CAIRO SECURITY CLAN

STARKWARE UTILS

SECURITY ASSESEMENT REPORT

APRIL 2025

Prepared for

STARKWARE

Contents

1 About Cairo Security Clan 2

2 Disclaimer 2

3 Executive Summary 3

4 Summary of Audit 4
- 4.1 Scoped Files 4
- 4.2 Issues 4

5 Risk Classification 5

6 Issues by Severity Levels 6
- 6.1 Informational 6
  - 6.1.1 Lack of Support for CamelCase Naming in ERC20 Tokens 6
  - 6.1.2 Unchecked Return Values of ERC20 Transfers 6
- 6.2 Best Practices 7
  - 6.2.1 Missing Event Emission in `register_token()` 7

7 Test Evaluation 8
- 7.1 Compilation Output 8
- 7.2 Tests Output 8

1 About Cairo Security Clan

Cairo Security Clan is a leading force in the realm of blockchain security, dedicated to fortifying the foundations of the digital age. As pioneers in the field, we specialize in conducting meticulous smart contract security audits, ensuring the integrity and reliability of decentralized applications built on blockchain technology.

At Cairo Security Clan, we boast a multidisciplinary team of seasoned professionals proficient in blockchain security, cryptography, and software engineering. With a firm commitment to excellence, our experts delve into every aspect of the Web3 ecosystem, from foundational layer protocols to application-layer development. Our comprehensive suite of services encompasses smart contract audits, formal verification, and real-time monitoring, offering unparalleled protection against potential vulnerabilities.

Our team comprises industry veterans and scholars with extensive academic backgrounds and practical experience. Armed with advanced methodologies and cutting-edge tools, we scrutinize and analyze complex smart contracts with precision and rigor. Our track record speaks volumes, with a plethora of published research papers and citations, demonstrating our unwavering dedication to advancing the field of blockchain security.

At Cairo Security Clan, we prioritize collaboration and transparency, fostering meaningful partnerships with our clients. We believe in a customer-oriented approach, engaging stakeholders at every stage of the auditing process. By maintaining open lines of communication and soliciting client feedback, we ensure that our solutions are tailored to meet the unique needs and objectives of each project.

Beyond our core services, Cairo Security Clan is committed to driving innovation and shaping the future of blockchain technology. As active contributors to the ecosystem, we participate in the development of emerging technologies such as Starknet, leveraging our expertise to build robust infrastructure and tools. Through strategic guidance and support, we empower our partners to navigate the complexities of the blockchain landscape with confidence and clarity.

In summary, Cairo Security Clan stands at the forefront of blockchain security, blending technical prowess with a client-centric ethos to deliver unparalleled protection and peace of mind in an ever-evolving digital landscape. Join us in safeguarding the future of decentralized finance and digital assets with confidence and conviction.

2 Disclaimer

Disclaimer Limitations of this Audit:

This report is based solely on the materials and documentation provided by you to Cairo Security Clan for the specific purpose of conducting the security review outlined in the Summary of Audit and Scoped Files. The findings presented here may not be exhaustive and may not identify all potential vulnerabilities. Cairo Security Clan provides this review and report on an "as-is" and "as-available" basis. You acknowledge that your use of this report, including any associated services, products, protocols, platforms, content, and materials, occurs entirely at your own risk.

Inherent Risks of Blockchain Technology:

Blockchain technology remains in its developmental stage and is inherently susceptible to unknown risks and vulnerabilities. This review is specifically focused on the smart contract code and does not extend to the compiler layer, programming language elements beyond the reviewed code, or other potential security risks outside the code itself.

Report Purpose and Reliance:

This report should not be construed as an endorsement of any specific project or team, nor does it guarantee the absolute security of the audited smart contracts. No third party should rely on this report for any purpose, including making investment or purchasing decisions.

Liability Disclaimer:

To the fullest extent permitted by law, Cairo Security Clan disclaims all liability associated with this report, its contents, and any related services and products arising from your use. This includes, but is not limited to, implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Third-Party Products and Services:

Cairo Security Clan does not warrant, endorse, guarantee, or assume responsibility for any products or services advertised by third parties within this report, nor for any open-source or third-party software, code, libraries, materials, or information linked to, referenced by, or accessible through this report, its content, and related services and products. This includes any hyperlinked websites, websites or applications appearing on advertisements, and Cairo Security Clan will not be responsible for monitoring any transactions between you and third-party providers. It is recommended that you exercise due diligence and caution when considering any third-party products or services, just as you would with any purchase or service through any medium.

