SuperCoding

Boogle Project

Test Case

Version <1.1>

Revision History

Date	Version	Description	Author
29/04/24	1.0	Initial Draft	Brandon Dodge
02/05/24	1.1	Final Draft	Brandon Dodge Lisa Phan

Table of Contents

Test Case

1. Purpose

This Test Case Specification document outlines the testing protocols for the Boolean Calculator software. It details the specific test cases to validate the functionality and robustness of the software in evaluating boolean expressions. This document is intended to ensure that all features behave as expected under various scenarios.

2. Test case identifier

A. Testing single operation

Test Case ID	TC-01
Description	Testing AND
Test Data	T & T
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-02
Description	Testing AND
Test Data	T & F
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-03
Description	Testing AND
Test Data	F & F
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-04
Description	Testing OR
Test Data	T | T
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-05
Description	Testing OR
Test Data	T | F
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-06
Description	Testing OR
Test Data	F | F
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-07
Description	Testing NOT
Test Data	T ! F
Expected result	Error: Invalid syntax for binary operation
Actual result	Error: Invalid syntax for binary operation
Status	Pass

Test Case ID	TC-08
Description	Testing NAND
Test Data	T @ F
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-09
Description	Testing XOR
Test Data	T $ T
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-10
Description	Testing XOR
Test Data	T $ F
Expected result	True
Actual result	True
Status	Pass

B. Expressions contains two operators

Test Case ID	TC-11
Description	Testing &, &
Test Data	T & F & T
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-12
Description	Testing &, &
Test Data	F & T & F
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-13
Description	Testing |, |
Test Data	T | F | T
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-14
Description	Testing !, !
Test Data	T ! F ! T
Expected result	Error: Invalid syntax for binary operation
Actual result	Error: Invalid syntax for binary operation
Status	Pass

Test Case ID	TC-15
Description	Testing @, @
Test Data	T @ F @ T
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-16
Description	Testing $, $
Test Data	T $ F $ T
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-17
Description	Testing &, |
Test Data	T & F | T
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-18
Description	Testing |, !
Test Data	T | F ! T
Expected result	Error: Invalid syntax for binary operation
Actual result	Error: Invalid syntax for binary operation
Status	Pass

Test Case ID	TC-19
Description	Testing !, @
Test Data	T ! F @ T
Expected result	Error: Invalid syntax for binary operation
Actual result	Error: Invalid syntax for binary operation
Status	Pass

Test Case ID	TC-20
Description	Testing @, $
Test Data	T @ F $ T
Expected result	False
Actual result	False
Status	Pass

C. Complex expressions

Test Case ID	TC-21
Description	Testing a complex expression that contains different logical operators
Test Data	(((((T | F) & F) | (T & (T | F))) @ (T @ T)) $ (! (T | F)))
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-22
Description	Testing a complex expression that contains different logical operators
Test Data	((F $ ((T | F) & (F @ (T | F)))) | (T $ (T & F)))
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-23
Description	Testing a complex expression that contains different logical operators
Test Data	(((! (T $ F)) & (T @ T)) | ((F | T) & (T $ T)))
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-24
Description	Testing a complex expression that contains different logical operators
Test Data	(((T @ T) $ (F @ T)) | ((!T) & (T | (!T))))
Expected result	True
Actual result	The result of the expression is True
Status	Pass

Test Case ID	TC-25
Description	Testing a complex expression that contains different logical operators
Test Data	((F @ T) $ (T | (F & F))) & (T & (T @ (!T)))
Expected result	False
Actual result	The result of the expression is False
Status	Pass

Test Case ID	TC-26
Description	Testing a complex expression that contains different logical operators
Test Data	!(T & (F | T)) @ (!F $ (T | F))
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-27
Description	Testing a complex expression that contains different logical operators
Test Data	T & F | T @ F $!(T | F)
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-28
Description	Testing a complex expression that contains different logical operators
Test Data	(T & F) | (T @ F) & !(T $ F)
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-29
Description	Testing a complex expression that contains different logical operators
Test Data	((T | F) & (T & F)) | (!T & (T | F))
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-30
Description	Testing a complex expression that contains different logical operators
Test Data	T & F | T @ F $!(T | F) & !(F @ T)
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-31
Description	Testing a complex expression that contains different logical operators
Test Data	!(!(T & F) & !(T | F)) | (!(F | T) & !(T & F))
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-32
Description	Testing a complex expression that contains different logical operators
Test Data	(!(T & F) | !(T | F)) & ((!T & (F | T)) | !(F & T))
Expected result	True
Actual result	True
Status	Pass

Test Case ID	TC-33
Description	Testing a complex expression that contains different logical operators
Test Data	(T | (F & (T & F))) & ((T & F) | (T & F))
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-34
Description	Testing a complex expression that contains different logical operators
Test Data	(!(!(T & F) | !(T | F))) & (!(F | T) | !(T & F))
Expected result	False
Actual result	False
Status	Pass

Test Case ID	TC-35
Description	Testing a complex expression that contains different logical operators
Test Data	(T | F) & (T & F) $ (T @ F) | (F & T)
Expected result	True
Actual result	True
Status	Pass

