**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 №1.4

Тема: «Теоретико-множественные уравнения»

Выполнил:

студент группы ВТ-12

Воскобойников Илья Сергеевич

Проверил: Рязанов Ю. Д.

Белгород 2019

Цели: научиться решать теоретико-множественные уравнения с применением ЭВМ

Задания:

1. Преобразовать исходное уравнение (см. “варианты заданий”) в уравнение с пустой правой частью.
2. Преобразовать левую часть уравнения к виду , используя разложение Шеннона по неизвестному множеству Х

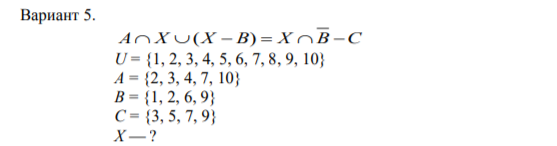
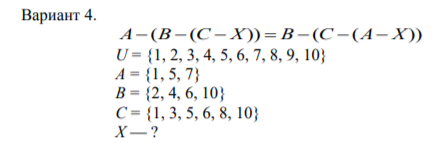


1. Написать программу, вычисляющую значения множеств и при заданных исходных множествах



1. Вычислить значения множеств C:\Users\500a5\AppData\Local\Microsoft\Windows\INetCache\Content.Word\18-04-2019 18-50-36.pngи C:\Users\500a5\AppData\Local\Microsoft\Windows\INetCache\Content.Word\18-04-2019 18-50-46.pngи сделать вывод о существовании решения уравнения. Если решения уравнения не существует, то выполнить 1-4 для следующего варианта
2. Определить мощность общего решения, найти некоторые или все частные решения, в том числе частные решения наименьшей и наибольшей мощности
3. Написать программу для проверки найденных решений

Вариант 4



1. Преобразовать исходное уравнение (см. “варианты заданий”) в уравнение с пустой правой частью.
2. Преобразовать левую часть уравнения к виду C:\Users\500a5\AppData\Local\Microsoft\Windows\INetCache\Content.Word\18-04-2019 18-49-33.png, используя разложение Шеннона по неизвестному множеству Х
3. Написать программу, вычисляющую значения множеств C:\Users\500a5\AppData\Local\Microsoft\Windows\INetCache\Content.Word\18-04-2019 18-50-36.pngи C:\Users\500a5\AppData\Local\Microsoft\Windows\INetCache\Content.Word\18-04-2019 18-50-46.pngпри заданных исходных множествах

Описание спецификаций функций

Спецификация функции f

1.Заголовок: int f (int \*A,int \*B, int\*C ,int\* U\_or\_noU,int nU\_or\_noU, int \*F1)

2.Назначение: Возвращает мощность множества F1 полученное в результате вычисления значение множества C:\Users\500a5\AppData\Local\Microsoft\Windows\INetCache\Content.Word\18-04-2019 18-50-36.png или , А –множество А ,



В –множество В , С –множество С , U\_or\_noU- универсум или пустое множество , nU\_or\_noU- мощность U\_or\_noU .

Функции для работы со множествами, хранящимися в виде отсортированного массива, используемые в функциях main, f описаны в лабораторной работе 1.1

#include **<stdio.h>**#include **"zagalovok\_arr\_sort.h"  
int const** NA=5;  
**int const** NB=4;  
**int const** NC=4;  
**int const** NU=10;  
**int const** NNoU =0;  
**int** f(**int** \*A,**int** \*B, **int**\*C ,**int**\* U\_or\_noU,**int** nU\_or\_noU, **int** \*F1){  
 **int** tmp1[10],ntmp1;  
 **int** tmp2[10],ntmp2;  
 **int** tmp3[10],ntmp3,  
 nF1;  
 ntmp1=A\_minus\_B(U\_or\_noU,B,nU\_or\_noU,NB,tmp1);  
 ntmp2=A\_and\_B(A,U\_or\_noU,NA,nU\_or\_noU,tmp2);  
 ntmp3=A\_or\_B(tmp1,tmp2,ntmp1,ntmp2,tmp3);  
 ntmp1=dop\_A(B,NB,tmp1);  
 ntmp2=A\_and\_B(U\_or\_noU,tmp1,nU\_or\_noU,ntmp1,tmp2);  
 ntmp1=A\_minus\_B(tmp2,C,ntmp2,NC,tmp1);  
 nF1=A\_sm\_minus\_B(tmp3,tmp1,ntmp3,ntmp1,F1);  
  