Disclaimer of Advice:

FOR THE AVOIDANCE OF DOUBT, THIS REPORT, ITS CONTENT, ACCESS, AND/OR USE, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHOULD NOT BE CONSIDERED OR RELIED UPON AS FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER PROFESSIONAL ADVICE.

3 Executive Summary

This document presents the security review performed by Cairo Security Clan on the Starkware Utils.

Starkware utils is a shared utility library for Starknet applications, providing common tools and abstractions used across the Starknet ecosystem.

The audit was performed using

- manual analysis of the codebase,
- automated analysis tools,
- simulation of the smart contract,
- analysis of edge test cases

3 points of attention, where 0 is classified as Critical, 0 is classified as High, 0 is classified as Medium,0 is classified as Low,2 are classified as Informational and 1 is classified as Best Practices. The issues are summarized in Fig. 1.

This document is organized as follows. Section 1 About Cairo Security Clan. Section 2 Disclaimer. Section 3 Executive Summary. Section 4 Summary of Audit. Section 5 Risk Classification. Section 6 Issues by Severity Levels. Section 7 Test Evaluation.

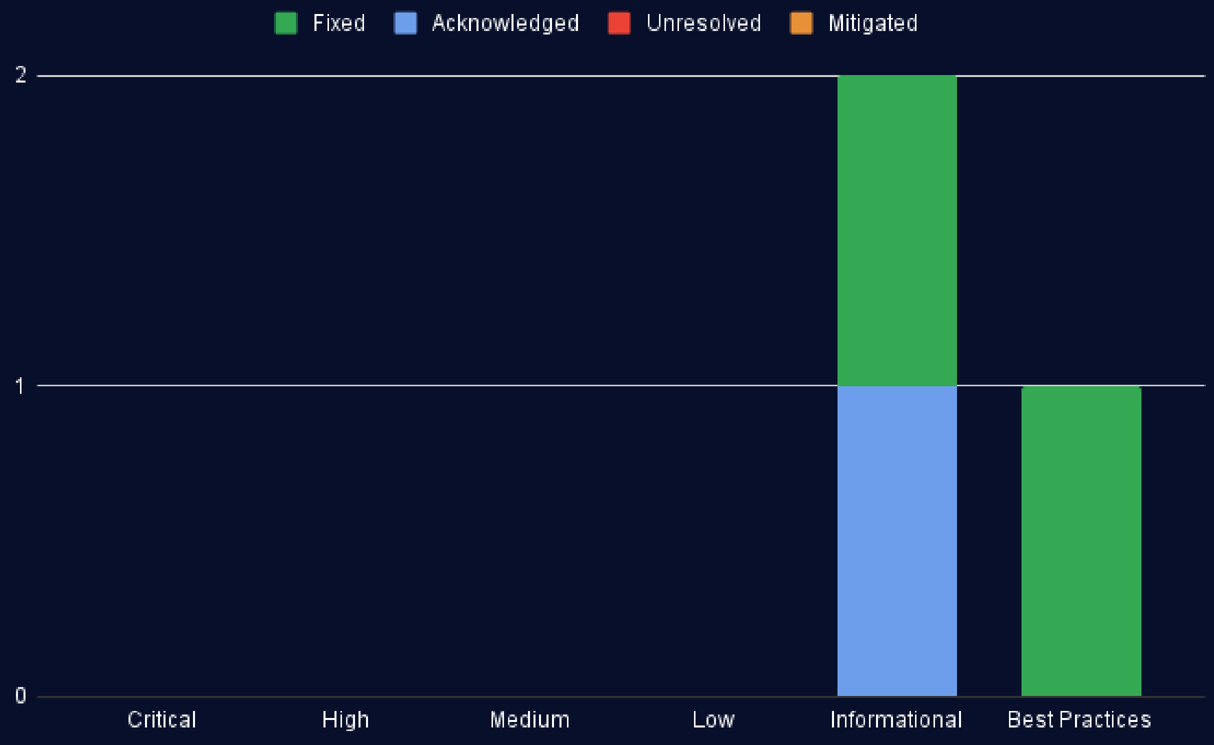

Fig 1: Distribution of issues: Critical (0), **High** (0), **Medium** (0), **Low** (0), **Informational** (2), **Best Practices** (1). **Distribution of status: Fixed** (2), **Acknowledged** (1), **Mitigated** (0), **Unresolved** (0).

