МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №5**

по дисциплине: Информатика

тема: «Работа с документами в MS Office Excel»

Выполнил: ст. группы ПВ-223

Пахомов Владислав Андреевич

Проверили: ст. пр.

Бондаренко Т. В.

Белгород 2022 г.

**Цель работы:** изучить основные принципы и получить практические навыки работы с документами в приложении MS Office Excel.

Вариант № 10 ПВ-223

**Задания к работе:**

1. Выполнить перевод целого положительного числа ― номера зачетной книжки (105223194) в двоичную систему счисления.

2. Составить таблицу значений логической функции от 5 логических переменных: f(X) = f(x1, x2, x3, x4, x5), вектор X = {x1, x2, x3, x4, x5}. Использовать 32 набора значений логических переменных xi, i=1,...,5, составить таблицу значений логической функции (см. табл. 8).



В качестве значений логической функции взять двоичное число, полученное в пункте 1, записать его поразрядно в ячейки столбца значений функции (см. табл. 6). Незаполненные клетки таблицы значений функции считать равными 0.

3. По полученной таблице значений логической функции выполнить построение СДНФ (совершенной дизъюнктивной нормальной формы) логической функции f(X).

4. Выполнить минимизацию полученной в пункте 3 логической функции (X), используя основные законы и аксиомы алгебры логики. В итоге будет получена логическая функция G(X).

5. Выполнить вычисление значений логической функции G(X) на заданных наборах значений логических переменных xi, i=1,...,5. Построить таблицу значений функции G(X). Сравнить полученные значения со значениями логической функции f(X).

6. Для логической функции G(X) выполнить построение комбинационной схемы.

7. Преобразовать полученную функцию G(X), перейдя к базису «И-НЕ» или «ИЛИ-НЕ». Выразить все логические операции, содержащиеся в функции G(X), используя операцию штрих Шеффера ― “и-не” или стрелка Пирса ― “или-не” (допустимо приведение функции к новому базису по частям).

8. По таблице значений логической функции выполнить построение СКНФ (совершенной конъюнктивной нормальной формы) логической функции f(X).

9. Выполнить упрощение полученной в пункте 8 логической функции, используя основные законы и аксиомы алгебры логики, уменьшая количество логических операций. В итоге будет получена логическая функция G1(X).

10.Выполнить вычисление значений логической функции G1(X) на заданных наборах значений логических переменных xi, i=1,...,5. Построить таблицу значений функции G1(X). Сравнить полученные значения функции G1(X) со значениями логической функции f(X).

11. Создать модуль, реализующий следующие логические функции: отрицание, конъюнкция, дизъюнкция, «исключающее или», эквивалентность, импликация, штрих Шеффера, стрелка Пирса.

12. Составить программу, выполняющую вычисление значений логических функций G(X) и G1(X) на заданных наборах значений логических переменных xi, i=1,...,5 и вывод на экран таблиц значений этих логических функций. Логические функции могут быть заданы в тексте программы.

**Задание 1 (номер книжной зачётки = 105223194)**

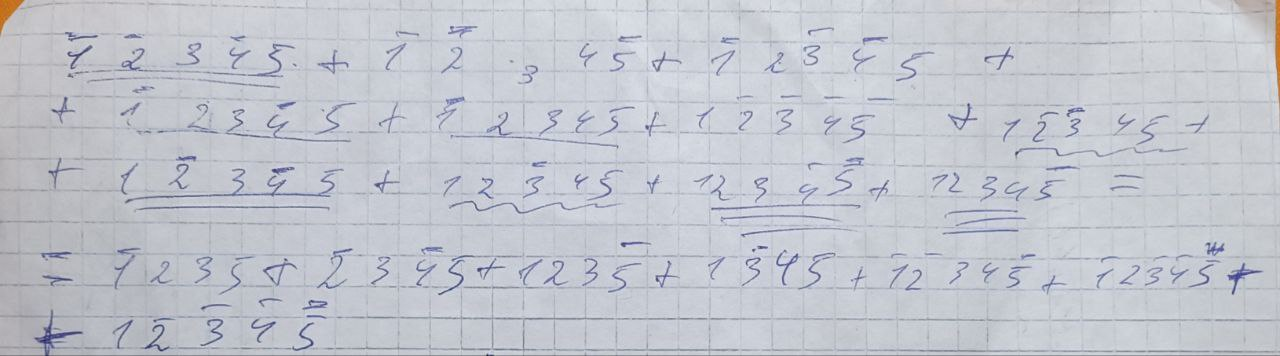
105223194 = 110 0100 0101 1001 0100 0001 1010

