

Краевая научно-практическая конференция
учебно-исследовательских работ учащихся 9-11 классов
«Прикладные и фундаментальные вопросы математики»

Прикладные вопросы математики

Логический калькулятор

Захаров Глеб,

11 кл., МБОУ «Гимназия №17», г. Пермь,

Банникова Лидия Борисовна,

педагог ДО высшей категории МАОУ ДОД
ДД(Ю)Т г. Пермь.

Пермь. 2014.

Содержание

Введение.....	3
Основная часть	
I. Постановка задачи.....	3
II. Немного о Булевой алгебре и математической логике.....	4
III. Описание комплекса.....	7
Необходимые программные и технические средства.....	11
Заключение.....	12
Использованная литература.....	12
Приложение	
Код функции вычисления логических выражений.....	13

Введение

Я занимаюсь в секции программирования уже 2 года. Все это время я изучил несколько языков программирования и писал программы на них. Часто я ошибался: писал не те циклы, объявлял не те переменные, выбирал не те типы данных. Приходилось возиться с системными функциями, искать их в справочниках и Интернете. Но больше всего я стонал, когда логика программы требовала создать сложное условие с союзами «И», «ИЛИ», «НЕ». Приходилось сидеть и думать, что получится в результате этого сложного условия: истина или ложь, так как от этого зависел ход программы. Поэтому я решил создать программу-калькулятор, которая помогла бы людям справляться с такой проблемой.

Как она бы выглядела? Пользователь мог бы ввести в строку логическое выражение, значения высказываний и получить ответ. Если бы ответ получался неправильный, пользователь посмотрел бы поочередно действия, которые выполняла программа для вычисления ответа, нашел бы ошибку и исправил ее.

Для создания этой программы мне пришлось изучить теорию Булевой алгебры и математической логики, узнать больше возможностей языка VisualBasic и разработать алгоритм.

Для кого нужна такая программа? Прежде всего, для людей, которым необходимо работать с математической логикой как с частью профессии, например физикам, математикам-теоретикам и т.д. Однако действительно нужна логика в таких областях как программирование и радиоэлектроника: в 1-ой она используется для написания программ, а во 2-ой для составления электронных плат и микроконтроллеров.

Основная часть

I. Постановка задачи

Требовалось создать программу, которая могла бы вычислять значение логических высказываний, а так же программу-справочник, которая помогла бы пользователям узнать немного об интересной науке «Булева алгебра». Для выполнения 1-ой задачи пришлось:

1. Изучить раздел VisualBasic'a, связанный с символами и логическими переменными.
2. Написать код вычисления значения логического высказывания.
3. Написать код проверки высказывания для любых значений переменных.

Для выполнения второй задачи:

1. Найти информацию о Булевой алгебре в Интернете и книгах.
2. Изучить теорию алгебры логики и математической логики.
3. Составить тест для проверки знаний пользователей.

Для создания такой программы было необходимо:

- Пополнить свои знания о языке VisualBasic.
- Изучить теорию Булевой алгебры и математической логики.
- Разработать алгоритм программы.
- Написать код программы.
- Протестировать программу.

II. Немного о Булевой алгебре и математической логике

Булеву алгебру, как ни странно, создал в начале XIX века английский математик Джордж Буль. Он изобрел своеобразную алгебру – систему правил и обозначений, которые применялись для высказываний. Собственно, высказывание – утверждение, которое может принимать значение истина (True, 1) или ложь (False, 0). Обычно высказывания обозначаются латинскими буквами A, B, C и т. д. Например «Сегодня на улице светит солнце» - высказывание, « $2 \times 3 = 359$ » - тоже высказывание, а «Когда закончится урок?» - не высказывание, так как является вопросом.