4 Summary of Audit

Audit Type	Security Review
Cairo Version	2.11.2
Final Report	05/05/2025
Repository	starkware-libs/starkware-starknet-utils
Initial Commit Hash	43451ba97e2bbcffa34c1cf984b75cf1c92f6fa1
Final Commit Hash	142ba4fd8525368690969f623c682e33230ce866
Test Suite Assessment	High

4.1 Scoped Files

	Contracts
1	packages/utils/src/components.cairo
2	packages/utils/src/constants.cairo
3	packages/utils/src/erc20_mocks.cairo
4	packages/utils/src/errors.cairo
5	packages/utils/src/interfaces.cairo
6	packages/utils/src/iterable_map.cairo
7	packages/utils/src/lib.cairo
8	packages/utils/src/math.cairo
9	packages/utils/src/message_hash.cairo
10	packages/utils/src/tests.cairo
11	packages/utils/src/trace.cairo
12	packages/utils/src/types.cairo
13	packages/utils/src/utils.cairo
14	packages/utils/src/components/deposit/deposit.cairo
15	packages/utils/src/components/deposit/errors.cairo
16	packages/utils/src/components/deposit/events.cairo
17	packages/utils/src/components/deposit/interface.cairo
18	packages/utils/src/components/nonce/interface.cairo
19	packages/utils/src/components/nonce/mock_contract.cairo
20	packages/utils/src/components/nonce/nonce.cairo
21	packages/utils/src/components/nonce/test.cairo
22	packages/utils/src/components/pausable/interface.cairo
23	packages/utils/src/components/pausable/pausable.cairo
24	packages/utils/src/components/request_approvals/errors.cairo
25	packages/utils/src/components/request_approvals/interface.cairo
26	packages/utils/src/components/request_approvals/request_approvals.cairo
27	packages/utils/src/components/deposit.cairo
28	packages/utils/src/components/nonce.cairo
29	packages/utils/src/components/pausable.cairo
30	packages/utils/src/components/request_approvals.cairo
31	packages/utils/src/interfaces/identity.cairo
32	packages/utils/src/interfaces/mintable_token.cairo
33	packages/utils/src/math/abs.cairo
34	packages/utils/src/math/fraction.cairo
35	packages/utils/src/math/utils.cairo
36	packages/utils/src/trace/errors.cairo
37	packages/utils/src/trace/mock.cairo
38	packages/utils/src/trace/test.cairo
39	packages/utils/src/trace/trace.cairo
40	packages/utils/src/types/time.cairo
41	packages/utils/src/types/time/errors.cairo
42	packages/utils/src/types/time/time.cairo

4.2 Issues

	Findings	**Severity**	**Update**
1	Lack of support camel-case naming ERC20 tokens	Informational	Acknowledged
2	Unchecked return values of ERC20 transfer	Informational	Fixed
3	Missing event emission on `register_token()` function	Best Practices	Fixed

5 Risk Classification

The risk rating methodology used by Cairo Security Clan follows the principles established by the CVSS risk rating methodology. The severity of each finding is determined by two factors: **Likelihood** and **Impact**.

Likelihood measures how likely an attacker will uncover and exploit the finding. This factor will be one of the following values:

a) **High**: The issue is trivial to exploit and has no specific conditions that need to be met;

b) **Medium**: The issue is moderately complex and may have some conditions that need to be met;

c) **Low**: The issue is very complex and requires very specific conditions to be met.

When defining the likelihood of a finding, other factors are also considered. These can include but are not limited to Motive, opportunity, exploit accessibility, ease of discovery, and ease of exploit.

Impact is a measure of the damage that may be caused if an attacker exploits the finding. This factor will be one of the following values:

a) **High**: The issue can cause significant damage such as loss of funds or the protocol entering an unrecoverable state;

b) **Medium**: The issue can cause moderate damage such as impacts that only affect a small group of users or only a particular part of the protocol;

c) **Low**: The issue can cause little to no damage such as bugs that are easily recoverable or cause unexpected interactions that cause minor inconveniences.

