

---

**Tech Titans**

---

**Boolean Logic Simulator**

**Test Case Specifications**

**Version 1.3**

Boolean Logic Simulator	Version: 1.3
Test Case Specifications: the Ideal input for the program is a boolean expression using only the characters (T, F, @, \$,  , &, !), with valid matching parentheses	Date: 01/05/2024

## Revision History

Date	Version	Description	Author
30/04/2024	1.0	Initial work on the document	Cole Charpentier
01/05/2024	1.1	Finished the purpose section, suggested edits to sections 6-8, added rows to table and added given test inputs	Cole Charpentier
01/05/2024	1.2	Ran group test cases and filled table	Justin Daugherty
01/05/2024	1.3	Additional test cases	Caden LeCluyse

Boolean Logic Simulator	Version: 1.3
Test Case Specifications: the Ideal input for the program is a boolean expression using only the characters (T, F, @, \$,  , &, !), with valid matching parentheses	Date: 01/05/2024

## Table of Contents

1. Purpose	4
2. Continuous Mode	4
3. Test Mode	6

Boolean Logic Simulator	Version: 1.3
Test Case Specifications: the Ideal input for the program is a boolean expression using only the characters (T, F, @, \$,  , &, !), with valid matching parentheses	Date: 01/05/2024

## Test Case

### 1. Purpose.

The purpose of this Test Case Specifications document is to document and define the test cases for the Boolean Logic Simulator. It provides a structured outline of the test scenarios used to verify that the Boolean Logic Simulator behaves as expected when given various different inputs. The test cases are represented in a table and have clear identifiers, a description, the input, expected output, actual output, and whether the input passed or failed the test.

### 2. Continuous Mode

The following test cases were ran in continuous mode, entering by hand.

Test Case ID	Test Case Description	Input	Expected Output	Actual Output	Pass/Fail Status
TC01	Valid	(T   F) \$ F	True	True	Pass
TC02	Valid	! (T & T)	False	False	Pass
TC03	Valid	(F @ T)   (T @ F)	True	True	Pass
TC04	Valid	(T \$ T) & F	False	False	Pass
TC05	Valid	! F   ! T	True	True	Pass
TC06	Valid	(((((T   F) & F)   (T & (T   F))) @ (T @ T)) \$ (! (T   F)))	True	True	Pass
TC07	Valid	((F \$ ((T   F) & (F @ (T   F))))   (T \$ (T & F)))	True	True	Pass
TC08	Valid	(((! (T \$ F)) & (T @ T))   ((F   T) & (T \$ T)))	False	False	Pass
TC09	Valid	((((T @ T) \$ (F @ T))   ((!T) & (T   (!T))))	True	True	Pass
TC10	Valid	((F @ T) \$ (T   (F & F))) & (T & (T @ (!T)))	False	Fasle	Pass
TC11	Invalid, missing operand	! & T	Error: NOT applied after value	Error: NOT applied after value	Pass
TC12	Invalid, unknown operator	T ? T	Error: Expected &,  , !, @, \$, received ?	Error: Expected &,  , !, @, \$, received ?	Pass
TC13	Missing operator	(T  ) F	Error: Missing operator!	Error: Missing operator!	Pass

Boolean Logic Simulator	Version: 1.3
Test Case Specifications: the Ideal input for the program is a boolean expression using only the characters (T, F, @, \$,  , &, !), with valid matching parentheses	Date: 01/05/2024

TC14	Incorrect symbol '='	T = !(T & T)	Error: Expected &, !, @, \$, received =	Error: Expected &, !, @, \$, received =	Pass
TC15	Invalid, empty expression		Error: Empty expression received!	Error: Empty expression received!	Pass
TC16	Invalid, double operator	T && & F	Error: Two consecutive operators detected: & and &	Error: Two consecutive operators detected: & and &	Pass
TC17	Invalid, missing truth values	X   Y	Error: expected T or F, received: Y	Error: expected T or F, received: Y	Pass
TC18	Invalid, inconsistent characters	True   F	Error: expected T or F, received: e	Error: expected T or F, received: e	Pass
TC19	Invalid, operators after operand	True!	Error: expression ends with NOT!	Error: expression ends with NOT!	Pass
TC20	Invalid, invalid characters	a & b	Error: expected T or F, received: b	Error: expected T or F, received: b	Pass
TC21	Valid	(T&T)&T	True	True	Pass
TC22	Valid	(T F) (T&F)	True	True	Pass
TC23	Valid	(T&(F T)@(F (T&T)))	False	False	Pass
TC24	Valid	((T@F)\$ (F T))&((T&(F@T)) (F\$T))	False	False	Pass
TC25	Valid	((T@F)\$T) ((T&(F@T))@(F T))&(T F)	False	False	Pass
TC26	Valid	((T@(F&F))@(F T))&((T F)@(F&T))	False	False	Pass
TC27	Valid	(T@(F&T)) ((T F)@(F T))	True	True	Pass
TC28	Statement ends with not	(!T)!	Error: expression ends with NOT!	Error: expression ends with NOT!	Pass
TC29	Invalid: only whitespace characters		Error expression contains only spaces	Error expression contains only spaces	Pass
TC30	Valid	T & F	False	False	Pass

Boolean Logic Simulator	Version: 1.3
Test Case Specifications: the Ideal input for the program is a boolean expression using only the characters (T, F, @, \$,  , &, !), with valid matching parentheses	Date: 01/05/2024

## 2. Test Mode

The following tests were ran in test mode, using the -t flag.