Из высказываний можно строить сложные выражения с помощью логических операций:

1. **Инверсия** (или логическое отрицание). Обозначается частицей НЕ, not, знаком « \neg » или «-» или чертой над высказыванием (\bar{A}). В результате операции высказывание меняет свое значение на противоположное.
2. **Конъюнкция** (или логическое умножение). Обозначается союзом И, and, знаками «&», «*» и « \wedge ». Конъюнкция имеет значение 1, если оба высказывания имеют значения 1.
3. **Дизъюнкция** (или логическое сложение). Обозначается союзом ИЛИ, or, знаками «+» и « \vee ». Дизъюнкция имеет значение 1, если хотя бы одно из высказываний имеет значение 1.
4. **Импликация** (условное высказывание) ($A \rightarrow B, A \supset B$) - в русском языке этой связке соответствует выражение «если A, то B». Имеет значение 0 только если $A=1$, а $B=0$ (Из истины не может следовать ложь).
5. **Эквивалентность** ($A \equiv B, A \leftrightarrow B, A=B$) - в русском языке этой связке соответствует выражение «если и только если», «тогда и только».

тогда». Имеет значение 1 когда оба высказывания имеют одинаковые значения.

6. **Разделительное ИЛИ** (сложение по модулю 2). Обозначается знаком XOR. Противоположная эквивалентности операция, имеет значение 1 когда высказывания противоположных значений. Ее можно записать как $\neg(A=B)$.

7. Также в Булевой алгебре могут использоваться скобки.

Правила выполнения рассмотренных логических операций можно отразить с помощью **таблицы истинности**:

A	B	A*B	A+B	-A	AxorB	A=B	A>B
1	1	1	1	0	0	1	1
1	0	0	1	0	1	0	0
0	1	0	1	1	1	0	1
0	0	0	0	1	0	1	1

Последовательность выполнения операций в логических формулах определяется старшинством операций. Приоритеты операций:

- выполнение действий в скобках,
- инверсия,
- конъюнкция,
- дизъюнкция, разделительное ИЛИ, эквивалентность.

В Булевой алгебре также существуют формулы, похожие на формулы сокращенного умножения.

Основные формулы преобразования логических выражений:

Двойное отрицание	$\neg\neg A=A$
Закон де Моргана	$\neg(A+B)=\neg A*\neg B, \neg(A*B)=\neg A+\neg B$
Закон поглощения	$A*(A+B)=A, A+(A*B)=A$

Закон Порецкого	$-A*(A+B) = -A*B, A+(-A*B) = A+B$
Закон коммутативности	$A \& B = B \& A$ или $(A*B = B*A)$ $A \vee B = B \vee A$ или $(A+B = B+A)$
Закон ассоциативности	$(A \vee B) \vee C = A \vee (B \vee C)$ или $A + (B + C) = (A + B) + C$ $(A \& B) \& C = A \& (B \& C)$ или $A * (B * C) = (A * B) * C$
Закон идемпотентности	$A \vee A \vee A = A$ или $A + A + A = A$, $A \& A \& A = A$ или $A * A * A = A$
Закон дистрибутивности	$A*(B+C) = A*B + A*C$ $A+(B*C) = (A+B)*(A+C)$
Законы работы с константами	$A + 1 = 1, A + 0 = A, A*1 = A, A*0 = 0,$ $A + -A = 1, A * -A = 0$
Избавление от импликации	$A > B = -A + B$
Избавление от эквивалентности	$A = B = A*B + -A*-B$ $-(A=B) = (A \oplus B) = A*-B + -A*B$

III. Описание программного комплекса

Комплекс программ - подпрограммы, объединенные в систему с помощью меню и интерфейса. В этой программе пользователь сможет посмотреть полезную информацию, пройти небольшой тест и вычислить логическое выражение.

Запустив программу **Calculator_Of_Logic.exe**, пользователь попадает в главное меню (Рис. 1). На нем расположены кнопки для перехода по программе, а справа – изображен портрет Джорджа Буля.



Рис. 1

1. «Справка». Нажав на эту кнопку, пользователь попадет в окно справки (рис.2), где сможет ознакомиться с информацией о Булевой алгебре, а также перейти на полезные сайты по этой теме.

