

---

**Incomplete Inc.**

---

**Boolean Algebra Calculator**

**User's Manual**

**Version 1.0**

Boolean Algebra Calculator	Version: 1.0
User's Manual	Date: 02/May/2024
Manual	

## Revision History

Date	Version	Description	Author
01/May/2024	0.1	Filling out Introduction and Getting Started	Ibrahim Sufi
02/May/2024	0.2	Filling out Purpose and Advanced Features	Ben Weinzirl
02/May/2024	0.3	Filling out Examples and Glossary Terms	Del Endecott
02/May/2024	1.0	Filling out Troubleshooting and FAQ along with consolidating previous versions	Ben Weinzirl

Boolean Algebra Calculator	Version: 1.0
User's Manual	Date: 02/May/2024
Manual	

# Table of Contents

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

Boolean Algebra Calculator	Version: 1.0
User's Manual	Date: 02/May/2024
Manual	

# User Manual

## 1. Purpose

The purpose of this manual is to provide users with the ability to effectively use the Boolean Algebra Calculator program. The User Manual contains all essential information for the user to solve Boolean expressions. This manual includes a description of the functions and capabilities of the program.

## 2. Introduction

This repository contains the C++ code for a Boolean algebra calculator. The calculator takes an input Boolean algebra expression and outputs the expression's truth value.

To install, run `git clone https://github.com/SneakyNinja81/eecs348_FinalProject.git` in the terminal. Then, compile the code with `g++ Project.cpp -o Project.exe` on Linux or Windows and `g++ Project.cpp -std=c++11 -o Project.exe` on Mac. Finally, run the program with `./Project.exe`

## 3. Getting started

Enter a logical expression using T to represent True, F to represent False, and the following symbols as logical operators.

Symbol	Logical Operation
&	AND
	OR
!	NOT
@	NAND
\$	XOR

This calculator is programmed to handle parentheses/nested expressions.

## 4. Advanced features

The program does not currently contain any advanced features. However, improvements may come in the future, in which case this document will be updated.

## 5. Troubleshooting

Attempting to input a variable other than 'T' or 'F' will result in an error. Avoid using variables that would need to be user-defined.

If a 'missing parenthesis' error occurs, ensure that parentheses are balanced.

Make sure to include operators between truth values. Values such as 'FF' or 'TF' will result in an invalid expression.

## 6. Examples

(T | F) \$ F First, the value of the expression in parentheses is determined to be True. Next, this value is compared to the next Boolean in the input using the XOR operator, which evaluates as True. The whole statement has been processed, so the calculator returns True.

Boolean Algebra Calculator	Version: 1.0
User's Manual	Date: 02/May/2024
Manual	

## 7. Glossary of terms

**AND:** A logical operator represented by the & symbol. AND evaluates to True if and only if both operands are True.

**Boolean:** A binary variable that can be either True or False. True may also be represented as T or 1. False may also be represented as F or 0.

**Boolean Algebra:** A type of mathematics using Boolean values and logical operators such as AND and OR.

**Command Line:** The designated area in a terminal where the user can enter commands to interact with the computer.

**EECS:** The Electrical Engineering and Computer Science department at the University of Kansas.

**Expression:** A statement containing both values and operators that can be evaluated to find a specific answer.

**Logical Operator:** A symbol connected to one or more operands to produce a new result. Logical operators included in this calculator are AND, OR, NAND, NOT, and XOR.

**NAND:** A logical operator represented by the @ symbol. NAND evaluates to True if and only if both operands are False.

**NOT:** A logical operator represented by the ! symbol placed in front of its operand. NOT reverses the value of its operand, so a True operand will instead evaluate to False and vice versa.

**Operand:** A quantity that will undergo an operation. An operand may be a Boolean value or an expression.

**OR:** A logical operator represented by the | symbol. OR evaluates to True if one or both operands are True.

**Repository:** A collection of files containing or related to a certain program.

**Terminal:** A text-based interface that enables interaction with a computer.

**Truth Value:** The value represented by a Boolean. A truth value must either be True (AKA T, 1) or False (AKA F, 0).

**XOR:** A logical operator represented by the \$ value. XOR evaluates to True if and only if exactly one operand is True.

## 8. FAQ

Q: How do I run the code?

A: Refer to the 'Getting Started' section.

Q: Why is my input creating an error?

A: The input is most likely flawed. Please refer to the 'Troubleshooting' section.

Q: Will there be an implementation for more Boolean expressions?

A: There may be more in the future, but please refer to 'Getting Started' to see which operators are supported.

Q: How can I get in contact with the team behind this project?

A: Refer to the '01-Project-Plan-marked.pdf' in GitHub for contact information.