task_1_tests.py::test_parse_calls[singleCall_singleMonth] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_parse_calls[multipleCall_singleMonth] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_parse_calls[multipleCall_multipleMonth] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_parse_calls[multipleCall_singleMonth_withSMS] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_parse_calls[multipleCall_multipleMonth_withSMS] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_parse_calls[one_month_dataset] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_parse_calls[two_month_dataset] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_parse_calls[six_month_dataset] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_parse_calls[large_dataset] (test reading in a set of calls/sms objects and ensure the correct methods are called)

pass

1 / 1

task_1_tests.py::test_call_typecontract[singleCall_singleMonth] (Test to make sure all parts of Call are init properly)

fail

0 / 1

task_1_tests.py::test_call_typecontract[multipleCall_singleMonth] (Test to make sure all parts of Call are init properly)

fail

0 / 1

```
task_1_tests.py::test_call_typecontract[multipleCall_multipleMonth] ( Test to make sure all parts of Call are init properly )
fail
0 / 1
task_1_tests.py::test_call_typecontract[multipleCall_singleMonth_withSMS] ( Test to make sure all parts of Call are init properly )
fail
0 / 1
task_1_tests.py::test_call_typecontract[multipleCall_multipleMonth_withSMS] ( Test to make sure all parts of Call are init properly )
fail
0 / 1
task_2_tests.py::test_callhistory_import[singleCall_singleMonth] ( Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers )
pass
1 / 1
task_2_tests.py::test_callhistory_import[multipleCall_singleMonth] ( Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers )
pass
1 / 1
task_2_tests.py::test_callhistory_import[multipleCall_multipleMonth] ( Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers )
pass
1 / 1
task_2_tests.py::test_callhistory_import[multipleCall_singleMonth_withSMS] ( Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers )
pass
1 / 1
task_2_tests.py::test_callhistory_import[multipleCall_multipleMonth_withSMS] ( Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers )
pass
1 / 1
task_2_tests.py::test_callhistory_import[one_month_dataset] ( Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers )
pass
1 / 1
task_2_tests.py::test_callhistory_import[two_month_dataset] ( Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers )
pass
1 / 1
task_2_tests.py::test_callhistory_import[six_month_dataset] ( Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers )
pass
1 / 1
```

task_2_tests.py::test_callhistory_import[large_dataset] (Test the callhistory by importing datasets and ensuring the calls are placed with the correct customers)

pass

1 / 1

task_2_basic_tests.py::test_callhistory_incoming_first_call (Test register incoming call with one call)

pass

1 / 1

task_2_basic_tests.py::test_callhistory_incoming_multiple_months_and_calls (Test register incoming call with multiple calls)

pass

1 / 1

task_2_basic_tests.py::test_callhistory_outgoing_first_call (Test register outgoing call with one call)

pass

1 / 1

task_2_basic_tests.py::test_callhistory_outgoing_multiple_months_and_calls (Test register outgoing call with multiple calls)

pass

1 / 1

task_2_basic_tests.py::test_phoneline_makecall_1_call (Test phoneline makecall with 1 call)

fail

0 / 1

task_2_basic_tests.py::test_phoneline_makecall_multiple_calls (Test phoneline makecall with multiple calls)

fail

0 / 1

task_2_basic_tests.py::test_phoneline_recvcall_1_call (Test phoneline recvcall with 1 call)

fail

0 / 1

task_2_basic_tests.py::test_phoneline_revcall_multiple_calls (Test phoneline recvcall with multiple calls)

fail

0 / 1

task_2_basic_tests.py::test_customer_makecall_1_phoneline (Test customer makercall with 1 phoneline)

fail

0 / 1

task_2_basic_tests.py::test_customer_makecall_multiple_phonelines (Test customer makecall with multiple phonelines)

