User Manual

Truthy™

Introduction:

Truthy is a Boolean logic simulator that allows users to input Boolean expressions. A Boolean expression is a mathematical expression that only operates on binary values known as Booleans, where 0 = false, 1 = true. The program will evaluate the expression for the user and optionally show the truth table.

Installation:

You can download the latest version of Truthy by going to this link and downloading the executable.

Getting Started:

Launch the application found at the link. From there a command line will appear and allow you to type in your Boolean expression.

Valid Operators: Truthy™ only supports True and False values that are to be represented by T and F respectively. Giving any other input will result in error.

Operand	**Truthy™ Equivalent**
0	F
1	T
true	T
false	F

Valid Operands:

Logical Operator	**Truthy™ equivalent**
NOT	!
AND	&
OR	|
NAND	@
XOR	$

Parenthesis: Boolean expressions have order of operations. Truthy™ Allows for users to use parenthesis to manage order of operations.

Additional Features (Beta):

Exit: typing exit into the terminal will close the application

Help: typing help into the terminal will show a list of operators and examples

Truth Table (Beta): typing ‘table’ before your Boolean expression will print a truth table. Note: Since variables are not implemented as of 4/25/2024, when you input F/T. You can think of T as var A, and F as var B. This means that 2 variable truth tables are the only kind implemented as of now.

Glossary:

NOT: does the logical inverse

AND: only evaluates true when both statements are true. I.E. When A is true, and B is true

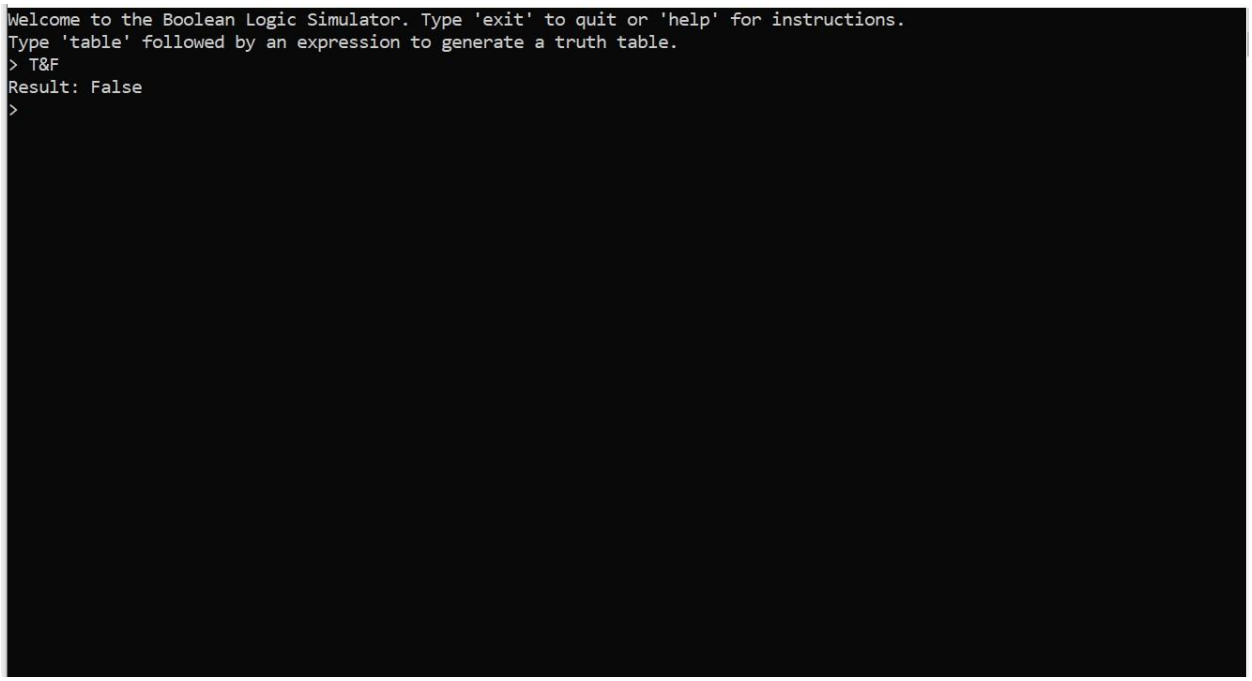
OR: evaluate true when either statements are true. I.E. When A is true, or B is true.