**Задание 2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | **0** |  |  |
| 0 | 0 | 0 | 0 | 1 | **0** |  |  |
| 0 | 0 | 0 | 1 | 0 | **0** |  |  |
| 0 | 0 | 0 | 1 | 1 | **0** |  |  |
| 0 | 0 | 1 | 0 | 0 | **0** |  |  |
| 0 | 0 | 1 | 0 | 1 | **1** | = |  |
| 0 | 0 | 1 | 1 | 0 | **1** | = |  |
| 0 | 0 | 1 | 1 | 1 | **0** |  |  |
| 0 | 1 | 0 | 0 | 0 | **0** |  |  |
| 0 | 1 | 0 | 0 | 1 | **1** | = |  |
| 0 | 1 | 0 | 1 | 0 | **0** |  |  |
| 0 | 1 | 0 | 1 | 1 | **0** |  |  |
| 0 | 1 | 1 | 0 | 0 | **0** |  |  |
| 0 | 1 | 1 | 0 | 1 | **1** | = |  |
| 0 | 1 | 1 | 1 | 0 | **0** |  |  |
| 0 | 1 | 1 | 1 | 1 | **1** | = |  |
| 1 | 0 | 0 | 0 | 0 | **1** | = |  |
| 1 | 0 | 0 | 0 | 1 | **0** |  |  |
| 1 | 0 | 0 | 1 | 0 | **0** |  |  |
| 1 | 0 | 0 | 1 | 1 | **1** | = |  |
| 1 | 0 | 1 | 0 | 0 | **0** |  |  |
| 1 | 0 | 1 | 0 | 1 | **1** | = |  |
| 1 | 0 | 1 | 1 | 0 | **0** |  |  |
| 1 | 0 | 1 | 1 | 1 | **0** |  |  |
| 1 | 1 | 0 | 0 | 0 | **0** |  |  |
| 1 | 1 | 0 | 0 | 1 | **0** |  |  |
| 1 | 1 | 0 | 1 | 0 | **0** |  |  |
| 1 | 1 | 0 | 1 | 1 | **1** | = |  |
| 1 | 1 | 1 | 0 | 0 | **1** | = |  |
| 1 | 1 | 1 | 0 | 1 | **0** |  |  |
| 1 | 1 | 1 | 1 | 0 | **1** | = |  |
| 1 | 1 | 1 | 1 | 1 | **0** |  |  |

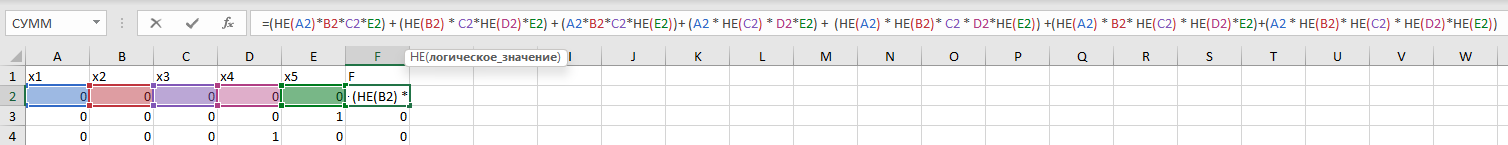
**Задание 3-4**

Для удобства в вычислениях заменил x1 = 1, x2 = 2 и т.д.