fail

0 / 1

task_2_basic_tests.py::test_customer_recvcall_1_phoneline (Test customer recvcall with 1 phoneline)

```
fail
0 / 1
task_2_basic_tests.py::test_customer_recvcall_multiple_phonelines (Test customer recvcall with multiple phonelines)
fail
0 / 1
task_3_inheritance_tests.py::test_MTMContract_inheritance (Test inheritance in MTMContract subclasses)
pass
1 / 1
task_3_inheritance_tests.py::test_TermContract_inheritance (Test inheritance in TermContract subclasses)
pass
1 / 1
task_3_inheritance_tests.py::test_PrepaidContract_inheritance (Test inheritance in PrepaidContract subclasses)
pass
1 / 1
task_3_basic_tests.py::test_TermContract_new_month (Test TermContract new month)
pass
1 / 1
task_3_basic_tests.py::test_TermContract_bill_call_free_minutes (Test termContract bill_call using only free minutes)
pass
1 / 1
task_3_basic_tests.py::test_TermContract_bill_call_all_free_minutes (Test termConract bill_call using all free minutes)
pass
1 / 1
task_3_basic_tests.py::test_TermContract_bill_call_all_free_minutes_and_more (Test termConract bill_call using all free minutes and then billed minutes)
pass
1 / 1
task_3_basic_tests.py::test_TermContract_cancel_contract_no_return (Test termContract cancel contract no return)
pass
1 / 1
task_3_basic_tests.py::test_TermContract_cancel_contract_deposit_return (Test termContract cancel contract with left over deposit money returned)
pass
1 / 1
task_3_basic_tests.py::test_MTMContract_new_month (Test MTMContract new_month)
pass
1 / 1
```

```
task_3_basic_tests.py::test_MTMContract_bill_call (Test MTMContract bill call)
pass
1 / 1
task_3_basic_tests.py::test_MTMContract_cancel_contract (Test MTMContract cancel contract)
pass
1 / 1
task_3_basic_tests.py::test_prepaidContract_new_month ("Test prepaidContract initial new_month)
pass
1 / 1
task_3_basic_tests.py::test_prepaidContract_bill_call (Test prepaidContract bill call)
pass
1 / 1
task_3_basic_tests.py::test_prepaidContract_new_month_topup (Test prepaidContract bill call with a topup)
pass
1 / 1
task_3_basic_tests.py::test_prepaidContract_new_month_carryover (Test prepaidContract bill call with a carry over)
pass
1 / 1
task_3_basic_tests.py::test_prepaidContract_cancel_contract_owing (Test prepaidContract cancel contract with a positive balance)
fail
0 / 1
task_3_basic_tests.py::test_prepaidContract_cancel_contract_forfeit (Test prepaidContract cancel contract with leftover credit forfeited)
pass
1 / 1
task_3_docstring_test.py::test_MTM_class_docstring ( Tests that this class has at least a basic description. )
pass
1 / 1
task_3_docstring_test.py::test_MTM_init_new_month_docstrings ( Tests that the init and new_month methods have a docstring. )
fail
0 / 1
task_3_docstring_test.py::test_MTM_bill_call_cancel_contract_docstrings ( Tests that the bill_call and cancel_contract methods have a docstring. )
pass
1 / 1
task_3_docstring_test.py::test_TermContract_class_docstring ( Tests that this class has at least a basic description. )
pass
```

1 / 1
task_3_docstring_test.py::test_TermContract_init_new_month_docstrings (Tests that the init and new_month methods have a docstring.)
fail
0 / 1
task_3_docstring_test.py::test_TermContract_bill_call_cancel_contract_docstrings (Tests that the bill_call and cancel_contract methods have a docstring.)
fail
0 / 1
task_3_docstring_test.py::test_PrepaidContract_class_docstring (Tests that this class has at least a basic description.)
pass
1 / 1
task_3_docstring_test.py::test_PrepaidContract_init_new_month_docstrings (Tests that the init and new_month methods have a docstring.)
fail
0 / 1
task_3_docstring_test.py::test_PrepaidContract_bill_call_cancel_contract_docstrings (Tests that the bill_call and cancel_contract methods have a docstring.)
fail
0 / 1
task_3_tests.py::test_billing[mtm-singleCall_singleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[mtm-multipleCall_singleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[mtm-multipleCall_multipleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[mtm-multipleCall_singleMonth_withSMS] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[mtm-multipleCall_multipleMonth_withSMS] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass

1 / 1

task_3_tests.py::test_billing[mtm-one_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[mtm-two_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[mtm-six_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[mtm-large_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-singleCall_singleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-multipleCall_singleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-multipleCall_multipleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-multipleCall_singleMonth_withSMS] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-multipleCall_multipleMonth_withSMS] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-one_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-two_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-six_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[term-large_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[prepaid-singleCall_singleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[prepaid-multipleCall_singleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[prepaid-multipleCall_multipleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[prepaid-multipleCall_singleMonth_withSMS] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[prepaid-multipleCall_multipleMonth_withSMS] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[prepaid-one_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[prepaid-two_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass
1 / 1
task_3_tests.py::test_billing[prepaid-six_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[prepaid-large_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[Mixed-singleCall_singleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[Mixed-multipleCall_singleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[Mixed-multipleCall_multipleMonth] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[Mixed-multipleCall_singleMonth_withSMS] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[Mixed-multipleCall_multipleMonth_withSMS] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[Mixed-one_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[Mixed-two_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)
pass
1 / 1
task_3_tests.py::test_billing[Mixed-six_month_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_billing[Mixed-large_dataset] (Import a series of calls/sms objects and ensure the billing is correct. May also check for months which have no calls in the dataset.)

pass

1 / 1

task_3_tests.py::test_prepaid_topup[singleCall_singleMonth] (Test that the topup is performed correctly)

pass

1 / 1

task_3_tests.py::test_prepaid_topup[multipleCall_singleMonth] (Test that the topup is performed correctly)

pass

1 / 1

task_3_tests.py::test_prepaid_topup[multipleCall_multipleMonth] (Test that the topup is performed correctly)

pass

1 / 1

task_3_tests.py::test_prepaid_topup[multipleCall_singleMonth_withSMS] (Test that the topup is performed correctly)

pass

1 / 1

task_3_tests.py::test_prepaid_topup[multipleCall_multipleMonth_withSMS] (Test that the topup is performed correctly)

pass

1 / 1

task_3_tests.py::test_cancel_contract[mtm-one_month_dataset] (test the cancelling of the contract)

pass

1 / 1

task_3_tests.py::test_cancel_contract[mtm-two_month_dataset] (test the cancelling of the contract)

pass

1 / 1

task_3_tests.py::test_cancel_contract[mtm-six_month_dataset] (test the cancelling of the contract)

pass

1 / 1

task_3_tests.py::test_cancel_contract[mtm-large_dataset] (test the cancelling of the contract)

pass

1 / 1

task_3_tests.py::test_cancel_contract[term-one_month_dataset] (test the cancelling of the contract)

pass

```
1 / 1
task_3_tests.py::test_cancel_contract[term-two_month_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[term-six_month_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[term-large_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[prepaid-one_month_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[prepaid-two_month_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[prepaid-six_month_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[prepaid-large_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[Mixed-one_month_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[Mixed-two_month_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[Mixed-six_month_dataset] ( test the cancelling of the contract )
pass
1 / 1
task_3_tests.py::test_cancel_contract[Mixed-large_dataset] ( test the cancelling of the contract )
pass
1 / 1
```

task_3_tests.py::test_contract[Term] (Make sure the contracts are created with the correct set of parameters)

pass

1 / 1

task_3_tests.py::test_contract[MTM] (Make sure the contracts are created with the correct set of parameters)

pass

1 / 1

task_3_tests.py::test_contract[Prepaid] (Make sure the contracts are created with the correct set of parameters)

pass

1 / 1

task_4_basic_tests.py::test_CustomerFilter_apply_valid (Test customerFilter apply method proper filter)

pass

1 / 1

task_4_basic_tests.py::test_CustomerFilter_apply_invalid (test customer filter with an invalid filter)

pass

1 / 1

task_4_basic_tests.py::test_DurationFilter_less (Test the duration filter for less than filter string)

pass

1 / 1

task_4_basic_tests.py::test_DurationFilter_greater (Test the duration filter with a greater than filter string)

pass

1 / 1

task_4_basic_tests.py::test_DurationFilter_invalid (Test the duration filter with invalid filter string)

pass

1 / 1

task_4_basic_tests.py::test_DurationFilter_approx_greater (Test the duration filter with a valid string in the case wrong comparator was used for greater)

pass

1 / 1

task_4_basic_tests.py::test_DurationFilter_approx_less (Test the duration filter with a valid string in the case wrong comparator was used for less)

pass

1 / 1

task_4_basic_tests.py::test_LocationFilter_valid (Test LocationFilter apply with a valid filter string.)

pass

1 / 1

task_4_basic_tests.py::test_LocationFilter_invalid (Test LocationFilter apply with a valid filter string.)