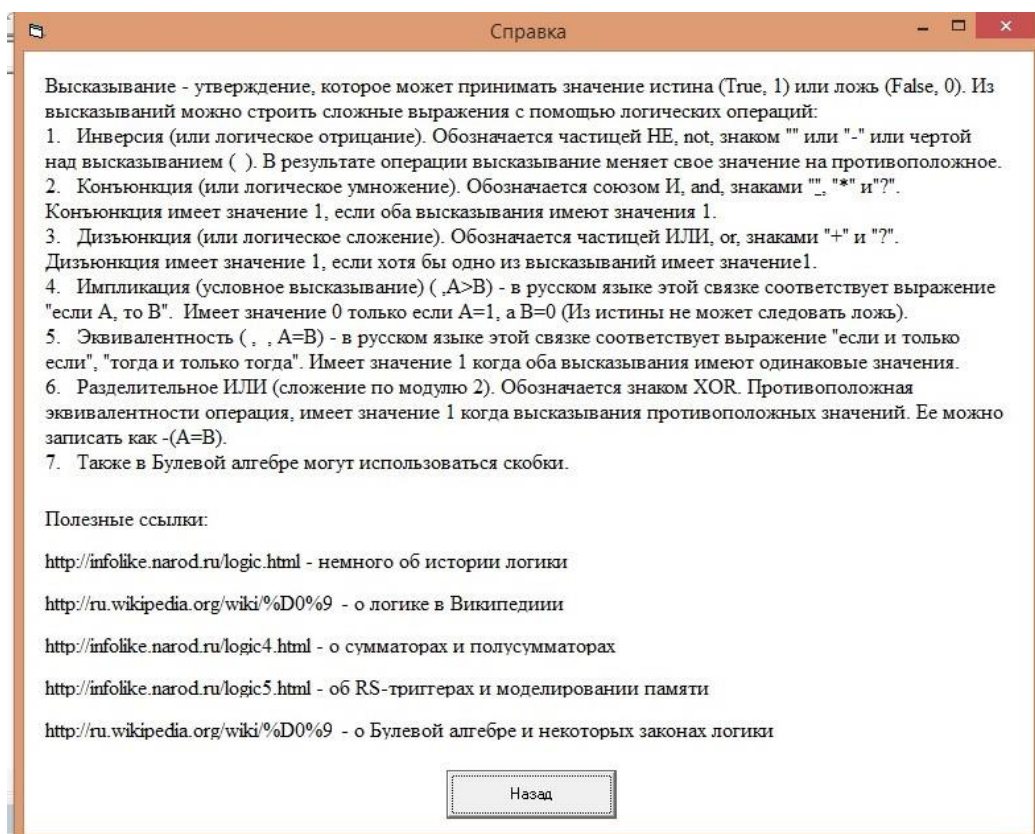


Рис. 2

2. «Тесты для самопроверки». Нажав на эту кнопку, пользователь попадет в окно тестов(Рис. 3).
- Все вопросы основаны на информации из справки. Поэтому каждый пользователь сможет пройти весь тест, если прочитает справку полностью.
 - Каждый вопрос находится в отдельном файле в корне программы. Вопросы, варианты ответов и сами ответы может добавлять сам пользователь программы (Например, учитель добавит в программу дополнительные вопросы для урока)
 - Пока в базе около 30-ти вопросов.
 - Вопросы для теста выбираются случайным образом. Это нужно для выбора 20-ти вопросов из базы данных.
 - После теста выводится сообщение о количестве правильных ответов.