When defining the impact of a finding other factors are also considered. These can include but are not limited to Data/state integrity, loss of availability, financial loss, and reputation damage. After defining the likelihood and impact of an issue, the severity can be determined according to the table below.

		Likelihood		
		High	**Medium**	**Low**
Impact	**High**	Critical	High	Medium
	Medium	High	Medium	Low
	Low	Medium	Low	Info/Best Practices

To address issues that do not fit a High/Medium/Low severity, Cairo Security Clan also uses three more finding severities: **Informational**, **Best Practices** and **Gas**

a) **Informational** findings do not pose any risk to the application, but they carry some information that the audit team intends to formally pass to the client;

b) **Best Practice** findings are used when some piece of code does not conform with smart contract development best practices;

c) **Gas** findings are used when some piece of code uses more gas than it should be or have some functions that can be removed to save gas.

6 Issues by Severity Levels

6.1 Informational

6.1.1 Lack of Support for CamelCase Naming in ERC20 Tokens

File(s): packages/utils/src/components/deposit/deposit.cairo

Description: The Deposit component is designed to interface with ERC20 tokens that expose their methods using `snake_case` naming conventions. However, many ERC20 tokens on Starknet still follow the `camelCase` naming style, particularly for functions such as `transferFrom`.

The current implementation assumes all token interfaces use `snake_case`, which leads to reverts when interacting with tokens that follow the legacy `camelCase` format.

```
token_contract
    .transfer_from(
        sender: caller_address,
        recipient: get_contract_address(),
        amount: unquantized_amount.into(),
    );
```

Recommendation(s): Enhance compatibility by supporting both `snake_case` and `camelCase` naming conventions for ERC20 token interactions.

Status: Acknowledged

Update from the client: This issue was acknowledged by the developer team.

6.1.2 Unchecked Return Values of ERC20 Transfers

File(s): packages/utils/src/components/deposit/deposit.cairo

Description: Within the Deposit component, ERC20 token transfers are performed using the `transfer()` and `transfer_from()` functions. These functions return a boolean indicating whether the transfer was successful. However, in the current implementation, the return values are not checked.

This can lead to unexpected behavior, particularly with tokens that do not revert on failure but instead return `false` to indicate unsuccessful execution.

```
token_contract
    .transfer_from(
        sender: caller_address,
        recipient: get_contract_address(),
        amount: unquantized_amount.into(),
    );

token_contract.transfer(recipient: caller_address, amount: unquantized_amount.into()); // @audit unsafe transfer
```

Recommendation(s): Consider asserting the boolean return values of `transfer()` and `transfer_from()` to ensure that the operations succeed and prevent silent failures.

Status: Fixed

Update from the client: Fixed in this commit.

6.2 Best Practices

6.2.1 Missing Event Emission in register_token()

File(s): packages/utils/src/components/deposit/deposit.cairo

Description: The register_token() function introduces a critical state change by registering a new token via mapping an asset_id to its token_address and quantum. However, the function does not emit any event to signal this registration. In blockchain development, emitting events on important state changes is considered best practice, as it facilitates off-chain indexing, monitoring, and transparency.

```
fn register_token(
    ref self: ComponentState<TContractState>,
    asset_id: felt252,
    token_address: ContractAddress,
    quantum: u64,
) {
    let (_token_address, _) = self.asset_info.read(asset_id);
    assert(_token_address.is_zero(), errors::ASSET_ALREADY_REGISTERED);
    assert(token_address.is_non_zero(), errors::INVALID_ZERO_TOKEN_ADDRESS);
    assert(quantum.is_non_zero(), errors::INVALID_ZERO_QUANTUM);
    self.asset_info.write(key: asset_id, value: (token_address, quantum));
}
```

Recommendation(s): Consider emitting an event, such as TokenRegistered, when a token is successfully registered. The event should include key fields like asset_id, token_address, and quantum for complete traceability.

Status: Fixed

Update from the client: Fixed in this commit.