**Задание 5**

Составил выражение G(x) в Excel

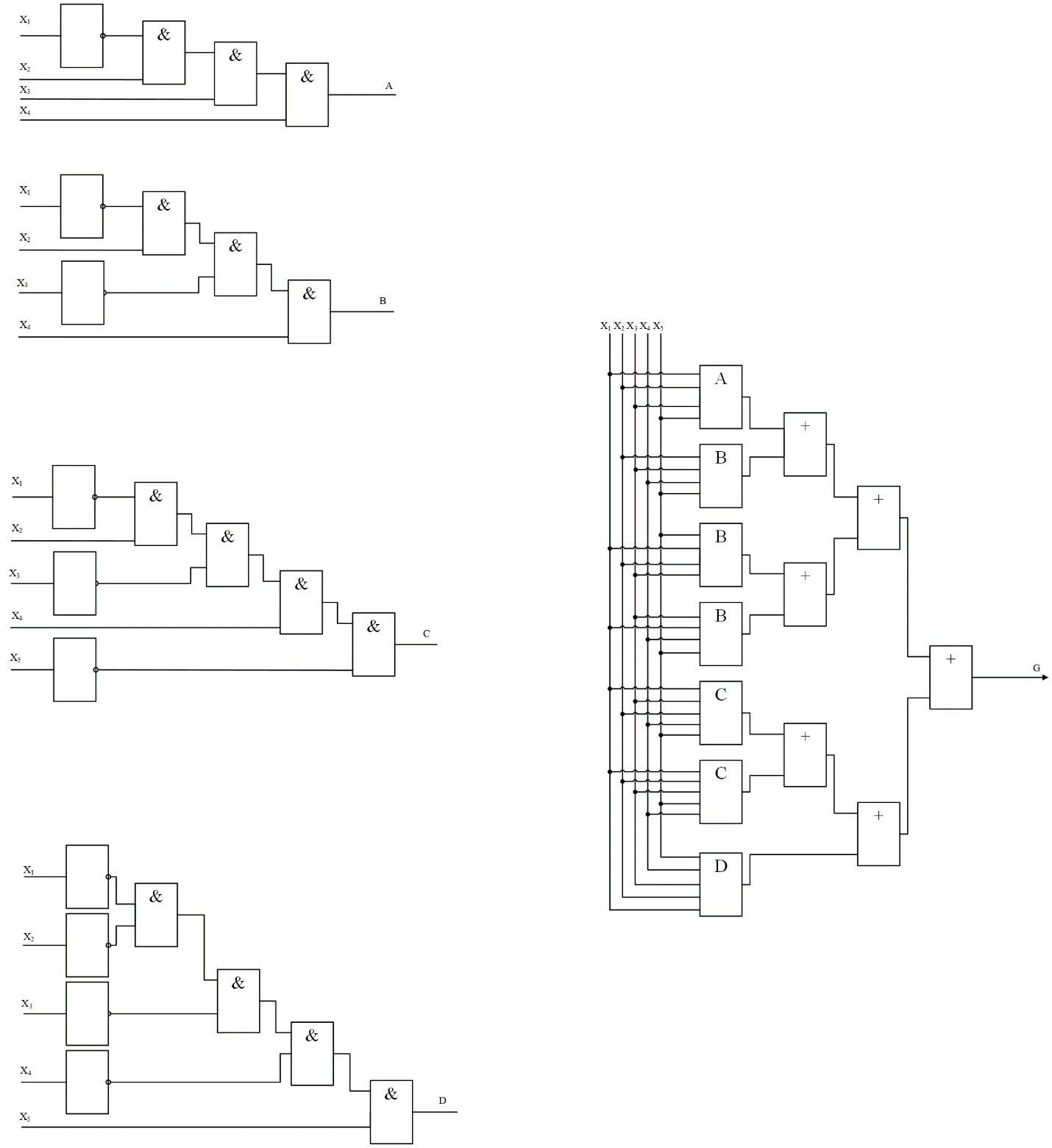




Полученная таблица истинности совпадает с исходной, формула составлена верно

**Задание 6**

Заменил повторяющиеся комбинаторные схемы, получил необходимые функции путём изменения порядка входных данных.

