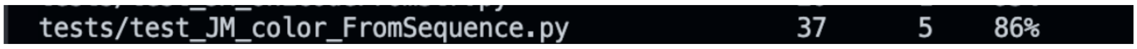
Reason for Coverage Improvement:

The coverage result has improved by 1%. Even though it was already a high result, I was able to upgrade the code by adding new test cases. When I first implemented the test cases I did not give much attention to negative number cases. Therefore, to be able to increase my coverage result I added negative number cases. In addition to that I added a function that the number is below 0 and also not divisible by 90 when I added 360 to the negative number. So, I tried to hit every branch with my new test cases.

test_JM_color_FromSequence:

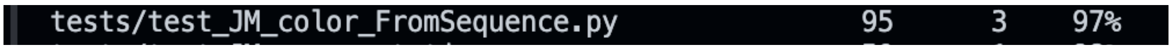
Link: https://github.com/misraozoktas/sep_group36_pymupdf/commit/d089b154783fa50ce27508faf8d5260c8508b9cb

Screenshot of the Old Coverage Results:



```
tests/test_JM_color_FromSequence.py                 37       5     86%
```

Screenshot of the New Coverage Results:



```
tests/test_JM_color_FromSequence.py                 95       3     97%
```

Reason for Coverage Improvement:

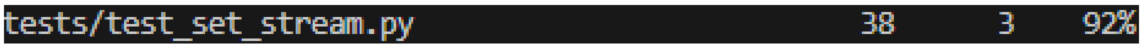
Firstly, the coverage result was 86% when I first ran the test function. While I was trying to improve the result I noticed that the test cases do not hit branch 2. Therefore, I had to add a new test case for branch 2. When I created the test case for branch 2, I observed a significant raise to 95%. Although I added a lot of test cases for this test function, I could increase it to 97%. Overall, the coverage result was increased by 11%.

Erin Elagoz:

set_stream:

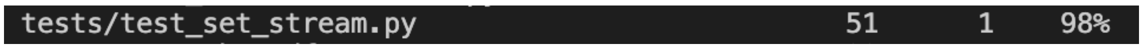
Link:
https://github.com/misraozoktas/sep_group36_pymupdf/commit/aea7bb3128d968662b2deaf6db96459ee153aab2

Screenshot of the Old Coverage Results:



```
tests/test_set_stream.py                38         3     92%
```

Screenshot of the New Coverage Results:



```
tests/test_set_stream.py                51         1     98%
```

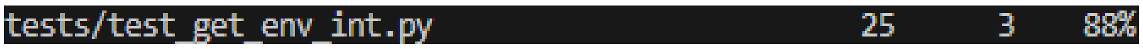
Reason for Coverage Improvement:

The coverage results were increased by %6 after making functions for each test and calling them. All the if cases were being executed except the first one which was "If t is None". After adding the last test case where t has the value "none", the first case is also executed and the coverage further increased.

get_env_int:

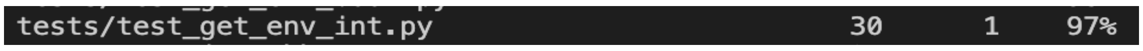
Link:
https://github.com/misraozoktas/sep_group36_pymupdf/commit/aea7bb3128d968662b2deaf6db96459ee153aab2

Screenshot of the Old Coverage Results:



```
tests/test_get_env_int.py               25         3     88%
```

Screenshot of the New Coverage Results:



```
tests/test_get_env_int.py               30         1     97%
```

Reason for Coverage Improvement:

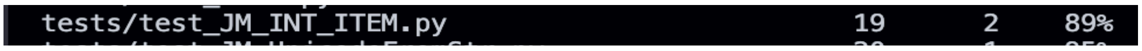
The coverage results were also increased by %9 in this test after writing separate functions for each test and calling them, which was the main reason. Additionally, I have added a test case to further improve the coverage result. I have changed the variable so that the returned integer is different, making sure all the if cases have been executed.

Misra Ozoktash:

JM_INT_ITEM:

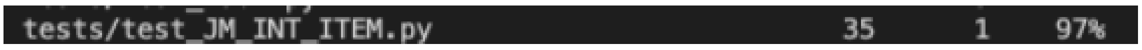
Link:
https://github.com/misraozoktas/sep_group36_pymupdf/commit/4a3da66b36ba65331044bdedd36796cc83f19353

Screenshot of the Old Coverage Results:



```
tests/test_JM_INT_ITEM.py                      19      2    89%
```

Screenshot of the New Coverage Results:



```
tests/test_JM_INT_ITEM.py                      35      1    97%
```

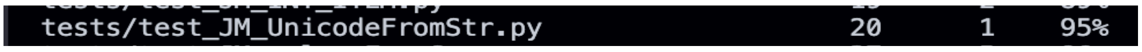
Reason for Coverage Improvement:

As it can be seen from the screenshots above, there has been a 8% increase on the coverage test result from 89% to 97%. This is a result of adding a number of functions of different test cases that are for covering different branches. I have made sure to add at least one test case for each branch. For example, there was an improvement in the coverage when a function was added to check the test case for an empty object.