7 Test Evaluation

7.1 Compilation Output

```
scarb build
 Downloading snforge_scarb_plugin v0.38.3
 Downloading snforge_std v0.38.3
 Downloading openzeppelin_finance v1.0.0
 Downloading openzeppelin v1.0.0
 Downloading openzeppelin_token v1.0.0
 Downloading openzeppelin_account v1.0.0
 Downloading openzeppelin_security v1.0.0
 Downloading openzeppelin_upgrades v1.0.0
 Downloading openzeppelin_utils v1.0.0
 Downloading openzeppelin_testing v2.0.0
 Downloading openzeppelin_merkle_tree v1.0.0
 Downloading openzeppelin_governance v1.0.0
 Downloading openzeppelin_presets v1.0.0
 Downloading openzeppelin_access v1.0.0
 Downloading openzeppelin_introspection v1.0.0
   Compiling lib(starkware_utils) starkware_utils v0.1.1 (/home/runner/work/049-Starknet-Staking/049-Starknet-Staking/contracts/packages/utils/Scarb.toml)
   Compiling starknet-contract(starkware_utils) starkware_utils v0.1.1 (/home/runner/work/049-Starknet-Staking/049-Starknet-Staking/contracts/packages/utils/Scarb.toml)
   Compiling starkware_utils_testing v1.0.0 (/home/runner/work/049-Starknet-Staking/049-Starknet-Staking/contracts/packages/testing/Scarb.toml)
    Finished `dev` profile target(s) in 2 minutes
```