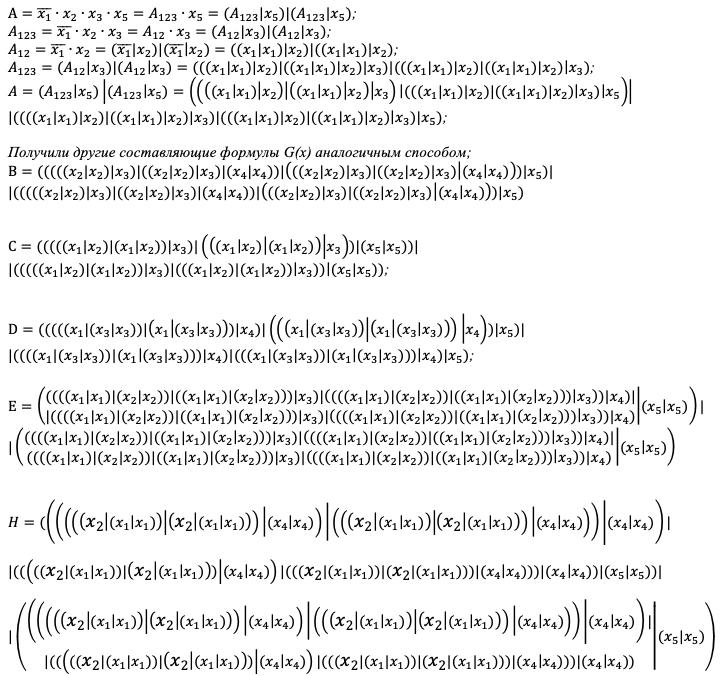


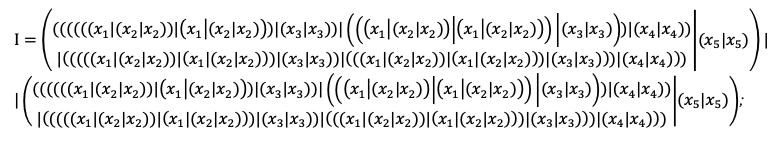
**Задание 7**

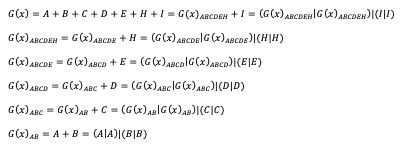
A=; B=; C=; D=;