Test Case ID	Test Case Description	Input	Expected Output	Actual Output	Pass/Fail Status
TC31	Ending with open parentheses	(T   F) & (T & F)(	Expression ends with open parentheses!	Expression ends with open parentheses!	Pass
TC32	Beginning with closed parentheses	)(T \$ F) & T	Expression begins with closed parentheses!	Expression begins with closed parentheses!	Pass
TC33	Ending with NOT	T & F   (T @ F)!	Expression ends with NOT!	Expression ends with NOT!	Pass
TC34	Consecutive operands	(T   F) \$ !T   (TF)	Two consecutive operands detected: T and F	Two consecutive operands detected: T and F	Pass
TC35	Empty parentheses	(T @ F) \$ T   ( )	Empty parentheses detected!	Empty parentheses detected!	Pass
TC36	Missing operator	(T & F) (T \$ T)	Missing operator!	Missing operator!	Pass
TC37	Valid	((T@F)\$ (F T))&((T&(F@T)) (F\$T)) & (T@(F&T)) ((T F)@(F T))	False	False	Pass
TC38	Ending with NOT	(T   F) \$ T   (T\$F)!	Expression ends with NOT!	Expression ends with NOT!	
TC39	Valid	(T \$ F) & F   (F @ T)	True	True	Pass
TC40	Invalid character	!X & F   T \$ z	Expected T or F, received: z	Expected T or F, received: z	Pass
TC41	Empty parentheses	( )	Empty parentheses detected!	Empty parentheses detected!	Pass
TC42	Valid, simple expression	T @ F	True	True	Pass
TC43	Valid, simple expression	T \$ F	True	True	Pass
TC44	Valid, simple expression	T \$ T	False	False	Pass
TC45	Missing operator	( T F)	Two consecutive	Two consecutive	Pass

Boolean Logic Simulator	Version: 1.3
Test Case Specifications: the Ideal input for the program is a boolean expression using only the characters (T, F, @, \$,  , &, !), with valid matching parentheses	Date: 01/05/2024

			operands detected: T and F	operands detected: T and F	
TC46	Not after variable	(T & F)   (F! @ T)	NOT applied after value!	NOT applied after value!	Pass
TC47	Valid	(F @ T)   (T @ F) \$ F	True	True	Pass
TC48	Valid	F   F	False	False	Pass
TC49	Valid	((T@F)\$ (F!T))&((T&(F@T)) (F\$T)) & (T@ (F&T)) ( (T!F)@ (F!T)) & (T   F)	False	False	Pass
TC50	White space		Expression contains only spaces!	Expression contains only spaces!	Pass
TC51	Missing closed parentheses	(( (T @ F) @ T)	Missing closing parentheses!	Missing closing parentheses!	Pass
TC52	Missing open parentheses	F   (T & F) & F)	Missing open parentheses!	Missing open parentheses!	Pass
TC53	Invalid operator	F & T \$ F * (F @ T)	Expected &,  , !, @, \$, received: *	Expected &,  , !, @, \$, received: *	Pass
TC54	Beginning with operator	& T@F \$ T   T	Expression begins with an operator	Expression begins with an operator	Pass
TC55	Ending with operator	((F   !T) @	Expression ends with an operator!	Expression ends with an operator!	Pass
TC56	Not added to beginning of TC24	!((T@F)\$ (F!T))&((T &(F@T)) (F\$T))	True	True	Pass
TC57	Valid	F @ T   !T	True	True	Pass
TC58	Empty parentheses	((((( ((((((( )))))))))	Empty parentheses detected!	Empty parentheses detected!	Pass
TC59	NOT after variable	F @ T   (F & T!)	NOT applied after value!	NOT applied after value!	Pass
TC60	Missing operand	((!F   T) @)	Missing an operand!	Missing an operand!	Pass
TC61	Valid simple expression	T @ T	False	False	Pass
TC62	Valid expression	(( (T@ (F&F))@ (F!T)) & ((T!F)@ (F&T))) & ((( (T   F) & F)   (T &	False	False	Pass

Boolean Logic Simulator	Version: 1.3
Test Case Specifications: the Ideal input for the program is a boolean expression using only the characters (T, F, @, \$,  , &, !), with valid matching parentheses	Date: 01/05/2024

		(T   F)) @ (T @ T) \$ (! (T   F))			
--	--	--------------------------------------	--	--	--