pass

1 / 1

task_4_basic_tests.py::test_LocationFilter_approx_valid (Test LocationFilter apply in the case wrong comparator was used .)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid1-singleCall_singleMonth] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid1-multipleCall_singleMonth] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid1-multipleCall_multipleMonth] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid1-multipleCall_multipleMonth_withSMS] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid1-one_month_dataset] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid1-two_month_dataset] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid1-six_month_dataset] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid1-large_dataset] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

task_4_tests.py::test_Location_filter[Valid2-singleCall_singleMonth] (Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568)

pass

1 / 1

```
task_4_tests.py::test_Location_filter[Valid2-multipleCall_singleMonth] ( Test the location for
correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Valid2-multipleCall_multipleMonth] ( Test the location for
correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Valid2-multipleCall_multipleMonth_withSMS] ( Test the
location for correct operation under varying inputs -79.697878, 43.576959 -79.196382,
43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Valid2-one_month_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Valid2-two_month_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Valid2-six_month_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Valid2-large_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid1-singleCall_singleMonth] ( Test the location for
correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid1-multipleCall_singleMonth] ( Test the location for
correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid1-multipleCall_multipleMonth] ( Test the location for
correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid1-multipleCall_multipleMonth_withSMS] ( Test the
location for correct operation under varying inputs -79.697878, 43.576959 -79.196382,
43.799568 )
```

```
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid1-one_month_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid1-two_month_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid1-six_month_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid1-large_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid2-singleCall_singleMonth] ( Test the location for
correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid2-multipleCall_singleMonth] ( Test the location for
correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid2-multipleCall_multipleMonth] ( Test the location for
correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid2-multipleCall_multipleMonth_withSMS] ( Test the
location for correct operation under varying inputs -79.697878, 43.576959 -79.196382,
43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid2-one_month_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid2-two_month_dataset] ( Test the location for correct
operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
```

```
task_4_tests.py::test_Location_filter[Invalid2-six_month_dataset] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid2-large_dataset] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid3-singleCall_singleMonth] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid3-multipleCall_singleMonth] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid3-multipleCall_multipleMonth] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid3-multipleCall_multipleMonth_withSMS] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid3-one_month_dataset] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid3-two_month_dataset] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid3-six_month_dataset] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Location_filter[Invalid3-large_dataset] ( Test the location for correct operation under varying inputs -79.697878, 43.576959 -79.196382, 43.799568 )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Valid1-singleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
```

1 / 1

task_4_tests.py::test_Duration_filter[Valid1-multipleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid1-multipleCall_multipleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid1-multipleCall_multipleMonth_withSMS] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid1-one_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid1-two_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid1-six_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid1-large_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid2-singleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid2-multipleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid2-multipleCall_multipleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid2-multipleCall_multipleMonth_withSMS] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid2-one_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid2-two_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid2-six_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Valid2-large_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Invalid1-singleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Invalid1-multipleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Invalid1-multipleCall_multipleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Invalid1-multipleCall_multipleMonth_withSMS] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Invalid1-one_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Invalid1-two_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Duration_filter[Invalid1-six_month_dataset] (Test the location for correct operation under varying inputs)

pass

```
1 / 1
task_4_tests.py::test_Duration_filter[Invalid1-large_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid2-singleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid2-multipleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid2-multipleCall_multipleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid2-multipleCall_multipleMonth_withSMS] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid2-one_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid2-two_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid2-six_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid2-large_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid3-singleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid3-multipleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
```

```
1 / 1
task_4_tests.py::test_Duration_filter[Invalid3-multipleCall_multipleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid3-multipleCall_multipleMonth_withSMS] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid3-one_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid3-two_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid3-six_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Duration_filter[Invalid3-large_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid1-singleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid1-multipleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid1-multipleCall_multipleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid1-multipleCall_multipleMonth_withSMS] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid1-one_month_dataset] ( Test the location for correct operation under varying inputs )
fail
```

```
0 / 1
task_4_tests.py::test_Customer_filter[Valid1-two_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid1-six_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid1-large_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid2-singleCall_singleMonth] ( Test the location for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_Customer_filter[Valid2-multipleCall_singleMonth] ( Test the location for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_Customer_filter[Valid2-multipleCall_multipleMonth] ( Test the location for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_Customer_filter[Valid2-multipleCall_multipleMonth_withSMS] ( Test the location for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_Customer_filter[Valid2-one_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid2-two_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid2-six_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Valid2-large_dataset] ( Test the location for correct operation under varying inputs )
pass
```