E=; H=; I=

G(x)=A+B+C+D+E+H+I



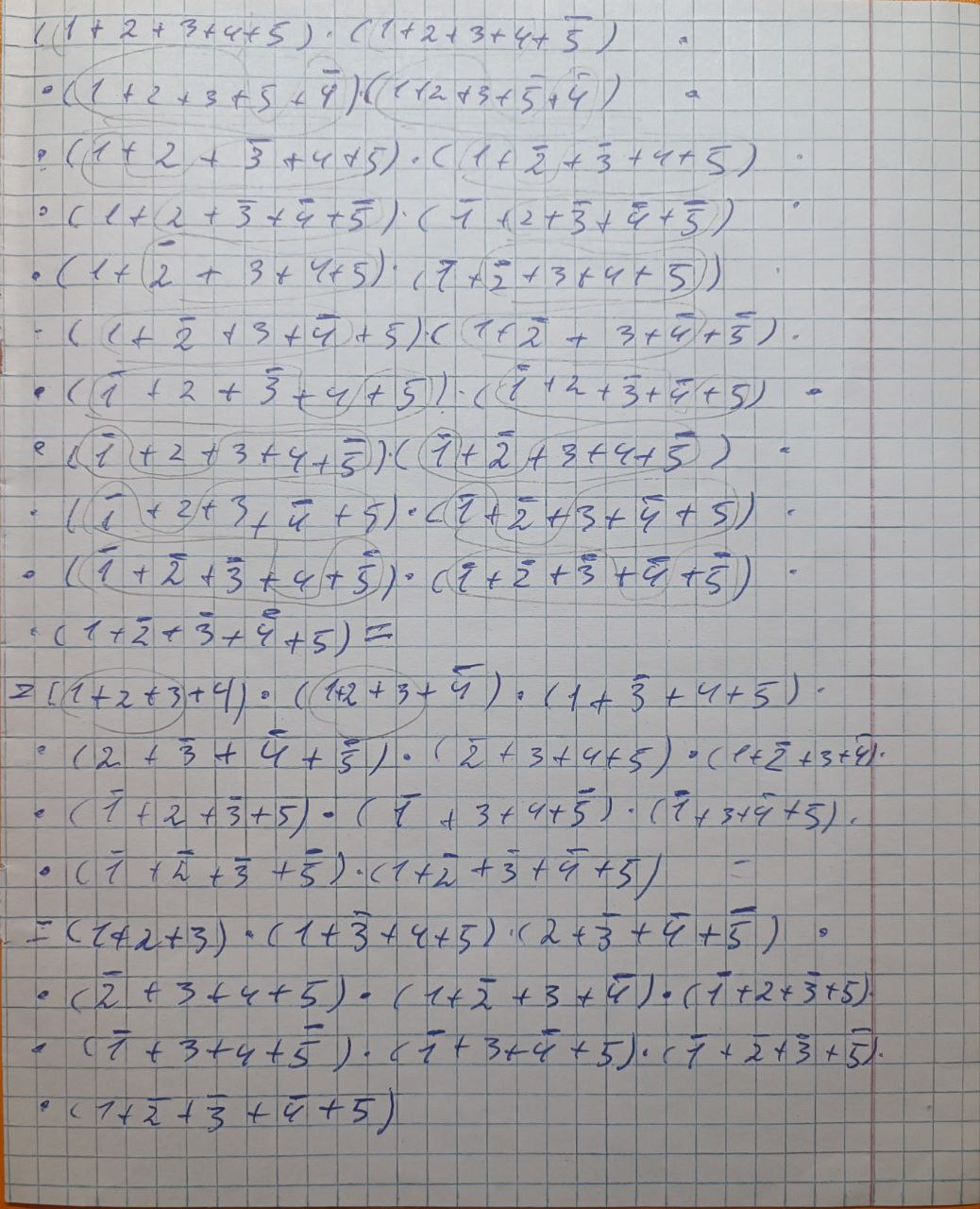




**Задание 8-9**

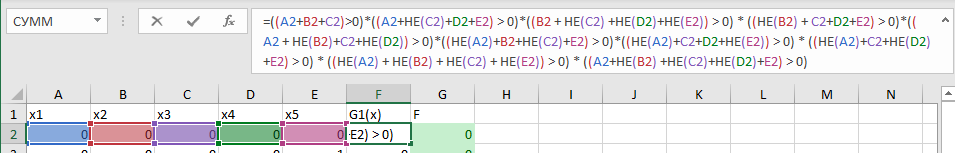
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | **0** | = |  |
| 0 | 0 | 0 | 0 | 1 | **0** | = |  |
| 0 | 0 | 0 | 1 | 0 | **0** | = |  |
| 0 | 0 | 0 | 1 | 1 | **0** | = |  |
| 0 | 0 | 1 | 0 | 0 | **0** | = |  |
| 0 | 0 | 1 | 0 | 1 | **1** |  |  |
| 0 | 0 | 1 | 1 | 0 | **1** |  |  |
| 0 | 0 | 1 | 1 | 1 | **0** | = |  |
| 0 | 1 | 0 | 0 | 0 | **0** | = |  |
| 0 | 1 | 0 | 0 | 1 | **1** |  |  |
| 0 | 1 | 0 | 1 | 0 | **0** | = |  |
| 0 | 1 | 0 | 1 | 1 | **0** | = |  |
| 0 | 1 | 1 | 0 | 0 | **0** | = |  |
| 0 | 1 | 1 | 0 | 1 | **1** |  |  |
| 0 | 1 | 1 | 1 | 0 | **0** | = |  |
| 0 | 1 | 1 | 1 | 1 | **1** |  |  |
| 1 | 0 | 0 | 0 | 0 | **1** |  |  |
| 1 | 0 | 0 | 0 | 1 | **0** | = |  |
| 1 | 0 | 0 | 1 | 0 | **0** | = |  |
| 1 | 0 | 0 | 1 | 1 | **1** |  |  |
| 1 | 0 | 1 | 0 | 0 | **0** | = |  |
| 1 | 0 | 1 | 0 | 1 | **1** |  |  |
| 1 | 0 | 1 | 1 | 0 | **0** | = |  |
| 1 | 0 | 1 | 1 | 1 | **0** | = |  |
| 1 | 1 | 0 | 0 | 0 | **0** | = |  |
| 1 | 1 | 0 | 0 | 1 | **0** | = |  |
| 1 | 1 | 0 | 1 | 0 | **0** | = |  |
| 1 | 1 | 0 | 1 | 1 | **1** |  |  |
| 1 | 1 | 1 | 0 | 0 | **1** |  |  |
| 1 | 1 | 1 | 0 | 1 | **0** | = |  |
| 1 | 1 | 1 | 1 | 0 | **1** |  |  |
| 1 | 1 | 1 | 1 | 1 | **0** | = |  |

(Для удобства в вычислениях x1 = 1, x2 = 2, x3 = 3, x4 = 4, x5 = 5).



**Задание 10**

Составил выражение G1(x) в Excel





