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:**

Currently, the only way to download the program is by cloning the git repository from our GitHub page.

[KU-EECS-384-group-R6/Truthy (github.com)](https://github.com/KU-EECS-384-group-R6/Truthy)

From here you can clone the repository and run the application. You can find the executable application at your\_download\_location/Truthy/src/Debug/Truthy.exe

**Getting Started:**

Launch the application found at your\_download\_location/Truthy/src/Debug/Truthy.exe. 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 examlpes

**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 evalutes true when both statements are true. I.E. When A is true and B is true

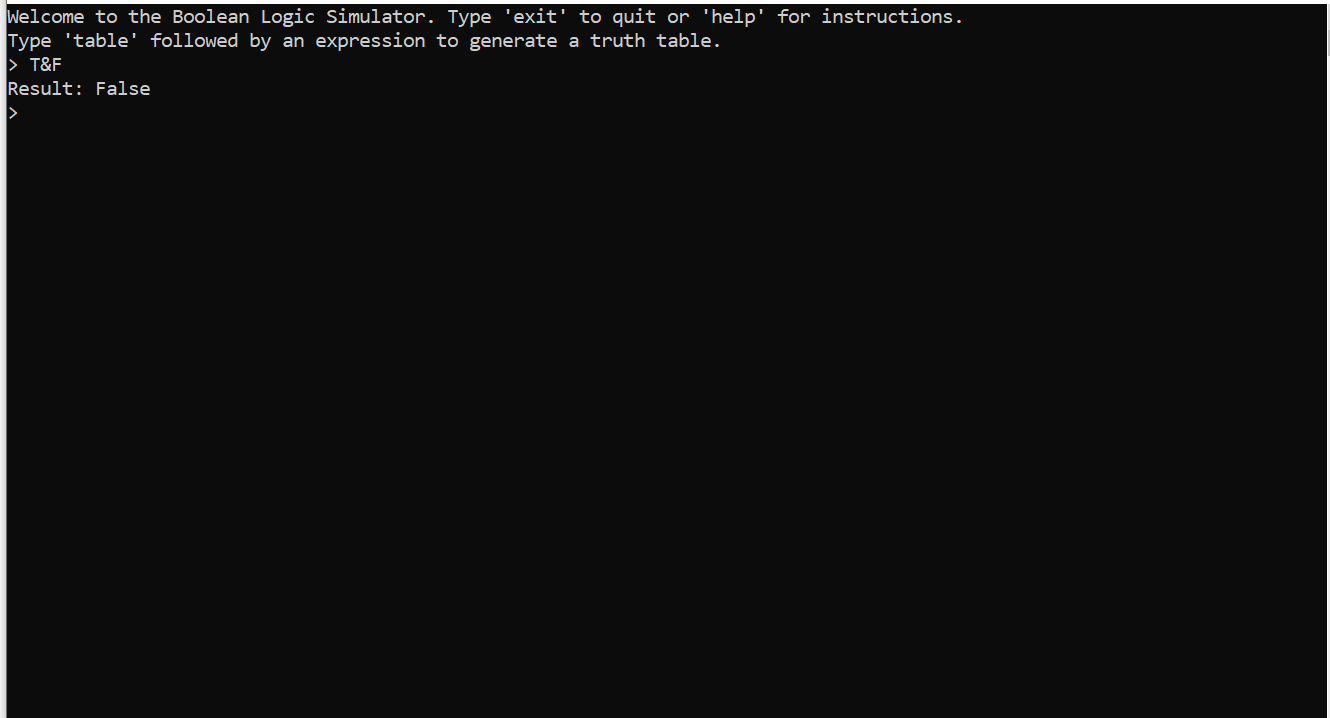
OR: evaluate true when either statements are true. I.E. When A is true or B is true.

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:**

**A screen shot of a computer

Description automatically generated**

**FAQ: //tbd**

**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 a-z | 🙂 | Use a letter between a and z 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 closing parenthesis** | Parenthesis were closed before they were opened | T) | Open the parentheses, or remove the extra closing parenthesis |