**Лабораторная работа №2 по теме «Работа с множествами с помощью динамических списков»**

**Вариант 18**

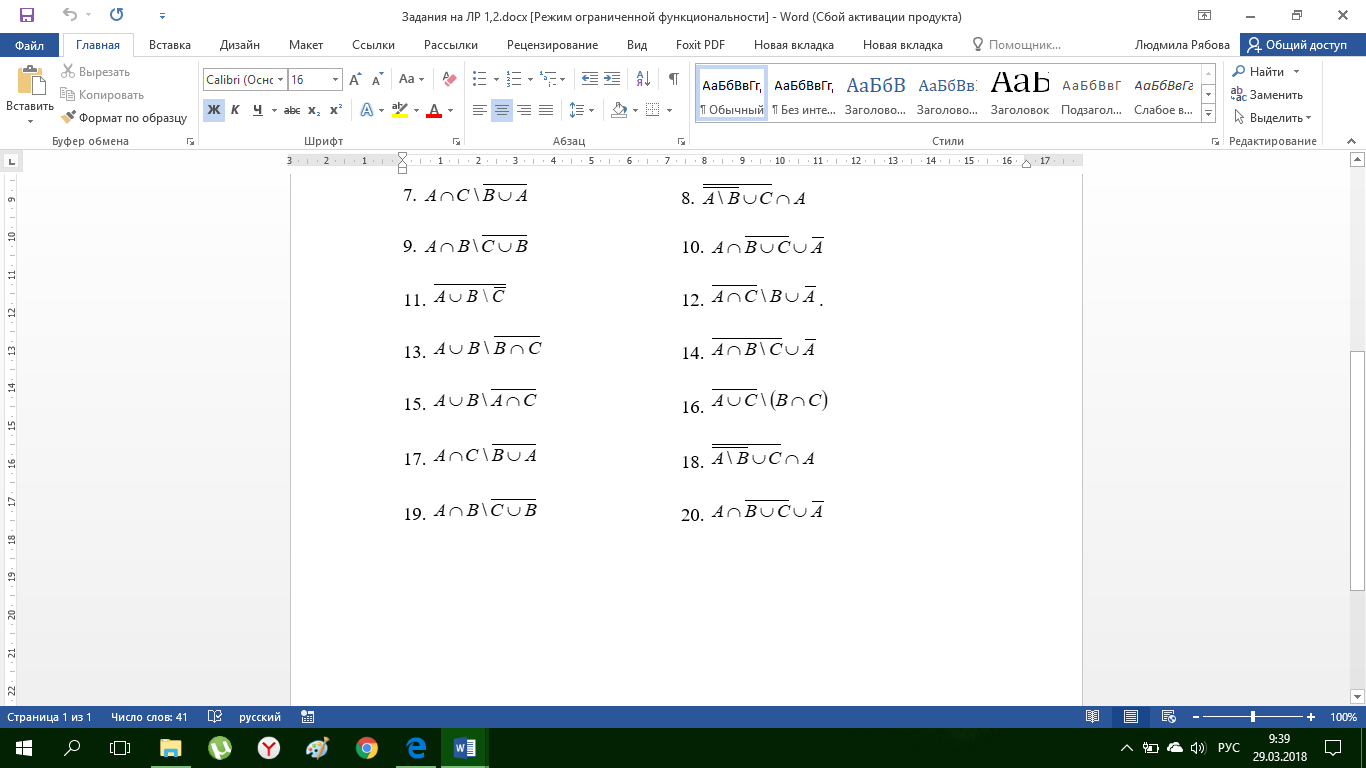
Выполнила студент гр. ПО(б)-61

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, списки отсортированны.

