**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**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

автоматизированныхсистем

Дисциплина: Дискретная математика

Лабораторная работа №1.3

Тема: «Теоретико-множественные тождества»

Выполнил:

студент группы ВТ-12

Воскобойников Илья Сергеевич

Проверил: Рязанов Ю. Д.

Белгород 2019

**Цель работы**: изучить методы доказательства теоретико-множественных тождеств.

1. На рис.1 изображены круги Эйлера, соответствующие множествам А, В и С, с пронумерованными элементарными областями (не содержащими внутри себя других областей). Заштриховать элементарные области в соответствии с вариантом задания (см. табл.2).

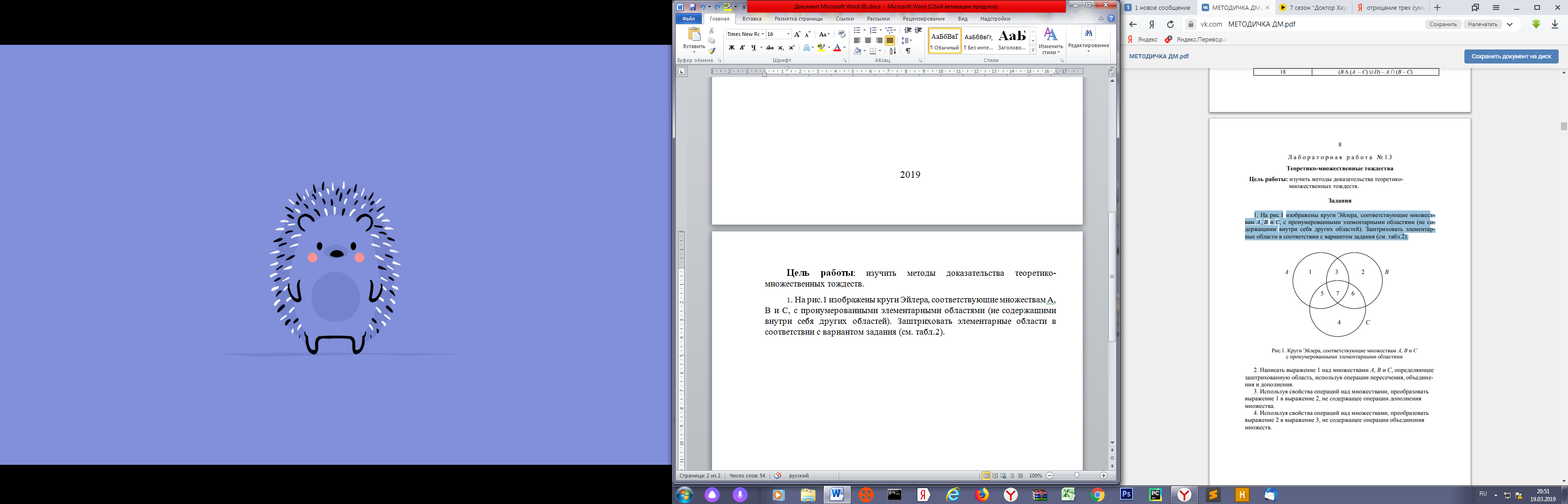


Рис.1. Круги Эйлера, соответствующие множествам А, В и С с пронумерованными элементарными областями

2. Написать выражение 1 над множествами А, В и С, определяющее заштрихованную область, используя операции пересечения, объединения и дополнения.

3. Используя свойства операций над множествами, преобразовать выражение 1 в выражение 2, не содержащее операции дополнения множества.

4. Используя свойства операций над множествами, преобразовать выражение 2 в выражение 3, не содержащее операции объединения множеств.

5. Используя свойства операций над множествами, преобразовать выражение 3 в выражение 4, не содержащее операции пересечения множеств.

6. Доказать тождественность выражений 2 и 3 методом характеристических функций.

7. Доказать тождественность выражений 2 и 4 методом логических функций. Для автоматизации доказательства написать программу, которая получает и сравнивает таблицы истинности логических функций.

Описание спецификаций функций

Спецификация функции Log\_F\_2

1.Заголовок: void Log\_F\_2(int \*a)

2.Назначение:Вычисляет значение таблицы истинности для выражения 2 описанного выше и записывает его в а

Спецификация функции Log\_F\_4

1.Заголовок: void Log\_F\_2(int \*a)

2.Назначение: Вычисляет значение таблицы истинности для выражения 4 описанного выше и записывает его в а

Спецификация функции PrintTable

1.Заголовок: void PrintTable(int \*a,int \*b)

2.Назначение: Выводит таблицу истинности для a и b, которые вычисляются в Log\_F\_4 и Log\_F\_2