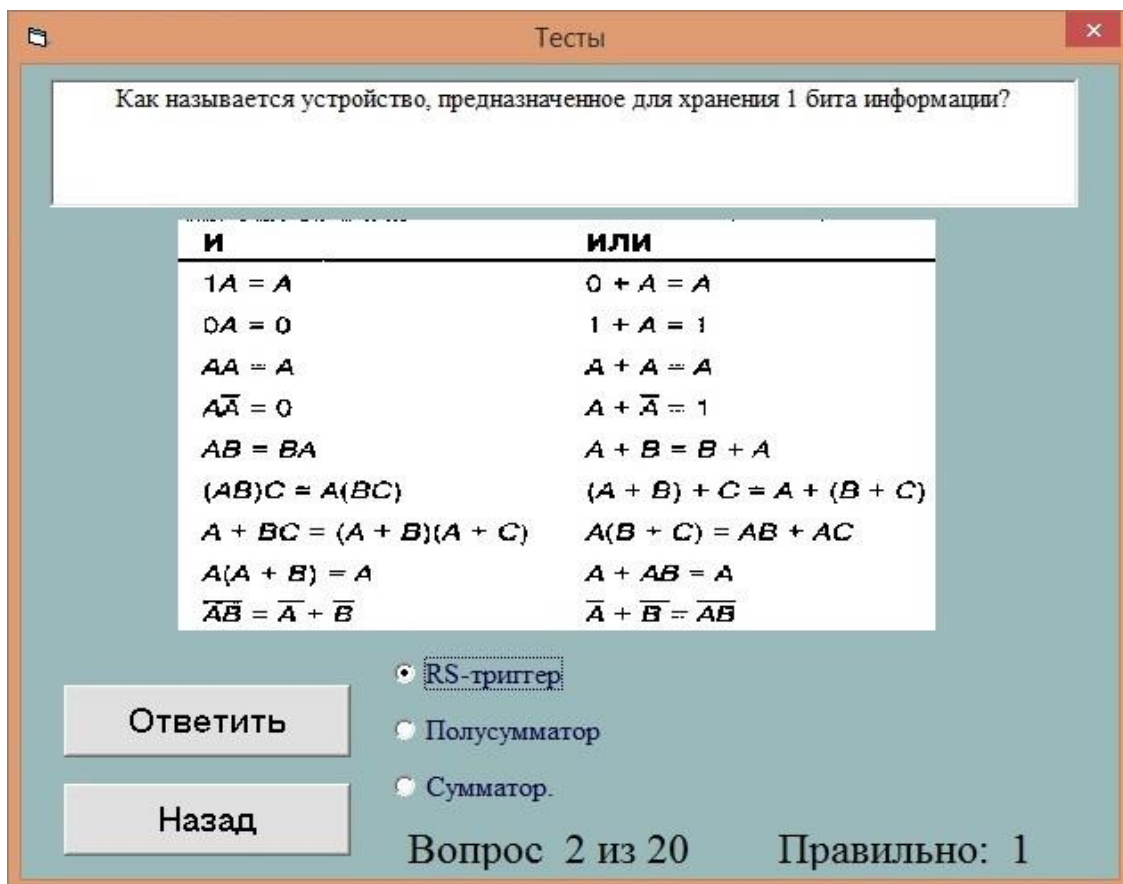


Рис. 3

3. «Логический калькулятор». Нажав на эту кнопку, пользователь попадет в окно калькулятора (Рис. 4).

