--- Step 3: Running mutation testing (Attempt 4/30) ---

[INFO] Running MutPy for target: mutation\_output\source\_to\_mutate.py, tests: mutation\_output\test\_generated\_mutants.py

[\*] Start mutation process:

- targets: source\_to\_mutate

- tests: test\_generated\_mutants

[\*] 24 tests passed:

- test\_generated\_mutants [0.09807 s]

[\*] Start mutants generation and execution:

- [# 1] AOD source\_to\_mutate: [0.09490 s] killed by test\_generated\_mutants.py::test\_negative\_numbers

- [# 2] AOD source\_to\_mutate: [0.06367 s] killed by test\_generated\_mutants.py::test\_negative\_numbers

- [# 3] AOR source\_to\_mutate: [0.05372 s] killed by test\_generated\_mutants.py::test\_negative\_numbers

- [# 4] AOR source\_to\_mutate: [0.06203 s] killed by test\_generated\_mutants.py::test\_negative\_numbers

- [# 5] AOR source\_to\_mutate: [0.05426 s] killed by test\_generated\_mutants.py::test\_negative\_large\_numbers

- [# 6] AOR source\_to\_mutate: [0.05968 s] killed by test\_generated\_mutants.py::test\_negative\_numbers

- [# 7] AOR source\_to\_mutate: [0.07206 s] killed by test\_generated\_mutants.py::test\_negative\_numbers

- [# 8] AOR source\_to\_mutate: [0.06237 s] survived

- [# 9] AOR source\_to\_mutate: [0.06242 s] survived

- [# 10] AOR source\_to\_mutate: [0.05611 s] killed by test\_generated\_mutants.py::test\_negative\_numbers

- [# 11] COI source\_to\_mutate: [0.05524 s] killed by test\_generated\_mutants.py::test\_positive\_numbers

- [# 12] ROR source\_to\_mutate: [0.06114 s] killed by test\_generated\_mutants.py::test\_positive\_numbers

- [# 13] ROR source\_to\_mutate: [0.06239 s] survived

- [# 14] ROR source\_to\_mutate: [0.05325 s] killed by test\_generated\_mutants.py::test\_positive\_numbers

- [# 15] ROR source\_to\_mutate: [0.06105 s] killed by test\_generated\_mutants.py::test\_mixed\_numbers

[\*] Mutation score [1.09384 s]: 80.0%

- all: 15

- killed: 12 (80.0%)

- survived: 3 (20.0%)

- incompetent: 0 (0.0%)

- timeout: 0 (0.0%)

[SUCCESS] Initial tests passed. Now calculating coverage and mutation score.

--- Step 4: Calculating test coverage ---

[INFO] Running coverage for target: mutation\_output\source\_to\_mutate.py, tests: mutation\_output\test\_generated\_mutants.py

Name Stmts Miss Branch BrPart Cover Missing

-----------------------------------------------------------------

source\_to\_mutate.py 8 0 0 0 100%

-----------------------------------------------------------------

TOTAL 8 0 0 0 100%

--- Step 5: Final Results ---

[INFO] Test Coverage: 100%

[INFO] Mutation Score: 80.00%

--- Analysis Finished ---