1 / 1

task_4_tests.py::test_Customer_filter[Invalid1-singleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid1-multipleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid1-multipleCall_multipleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid1-multipleCall_multipleMonth_withSMS] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid1-one_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid1-two_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid1-six_month_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid1-large_dataset] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid2-singleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid2-multipleCall_singleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_Customer_filter[Invalid2-multipleCall_multipleMonth] (Test the location for correct operation under varying inputs)

pass

1 / 1
task_4_tests.py::test_Customer_filter[Invalid2-multipleCall_multipleMonth_withSMS] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid2-one_month_dataset] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid2-two_month_dataset] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid2-six_month_dataset] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid2-large_dataset] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid3-singleCall_singleMonth] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid3-multipleCall_singleMonth] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid3-multipleCall_multipleMonth] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid3-multipleCall_multipleMonth_withSMS] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid3-one_month_dataset] (Test the location for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid3-two_month_dataset] (Test the location for correct operation under varying inputs)
pass

```
1 / 1
task_4_tests.py::test_Customer_filter[Invalid3-six_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid3-large_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid4-singleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid4-multipleCall_singleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid4-multipleCall_multipleMonth] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid4-multipleCall_multipleMonth_withSMS] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid4-one_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid4-two_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid4-six_month_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_Customer_filter[Invalid4-large_dataset] ( Test the location for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Valid1-singleCall_singleMonth] ( Test all filters for correct operation under varying inputs )
pass
```

1 / 1

task_4_tests.py::test_stacked_filters[Valid1-multipleCall_singleMonth] (Test all filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_stacked_filters[Valid1-multipleCall_multipleMonth] (Test all filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_stacked_filters[Valid1-multipleCall_multipleMonth_withSMS] (Test all filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_stacked_filters[Valid1-one_month_dataset] (Test all filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_stacked_filters[Valid1-two_month_dataset] (Test all filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_stacked_filters[Valid1-six_month_dataset] (Test all filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_stacked_filters[Valid1-large_dataset] (Test all filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_stacked_filters[Valid2-singleCall_singleMonth] (Test all filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_stacked_filters[Valid2-multipleCall_singleMonth] (Test all filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_stacked_filters[Valid2-multipleCall_multipleMonth] (Test all filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_stacked_filters[Valid2-multipleCall_multipleMonth_withSMS] (Test all filters for correct operation under varying inputs)

fail

```
0 / 1
task_4_tests.py::test_stacked_filters[Valid2-one_month_dataset] ( Test all filters for correct
operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Valid2-two_month_dataset] ( Test all filters for correct
operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Valid2-six_month_dataset] ( Test all filters for correct
operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Valid2-large_dataset] ( Test all filters for correct operation
under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid1-singleCall_singleMonth] ( Test all filters for correct
operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid1-multipleCall_singleMonth] ( Test all filters for
correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid1-multipleCall_multipleMonth] ( Test all filters for
correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid1-multipleCall_multipleMonth_withSMS] ( Test all
filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid1-one_month_dataset] ( Test all filters for correct
operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid1-two_month_dataset] ( Test all filters for correct
operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid1-six_month_dataset] ( Test all filters for correct
operation under varying inputs )
pass
```

```
1 / 1
task_4_tests.py::test_stacked_filters[Invalid1-large_dataset] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid2-singleCall_singleMonth] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid2-multipleCall_singleMonth] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid2-multipleCall_multipleMonth] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid2-multipleCall_multipleMonth_withSMS] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid2-one_month_dataset] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid2-two_month_dataset] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid2-six_month_dataset] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid2-large_dataset] ( Test all filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_stacked_filters[Invalid3-singleCall_singleMonth] ( Test all filters for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_stacked_filters[Invalid3-multipleCall_singleMonth] ( Test all filters for correct operation under varying inputs )
fail
```