Полученная таблица истинности совпадает с исходной, формула составлена верно

**Задание 11**

Составил функции в Си, написал тесты, соответствующие таблице истинности каждой из функций.

#include <stdbool.h>  
#include <assert.h>  
  
**bool** not(**bool** value) {  
 return !value;  
}  
  
**bool** and(**bool** valueA, **bool** valueB) {  
 return valueA && valueB;  
}  
  
**bool** or(**bool** valueA, **bool** valueB) {  
 return valueA || valueB;  
}  
  
**bool** xor(**bool** valueA, **bool** valueB) {  
 return and(or(not(valueA),  
 not(valueB)),  
 or(valueA,  
 valueB));  
}  
  
**bool** equal(**bool** valueA, **bool** valueB) {  
 return or(and(not(valueA),  
 not(valueB)),  
 and(valueA,  
 valueB));  
}  
  
**bool** implication(**bool** valueA, **bool** valueB) {  
 return or(not(valueA), valueB);  
}  
  
**bool** shefferStroke(**bool** valueA, **bool** valueB) {  
 return not(and(valueA, valueB));  
}  
  
**bool** pierceArrow(**bool** valueA, **bool** valueB) {  
 return not(or(valueA, valueB));  
}  
  
void testnot() {  
 **assert**(not(true) == false);  
 **assert**(not(false) == true);  
}  
  
void testand() {  
 **assert**(and(false, false) == false);  
 **assert**(and(false, true) == false);  
 **assert**(and(true, false) == false);  
 **assert**(and(true, true) == true);  
}  
  
void testor() {  
 **assert**(or(false, false) == false);  
 **assert**(or(false, true) == true);  
 **assert**(or(true, false) == true);  
 **assert**(or(true, true) == true);  
}  
  