JM_UnicodeFromStr:

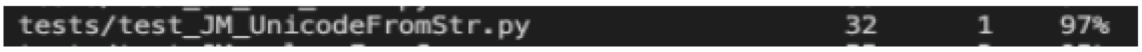
Link:
https://github.com/misraozoktas/sep_group36_pymupdf/commit/4a3da66b36ba65331044bdedd36796cc83f19353

Screenshot of the Old Coverage Results:



```
tests/test_JM_UnicodeFromStr.py                20      1    95%
```

Screenshot of the New Coverage Results:



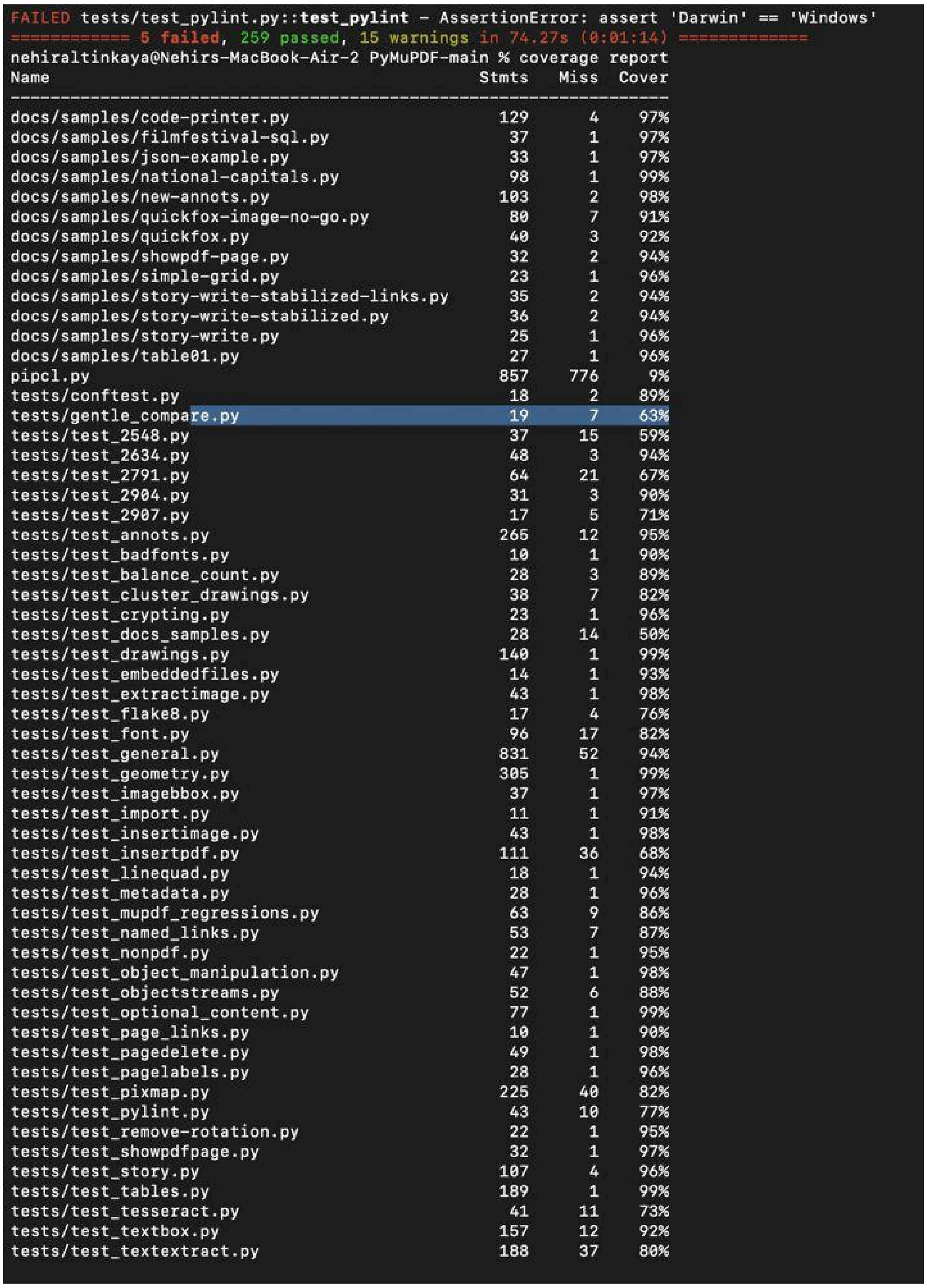
```
tests/test_JM_UnicodeFromStr.py                32      1    97%
```

Reason for Coverage Improvement:

An improvement can be seen from the screenshots above that the percentage result of the coverage test increased from 95% to 97%, a 2% increase. Even though the test was already a high result, with the improvements made to the code, a percentage rise is achieved. To achieve these results, I have added different functions that have at least one test case for each branch, making sure that all branches have been covered. For example, a test case has been written for a byte that will return a result of the decoded version of the byte. This has helped make an improvement in the coverage test.