 **return** nF1;  
}  
**int** main() {  
 **int** U[]={1,2,3,4,5,6,7,8,9,10},  
 NoU[]={},  
 A[]={2,3,4,7,10},  
 B[]={1,2,6,9},  
 C[]={3,5,7,9},  
 F0[10],  
 F1[10],nf0,nf1,tmp[10],ntmp;  
 nf0=f(A,B,C,NoU,NNoU,F0);  
 output(F0,nf0);  
 ntmp=f(A,B,C,U,NU,tmp);  
 nf1=dop\_A(tmp,ntmp,F1);  
 output(F1,nf1);  
 **return** 0;  
}

1. Вычислить значения множеств C:\Users\500a5\AppData\Local\Microsoft\Windows\INetCache\Content.Word\18-04-2019 18-50-36.pngи C:\Users\500a5\AppData\Local\Microsoft\Windows\INetCache\Content.Word\18-04-2019 18-50-46.pngи сделать вывод о существовании решения уравнения. Если решения уравнения не существует, то выполнить 1-4 для следующего варианта
2. Определить мощность общего решения, найти некоторые или все частные решения, в том числе частные решения наименьшей и наибольшей мощности
3. Написать программу для проверки найденных решений

Описание спецификаций функций

Спецификация функции f1

1.Заголовок: int f1(int \*A,int \*B, int\*C ,int\* X,int nX, int \*F1)

2.Назначение: Возвращает мощность множества F1 полученное в результате вычисления левой части равенства , А –множество А ,

В –множество В , С –множество С X- множество X , nX- мощность X .

Спецификация функции f2

1.Заголовок: int f2(int \*A,int \*B, int\*C , int\* X,int nX, int \*F2)

2.Назначение: Возвращает мощность множества F2 полученное в результате вычисления правой части равенства , А –множество А ,

В –множество В , С –множество С X- множество X , nX- мощность X .

Функции для работы со множествами, хранящимися в виде отсортированного массива, используемые в функциях main, f описаны в лабораторной работе 1.1

#include **<stdio.h>**#include **"zagalovok\_arr\_sort.h"  
int const** NA=5;  
**int const** NB=4;  
**int const** NC=4;  
**int** f1(**int** \*A,**int** \*B, **int**\*C ,**int**\* X,**int** nX, **int** \*F1){  
 **int** tmp1[10],ntmp1;  
 **int** tmp2[10],ntmp2;  
 **int** nF1;  
 ntmp1=A\_minus\_B(X,B,nX,NB,tmp1);  
 ntmp2=A\_and\_B(A,X,NA,nX,tmp2);  
 nF1=A\_or\_B(tmp1,tmp2,ntmp1,ntmp2,F1);  
  
  
 **return** nF1;  
}  
**int** f2(**int** \*A,**int** \*B, **int**\*C ,**int**\* X,**int** nX, **int** \*F2){  
 **int** tmp1[10],ntmp1;  
 **int** tmp2[10],ntmp2;  
 **int** nF2;  
 ntmp1=dop\_A(B,NB,tmp1);  
 ntmp2=A\_and\_B(X,tmp1,nX,ntmp1,tmp2);  
 nF2=A\_minus\_B(tmp2,C,ntmp2,NC,F2);  
 **return** nF2;  
}  
  
**int** main() {  
 **int** A[]={2,3,4,7,10},  
 B[]={1,2,6,9},  
 C[]={3,5,7,9},  
 F1[10],nf1,nf2,F2[10],X[10],nX;  
 printf(**"введите мощьность Х "**);  
 scanf(**"%d"**,&nX);  
 printf(**"введите множество Х "**);  
 input(X,&nX);  
 nf1=f1(A,B,C,X,nX,F1);  
 nf2=f2(A,B,C,X,nX,F2);  
 **if**(A\_ravn\_B(F1,F2,nf1,nf2)){  
 printf(**"решение подходит"**);  
 } **else**{  
 printf(**"решение не подходит"**);  
 }  
  
 **return** 0;  
}