void testxor() {  
 **assert**(xor(false, false) == false);  
 **assert**(xor(false, true) == true);  
 **assert**(xor(true, false) == true);  
 **assert**(xor(true, true) == false);  
}  
  
void testequal() {  
 **assert**(equal(false, false) == true);  
 **assert**(equal(false, true) == false);  
 **assert**(equal(true, false) == false);  
 **assert**(equal(true, true) == true);  
}  
  
void testimplication() {  
 **assert**(implication(false, false) == true);  
 **assert**(implication(false, true) == true);  
 **assert**(implication(true, false) == false);  
 **assert**(implication(true, true) == true);  
}  
  
void testShefferStroke() {  
 **assert**(shefferStroke(false, false) == true);  
 **assert**(shefferStroke(false, true) == true);  
 **assert**(shefferStroke(true, false) == true);  
 **assert**(shefferStroke(true, true) == false);  
}  
  
void testPierceArrow() {  
 **assert**(pierceArrow(false, false) == true);  
 **assert**(pierceArrow(false, true) == false);  
 **assert**(pierceArrow(true, false) == false);  
 **assert**(pierceArrow(true, true) == false);  
}  
  
void test() {  
 testnot();  
 testand();  
 testor();  
 testxor();  
 testequal();  
 testimplication();  
 testShefferStroke();  
 testPierceArrow();  
}  
  
int main() {  
 test();  
  
 return 0;  
}

**Задание 12**

#include <stdbool.h>  
#include <stdio.h>  
#include <windows.h>  
  
**bool** G(**bool** x1, **bool** x2, **bool** x3, **bool** x4, **bool** x5) {  
 return (!(x1) && x2 && x3 && x5) || (!(x2) && x3 && !(x4) && x5) || (x1 && x2 && x3 && !(x5)) ||  
 (x1 && !(x3) && x4 && x5) || (!(x1) && !(x2) && x3 && x4 && !(x5)) ||  
 (!(x1) && x2 && !(x3) && !(x4) && x5) || (x1 && !(x2) && !(x3) && !(x4) && !(x5));  
}  
  
**bool** G1(**bool** x1, **bool** x2, **bool** x3, **bool** x4, **bool** x5) {  
 return (x1 || x2 || x3) && (x1 || !(x3) || x4 || x5) && (x2 || !(x3) || !(x4) || !(x5)) &&  
 (!(x2) || x3 || x4 || x5) && (x1 || !(x2) || x3 || !(x4)) && (!(x1) || x2 || !(x3) || x5) &&  
 (!(x1) || x3 || x4 || !(x5)) && (!(x1) || x3 || !(x4) || x5) && (!(x1) || !(x2) || !(x3) || !(x5)) &&  
 (x1 || !(x2) || !(x3) || !(x4) || x5);  
}  
  
int main() {  
 SetConsoleOutputCP(**CP\_UTF8**);  
 printf("╔════╦════╦════╦════╦════╦════╦═════╗\n");  
 printf("║ x1 ║ x2 ║ x3 ║ x4 ║ x5 ║ Gx ║ G1x ║\n");  
  
 for (int x1 = 0; x1 < 2; x1++)  
 for (int x2 = 0; x2 < 2; x2++)  
 for (int x3 = 0; x3 < 2; x3++)  
 for (int x4 = 0; x4 < 2; x4++)  
 for (int x5 = 0; x5 < 2; x5++) {  
 printf("╠════╬════╬════╬════╬════╬════╬═════╣\n");  
 printf("║ %d ║ %d ║ %d ║ %d ║ %d ║ %d ║ %d ║\n",  
 x1, x2, x3, x4, x5,  
 G(x1, x2, x3, x4, x5),  
 G1(x1, x2, x3, x4, x5));  
 }  
  
 printf("╚════╩════╩════╩════╩════╩════╩═════╝\n");  
 return 0;  
}

Вывод: в ходе лабораторной работы изучил основные принципы и получить практические навыки работы с документами в приложении MS Office Excel.