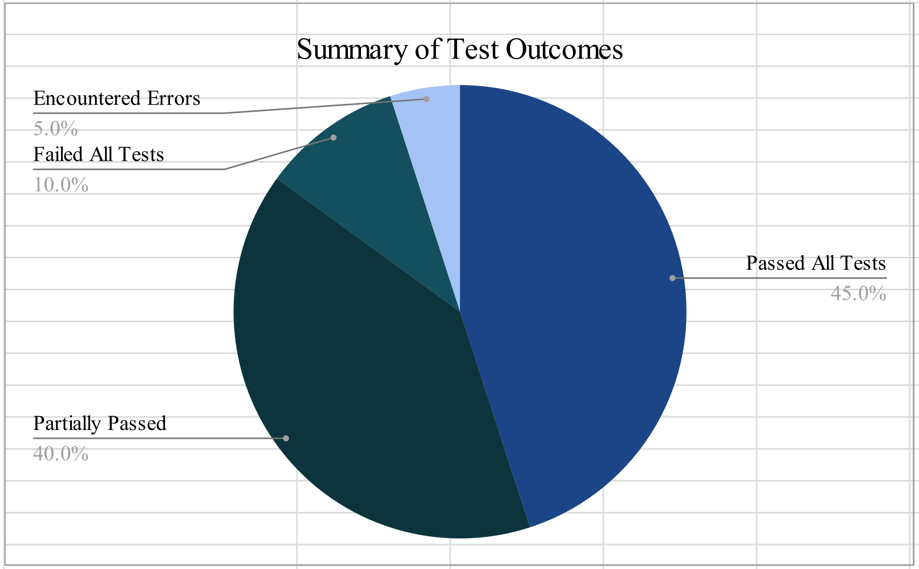
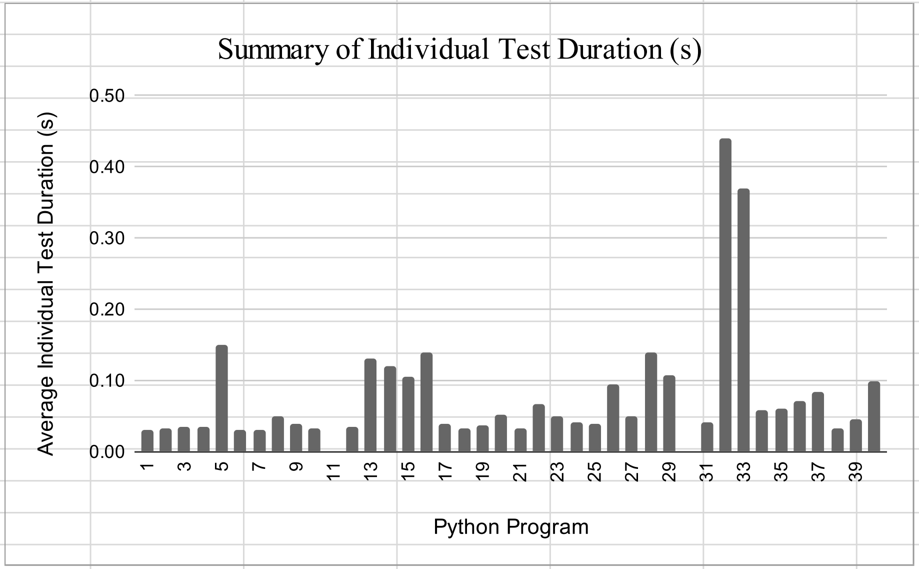
Prompt Number	Program Code	LLM Output	Modified Program Code	Modified Unit Test Code	Tests Passed?	Number of Tests Passed	Number of Tests Failed	Total Number of Tests	Pass Percentage (%)	Branch Coverage (%)	Entire Test Duration (s)	Individual Test Duration (s)
Description	Extracted from the Humaneval Dataset. Used as an Input into Claude-2's API	* Large Language Model (LLM) Generated by Claude-2's API, **Prompt: "**Write unit tests for the following Python code using Pytest. First provide the function followed by the test code. The code is: {Column_B}"	The programs extracted from the Humaneval dataset do not contain function definitions. This column represents the code refactored with function declariration.	Since the output response from Claude contains the contents from Coloumn E, adjustmnets needed to be made to the Unit Test files.								
1	for idx, elem in enumerate	Here is the test code and function to test	def has_close_eleme	import pytest	☐	3	1	4	75.00	100.00	0.12	0.03
2	result = [] current_string	Here are some Pytest unit tests	def decompose (paren_string):	import pytest from program2	☐	3	2	5	60.00	97.00	0.16	0.03
3	return number % 1.0	Here is the test code using pytest:	def is_integer (number):	from program3 import	☑	4	0	4	100.00	94.00	0.14	0.04
4	balance = 0	# Function	def check_negative_b	from program4 import	☑	4	0	4	100.00	100.00	0.14	0.04
5	mean = sum (numbers) / len	def mean_absolute_deviation (numbers):	def mean_absolute_d	from program5 import	☑	1	0	1	100.00	100.00	0.15	0.15
6	if not numbers: return []	Here are unit tests for the function using pytest:	def join_list (numbers,	from program6 import join_list	☑	5	0	5	100.00	100.00	0.15	0.03
7	def parse_paren_grou	Here are some Pytest unit tests for the parse_paren_group	def parse_paren_grou	from program7 import	☐	3	4	7	42.86	100.00	0.21	0.03
8	return [x for x in strings if	Here is a pytest test for that function:	def find_strings_with	from program8 import	☑	3	0	3	100.00	100.00	0.15	0.05
9	sum_value = 0 prod_value = 1	Here is the test code using pytest:	def sum_and_prod	from program9 import	☐	3	1	4	75.00	100.00	0.16	0.04
10	running_max = None	Here are some Pytest unit tests for this function:	def running_max (numbers):	import pytest from program10	☑	5	0	5	100.00	100.00	0.16	0.03
11	if not string: return ''	Here are some unit tests for the function using pytest:	def is_palindrome	import pytest from program11		-	-	0	-	-	-	-
12	def xor(i, j): if i == j:	Here are some pytest unit tests for the xor function:	def xor(i, j): if i == j:	from program12 import xor	☐	2	3	5	40.00	100.00	0.18	0.04
13	if not strings: return None	Here is the test code using Pytest:	def longest (strings):	from program13	☑	1	0	1	100.00	96.00	0.13	0.13
14	while b: a, b = b, a %	def test_gcd(): from math import gcd	def my_gcd(a, b): while b:	from program14 import my_gcd	☑	1	0	1	100.00	100.00	0.12	0.12
15	result = []	Here are some pytest unit tests for the function:	def string_subsets (string):	from program15 import	☐	0	4	4	0.00	100.00	0.42	0.11
16	return ' '.join ([str(x) for x in	def test_join_range(): # Basic test	def join_range (n):	from program16 import	☐	0	1	1	0.00	67.00	0.14	0.14
17	return len(set (string.lower()))	Here is the function to test followed by sample pytest test	def count_unique_lett	from program17 import	☐	2	5	7	28.57	100.00	0.27	0.04
18	note_map = {'o': 4, 'o|': 2, '.|':	Here is an example test suite using pytest:	def get_note_duratio	from program18 import	☑	5	0	5	100.00	100.00	0.16	0.03
19	times = 0	Here are some unit tests for the code using pytest:	def substring_count	from program19 import	☑	5	0	5	100.00	100.00	0.19	0.04
20	value_map = { 'zero': 0,	Here are some Pytest unit tests for the function:	def sort_numbers (numbers):	from program20 import	☐	3	1	4	75.00	100.00	0.21	0.05
21	closest_pair = None	Here are some pytest test cases for the code:	def closest_pair (numbers):	from program21 import	☐	3	4	7	42.86	100.00	0.23	0.03