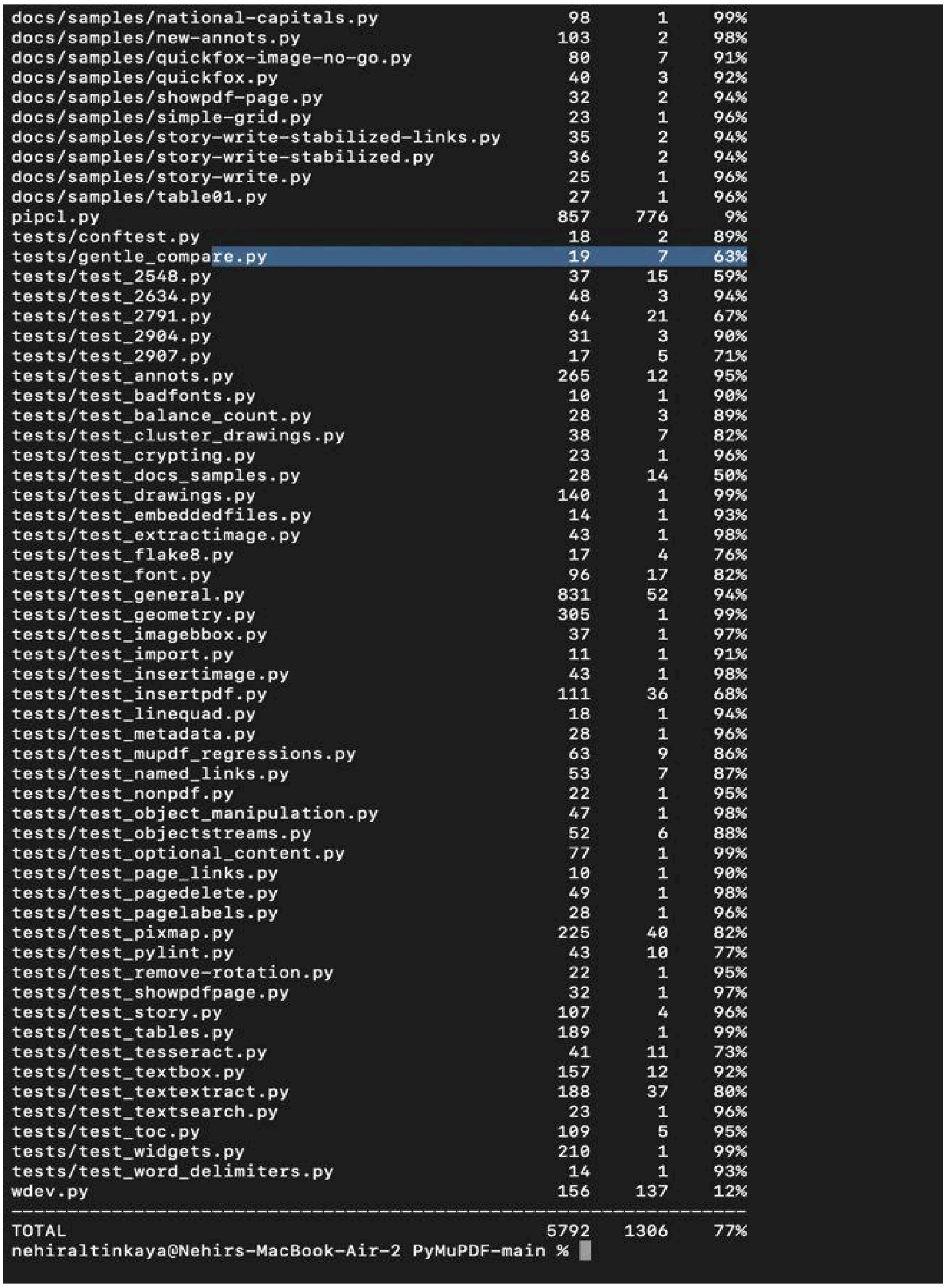
### Overall

Here is the screenshot of the old coverage results by running an existing tool:



```
FAILED tests/test_pylint.py::test_pylint - AssertionError: assert 'Darwin' == 'Windows'
============ 5 failed, 259 passed, 15 warnings in 74.27s (0:01:14) =============
nehiraltinkaya@Nehirs-MacBook-Air-2 PyMuPDF-main % coverage report
Name                                               Stmts   Miss  Cover
-------------------------------------------------------------------
docs/samples/code-printer.py                         129      4    97%
docs/samples/filmfestival-sql.py                      37      1    97%
docs/samples/json-example.py                          33      1    97%
docs/samples/national-capitals.py                     98      1    99%
docs/samples/new-annots.py                           103      2    98%
docs/samples/quickfox-image-no-go.py                  80      7    91%
docs/samples/quickfox.py                              40      3    92%
docs/samples/showpdf-page.py                          32      2    94%
docs/samples/simple-grid.py                           23      1    96%
docs/samples/story-write-stabilized-links.py          35      2    94%
docs/samples/story-write-stabilized.py                36      2    94%
docs/samples/story-write.py                           25      1    96%
docs/samples/table01.py                               27      1    96%
pipcl.py                                             857    776     9%
tests/conftest.py                                     18      2    89%
tests/gentle_compare.py                               19      7    63%
tests/test_2548.py                                    37     15    59%
tests/test_2634.py                                    48      3    94%
tests/test_2791.py                                    64     21    67%
tests/test_2904.py                                    31      3    90%
tests/test_2907.py                                    17      5    71%
tests/test_annots.py                                 265     12    95%
tests/test_badfonts.py                                10      1    90%
tests/test_balance_count.py                           28      3    89%
tests/test_cluster_drawings.py                        38      7    82%
tests/test_crypting.py                                23      1    96%
tests/test_docs_samples.py                            28     14    50%
tests/test_drawings.py                               140      1    99%
tests/test_embeddedfiles.py                           14      1    93%
tests/test_extractimage.py                            43      1    98%
tests/test_flake8.py                                  17      4    76%
tests/test_font.py                                    96     17    82%
tests/test_general.py                                831     52    94%
tests/test_geometry.py                               305      1    99%
tests/test_imagebbox.py                               37      1    97%
tests/test_import.py                                  11      1    91%
tests/test_insertimage.py                             43      1    98%
tests/test_insertpdf.py                              111     36    68%
tests/test_linequad.py                                18      1    94%
tests/test_metadata.py                                28      1    96%
tests/test_mupdf_regressions.py                       63      9    86%
tests/test_named_links.py                             53      7    87%
tests/test_nonpdf.py                                  22      1    95%
tests/test_object_manipulation.py                     47      1    98%
tests/test_objectstreams.py                           52      6    88%
tests/test_optional_content.py                       77      1    99%
tests/test_page_links.py                              10      1    90%
tests/test_pagedelete.py                              49      1    98%
tests/test_pagelabels.py                              28      1    96%
tests/test_pixmap.py                                 225     40    82%
tests/test_pylint.py                                  43     10    77%
tests/test_remove-rotation.py                         22      1    95%
tests/test_showpdfpage.py                             32      1    97%
tests/test_story.py                                  107      4    96%
tests/test_tables.py                                 189      1    99%
tests/test_tesseract.py                               41     11    73%
tests/test_textbox.py                                157     12    92%
tests/test_textextract.py                            188     37    80%
```

