МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №5

По дисциплине «Дискретная математика»

За 3 семестр

Тема: «Комбинаторика. Группы»

Выполнила:

студентка 2 курса

группы АС-56

Карпенко М.В.

Проверил:

Глущенко Т.А.

Брест 2020

*Вариант 7. Комбинаторика*

1. Вычислить максимальное количество чисел, которые можно представить в шестнадцатеричной системе четырехразрядным числом.

Самое большое число с четырьмя разрядами в шестнадцатеричной системе: FFFF16 = 6553510. Самое маленькое число с четырьмя разрядами в шестнадцатеричной системе: 100016 = 409610. Таким образом, можно вычислить:

*Ответ: 61439.*

2. Сколькими способами можно составить трехцветный полосатый флаг, если имеется материал пяти различных цветов и среди полос одна обязательно должна быть красной.

Допустим, что верхняя полоса – красного цвета, тогда для средней полосы существует 5 возможных цветов, для нижней – 4. Таким образом, . Допуская, что среднюю и нижнюю полосы также можно взять за полосы красного цвета, умножим полученное число на 3:

*Ответ:* *60.*

3. Сколькими способами можно распределить 15 студентов по 3-м учебным группам по 5 студентов в каждой?

Воспользуемся формулой: , где n – все студенты, x – количество групп, y – количество людей в группе. Подставим свои числа: .

*Ответ:* *126126.*

4. Решить линейное однородное рекуррентное соотношение 2-го порядка:   
.

*Ответ: .*

*Код программы:*

[#include](https://vk.com/im?sel=274580394&st=%23include) <iostream>  
  
using namespace std;  
  
int func(int x)  
{  
if (x == 0)  
return 2;  
if (x == 1)  
return 5;  
  
return 7 \* (func(x - 1)) - 12 \* (func(x - 2));  
}  
int main()  
{  
cout « func(5) « endl;  
system("pause");  
return 0;  
}

*Результаты работы программы:*

https://sun9-73.userapi.com/impg/MXkRiWYZQjeEwvZDoZ8dtooxFlG6jHlaXKlnxw/_jUJvcqzEEQ.jpg?size=378x43&quality=96&proxy=1&sign=a6d62ce971c033f6352cdd7f608cee33&type=album

5. Вычислить коэффициенты при произведении в разложении

При разложении степени коэффициенты рассчитываются по формуле: . Вычислим коэффициент при данном произведении в данном разложении: .

*Ответ: 560.*

6. Написать программу генерации k- элементных подмножеств n-элементного множества в лексикографическом порядке. Вход: k – мощность подмножества, n - мощность множества.

*Код программы:*

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <iostream>

using namespace std;

void anti(int \* mn, int k,int kol);

int main(int argc, char\* argv[])

{

int\* mn;

int k, i, kol;

cout<<"vvedite kolichestvo elementov"<<endl;

cin >> k;

kol = k;

mn = new int [k];

k--;

for (i = 0; i <kol; i++)

(mn[i] = i + 1);

anti(mn, k, kol);

delete[] mn;

system("pause");

return 0;

}

void anti(int \* mn, int k,int kol)

{

int i;

if (k == 0)

{

for (i = 0; i<kol; i++)

cout<<" "<< mn[i];

cout << endl;

}

else

{

for (i = 0; i <= k; ++i)

{

anti(mn, k - 1,kol);

if (i<k)

{

swap(mn[i], mn[k]);

for (int i = 0, j = (k-1); i < j; i++, j--)

swap(mn[i], mn[j]);

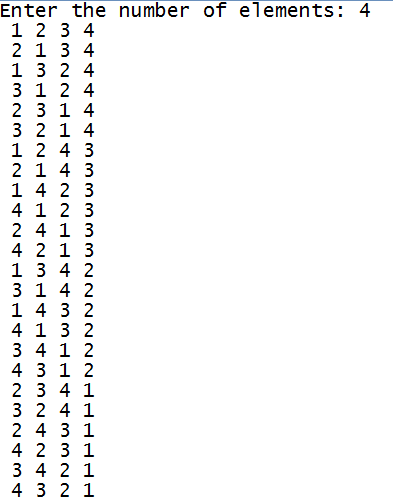
}

}

}

}

*Результаты работы программы:*



*Вариант 7. Группы*

Задание 1. Для заданной в таблице 1 группы:

1. Построить таблицу Кэли. Определить нейтральный элемент. Для каждого элемента группы указать обратный элемент.

