**Лабораторная работа №1 по теме «Работа с множествами с помощью битовых шкал»**

**Вариант 18**

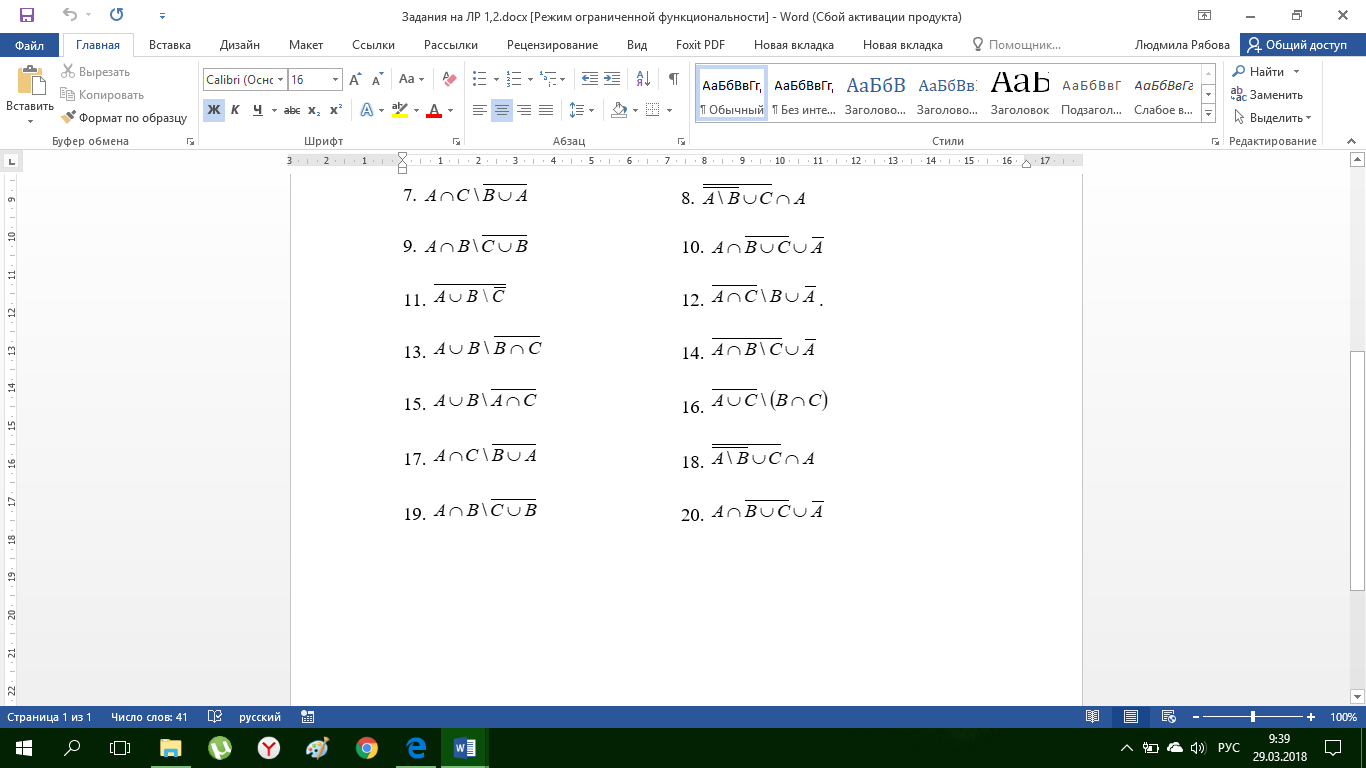
Выполнила студент гр. ПО(б)-61

Рябовой Л. А.

1. Постановка задачи

**Задание:** 1) Реализовать операции: объединение, пересечение, разность, дополнение. Продемонстрировать их выполнение. 2) Вычислить значение выражения. Ввод исходных множеств производится из текстового файла.

Исходное выражение:



2. Описание разработанных функций

Функция преобразования множества в битовую шкалу - PREO. Входная информация: данное множество типа string – dano, универсальное множество типа string – univ\_mn. Выходная информация: данное множество, представленное в виде битовой шкалы типа string.

string PREO (string dano, string univ\_mn)

{

string str;

bool ravenstvo=false;

for(int i=0;i<univ\_mn.size();i++)

{

for(int j=0;j<dano.size();j++)

if (dano[j]==univ\_mn[i])

{

str+='1';//устанавливаем единичку т.к. элементы множеств совпали

ravenstvo=true;

}

if(!ravenstvo)

str+='0';

ravenstvo=false;

}

return str;

}

Функция объединения множеств - obedinenie. Входная информация: два данных множеств в формате битовых шкал типа string – mn\_1, mn\_2. Выходная информация: множество, полученное после объединения двух данных множеств типа string.

string obedinenie(string mn\_1,string mn\_2)

{

string otvet;

for(int i=0;i<univ\_mn.size();i++)

{

if (mn\_1[i]=='0')

{

if(mn\_2[i]=='1')

otvet+= '1';

else

otvet+= '0';

}

else otvet+='1';

}

return otvet;

}

Функция пересечения множеств peresechenie. Входная информация: два данных множеств в формате битовых шкал типа string – mn\_1, mn\_2. Выходная информация: множество, полученное после пересечения двух данных множеств типа string.

string peresechenie(string mn\_1,string mn\_2)

{

string otvet;

for(int i=0;i<univ\_mn.size();i++)

{

if (mn\_1[i]=='0')

otvet+= '0';

else if (mn\_2[i]=='0')

otvet+= '0';

else otvet+='1';

}

return otvet;

}

Функция разности множеств - raznost. Входная информация: два данных множеств в формате битовых шкал типа string – mn\_1, mn\_2. Выходная информация: множество, полученное после разности двух данных множеств типа string.

string raznost(string mn\_1,string mn\_2)

{

string otvet;

for(int i=0;i<univ\_mn.size();i++)

{

if (mn\_1[i]=='0')

otvet+= '0';

else if (mn\_2[i]=='1')

otvet+= '0';

else otvet+='1';

}

return otvet;

}

Функция дополнения множества - dopolnenie. Входная информация: данное множество в формате битовой шкалы типа string – mn\_1. Выходная информация: множество, полученное после дополнения данного множеств типа string.

string dopolnenie(string mn\_1)

{

string otvet;

for(int i=0;i<univ\_mn.size();i++)

{

if (mn\_1[i]=='1')

otvet+= '0';

else otvet+= '1';

}

return otvet;

}

Функция вычисления выражения - vuragen. Входная информация: данные множества в формате битовых шкал типа string – А, В, С. Выходная информация: множество, полученное после вычисления выражения типа string.

string vuragen(string A,string B, string C)

{

string otvet;

otvet=peresechenie(dopolnenie(obedinenie(dopolnenie(raznost(A,B)),C)),A);

return otvet;

}

3. Пример

Исходная информация, представленная в текстовом файле:

ABCDEFG – универсальное множество

ABC – множество А

EG – множество В

CD – множество С