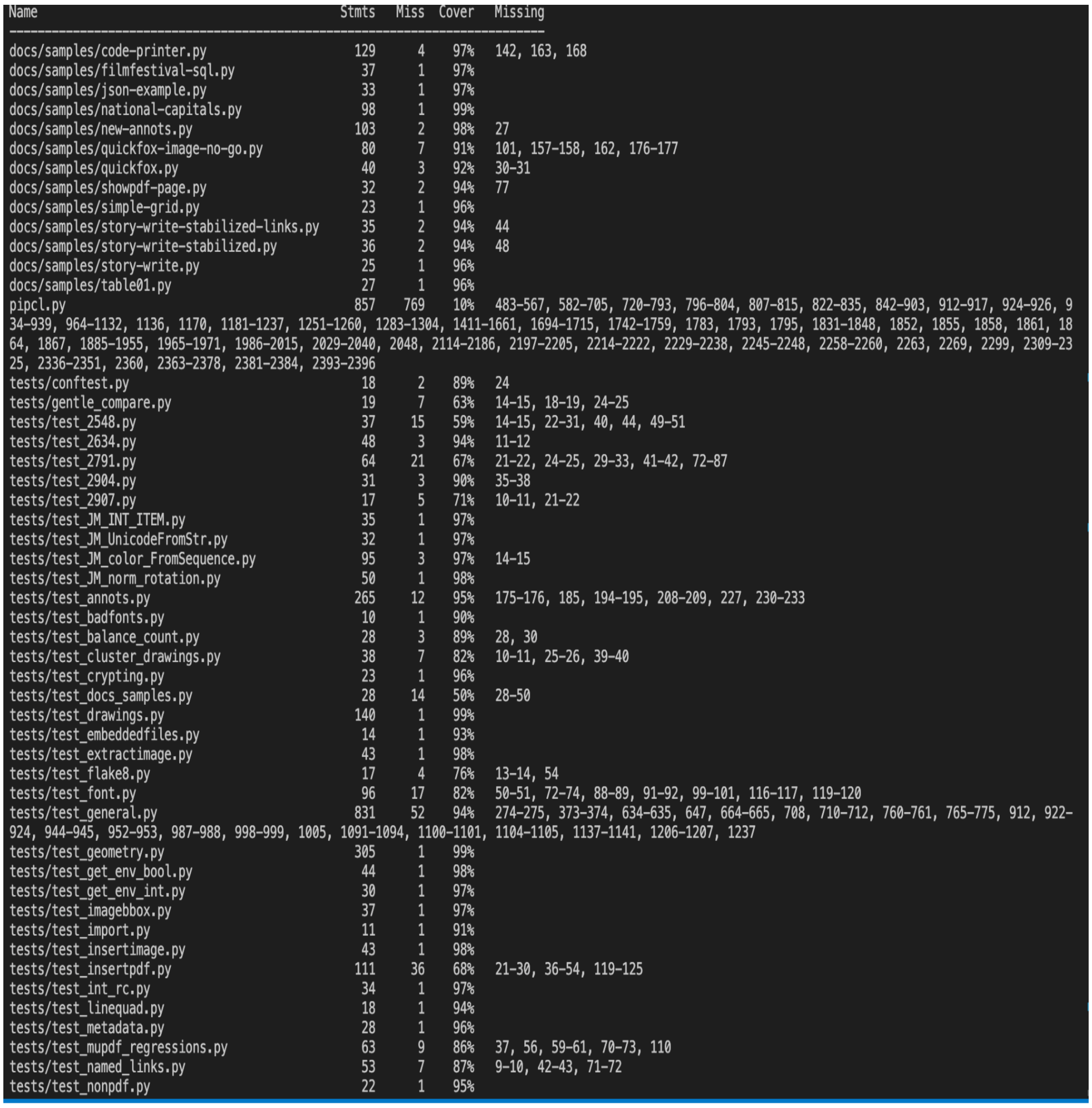
Figure 3: Old Coverage Result by Existing Tool (first part)



```
docs/samples/national-capitals.py                   98      1    99%
docs/samples/new-annots.py                         103      2    98%
docs/samples/quickfox-image-no-go.py                80      7    91%
docs/samples/quickfox.py                            40      3    92%
docs/samples/showpdf-page.py                        32      2    94%
docs/samples/simple-grid.py                         23      1    96%
docs/samples/story-write-stabilized-links.py        35      2    94%
docs/samples/story-write-stabilized.py              36      2    94%
docs/samples/story-write.py                         25      1    96%
docs/samples/table01.py                             27      1    96%
pipcl.py                                           857    776     9%
tests/conftest.py                                   18      2    89%
tests/gentle_compare.py                             19      7    63%
tests/test_2548.py                                  37     15    59%
tests/test_2634.py                                  48      3    94%
tests/test_2791.py                                  64     21    67%
tests/test_2904.py                                  31      3    90%
tests/test_2907.py                                  17      5    71%
tests/test_annots.py                               265     12    95%
tests/test_badfonts.py                              10      1    90%
tests/test_balance_count.py                         28      3    89%
tests/test_cluster_drawings.py                      38      7    82%
tests/test_crypting.py                              23      1    96%
tests/test_docs_samples.py                          28     14    50%
tests/test_drawings.py                             140      1    99%
tests/test_embeddedfiles.py                         14      1    93%
tests/test_extractimage.py                          43      1    98%
tests/test_flake8.py                                17      4    76%
tests/test_font.py                                  96     17    82%
tests/test_general.py                              831     52    94%
tests/test_geometry.py                             305      1    99%
tests/test_imagebbox.py                             37      1    97%
tests/test_import.py                                11      1    91%
tests/test_insertimage.py                           43      1    98%
tests/test_insertpdf.py                            111     36    68%
tests/test_linequad.py                              18      1    94%
tests/test_metadata.py                              28      1    96%
tests/test_mupdf_regressions.py                     63      9    86%
tests/test_named_links.py                           53      7    87%
tests/test_nonpdf.py                                22      1    95%
tests/test_object_manipulation.py                   47      1    98%
tests/test_objectstreams.py                         52      6    88%
tests/test_optional_content.py                      77      1    99%
tests/test_page_links.py                            10      1    90%
tests/test_pagedelete.py                            49      1    98%
tests/test_pagelabels.py                            28      1    96%
tests/test_pixmap.py                               225     40    82%
tests/test_pylint.py                                43     10    77%
tests/test_remove-rotation.py                       22      1    95%
tests/test_showpdfpage.py                           32      1    97%
tests/test_story.py                                107      4    96%
tests/test_tables.py                               189      1    99%
tests/test_tesseract.py                             41     11    73%
tests/test_textbox.py                              157     12    92%
tests/test_textextract.py                          188     37    80%
tests/test_textsearch.py                            23      1    96%
tests/test_toc.py                                  109      5    95%
tests/test_widgets.py                              210      1    99%
tests/test_word_delimiters.py                       14      1    93%
wdev.py                                            156    137    12%
-------------------------------------------------------------------
TOTAL                                             5792   1306    77%
nehiraltinkaya@Nehirs-MacBook-Air-2 PyMuPDF-main %
```