Исходное выражение:



2. Описание разработанных функций

Функция, выводящая все элементы списка – множества на экран (void) - all\_elem. Входная информация: указатель на начало списка - in.

void all\_elem(MNOGESTVO\* in)

{

if (!in)

{

cout<<"Елементов нет "<<endl;

}

else

{

MNOGESTVO \*node = in;

while(node)

{

cout<<node->elem <<"->";

node=node->next;

}

cout<<endl;

}

}

Функция для добавления элемента в конец множества - inser. Входная информация: указатель на начало множества - in, добавляемый элемент типа char - elem. Выходная информация: указатель на начало множества (с добавленным элементом)

MNOGESTVO\* inser (MNOGESTVO\* in, char elem)

{

MNOGESTVO \*new\_node = new MNOGESTVO; //Выделяем участок памяти

new\_node -> elem = elem;

new\_node->next = NULL;

if (!in)//список пустой

{

return new\_node;

}

else

{

MNOGESTVO \*cur\_node = in; //Временный указатель на начало списка

while (cur\_node->next)//Пока не конец списка

{

cur\_node=cur\_node->next;

}

cur\_node->next = new\_node;

return in;

}

}

Функция преобразования множества в динамический список - PREO. Входная информация: исходное множество типа string – dano, указатель на начало списка-множества – in. Выходная информация: указатель на начало списка-множества.

MNOGESTVO\* PREO(string dano, MNOGESTVO\* in)

{

if (!dano.size())//Множество пустое

{

cout<<"Множество пустое"<<endl;

return in;

}

else

{

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

{

in=inser(in, dano[i]);

}

}

return in;

}

Функция объединения множеств - obedinenie. Входная информация: указатель на ответ – otvet, два данных множеств в формате динамических списков – mn\_1, mn\_2. Выходная информация: указатель на множество, полученное после объединения двух данных множеств.

MNOGESTVO\* obedinenie(MNOGESTVO\* otvet,MNOGESTVO\* mn\_1, MNOGESTVO\* mn\_2)

{

otvet=NULL;

MNOGESTVO \*cur\_mn1 = mn\_1, \*cur\_mn2 = mn\_2; //Временные указатели на начало списков

//Если одно из множеств пусто

if (cur\_mn1 == NULL) {

otvet = mn\_2;

return otvet;

}

if (cur\_mn2 == NULL) {

otvet = mn\_1;

return otvet;

}

if (cur\_mn1->elem < cur\_mn2->elem)

{

otvet=inser(otvet,cur\_mn1->elem );

cur\_mn1 = cur\_mn1->next;

}

else

{

otvet=inser(otvet,cur\_mn2->elem );

cur\_mn2 = cur\_mn2->next;

}

while (cur\_mn1 && cur\_mn2)

{

if (cur\_mn1->elem < cur\_mn2->elem)

{

otvet=inser(otvet,cur\_mn1->elem );

cur\_mn1 = cur\_mn1->next;

}

else if (cur\_mn1->elem == cur\_mn2->elem)

{

cur\_mn1 = cur\_mn1->next;

}

else

{

otvet=inser(otvet,cur\_mn2->elem );

cur\_mn2 =cur\_mn2->next;

}

}

if (cur\_mn1)

{

while (cur\_mn1)

{

otvet=inser(otvet,cur\_mn1->elem );

cur\_mn1 =cur\_mn1->next;

}

}

if (cur\_mn2) {

while (cur\_mn2) {

otvet=inser(otvet,cur\_mn2->elem );

cur\_mn2 = cur\_mn2->next;

}

}

return otvet;

}

Функция пересечения множеств - peresechenie. Входная информация: указатель на ответ – otvet, два данных множеств в формате динамических списков – mn\_1, mn\_2. Выходная информация: указатель на множество, полученное после пересечения двух данных множеств.

