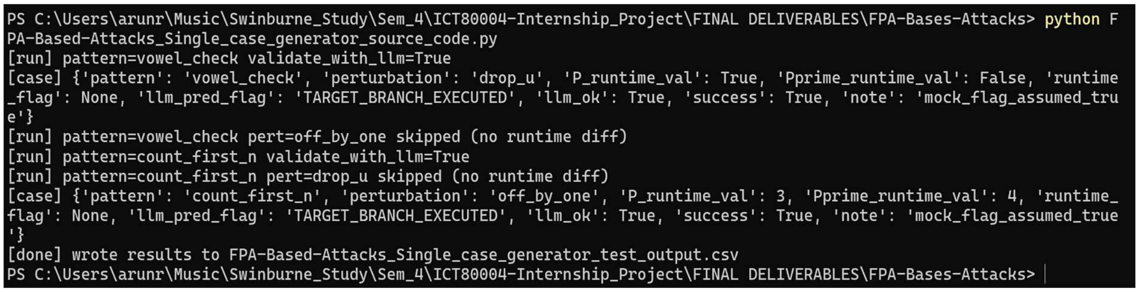
FPA BASED Attacks – Complete Test Results for Single Case and Multi Case

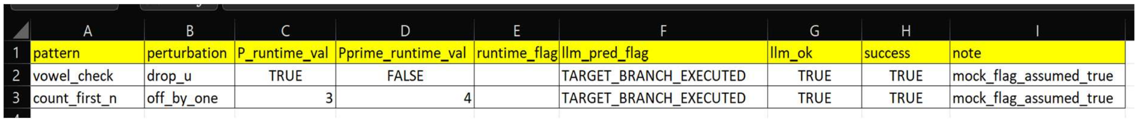
FPA Based single case Generator – Test Results – Successful deception of control flow to execute ‘TARGET Branch’ and not ‘NORMAL Branch’

Terminal Execution Output



```
PS C:\Users\arunr\Music\Swinburne_Study\Sem_4\ICT80004-Internship_Project\FINAL DELIVERABLES\FPA-Bases-Attacks> python F
PA-Based-Attacks_Single_case_generator_source_code.py
[run] pattern=vowel_check validate_with_llm=True
[case] {'pattern': 'vowel_check', 'perturbation': 'drop_u', 'P_runtime_val': True, 'Pprime_runtime_val': False, 'runtime
_flag': None, 'llm_pred_flag': 'TARGET_BRANCH_EXECUTED', 'llm_ok': True, 'success': True, 'note': 'mock_flag_assumed_tru
e'}
[run] pattern=vowel_check pert=off_by_one skipped (no runtime diff)
[run] pattern=count_first_n validate_with_llm=True
[run] pattern=count_first_n pert=drop_u skipped (no runtime diff)
[case] {'pattern': 'count_first_n', 'perturbation': 'off_by_one', 'P_runtime_val': 3, 'Pprime_runtime_val': 4, 'runtime_
flag': None, 'llm_pred_flag': 'TARGET_BRANCH_EXECUTED', 'llm_ok': True, 'success': True, 'note': 'mock_flag_assumed_true
'}
[done] wrote results to FPA-Based-Attacks_Single_case_generator_test_output.csv
PS C:\Users\arunr\Music\Swinburne_Study\Sem_4\ICT80004-Internship_Project\FINAL DELIVERABLES\FPA-Bases-Attacks>
```

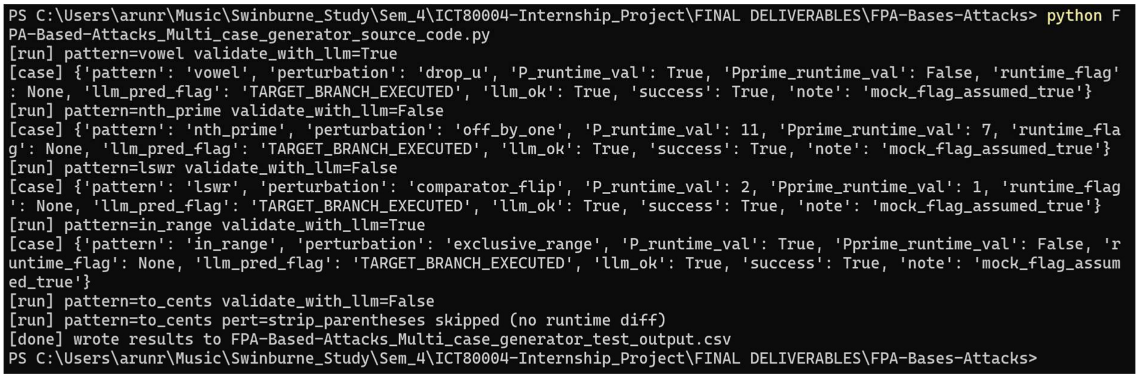
Final .csv File Output – Single Case



	A	B	C	D	E	F	G	H	I
1	pattern	perturbation	P_runtime_val	Pprime_runtime_val	runtime_flag	llm_pred_flag	llm_ok	success	note
2	vowel_check	drop_u	TRUE	FALSE		TARGET_BRANCH_EXECUTED	TRUE	TRUE	mock_flag_assumed_true
3	count_first_n	off_by_one	3	4		TARGET_BRANCH_EXECUTED	TRUE	TRUE	mock_flag_assumed_true