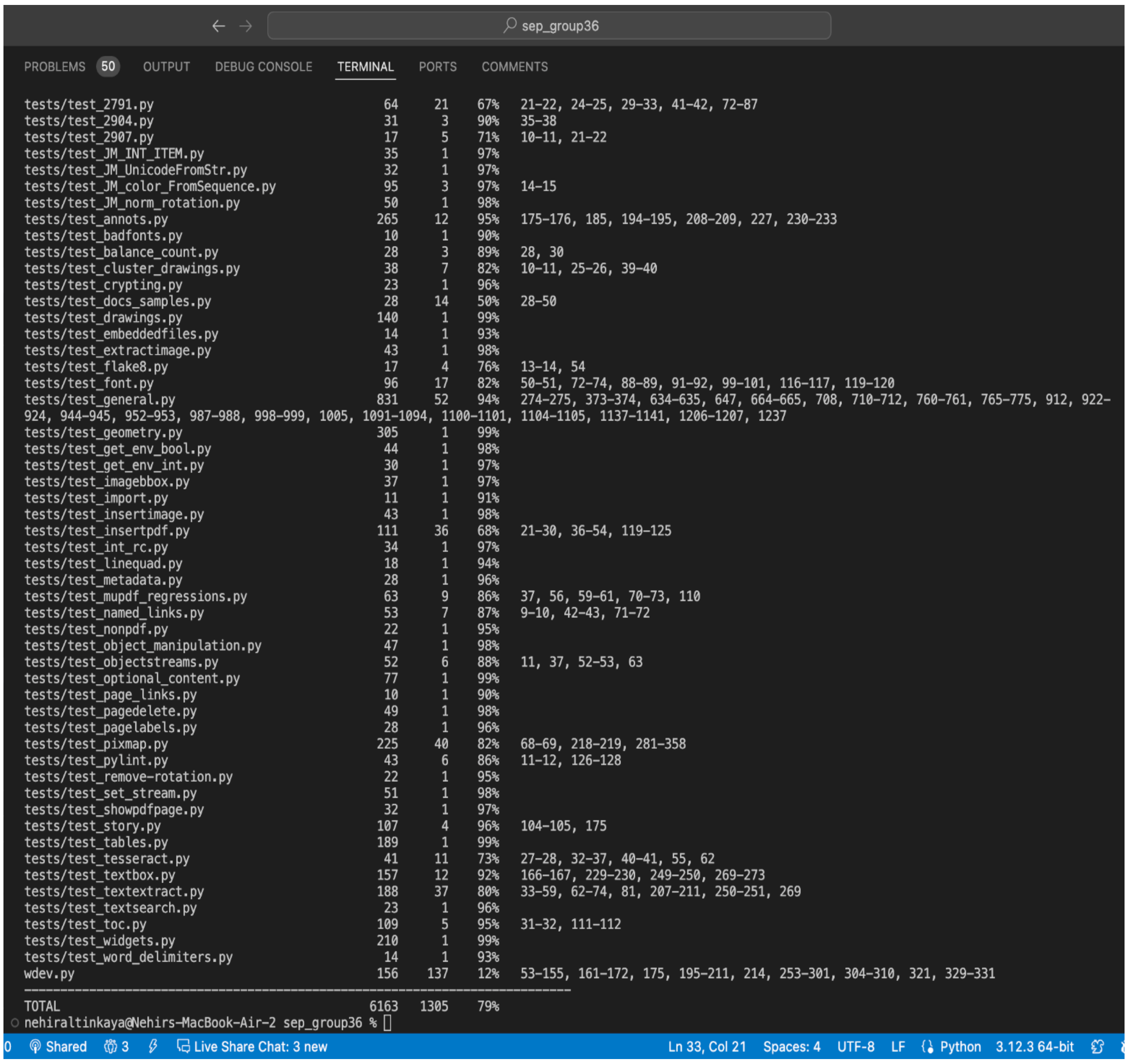
Figure 4: Old Coverage Result by Existing Tool (second part)