MNOGESTVO\* peresechenie(MNOGESTVO\* otvet,MNOGESTVO\* mn\_1, MNOGESTVO\* mn\_2)

{

otvet=NULL;

MNOGESTVO \*cur\_mn1 = mn\_1, \*cur\_mn2 = mn\_2; //Временные указатели на начало списков

//Если одно из множеств пусто

if ((cur\_mn1 == NULL) || (cur\_mn2 == NULL) )

{

return otvet;

}

while (cur\_mn1 && cur\_mn2)

{

if (cur\_mn1->elem == cur\_mn2->elem)

{

otvet=inser(otvet,cur\_mn1->elem );

cur\_mn1 = cur\_mn1->next;

cur\_mn2 = cur\_mn2->next;

}

else if (cur\_mn1->elem < cur\_mn2->elem)

{

cur\_mn1 = cur\_mn1->next;

}

else

{

cur\_mn2 =cur\_mn2->next;

}

}

return otvet;

}

Функция разности множеств - raznost. Входная информация: указатель на ответ – otvet, два данных множеств в формате динамических списков – mn\_1, mn\_2. Выходная информация: указатель на множество, полученное после разности двух данных множеств.

MNOGESTVO\* raznost(MNOGESTVO\* otvet,MNOGESTVO\* mn\_1, MNOGESTVO\* mn\_2)

{

otvet=NULL;

MNOGESTVO \*cur\_mn1 = mn\_1, \*cur\_mn2 = mn\_2; //Временные указатели на начало списков

//Если второе множество пусто

if (cur\_mn2 == NULL)

{

otvet = mn\_1;

return otvet;

}

//Если первое множество пусто

if (cur\_mn1 == NULL)

{

return otvet;

}

while (cur\_mn1 && cur\_mn2)

{

if (cur\_mn1->elem == cur\_mn2->elem)

{

cur\_mn1 = cur\_mn1->next;

cur\_mn2 = cur\_mn2->next;

}

else if (cur\_mn1->elem < cur\_mn2->elem)

{

otvet=inser(otvet,cur\_mn1->elem );

cur\_mn1 = cur\_mn1->next;

}

else if (cur\_mn1->elem > cur\_mn2->elem)

{

cur\_mn2 = cur\_mn2->next;

}

else

{

otvet=inser(otvet,cur\_mn1->elem );

cur\_mn2 =cur\_mn2->next;

}

}

if (cur\_mn1)

{

while (cur\_mn1)

{

otvet=inser(otvet,cur\_mn1->elem );

cur\_mn1 =cur\_mn1->next;

}

}

return otvet;

}

Функция дополнения множества - dopolnenie. Входная информация: указатель на ответ – otvet, данное множеств в формате динамического списка – mn\_1. Выходная информация: указатель на множество, полученное после дополнения данного множества.

MNOGESTVO\* dopolnenie (MNOGESTVO\* otvet, MNOGESTVO\* mn\_1)

{

otvet=NULL;

MNOGESTVO \*cur\_mn1 = mn\_1, \*cur\_mn2 = univ\_spisok; //Временные указатели на начало списков

//Если множествo пусто

if (cur\_mn1 == NULL)

{

otvet = univ\_spisok;

return otvet;

}

if (cur\_mn2 == NULL)

{

cout<<"Универсальное множество пустое !!!";

return NULL;

}

while (cur\_mn1)

{

if (cur\_mn1->elem > cur\_mn2->elem)

{

otvet=inser(otvet,cur\_mn2->elem );

cur\_mn2 = cur\_mn2->next;

}

else if (cur\_mn1->elem == cur\_mn2->elem)

{

cur\_mn1 = cur\_mn1->next;

cur\_mn2 = cur\_mn2->next;

}

}

if (cur\_mn2)

{

while (cur\_mn2)

{

otvet=inser(otvet,cur\_mn2->elem );

cur\_mn2 =cur\_mn2->next;

}

}

return otvet;

}

Функция вычисления выражения - vuragen. Входная информация: указатель на ответ – otvet, данные множества в формате динамического списка – А, В, С. Выходная информация: указатель на начало множества, полученного после вычисления выражения.

MNOGESTVO\* vuragen(MNOGESTVO\* A,MNOGESTVO\* B, MNOGESTVO\* C)

{

otvet=NULL;

otvet=peresechenie(otvet,dopolnenie(otvet,obedinenie(otvet,dopolnenie(otvet,raznost(otvet,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