22	min_number = min(numbers)	Here are some pytest unit tests for the code:	def normalize_numbe	from program22 import	☐	1	5	6	16.67	100.00	0.40	0.07
23	return [x for x in values if	
		```
		python
		import pytest
		```
			def filter_ints (values):	from program23 import filter_ints	☑	4	0	4	100.00	100.00	0.20	0.05
24	return len (string)	Here is the function to test:	def string_length (string):	from program24 import	☑	4	0	4	100.00	100.00	0.17	0.04
25	for i in reversed(range	```python	def largest_factor (n):	from program25	☐	2	3	5	40.00	100.00	0.20	0.04
26	import math fact = []	Here are some Pytest unit tests for this code:	import math	from program26	☑	2	0	2	100.00	100.00	0.19	0.10
27	import collections	Here are some unit tests for the code using pytest:	import collections	from program27 import	☑	4	0	4	100.00	86.00	0.20	0.05
28	return string. swapcase()	Here is a Pytest test case for this	def swap_case (text):	from program28 import	☐	0	1	1	0.00	80.00	0.14	0.14
29	return ''.join (strings)	Here is the unit test code using pytest:	def concat_strings	from program29 import	☑	4	0	4	100.00	100.00	0.43	0.11
30	return [x for x in strings if x.	def test_startswith(): strings = ['apple', 'banana', 'cat']	def func(strings, prefix):	from program30 import func		-	-	0	-	-	-	-
31	return [e for e in l if e > 0]	Here is the function to test along with some sample pytest unit	def positive_list (l):	from program31 import	☑	5	0	5	100.00	100.00	0.21	0.04
32	if n < 2: return False	import pytest def is_prime(n):	def is_prime(n): if n < 2:	from program32 import is_prime	☑	1	0	1	100.00	93.00	0.44	0.44
33	begin, end = -1., 1.	Here are Pytest unit tests for this code:	def bisect(xs):	from program33 import bisect,	☐	0	1	1	0.00	43.00	0.37	0.37
34	l = list(l) l[::3] = sorted(l	Here are Pytest unit tests for that function:	def sort_list_every_th	from program34	☐	4	2	6	66.67	100.00	0.35	0.06
35	return sorted (list(set(l)))	Here is the function followed by some sample pytest test cases:	def remove_duplicate	from program35 import	☐	4	1	5	80.00	100.00	0.30	0.06
36	m = l[0] for e in l:	Here is a set of pytest unit tests for the code:	def max_element (l):	from program36 import	☐	6	1	7	85.71	100.00	0.50	0.07
37	ns = [] for i in range	Here are some unit tests for this code using pytest:	def count_sevens_in_	from program37 import	☐	2	2	4	50.00	100.00	0.34	0.09
38	evens = l[::2] odds = l[1::2]	Here are some pytest unit tests for the function:	def sort_even_odd(l):	from program38	☐	4	2	6	66.67	100.00	0.20	0.03
39	return encode_cyclic	Here is the function to test:	def encode(s: str) -> str:	from program39 import encode	☐	3	3	6	50.00	96.00	0.28	0.05
40	import math	def test_is_prime(): # Test some prime numbers	import math	from program40 import is_prime	☑	1	0	1	100.00	100.00	0.10	0.10
Total Averages (%)					45.00					96.11	0.23	0.08

Passed All Tests	18
Partially Passed	16
Failed All Tests	4
Encountered Errors	2


Summary of Test Outcomes
Encountered Errors
Failed All Tests
Passed All Tests
Partially Passed


Summary of Individual Test Duration (s)
Average Individual Test Duration (s)
Python Program