Here is the screenshot of the new coverage results by running the existing tool using all test modifications made:



```
Name                                         Stmts   Miss  Cover   Missing
--------------------------------------------------------------------------
docs/samples/code-printer.py                   129      4    97%   142, 163, 168
docs/samples/filmfestival-sql.py                37      1    97%
docs/samples/json-example.py                    33      1    97%
docs/samples/national-capitals.py               98      1    99%
docs/samples/new-annots.py                     103      2    98%   27
docs/samples/quickfox-image-no-go.py            80      7    91%   101, 157-158, 162, 176-177
docs/samples/quickfox.py                        40      3    92%   30-31
docs/samples/showpdf-page.py                    32      2    94%   77
docs/samples/simple-grid.py                     23      1    96%
docs/samples/story-write-stabilized-links.py    35      2    94%   44
docs/samples/story-write-stabilized.py          36      2    94%   48
docs/samples/story-write.py                     25      1    96%
docs/samples/table01.py                         27      1    96%
pipcl.py                                       857    769    10%   483-567, 582-705, 720-793, 796-804, 807-815, 822-835, 842-903, 912-917, 924-926, 9
34-939, 964-1132, 1136, 1170, 1181-1237, 1251-1260, 1283-1304, 1411-1661, 1694-1715, 1742-1759, 1783, 1793, 1795, 1831-1848, 1852, 1855, 1858, 1861, 18
64, 1867, 1885-1955, 1965-1971, 1986-2015, 2029-2040, 2048, 2114-2186, 2197-2205, 2214-2222, 2229-2238, 2245-2248, 2258-2260, 2263, 2269, 2299, 2309-23
25, 2336-2351, 2360, 2363-2378, 2381-2384, 2393-2396
tests/conftest.py                               18      2    89%   24
tests/gentle_compare.py                         19      7    63%   14-15, 18-19, 24-25
tests/test_2548.py                              37     15    59%   14-15, 22-31, 40, 44, 49-51
tests/test_2634.py                              48      3    94%   11-12
tests/test_2791.py                              64     21    67%   21-22, 24-25, 29-33, 41-42, 72-87
tests/test_2904.py                              31      3    90%   35-38
tests/test_2907.py                              17      5    71%   10-11, 21-22
tests/test_JM_INT_ITEM.py                       35      1    97%
tests/test_JM_UnicodeFromStr.py                 32      1    97%
tests/test_JM_color_FromSequence.py             95      3    97%   14-15
tests/test_JM_norm_rotation.py                  50      1    98%
tests/test_annots.py                           265     12    95%   175-176, 185, 194-195, 208-209, 227, 230-233
tests/test_badfonts.py                          10      1    90%
tests/test_balance_count.py                     28      3    89%   28, 30
tests/test_cluster_drawings.py                  38      7    82%   10-11, 25-26, 39-40
tests/test_crypting.py                          23      1    96%
tests/test_docs_samples.py                      28     14    50%   28-50
tests/test_drawings.py                         140      1    99%
tests/test_embeddedfiles.py                     14      1    93%
tests/test_extractimage.py                      43      1    98%
tests/test_flake8.py                            17      4    76%   13-14, 54
tests/test_font.py                              96     17    82%   50-51, 72-74, 88-89, 91-92, 99-101, 116-117, 119-120
tests/test_general.py                          831     52    94%   274-275, 373-374, 634-635, 647, 664-665, 708, 710-712, 760-761, 765-775, 912, 922-
924, 944-945, 952-953, 987-988, 998-999, 1005, 1091-1094, 1100-1101, 1104-1105, 1137-1141, 1206-1207, 1237
tests/test_geometry.py                         305      1    99%
tests/test_get_env_bool.py                      44      1    98%
tests/test_get_env_int.py                       30      1    97%
tests/test_imagebbox.py                         37      1    97%
tests/test_import.py                            11      1    91%
tests/test_insertimage.py                       43      1    98%
tests/test_insertpdf.py                        111     36    68%   21-30, 36-54, 119-125
tests/test_int_rc.py                            34      1    97%
tests/test_linequad.py                          18      1    94%
tests/test_metadata.py                          28      1    96%
tests/test_mupdf_regressions.py                 63      9    86%   37, 56, 59-61, 70-73, 110
tests/test_named_links.py                       53      7    87%   9-10, 42-43, 71-72
