
SQUISHY CATS

LogiCalc
User's Manual
Version 1.0

LogiCalc	Version: 1.0
User's Manual	Date: 04/29/2024
E4	

Revision History

Date	Version	Description	Author
04/29/2024	0.1	Document initialization	Olivia
05/02/2024	1.0	Document completed	Yara, Mariam

LogiCalc	Version: 1.0
User's Manual	Date: 04/29/2024
E4	

Table of Contents

1.	Purpose	4
2.	Introduction	4
3.	Getting started	4
4.	Advanced features	4
5.	Troubleshooting	5
6.	Example of uses	5
7.	Glossary	5
8.	FAQ	5

LogiCalc	Version: 1.0
User's Manual	Date: 04/29/2024
E4	

Test Case

1. Purpose

This User's Manual Document for LogiCalc is an easy-to-understand guide on how to use the software. The following document includes a brief overview of the software, a step-by-step guide to use the software, advanced features, troubleshooting, and example input.

2. Introduction

The LogiCalc software is a Boolean Expression Calculator with a Command Line Interface. The program allows users to input different Boolean Algebra expressions, where it interprets the correct output, and then prints said output.

3. Getting started

To get started with the program, follow these steps:

1. Clone the repository onto your device and navigate into the top directory of the project files

```
git clone https://github.com/aaatipamula/bool_expr_calc.git
cd bool_expr_calc
```

2. Compile the boolean expression calculator using the MakeFile
(note: your device must use g++ for the compilation to work, clang will not work)

```
make
```

3. Run commands using the following format:

```
./bin/lcalc "[your_bool_expr]"
```

Valid expressions use the following symbols:

Symbol	Operation
()	parenthesis
T	"True"
F	"False"
&	AND
	OR
!	NOT
@	NAND
\$	XOR
quit	Quits program

The program will throw an error when any of the following inputs are given:

1. incomplete parenthesis. E.g: "(T | F"

LogiCalc	Version: 1.0
User's Manual	Date: 04/29/2024
E4	

2. Sequencing boolean values without using any operators. E.g: "T F"
3. Using operators without operands. E.g: "T | "
4. Using undefined operators. E.g: "??"
5. Blank expressions. E.g: " "

4. Troubleshooting

Problem: Invalid Expression Error

Symptoms: The simulator displays an error message indicating that the entered expression is invalid.

Solution: Check the expression for syntax errors, unknown operators, or missing parentheses. Ensure that all operators and variables are correctly spelled and formatted according to the simulator's guidelines

Problem: Performance Issues with Complex Expressions

Symptoms: The simulator slows down or becomes unresponsive when handling extremely long or complex expressions. Solution: Break down complex expressions into smaller, more manageable parts. Use parentheses to group operations and improve readability. Consider optimizing the expressions for efficiency, especially when dealing with many variables and operators.

Problem: Interface Display Issues

Symptoms: The simulator's interface appears distorted or functions improperly.

Solution: Refresh the simulator interface or restart the program to resolve display issues. Check for any browser or system compatibility issues that may affect the simulator's performance. Update your browser or software to the latest version for optimal functionality.

5. Examples

Explore the following examples to understand how to use the Boolean Logic Simulator effectively:

Example 1: Evaluating a simple Boolean expression (e.g., A & B | C).

Example 2: Handling complex expressions with parentheses (e.g., (A & B) | (C & D)).

Example 3: Detecting mismatched parenthesis in an expression (e.g., F | T))).

Example 4: Evaluate expressions with multiple operators and NOTs (e.g., !(T | F) \$!(T & (!F | T))).

Example 5: Detecting a missing operand in the expression (e.g., ! & T).

Example 6: Detecting a missing operator in the expression (e.g., T T)

6. Glossary of terms

- Boolean Expression: A mathematical expression representing logical operations using variables and operators.
- Operator Precedence: The order in which operators are evaluated in an expression.
- Truth Value: The result of evaluating a Boolean expression, either True or False.
- Parentheses Handling: The ability of the program to correctly interpret expressions enclosed within parentheses.

LogiCalc	Version: 1.0
User's Manual	Date: 04/29/2024
E4	

7. FAQ

Q: How do I input a Boolean expression in the simulator?

A: Use infix notation with supported operators (&, |, @, \$), prefix notation for (!), and parentheses for grouping operations.

Q: What happens if I input an invalid expression?

A: The simulator provides informative error messages and guides you to correct the expression.

Q: How do I handle operator precedence in complex Boolean expressions?

A: The simulator follows standard operator precedence rules. You can use parentheses to explicitly specify the order of operations. For example, (A & B) | C ensures that the AND operation is evaluated before the OR operation.

Q: Is there a limit to the length or complexity of Boolean expressions I can input?

A: The simulator is designed to handle a wide range of expressions, including complex ones with multiple operators and variables. However, extremely long or convoluted expressions may affect performance. It's recommended to break down complex expressions into manageable parts for clarity and efficiency.