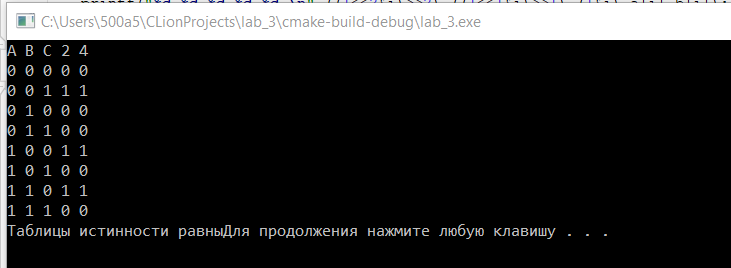
Спецификация функции A\_ravn\_B

**1.**Заголовок: \_Bool A\_ravn\_B (int\* a, int\* b, int n )

2.Назначение: возвращает true, если множества А и В равны и false в противном случае;

#include **<stdio.h>**#include **<stdbool.h>**#include **<process.h>  
  
void** Log\_F\_2(**int** \*a){  
 **for** (**int** i=0;i<8;i++){  
 **int** A=(1<<2&i)>>2;  
 **int** B=(1<<1&i)>>1;  
 **int** C=1&i;  
 a[i]=C&~A&~B|A&~C;  
 }  
}  
  
**void** Log\_F\_4(**int** \*a){  
 **for** (**int** i=0;i<8;i++){  
 **int** A=(1<<2&i)>>2;  
 **int** B=(1<<1&i)>>1;  
 **int** C=1&i;  
 a[i]=C&~A&~B|A&~C;  
 }  
}  
  
**\_Bool** A\_ravn\_B (**\_Bool** \* a, **\_Bool** \* b, **int** n ){  
 **int** i=0;  
 **\_Bool** F=**true**;  
 **while** ((i<n)&&(F==**true**)){  
 F=(a[i]==b[i]);  
 i++;  
 }  
 **return** F;  
}  
**void** PrintTable(**int** \*a,**int** \*b){  
 printf(**"A B C 2 4""\n"**);  
 **for**(**int** i=0;i<8;i++){  
 printf(**"%d %d %d %d %d \n"**,((1<<2&i)>>2),((1<<1&i)>>1),(1&i),a[i],b[i]);  
 }  
}  
  
**int** main() {  
 **int** n=8;  
 **int** a[n],b[n];  
 Log\_F\_2(a);  
 Log\_F\_4(b);  
 PrintTable(a,b);  
 **if**(A\_ravn\_B(a,b,n)){  
 printf(**"Таблицы истинности равны"**);  
 }  
 **else**{  
 printf(**"Таблицы истинности не равны"**);  
 }  
 system(**"pause"**);  
 **return** 0;  
}

Результат



8. Доказать тождественность выражений 3 и 4 теоретико-множественным методом. Для автоматизации доказательства написать программу, в которой вычисляются и сравниваются значения выражений 3 и 4 при А = {1,3,5,7}, B = {2,3,6,7} и C = {4,5,6,7}.

Спецификация функции F\_3

1.Заголовок:

void F\_3(bool \*A,bool \*B,bool \*C, bool \*Rez)

2.Назначение: Вычисляет значение выражения 2 описанного выше и записывает его в Rez

Спецификация функции F\_4

1.Заголовок:

void F\_4(bool \*A,bool \*B,bool \*C,bool \*Rez)

2.Назначение:Вычисляет значение выражения 4 описанного выше и записывает его в Rez

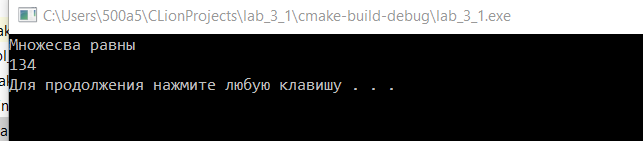
Функции для работы со множествами, хранящимися в виде массива Булевского типа, используемые в функциях main, F\_2, F\_4 описаны в лабораторной работе 1.1

#ifndef **LAB\_1\_3\_A\_ZAGALOVOK\_BOOL\_ARR\_H**#define **LAB\_1\_3\_A\_ZAGALOVOK\_BOOL\_ARR\_H  
  
void** input (**\_Bool**\* a, **int**\* n);  
**void** output (**\_Bool** \* a,**int** n);  
**\_Bool** A\_vkl\_B (**\_Bool** \* a, **\_Bool** \* b, **int** n );  
**\_Bool** A\_ravn\_B (**\_Bool** \* a, **\_Bool** \* b, **int** n );  
**\_Bool** A\_st\_vkl\_B (**\_Bool** \* a, **\_Bool** \* b, **int** n, **int** m );  
**void** A\_or\_B (**\_Bool**\* a, **\_Bool**\* b, **int** n, **\_Bool**\* c);  
**void** A\_and\_B (**\_Bool**\* a, **\_Bool**\* b, **int** n, **\_Bool**\* c );  
**void** A\_minus\_B (**\_Bool**\* a, **\_Bool**\* b, **int** n, **\_Bool**\* c);  
**void** A\_sm\_minus\_B (**\_Bool**\* a, **\_Bool**\* b, **int** n, **int** m , **\_Bool**\* c );  
**int** dop\_A (**\_Bool** \* a, **int** n, **\_Bool**\* c );  
#endif *//LAB\_1\_3\_A\_ZAGALOVOK\_BOOL\_ARR\_H*