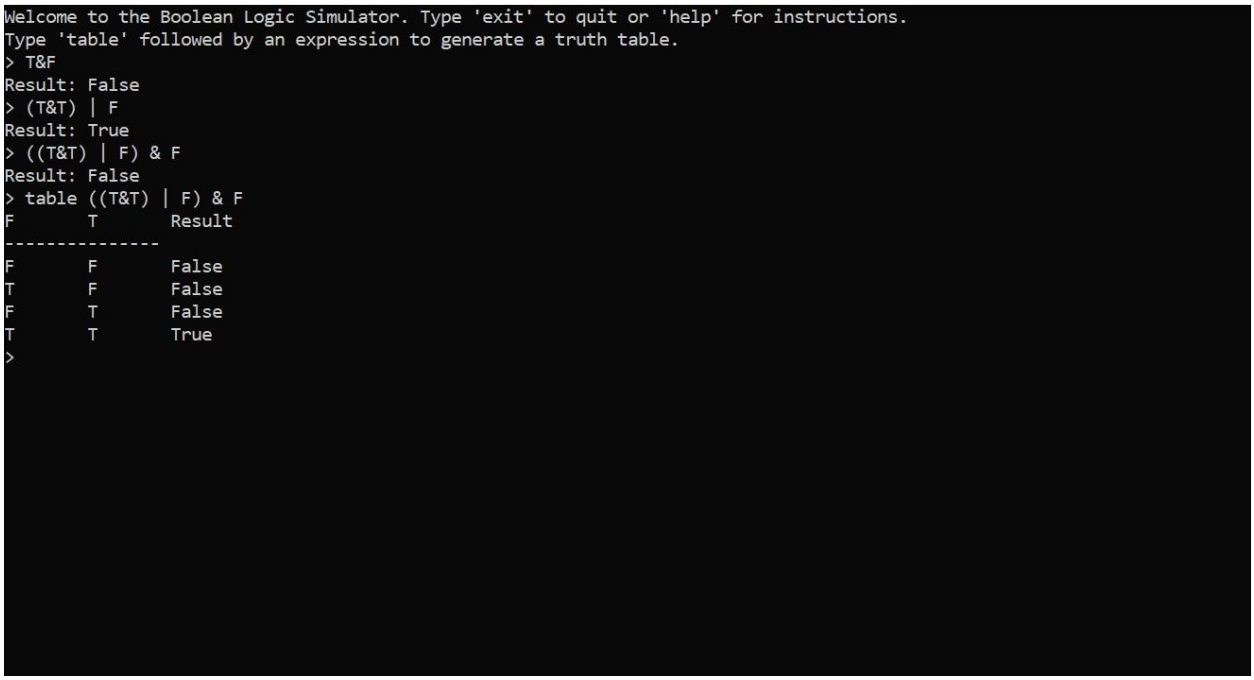
NAND: this is the inverse of AND

XOR: This stands for exclusive OR. It will be true if either statement is true, but not when both are true.

GitHub: A site that hosts online repositories. I.E. it stores a bunch of code.

Examples:


Welcome to the Boolean Logic Simulator. Type 'exit' to quit or 'help' for instructions.
Type 'table' followed by an expression to generate a truth table.
> T&F
Result: False
>


Welcome to the Boolean Logic Simulator. Type 'exit' to quit or 'help' for instructions.
Type 'table' followed by an expression to generate a truth table.
> T&F
Result: False
> (T&T) | F
Result: True
> ((T&T) | F) & F
Result: False
> table ((T&T) | F) & F
F T Result

F F False
T F False
F T False
T T True
>

FAQ:

What is truthy? Truthy is a Boolean logic simulator **What is a Boolean logic simulator?** A Boolean logic simulator is a simulator that allows inputs of Booleans, Boolean operators, and parenthesis, that outputs one final Boolean value (True or False).

What happens after my expression is inputted? Truthy parses your input, tokenizes it, then evaluates it.

What is a Boolean? A Boolean is a value that is either true or false.

What Boolean operators does Truthy support? Truthy supports and, or, not, nand, and xor, the last two being the opposite of and, and exclusive or, respectively (&, |, !, @, $).

What happens if nothing is inputted? An error is thrown.

Why is a syntax error being thrown when I try to use the not operator (!)? This is probably because you are trying to put it in between two Boolean values, and not just in front of one. For example T!T is invalid because, !T becomes F making the expression TF, which doesn't evaluate to one single value without another operator.

Is there a max size expression? Truthy allows for accurate expressions under 500 characters long, anything longer works in theory, but the runtime will increase as the characters are increased.

What types of errors does truthy check for? Invalid Expressions, which includes Missing operand, Unknown operator, Mismatched parentheses, Circular logic, Empty expression, Double operator, Missing truth values, Inconsistent characters, Operator after operand, Invalid characters.

How does truthy work? Truthy works by implementing c++ code that runs when the .exe file is opened/started.

Troubleshooting:

Error	Cause	Example Input	Solution
Missing operator	Two expressions next to each other without an operator	TT	Introduce an operator between the expressions
Tokenization error	A character that is not an operator or T/F	` ~ ^ * %	Use a valid operator or T/F for variables
Missing closing parenthesis	Parenthesis were opened, but not closed	(T	Add a matching closing parenthesis before the end of the expression
Missing operand	Two operators next to each other without an operator	||	Introduce an expression between the operators
Missing opening parenthesis	Parenthesis were closed before they were opened	T)	Open the parentheses, or remove the extra closing parenthesis