0 / 1
task_4_tests.py::test_stacked_filters[Invalid3-multipleCall_multipleMonth] (Test all filters for correct operation under varying inputs)
fail
0 / 1
task_4_tests.py::test_stacked_filters[Invalid3-multipleCall_multipleMonth_withSMS] (Test all filters for correct operation under varying inputs)
fail
0 / 1
task_4_tests.py::test_stacked_filters[Invalid3-one_month_dataset] (Test all filters for correct operation under varying inputs)
fail
0 / 1
task_4_tests.py::test_stacked_filters[Invalid3-two_month_dataset] (Test all filters for correct operation under varying inputs)
fail
0 / 1
task_4_tests.py::test_stacked_filters[Invalid3-six_month_dataset] (Test all filters for correct operation under varying inputs)
fail
0 / 1
task_4_tests.py::test_stacked_filters[Invalid3-large_dataset] (Test all filters for correct operation under varying inputs)
fail
0 / 1
task_4_tests.py::test_reversestacked_filters[Valid1-singleCall_singleMonth] (Test all filters with different ordering for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Valid1-multipleCall_singleMonth] (Test all filters with different ordering for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Valid1-multipleCall_multipleMonth] (Test all filters with different ordering for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Valid1-multipleCall_multipleMonth_withSMS] (Test all filters with different ordering for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Valid1-one_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)
pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid1-two_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid1-six_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid1-large_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid2-singleCall_singleMonth] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid2-multipleCall_singleMonth] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid2-multipleCall_multipleMonth] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid2-multipleCall_multipleMonth_withSMS] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid2-one_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid2-two_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid2-six_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Valid2-large_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

```
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid1-singleCall_singleMonth] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid1-multipleCall_singleMonth] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid1-multipleCall_multipleMonth] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid1-multipleCall_multipleMonth_withSMS] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid1-one_month_dataset] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid1-two_month_dataset] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid1-six_month_dataset] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid1-large_dataset] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid2-singleCall_singleMonth] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid2-multipleCall_singleMonth] ( Test all filters with different ordering for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_reversestacked_filters[Invalid2-multipleCall_multipleMonth] ( Test all filters with different ordering for correct operation under varying inputs )
pass
```

1 / 1

task_4_tests.py::test_reversestacked_filters[Invalid2-multipleCall_multipleMonth_withSMS] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Invalid2-one_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Invalid2-two_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Invalid2-six_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Invalid2-large_dataset] (Test all filters with different ordering for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_reversestacked_filters[Invalid3-singleCall_singleMonth] (Test all filters with different ordering for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_reversestacked_filters[Invalid3-multipleCall_singleMonth] (Test all filters with different ordering for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_reversestacked_filters[Invalid3-multipleCall_multipleMonth] (Test all filters with different ordering for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_reversestacked_filters[Invalid3-multipleCall_multipleMonth_withSMS] (Test all filters with different ordering for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_reversestacked_filters[Invalid3-one_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_reversestacked_filters[Invalid3-two_month_dataset] (Test all filters with different ordering for correct operation under varying inputs)

fail

```
0 / 1
task_4_tests.py::test_reversestacked_filters[Invalid3-six_month_dataset] ( Test all filters with different ordering for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_reversestacked_filters[Invalid3-large_dataset] ( Test all filters with different ordering for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_twostacked_filters[Valid1-singleCall_singleMonth] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid1-multipleCall_singleMonth] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid1-multipleCall_multipleMonth] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid1-multipleCall_multipleMonth_withSMS] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid1-one_month_dataset] ( Test two filters for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_twostacked_filters[Valid1-two_month_dataset] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid1-six_month_dataset] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid1-large_dataset] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid2-singleCall_singleMonth] ( Test two filters for correct operation under varying inputs )
fail
```

```
0 / 1
task_4_tests.py::test_twostacked_filters[Valid2-multipleCall_singleMonth] ( Test two filters for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_twostacked_filters[Valid2-multipleCall_multipleMonth] ( Test two filters for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_twostacked_filters[Valid2-multipleCall_multipleMonth_withSMS] ( Test two filters for correct operation under varying inputs )
fail
0 / 1
task_4_tests.py::test_twostacked_filters[Valid2-one_month_dataset] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid2-two_month_dataset] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid2-six_month_dataset] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Valid2-large_dataset] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Invalid1-singleCall_singleMonth] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Invalid1-multipleCall_singleMonth] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Invalid1-multipleCall_multipleMonth] ( Test two filters for correct operation under varying inputs )
pass
1 / 1
task_4_tests.py::test_twostacked_filters[Invalid1-multipleCall_multipleMonth_withSMS] ( Test two filters for correct operation under varying inputs )
pass
```

