Tuff Verification Certificate

Compiler version: 0.1.0 · Schema: 1.0

Issued: 2026-02-23T19:56:23.544Z

Compilation Outcome

Outcome: **PASSED**

Source Files

File	SHA-256
factorial.tuff	`9c1fe5adb691adff6fb349cc6bee4c61eaa0ec525c5b5187934c510145eeaf5a`

Combined SHA-256: `5fb89aece9cfe53b04be2cd05811d8cd539c82514438c97c7688409a94ab3b64`

Safety Properties

The following eight properties are enforced by the Tuff compiler for every successfully compiled program:

Property	Description	Enforcing Pass
No Buffer Overflows	Every array access is statically proven to lie within the bounds of its allocation. Accesses whose bounds cannot be proven at compile time are rejected with a compile-time error.	typecheck
No Null Dereferences	Nullable pointer types must be guarded before dereferencing. The type system tracks nullability and rejects any unguarded dereference of a nullable value.	typecheck
No Integer Overflow / Underflow	Arithmetic operations on fixed-width integer types are checked for overflow and underflow at compile time where possible. Expressions that could silently wrap are rejected.	typecheck
No Division by Zero	Integer division operations where the	typecheck

	divisor cannot be statically proven non-zero are rejected by the compiler.	
No Modulo by Zero	Integer modulo operations where the divisor cannot be statically proven non-zero are rejected by the compiler.	typecheck
No Data Races	The ownership and borrowing system ensures that mutable state is accessed by at most one part of the program at a time. Concurrent aliased mutation is structurally impossible in well-typed Tuff programs.	borrowcheck
No Use-After-Free / Double-Free	The borrow checker tracks ownership of every heap allocation. Reading or writing a moved value, and freeing the same allocation more than once, are both compile-time errors.	borrowcheck
No Undefined Control Flow (No Panics)	Well-typed Tuff programs do not contain reachable panic paths. The combination of the type checker and borrow checker eliminates the classes of runtime errors — out-of-bounds, null, overflow, zero-division, and bad aliasing — that are the root cause of panics in safe code.	typecheck, borrowcheck

What This Certificate Asserts

This certificate records that the Tuff compiler successfully parsed, type-checked, and borrow-checked the listed source file(s) under the strictSafety=true compilation mode. If compilationOutcome.success is true, the source satisfies all eight safety properties listed in this document as defined by the Tuff language specification §9.2.

What This Certificate Does Not Assert

This certificate does not assert the absence of logical errors, incorrect algorithms, business-logic defects, or security vulnerabilities outside the eight listed properties. It does not verify that the program produces correct output for any given input, nor that external C code called via FFI is itself safe. It does not constitute a formal mathematical proof; it records the outcome of a structural compiler analysis. A failed compilationOutcome means that one or more safety properties could not be established; the diagnosticCodes field identifies which checks triggered.

Generated by tuffc 0.1.0