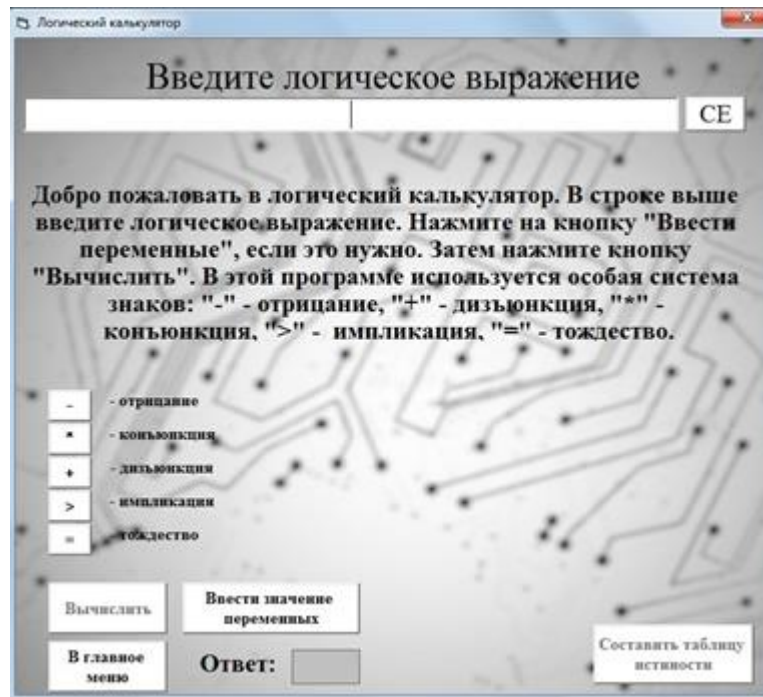


Рис. 4

- Вы можете ввести логическое выражение в строку. Оно должно состоять из латинских букв (A, B, C, D, E), 1 или 0, логических операций и скобок.
- Если вы хотите вычислить значение выражения для всех комбинаций значений переменных, нажмите кнопку «Составить таблицу истинности» (Рис. 5)
- Если же вы хотите вычислить значение выражения для одной комбинации значения переменных, то для начала введите значения переменных, нажав на кнопку «Ввести значения переменных», а затем кнопку «Вычислить» (Рис. 6)
- Если вы нажмете кнопку «CE», вы отчистите главную строку ввода.
- Слева для удобства расположены памятки-кнопки Булевых функций
- При вычислении отдельного значения программа также показывает порядок действий.