7.2 Tests Output

```
scarb test
     Running test starkware_utils (SNFORGE_BACKTRACE=1 snforge test)
   Compiling test(starkware_utils_unittest) starkware_utils v0.1.1 (/home/runner/work/049-Starknet-Staking/049-Starknet-Staking/contracts/packages/utils/Scarb.toml)
    Finished `dev` profile target(s) in 43 seconds


Collected 149 test(s) from starkware_utils package
Running 149 test(s) from tests/
[PASS] starkware_utils::bit_mask::tests::test_u128_inverse_bit_mask (gas: ~85)
[PASS] starkware_utils::bit_mask::tests::test_u128_two_to_the (gas: ~92)
[PASS] starkware_utils::bit_mask::tests::test_u16_two_to_the (gas: ~11)
[PASS] starkware_utils::bit_mask::tests::test_u16_inverse_bit_mask (gas: ~12)
[PASS] starkware_utils::bit_mask::tests::test_u32_inverse_bit_mask (gas: ~22)
[PASS] starkware_utils::bit_mask::tests::test_u32_two_to_the (gas: ~21)
[PASS] starkware_utils::bit_mask::tests::test_u64_inverse_bit_mask (gas: ~43)
[PASS] starkware_utils::bit_mask::tests::test_u64_two_to_the (gas: ~40)
[PASS] starkware_utils::bit_set::tests::test_count (gas: ~25)
[PASS] starkware_utils::bit_set::tests::test_get (gas: ~2)
[PASS] starkware_utils::bit_set::tests::test_get_out_of_bounds (gas: ~3)
[PASS] starkware_utils::bit_set::tests::test_get_set_bits_indices (gas: ~7)
[PASS] starkware_utils::bit_set::tests::test_len (gas: ~2)
[PASS] starkware_utils::bit_mask::tests::test_u8_inverse_bit_mask (gas: ~6)
[PASS] starkware_utils::bit_mask::tests::test_u8_two_to_the (gas: ~6)
[PASS] starkware_utils::bit_set::tests::test_any (gas: ~17)
[PASS] starkware_utils::bit_set::tests::test_all (gas: ~22)
[PASS] starkware_utils::bit_set::tests::test_clear (gas: ~4)
[PASS] starkware_utils::bit_set::tests::test_bit_set_store_packing (gas: ~3)
[PASS] starkware_utils::bit_set::tests::test_none (gas: ~22)
[PASS] starkware_utils::bit_set::tests::test_set (gas: ~4)
[PASS] starkware_utils::bit_set::tests::test_set_all (gas: ~6)
[PASS] starkware_utils::bit_set::tests::test_set_lower_bound (gas: ~3)
[PASS] starkware_utils::bit_set::tests::test_set_out_of_bounds (gas: ~3)
[PASS] starkware_utils::bit_set::tests::test_set_upper_bound (gas: ~4)
[PASS] starkware_utils::bit_set::tests::test_t_into_bit_set (gas: ~1)
[PASS] starkware_utils::bit_set::tests::test_span_try_into_bit_set (gas: ~2)
```

```
[PASS] starkware_utils::bit_set::tests::test_toggle (gas: ~3)
[PASS] starkware_utils::bit_set::tests::test_toggle_out_of_bounds (gas: ~3)
[PASS] starkware_utils::components::nonce::test::test_nonce_getter (gas: ~2)
[PASS] starkware_utils::components::nonce::test::test_use_checked_nonce (gas: ~163)
[PASS] starkware_utils::components::nonce::test::test_use_checked_nonce_invalid_current (gas: ~171)
[PASS] starkware_utils::components::nonce::test::test_use_nonce (gas: ~163)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_add_new_implementation (gas: ~1721)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_add_new_implementation_not_upgrade_governor (gas: ~1391)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_get_upgrade_delay (gas: ~1387)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_initialize (gas: ~260)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_initialize_already_initialized (gas: ~261)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_remove_implementation (gas: ~1551)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_remove_implementation_not_upgrade_governor (gas: ~1391)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_replace_to_already_final (gas: ~1634)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_replace_to_expire_impl (gas: ~1755)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_replace_to_final (gas: ~1752)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_replace_to_nonfinal_impl (gas: ~1653)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_replace_to_not_upgrade_governor (gas: ~1391)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_replace_to_remove_impl_on_replace (gas: ~1760)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_replace_to_unknown_implementation (gas: ~1504)
[PASS] starkware_utils::components::replaceability::test::ReplaceabilityTests::test_replace_to_with_eic (gas: ~1551)
[PASS] starkware_utils::components::roles::test::test_initialize_with_zero_address (gas: ~199)
[PASS] starkware_utils::components::roles::test::test_register_app_role_admin (gas: ~1347)
[PASS] starkware_utils::components::roles::test::test_register_app_governor (gas: ~1459)
[PASS] starkware_utils::components::roles::test::test_register_governance_admin (gas: ~1347)
[PASS] starkware_utils::components::roles::test::test_register_operator (gas: ~1459)
[PASS] starkware_utils::components::roles::test::test_register_security_admin (gas: ~1347)
[PASS] starkware_utils::components::roles::test::test_register_security_agent (gas: ~1347)
[PASS] starkware_utils::components::roles::test::test_register_token_admin (gas: ~1459)
[PASS] starkware_utils::components::roles::test::test_register_upgrade_governor (gas: ~1347)
[PASS] starkware_utils::components::roles::test::test_remove_app_role_admin (gas: ~1264)
[PASS] starkware_utils::components::roles::test::test_remove_app_governor (gas: ~1376)
[PASS] starkware_utils::components::roles::test::test_remove_governance_admin (gas: ~1265)
[PASS] starkware_utils::components::roles::test::test_remove_operator (gas: ~1376)
[PASS] starkware_utils::components::roles::test::test_remove_security_agent (gas: ~1264)
[PASS] starkware_utils::components::roles::test::test_remove_security_admin (gas: ~1264)
[PASS] starkware_utils::components::roles::test::test_remove_token_admin (gas: ~1376)
[PASS] starkware_utils::components::roles::test::test_remove_upgrade_governor (gas: ~1264)
[PASS] starkware_utils::components::roles::test::test_renounce (gas: ~1243)