#include **<stdio.h>**#include **<stdbool.h>  
  
void** input (**\_Bool**\* a, **int**\* n){  
 **int** j;  
 **for** (**int** i = 0; i <n ; ++i) {  
 a[i]=false;  
 }  
 **for** (**int** i = 0; i <n ; ++i) {  
 printf(**"%d"**,j);  
 a[j-1]=true;  
 }  
}  
  
  
**void** output (**\_Bool** \* a,**int** n) {  
 **for** (**int** i = 0; i < n; ++i) {  
 **if** (a[i]){  
 printf(**"%d"**,i+1);  
 }  
 }  
}  
  
  
  
**\_Bool** A\_vkl\_B (**\_Bool** \* a, **\_Bool** \* b, **int** n ){  
 **int** i=0;  
 **\_Bool** F=true;  
 **while** ((i<n)&&(F==true)){  
 F=(a[i]<=b[i]);  
 i++;  
 }  
 **return** F;  
}  
  
  
**\_Bool** A\_ravn\_B (**\_Bool** \* a, **\_Bool** \* b, **int** n ){  
 **int** i=0;  
 **\_Bool** F=true;  
 **while** ((i<n)&&(F==true)){  
 F=(a[i]==b[i]);  
 i++;  
 }  
 **return** F;  
}  
  
  
**\_Bool** A\_st\_vkl\_B (**\_Bool** \* a, **\_Bool** \* b, **int** n, **int** m ){  
 **\_Bool** F=(n < m) && A\_vkl\_B(a,b,n);  
 **return** F;  
}  
  
  
**void** A\_or\_B (**\_Bool**\* a, **\_Bool**\* b, **int** n, **\_Bool**\* c){  
 **for** (**int** i = 0; i < n; ++i) {  
 c[i]=(a[i]||b[i]);  
 }  
}  
  
  
**void** A\_and\_B (**\_Bool**\* a, **\_Bool**\* b, **int** n, **\_Bool**\* c ){  
 **for** (**int** i = 0; i < n; ++i) {  
 c[i]=(a[i]&&b[i]);  
 }  
}  
  
  
**void** A\_minus\_B (**\_Bool**\* a, **\_Bool**\* b, **int** n, **\_Bool**\* c){  
 **for** (**int** i = 0; i < n; ++i) {  
 c[i]=(a[i]>b[i]);  
 }  
}  
  
  
**void** A\_sm\_minus\_B (**\_Bool**\* a, **\_Bool**\* b, **int** n, **int** m , **\_Bool**\* c ){  
 **for** (**int** i = 0; i < n; ++i) {  
 c[i]=(a[i]!=b[i]);  
 }  
}  
  
  
**int** dop\_A (**\_Bool** \* a, **int** n, **\_Bool**\* c ){  
 **for** (**int** i = 0; i < n; ++i) {  
 c[i]=!a[i];  
 }  
}

#include **<stdio.h>**#include **<stdbool.h>**#include **<process.h>**#include **"zagalovok\_bool\_arr.h"  
int const** n=7;  
**\_Bool** A[7]={**true**,**false**,**true**,**false**,**true**,**false**,**true**},B[7]={**false**,**true**,**true**,**false**,**false**,**true**,**true**},C[7]={**false**,**false**,**false**,**true**,**true**,**true**,**true**};  
**void** F\_3(**\_Bool** \*A,**\_Bool** \*B,**\_Bool** \*C, **\_Bool** \*Rez){  
 **\_Bool** Q[n],M[n],Z[n];  
 A\_minus\_B(C,A,n,Q);  
 A\_minus\_B(Q,B,n,M);  
 dop\_A(M,n,Q);  
  
 A\_minus\_B(A,C,n,M);  
 dop\_A(M,n,Z);  
 A\_and\_B(Q,Z,n,M);  
 dop\_A(M,n,Rez);  
}  
**void** F\_4(**\_Bool** \*A, **\_Bool** \*B,**\_Bool** \*C, **\_Bool** \*Rez){  
 **\_Bool** Q[n],M[n],Z[n];  
 A\_minus\_B(C,A,n,Q);  
 A\_minus\_B(Q,B,n,M);  
 dop\_A(M,n,Q);  
  
 A\_minus\_B(A,C,n,M);  
 A\_minus\_B(Q,M,n,Z);  
 dop\_A(Z,n,Rez);  
}  
**int** main() {  
  
 **\_Bool** Rez3[n],Rez4[n];  
 F\_3(A,B,C,Rez3);  
 F\_4(A,B,C,Rez4);  
 **if** (A\_ravn\_B(Rez3,Rez4,n)) {  
 printf( **"Множесва равны\n"** );  
 output(Rez3,n);  
 }  
 **else**{  
 printf( **"Множесва не равны\n"** );  
 output(Rez3,n);  
 output(Rez4,n);  
 }  
 printf(**"\n"**);  
system(**"pause"**);  
 **return** 0;  
}

Результаты



Вывод: мы изучили методы доказательства теоретико-множественных тождеств.