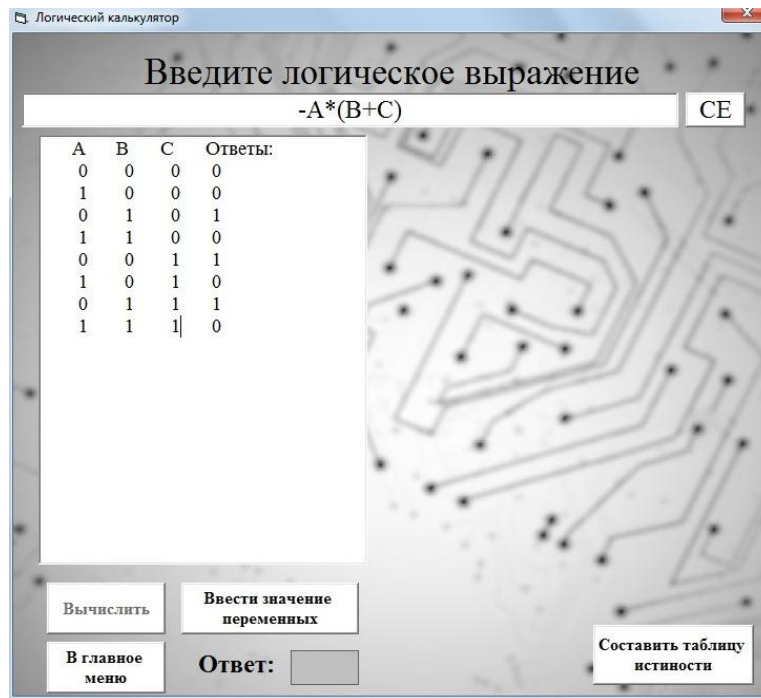


Рис. 5

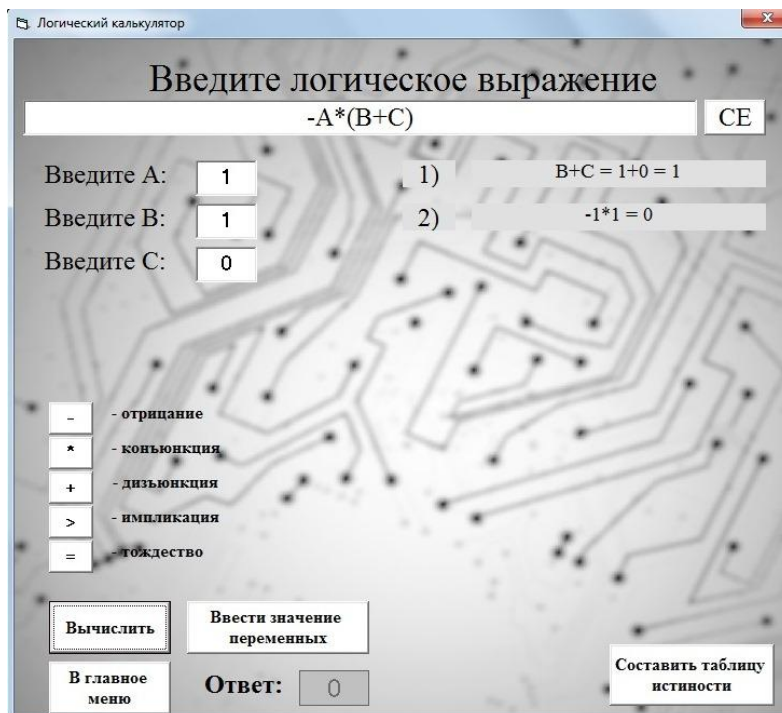


Рис. 6

Алгоритм программы «Калькулятор»:

- сначала программа разбирает выражение на отдельные модули без скобок.
- затем в каждом модуле меняет переменные, если они есть, на их численные значения.
- по очереди, согласно порядку действий вычисляет значение одного модуля, 2-го и так далее.
- программа по очереди соединяет модули и вычисляет общее выражение.

В таблицах истинности все это происходит в цикле, в котором постепенно меняется значение каждой переменной.

Необходимые программные и технические средства

Программа состоит из нескольких файлов, которые находятся в одной папке. Она не требует установки. Программа весит около 1 Мб, исполняемый файл называется Calculator_Of_Logic.exe.

Поэтому программа не требует особых программных средств.

Заключение

В результате работы была создана программа, которая позволяет пользователю вычислять логические выражения, проверять правильность написания этих выражений.

Кроме того, пользователь может при желании посмотреть информацию о математической логике и пройти тест.

Особенности программы:

- Дружественный интерфейс, с подсказками, памятками и неплохим дизайном;
- Сложный в написании алгоритм, интересный программистам;
- Справка о логике вместе с тестом для самопроверки, нужные для освежения информации о математической логике;
- Уникальность программы

Часть программы с калькулятором будет полезна программистам и инженерам, а часть программы с тестом и справкой может быть использована в школе как учебно-образовательная программа.

Эту программу тестировали мои друзья-программисты. Они решили, что эта полезная программа.

Использованная литература

В. И. Петрова. Учебный модуль «Логика» в школьном курсе преподавания информатики. - Пермь, 2010 г.

В. Л. Рвачев. Геометрические приложения алгебры логики. – К.: «ТЕХНИКА», 1967 г. – стр. 9 – 18.

Г. П. Мельников. Азбука математической логики. – М.: «ЗНАНИЕ», 1967 г. – стр. 63 – 78