tests/test_nonpdf.py                            22      1    95%
```

Figure 5: New Coverage Result by Existing Tool (first part)



```
tests/test_2791.py                              64     21    67%   21-22, 24-25, 29-33, 41-42, 72-87
tests/test_2904.py                              31      3    90%   35-38
tests/test_2907.py                              17      5    71%   10-11, 21-22
tests/test_JM_INT_ITEM.py                       35      1    97%
tests/test_JM_UnicodeFromStr.py                 32      1    97%
tests/test_JM_color_FromSequence.py             95      3    97%   14-15
tests/test_JM_norm_rotation.py                  50      1    98%
tests/test_annots.py                           265     12    95%   175-176, 185, 194-195, 208-209, 227, 230-233
tests/test_badfonts.py                          10      1    90%
tests/test_balance_count.py                     28      3    89%   28, 30
tests/test_cluster_drawings.py                  38      7    82%   10-11, 25-26, 39-40
tests/test_crypting.py                          23      1    96%
tests/test_docs_samples.py                      28     14    50%   28-50
tests/test_drawings.py                         140      1    99%
tests/test_embeddedfiles.py                     14      1    93%
tests/test_extractimage.py                      43      1    98%
tests/test_flake8.py                            17      4    76%   13-14, 54
tests/test_font.py                              96     17    82%   50-51, 72-74, 88-89, 91-92, 99-101, 116-117, 119-120
tests/test_general.py                          831     52    94%   274-275, 373-374, 634-635, 647, 664-665, 708, 710-712, 760-761, 765-775, 912, 922-
924, 944-945, 952-953, 987-988, 998-999, 1005, 1091-1094, 1100-1101, 1104-1105, 1137-1141, 1206-1207, 1237
tests/test_geometry.py                         305      1    99%
tests/test_get_env_bool.py                      44      1    98%
tests/test_get_env_int.py                       30      1    97%
tests/test_imagebbox.py                         37      1    97%
tests/test_import.py                            11      1    91%
tests/test_insertimage.py                       43      1    98%
tests/test_insertpdf.py                        111     36    68%   21-30, 36-54, 119-125
tests/test_int_rc.py                            34      1    97%
tests/test_linequad.py                          18      1    94%
tests/test_metadata.py                          28      1    96%
tests/test_mupdf_regressions.py                 63      9    86%   37, 56, 59-61, 70-73, 110
tests/test_named_links.py                       53      7    87%   9-10, 42-43, 71-72
tests/test_nonpdf.py                            22      1    95%
tests/test_object_manipulation.py               47      1    98%
tests/test_objectstreams.py                     52      6    88%   11, 37, 52-53, 63
tests/test_optional_content.py                  77      1    99%
tests/test_page_links.py                        10      1    90%
tests/test_pagedelete.py                        49      1    98%
tests/test_pagelabels.py                        28      1    96%
tests/test_pixmap.py                           225     40    82%   68-69, 218-219, 281-358
tests/test_pylint.py                            43      6    86%   11-12, 126-128
tests/test_remove-rotation.py                   22      1    95%
tests/test_set_stream.py                        51      1    98%
tests/test_showpdfpage.py                       32      1    97%
tests/test_story.py                            107      4    96%   104-105, 175
tests/test_tables.py                           189      1    99%
tests/test_tesseract.py                         41     11    73%   27-28, 32-37, 40-41, 55, 62
tests/test_textbox.py                          157     12    92%   166-167, 229-230, 249-250, 269-273
tests/test_textextract.py                      188     37    80%   33-59, 62-74, 81, 207-211, 250-251, 269
tests/test_textsearch.py                        23      1    96%
tests/test_toc.py                              109      5    95%   31-32, 111-112
tests/test_widgets.py                          210      1    99%
tests/test_word_delimiters.py                   14      1    93%
wdev.py                                        156    137    12%   53-155, 161-172, 175, 195-211, 214, 253-301, 304-310, 321, 329-331
------------------------------------------------------------------------------------
TOTAL                                         6163   1305    79%
nehiraltinkaya@Nehirs-MacBook-Air-2 sep_group36 % 
```