[PASS] starkware_utils::math::abs::tests::test_abs_i128 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_abs_i16 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_abs_i32 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_abs_i64 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_abs_i8 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_i16 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_i32 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_i64 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_i8 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_u128 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_u16 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_u32 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_u64 (gas: ~2)
[PASS] starkware_utils::math::abs::tests::test_wide_abs_diff_u8 (gas: ~2)
[PASS] starkware_utils::math::fraction::tests::fraction_denominator_test (gas: ~1)
[PASS] starkware_utils::math::fraction::tests::fraction_eq_test (gas: ~7)
[PASS] starkware_utils::math::fraction::tests::fraction_neg_test (gas: ~1)
[PASS] starkware_utils::math::fraction::tests::fraction_new_test_panic (gas: ~1)
```

```
[PASS] starkware_utils::math::fraction::tests::fraction_one_test (gas: ~3)
[PASS] starkware_utils::math::fraction::tests::fraction_numerator_test (gas: ~1)
[PASS] starkware_utils::math::fraction::tests::fraction_parial_ord_test (gas: ~24)
[PASS] starkware_utils::math::fraction::tests::fraction_zero_test (gas: ~1)
[PASS] starkware_utils::math::utils::tests::have_same_sign_test (gas: ~3)
[PASS] starkware_utils::math::utils::tests::u128_mul_wide_and_ceil_div_test (gas: ~3)
[PASS] starkware_utils::math::utils::tests::u128_mul_wide_and_ceil_div_test_panic (gas: ~2)
[PASS] starkware_utils::math::utils::tests::u128_mul_wide_and_div_test (gas: ~3)
[PASS] starkware_utils::math::utils::tests::u128_mul_wide_and_div_test_panic (gas: ~2)
[PASS] starkware_utils::math::utils::tests::u64_mul_wide_and_ceil_div_test (gas: ~2)
[PASS] starkware_utils::math::utils::tests::u64_mul_wide_and_ceil_div_test_panic (gas: ~2)
[PASS] starkware_utils::math::utils::tests::u64_mul_wide_and_div_test (gas: ~2)
[PASS] starkware_utils::math::utils::tests::u64_mul_wide_and_div_test_panic (gas: ~1)
[PASS] starkware_utils::tests::test_iterable_map::test_empty_map (gas: ~108)
[PASS] starkware_utils::tests::test_iterable_map::test_iterator (gas: ~1101)
[PASS] starkware_utils::tests::test_iterable_map::test_multiple_writes (gas: ~516)
[PASS] starkware_utils::tests::test_iterable_map::test_len (gas: ~1097)
[PASS] starkware_utils::tests::test_iterable_map::test_read_and_write (gas: ~501)
[PASS] starkware_utils::tests::test_message_hash::test_OffchainMessageHashImpl_Felt (gas: ~10)
[PASS] starkware_utils::tests::test_message_hash::test_StructHashStarknetDomainImpl (gas: ~2)
[PASS] starkware_utils::tests::test_message_hash::test_starknet_domain_type_hash (gas: ~1)
[PASS] starkware_utils::trace::test::test_at (gas: ~558)
[PASS] starkware_utils::trace::test::test_at_out_of_bounds (gas: ~3)
[PASS] starkware_utils::trace::test::test_insert (gas: ~568)
[PASS] starkware_utils::trace::test::test_insert_unordered_insertion (gas: ~360)
[PASS] starkware_utils::trace::test::test_is_empty (gas: ~358)
[PASS] starkware_utils::trace::test::test_latest (gas: ~556)
[PASS] starkware_utils::trace::test::test_latest_mutable (gas: ~556)
[PASS] starkware_utils::trace::test::test_latest_empty_trace (gas: ~3)
[PASS] starkware_utils::trace::test::test_latest_mutable_empty_trace (gas: ~3)
[PASS] starkware_utils::trace::test::test_length (gas: ~555)
[PASS] starkware_utils::trace::test::test_penultimate (gas: ~556)
[PASS] starkware_utils::trace::test::test_length_mutable (gas: ~555)
[PASS] starkware_utils::types::time::time::tests::test_days_overflow (gas: ~2)
[PASS] starkware_utils::trace::test::test_penultimate_not_exist (gas: ~3)
[PASS] starkware_utils::types::time::time::tests::test_time_add (gas: ~3)
[PASS] starkware_utils::types::time::time::tests::test_time_days (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_time_seconds (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_time_now (gas: ~8)
[PASS] starkware_utils::types::time::time::tests::test_time_weeks (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_add (gas: ~3)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_div (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_add_overflow (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_div_by_bigger (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_div_by_zero (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_into (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_eq (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_is_zero (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_is_non_zero (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_lt (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_sub (gas: ~3)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_sub_underflow (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timedelta_zero (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_add_assign (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_add_assign_overflow (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_add_overflow (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_eq (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_into (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_is_non_zero (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_lt (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_is_zero (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_sub_underflow (gas: ~2)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_sub (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_timestamp_zero (gas: ~1)
[PASS] starkware_utils::types::time::time::tests::test_weeks_overflow (gas: ~2)
Tests: 149 passed, 0 failed, 0 skipped, 0 ignored, 0 filtered out
```