Приложение

Код функции вычисления логических выражений:

```
Private Function Vich(k1 As Integer, j1 As Integer) As String
Dim s1 As String, s2 As String, s5 As String, s4 As String, s3 As String, i As Integer,
k As Integer, bool(4) As Boolean
Dim ber(2) As Boolean

    s1 = Mid$(SCif, k1 + 1, j1 - k1 - 1)
    For i = 0 To 4
bool(i) = False
    Next i
    For i = 1 To Len(s1) ' Проверка знаков
        s2 = Mid$(s1, i, 1)
        Select Case LCase(s2)
            Case "-":
bool(0) = True
            Case "*":
bool(1) = True
            Case "+":
bool(2) = True
            Case ">":
bool(3) = True
            Case "=":
bool(4) = True
        End Select
    Next i
    """"

i = 0
s2 = ""
While bool(0) = True 'Избавление от отрицаний
While s2 <> "-"
i = i + 1
    s2 = Mid(s1, i, 1)
Wend
s3 = Mid(s1, i + 1, 1)
Select Case s3
    Case "0":
        s3 = "1"
    Case "1":
        s3 = "0"
    Case "-":
```

```

        s3 = ""
    End Select
    s1 = Mid(s1, 1, i - 1) + s3 + Mid(s1, i + 2)
bool(0) = False
    For k = 1 To Le
        s2 = Mid(s1, k, 1)
        If s2 = "-" Then
bool(0) = True
i = 0
        End If
    Next k
Wend
""

i = 0
k = 0
s2 = ""
Whilebool(1) = True 'Избавление от конъюнкции
While s2 <> "*"
i = i + 1
    s2 = Mid(s1, i, 1)
Wend
    s3 = Mid$(s1, i - 1, 1)
    s4 = Mid$(s1, i + 1, 1)
    If s3 = "1" Then
ber(0) = True
    Else
ber(0) = False
    End If
    If s4 = "1" Then
ber(1) = True
    Else
ber(1) = False
    End If
    If (ber(0) And ber(1)) = True Then
        s3 = "1"
    Else
        s3 = "0"
    End If
    s1 = Mid(s1, 1, i - 2) + s3 + Mid(s1, i + 2)
bool(1) = False
    For k = 1 To Le
        s2 = Mid(s1, k, 1)
        If s2 = "*" Then
bool(1) = True

```

```

i = 0
    End If
Next k
Wend
*****
i = 0
k = 0
s2 = ""
Whilebool(2) = True 'Избавление от дизъюнкции
While s2 <> "+"
i = i + 1
    s2 = Mid(s1, i, 1)
Wend
    s3 = Mid$(s1, i - 1, 1)
    s4 = Mid$(s1, i + 1, 1)
    If s3 = "1" Then
ber(0) = True
    Else
ber(0) = False
    End If
    If s4 = "1" Then
ber(1) = True
    Else
ber(1) = False
    End If
    If (ber(0) Or ber(1)) = True Then
        s3 = "1"
    Else
        s3 = "0"
    End If
    s1 = Mid(s1, 1, i - 2) + s3 + Mid(s1, i + 2)
bool(2) = False
    For k = 1 To Le
        s2 = Mid(s1, k, 1)
        If s2 = "+" Then
bool(2) = True
i = 0
    End If
Next k
Wend
*****
i = 0
k = 0
s2 = ""
Whilebool(3) = True 'Избавление от коммутативности

```

```

While s2 <> ">"
i = i + 1
    s2 = Mid(s1, i, 1)
Wend
s3 = Mid$(s1, i - 1, 1)
s4 = Mid$(s1, i + 1, 1)
If (s3 = "1") And (s4 = "0") Then
    s3 = "0"
Else
    s3 = "1"
End If
s1 = Mid(s1, 1, i - 2) + s3 + Mid(s1, i + 2)
bool(3) = False
For k = 1 To Le
    s2 = Mid(s1, k, 1)
    If s2 = ">" Then
bool(3) = True
i = 0
        End If
    Next k
Wend
''''''
i = 0
k = 0
s2 = ""
Whilebool(4) = True 'Избавление от тождеств
While s2 <> "="
i = i + 1
    s2 = Mid(s1, i, 1)
Wend
    s3 = Mid$(s1, i - 1, 1)
s4 = Mid$(s1, i + 1, 1)
    If s3 = s4 Then
        s3 = "1"
    Else
        s3 = "0"
    End If
    s1 = Mid(s1, 1, i - 2) + s3 + Mid(s1, i + 2)
bool(4) = False
    For k = 1 To Le
        s2 = Mid(s1, k, 1)
        If s2 = "=" Then
bool(4) = True
i = 0
            End If
        Next k
    Wend

```



```
    Next k
Wend
****
Vich = s1
End Function
```