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid1-one_month_dataset] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid1-two_month_dataset] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid1-six_month_dataset] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid1-large_dataset] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid2-singleCall_singleMonth] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid2-multipleCall_singleMonth] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid2-multipleCall_multipleMonth] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid2-multipleCall_multipleMonth_withSMS] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid2-one_month_dataset] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid2-two_month_dataset] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid2-six_month_dataset] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid2-large_dataset] (Test two filters for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_twostacked_filters[Invalid3-singleCall_singleMonth] (Test two filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_twostacked_filters[Invalid3-multipleCall_singleMonth] (Test two filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_twostacked_filters[Invalid3-multipleCall_multipleMonth] (Test two filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_twostacked_filters[Invalid3-multipleCall_multipleMonth_withSMS] (Test two filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_twostacked_filters[Invalid3-one_month_dataset] (Test two filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_twostacked_filters[Invalid3-two_month_dataset] (Test two filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_twostacked_filters[Invalid3-six_month_dataset] (Test two filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_twostacked_filters[Invalid3-large_dataset] (Test two filters for correct operation under varying inputs)

fail

0 / 1

task_4_tests.py::test_durationstacked_filters[Valid1-singleCall_singleMonth] (Test stacking duration filter for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_durationstacked_filters[Valid1-multipleCall_singleMonth] (Test stacking duration filter for correct operation under varying inputs)

pass

```
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid1-multipleCall_multipleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid1-multipleCall_multipleMonth_withSMS] ( Test
stacking duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid1-one_month_dataset] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid1-two_month_dataset] ( Test stacking duration
filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid1-six_month_dataset] ( Test stacking duration
filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid1-large_dataset] ( Test stacking duration filter
for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid2-singleCall_singleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid2-multipleCall_singleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid2-multipleCall_multipleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid2-multipleCall_multipleMonth_withSMS] ( Test
stacking duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid2-one_month_dataset] ( Test stacking
duration filter for correct operation under varying inputs)
pass
```

```
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid2-two_month_dataset] ( Test stacking duration
filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid2-six_month_dataset] ( Test stacking duration
filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Valid2-large_dataset] ( Test stacking duration filter
for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid1-singleCall_singleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid1-multipleCall_singleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid1-multipleCall_multipleMonth] ( Test
stacking duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid1-multipleCall_multipleMonth_withSMS] (
Test stacking duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid1-one_month_dataset] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid1-two_month_dataset] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid1-six_month_dataset] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid1-large_dataset] ( Test stacking duration
filter for correct operation under varying inputs)
pass
```

```
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid2-singleCall_singleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid2-multipleCall_singleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid2-multipleCall_multipleMonth] ( Test
stacking duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid2-multipleCall_multipleMonth_withSMS] (
Test stacking duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid2-one_month_dataset] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid2-two_month_dataset] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid2-six_month_dataset] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid2-large_dataset] ( Test stacking duration
filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid3-singleCall_singleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid3-multipleCall_singleMonth] ( Test stacking
duration filter for correct operation under varying inputs)
pass
1 / 1
task_4_tests.py::test_durationstacked_filters[Invalid3-multipleCall_multipleMonth] ( Test
stacking duration filter for correct operation under varying inputs)
pass
```

1 / 1

task_4_tests.py::test_durationstacked_filters[Invalid3-multipleCall_multipleMonth_withSMS] (Test stacking duration filter for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_durationstacked_filters[Invalid3-one_month_dataset] (Test stacking duration filter for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_durationstacked_filters[Invalid3-two_month_dataset] (Test stacking duration filter for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_durationstacked_filters[Invalid3-six_month_dataset] (Test stacking duration filter for correct operation under varying inputs)

pass

1 / 1

task_4_tests.py::test_durationstacked_filters[Invalid3-large_dataset] (Test stacking duration filter for correct operation under varying inputs)

pass

1 / 1

sample_tests.py::test_customer_creation (Test for the correct creation of Customer, PhoneLine, and Contract classes)

pass

1 / 1

sample_tests.py::test_events (Test the ability to make calls, and ensure that the CallHistory objects are populated)

pass

1 / 1

sample_tests.py::test_contract_start_dates (Test the start dates of the contracts. Ensure that the start dates are the correct dates as specified in the given starter code.)

pass

1 / 1

sample_tests.py::test_filters (Test the functionality of the filters. We are only giving you a couple of tests here, you should expand both the dataset and the tests for the different types of applicable filters)

pass

1 / 1

-

Pyta customer.py

pass

10 / 10

-

Pyta phoneline.py

pass

10 / 10

- Pyta callhistory.py

  pass

  10 / 10

- Pyta filter.py

  pass

  10 / 10

- Pyta application.py

  pass

  10 / 10

- Pyta contract.py

  pass

  10 / 10