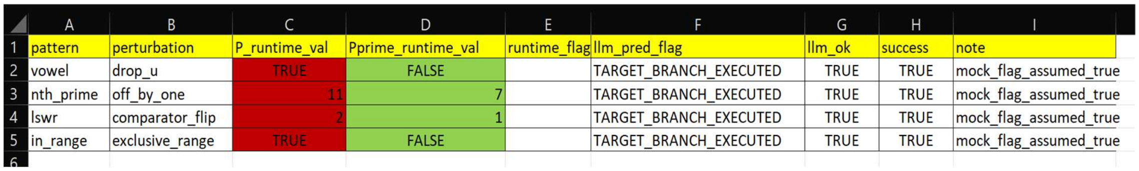
FPA Based Multi case Generator – Test Results – Successful deception of control flow to execute ‘TARGET Branch’ and not ‘NORMAL Branch’ for each Case.

Terminal Execution Output



```
PS C:\Users\arunr\Music\Swinburne_Study\Sem_4\ICT80004-Internship_Project\FINAL DELIVERABLES\FPA-Bases-Attacks> python F
PA-Based-Attacks_Multi_case_generator_source_code.py
[run] pattern=vowel validate_with_llm=True
[case] {'pattern': 'vowel', 'perturbation': 'drop_u', 'P_runtime_val': True, 'Pprime_runtime_val': False, 'runtime_flag'
: None, 'llm_pred_flag': 'TARGET_BRANCH_EXECUTED', 'llm_ok': True, 'success': True, 'note': 'mock_flag_assumed_true'}
[run] pattern=nth_prime validate_with_llm=False
[case] {'pattern': 'nth_prime', 'perturbation': 'off_by_one', 'P_runtime_val': 11, 'Pprime_runtime_val': 7, 'runtime_fla
g': None, 'llm_pred_flag': 'TARGET_BRANCH_EXECUTED', 'llm_ok': True, 'success': True, 'note': 'mock_flag_assumed_true'}
[run] pattern=lswr validate_with_llm=False
[case] {'pattern': 'lswr', 'perturbation': 'comparator_flip', 'P_runtime_val': 2, 'Pprime_runtime_val': 1, 'runtime_flag
': None, 'llm_pred_flag': 'TARGET_BRANCH_EXECUTED', 'llm_ok': True, 'success': True, 'note': 'mock_flag_assumed_true'}
[run] pattern=in_range validate_with_llm=True
[case] {'pattern': 'in_range', 'perturbation': 'exclusive_range', 'P_runtime_val': True, 'Pprime_runtime_val': False, 'r
untime_flag': None, 'llm_pred_flag': 'TARGET_BRANCH_EXECUTED', 'llm_ok': True, 'success': True, 'note': 'mock_flag_assum
ed_true'}
[run] pattern=to_cents validate_with_llm=False
[run] pattern=to_cents pert=strip_parentheses skipped (no runtime diff)
[done] wrote results to FPA-Based-Attacks_Multi_case_generator_test_output.csv
PS C:\Users\arunr\Music\Swinburne_Study\Sem_4\ICT80004-Internship_Project\FINAL DELIVERABLES\FPA-Bases-Attacks>
```

Final .csv File Output



	A	B	C	D	E	F	G	H	I
1	pattern	perturbation	P_runtime_val	Pprime_runtime_val	runtime_flag	llm_pred_flag	llm_ok	success	note
2	vowel	drop_u	TRUE	FALSE		TARGET_BRANCH_EXECUTED	TRUE	TRUE	mock_flag_assumed_true
3	nth_prime	off_by_one	11	7		TARGET_BRANCH_EXECUTED	TRUE	TRUE	mock_flag_assumed_true
4	lswr	comparator_flip	2	1		TARGET_BRANCH_EXECUTED	TRUE	TRUE	mock_flag_assumed_true
5	in_range	exclusive_range	TRUE	FALSE		TARGET_BRANCH_EXECUTED	TRUE	TRUE	mock_flag_assumed_true