Figure 6: New Coverage Result by Existing Tool (second part)

## Statement of individual contributions
All Members:
Firstly, all group members worked together to find a suitable project to be used by our group. When found, certain functions were identified and tests were created correspondingly. After the completion of all the tests, all team members contributed to necessary changes on all of the newly created test files whether related to names of the functions or altering test cases. Lastly, the coverage is measured and improvements are observed. After completion of the programming aspect of the assignment, all team members contributed to the “Readme” file of our Assignment 1 project.

Nehir Altinkaya:
During the assignment, I created the tests named as “test_int_rc.py” and “test_get_env_bool”. At first, the functions of these tests were found in the “__init__.py” file. The coverage was observed after the creation and addition to the test files. After the tests were made depending on the branches of the functions, different functions were added within the test file corresponding to different test cases. In addition, naming of these functions have been altered depending on the requirement mentioned on the Gitbook guide for Assignment 1. The coverage is then observed again for the last time.

Coverage Results: In general, the coverage result was 77 percent which improved to 79 percent at the end.
Device Used: Macbook Air M1

Turkan Badalzade:
I assisted my teammates to find a suitable project on github. After that, I called coverage html to observe the functions were not covered. I analysed the “__init.py__” file and found two functions to make the test functions. I created two files: “test_JM_norm_rotation.py” and “test_JM_color_fromSequence.py”. After creating the functions I improved the coverage results as much as I could by adding new test cases. Furthermore, I helped my teammates when they had questions about the test implementation part. In addition to that I revised the functions according to the requirements on gitbook. Finally, I also contributed to the “Readme” file as our last step in Assignment 1.

Coverage Results: In general, the coverage result was 78 percent which improved to 79 percent at the end.
Device Used: Macbook M1 Pro

Erin Elagoz:
In the beginning of the assignment I helped my team find a project that suits the guidelines and calculated the total lines of code using an existing tool (lizard). I have created “test_set_stream.py” and “test_get_env_int.py” files which contain 2 functions from the “__init__.py” file. Initially I calculated the coverage of these functions, then I improved them and calculated the coverage again to observe the increased coverage and uploaded the results as a screenshot to this document.

Coverage Results: In general, the coverage result was 77 percent which improved to 78 percent at the end.
Device Used: MSI - Windows 11

Misra Ozoktash:
At the beginning of the assignment, I helped my team find a suitable project on GitHub that meets the requirements of Assignment 1. During the assignment, I have created tests named “test_JM_INT_ITEM.py” and “test_JM_UnicodeFromStr.py”. The original version of the functions used to create these tests could be found in the “__init__.py” file. After creating these test files, the initial coverage tests were observed. Moreover, using the information gained from the coverage test results, several improvements have been made to both of the tests resulting in a rise of the coverage test percentages. To name some of these improvements, I have added several different test cases depending on the function, making sure that all of the branches have been tested at least once. Finally, I have checked the format of the code, files, functions and the function names to meet the requirements of Assignment 1. To make sure everything is in working order, the coverage test has been made once again.

Coverage Results: In general, the coverage result was 71 percent which improved to 72 percent at the end.
Device Used: Lenovo - Windows 11