D. Invalid Expressions

Test Case ID	TC-36
Description	Testing encountering an invalid syntax
Test Data	!&F
Expected result	Error: Invalid syntax
Actual result	Error: Invalid syntax for NOT operation.
Status	Pass

Test Case ID	TC-37
Description	Testing encountering an invalid character
Test Data	T ? T
Expected result	Error: Invalid character ?
Actual result	Error: Invalid character ?
Status	Pass

Test Case ID	TC-38
Description	Testing encountering a mismatched parentheses error
Test Data	(T&F|F
Expected result	Error: Mismatched parentheses
Actual result	Error: Mismatched parentheses
Status	Pass

Test Case ID	TC-39
Description	Testing encountering an expression with only one operand
Test Data	T = !(T & T)
Expected result	Error: Variable defined in terms of itself
Actual result	Error: Invalid character =
Status	Failed

Test Case ID	TC-40
Description	Testing encountering an empty expression
Test Data	
Expected result	Error: Invalid expression
Actual result	Error: Invalid expression
Status	Pass

Test Case ID	TC-41
Description	Testing encountering an expression with double operator
Test Data	T && & F
Expected result	Error: Invalid syntax
Actual result	Error: Invalid syntax for binary operation
Status	Pass

Test Case ID	TC-42
Description	Testing encountering an expression with unassigned variables
Test Data	X | Y
Expected result	Error: Invalid character
Actual result	Error: Invalid character X
Status	Pass

Test Case ID	TC-43
Description	Testing encountering an expression with inconsistent characters
Test Data	True | F
Expected result	Error: Invalid syntax
Actual result	Error: Invalid syntax
Status	Pass

Test Case ID	TC-44
Description	Testing encountering an expression missing operator
Test Data	T F
Expected result	Error: Invalid expression
Actual result	Error: Invalid expression
Status	Pass

Test Case ID	TC-45
Description	Testing encountering an expression missing operand
Test Data	$
Expected result	Error: Invalid expression
Actual result	Error: Invalid syntax
Status	Failed

Test Case ID	TC-46
Description	Testing encountering an expression with invalid character
Test Data	NOT F
Expected result	Error: Invalid character
Actual result	Error: Invalid character
Status	Pass

Test Case ID	TC-47
Description	Testing encountering an expression with inconsistent characters
Test Data	T & | F
Expected result	Error: Invalid syntax for binary operation
Actual result	Error: Invalid syntax for binary operation
Status	Pass

Test Case ID	TC-48
Description	Testing encountering an expression missing operator
Test Data	T & (F T)
Expected result	Error: Invalid expression
Actual result	Error: Invalid expression
Status	Pass

Test Case ID	TC-49
Description	Testing encountering an expression with invalid character
Test Data	T & (X | F)
Expected result	Error: Invalid character
Actual result	Error: Invalid character
Status	Pass

Test Case ID	TC-50
Description	Testing encountering an expression missing operand
Test Data	T & ()
Expected result	Error: Invalid syntax for binary operation
Actual result	Error: Invalid syntax for binary operation
Status	Pass

3. Test item

- Item: Boolean Expression Evaluation
- Features to be tested:
  - Input of boolean expressions using logical operators (AND, OR, NOT, etc.)
  - Evaluation of boolean expressions
  - Display of results (true/false)
  - Saving and loading expressions
- References:
  - Requirements Specification: Refer to the Software Requirements Document.
  - Design Specification: Refer to the Software Architecture Document.
  - User Guide: Refer to the Boolean Calculator User Manual.

4. Input specifications

- Inputs:
  - Direct inputs: Boolean expressions entered as text, e.g., (1 && 0), ((1 && 1) || (0 && !1))
  - Indirect inputs: Use of predefined variables and functions within expressions.
- Relationships:
  - Inputs must adhere to valid boolean syntax and supported operators.

5. Output specifications

- Expected Outputs:
  - For (1 && 0), expect 0 (False).
  - For ((1 && 1) || (0 && !1)), expect 1 (True).
  - Outputs should match the calculated result of the boolean expression.

6. Environmental needs

6.1.1 Hardware

- No specific hardware requirements.
- Hardware must be capable of handling any basic operating system.

6.1.2 Software

- Operating System: Compatible with Windows, macOS, and Linux.
- Required Software: None beyond the Boolean Calculator software itself.
- System Requirement: Must be capable of running .exe files.

6.1.3 Other

- Internet connection required for downloading the software.

7. Special procedural requirements

- Setup: Ensure the software is installed and running on a supported operating system.
- Operator Intervention: Minimal; users must manually enter expressions and initiate evaluation.
- Output Determination Procedures: Compare the software output directly against expected results.
- Wrap Up: Check for any errors logged during the test and ensure all temporary files are cleared.

8. Intercase dependencies

- Prior Test Cases:
  - TC-00: Installation and Setup Test to ensure the software is correctly installed and operational.
- Dependency Nature:
  - TC-01 depends on TC-00 for the environment setup and initial software functionality check.