2. Определить является ли группа циклической, указать порядок каждого элемента в группе.

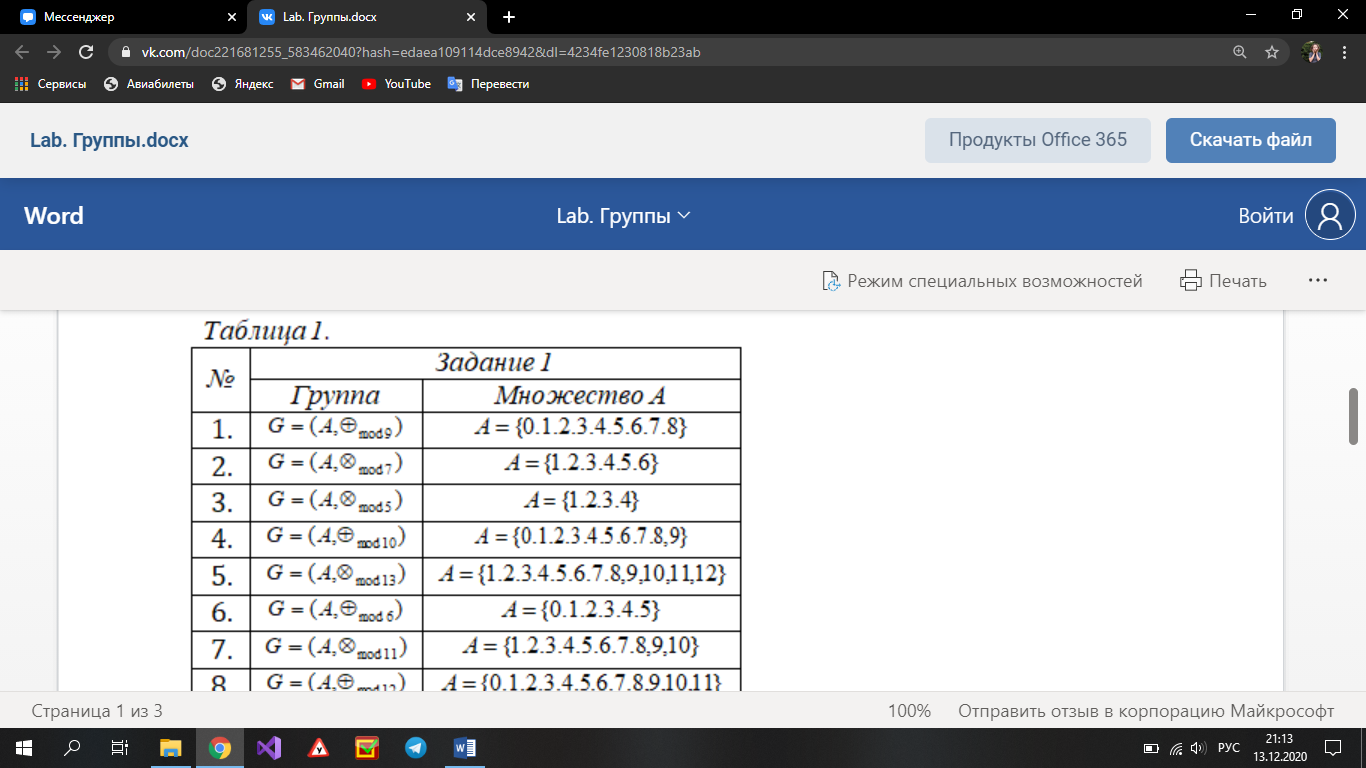
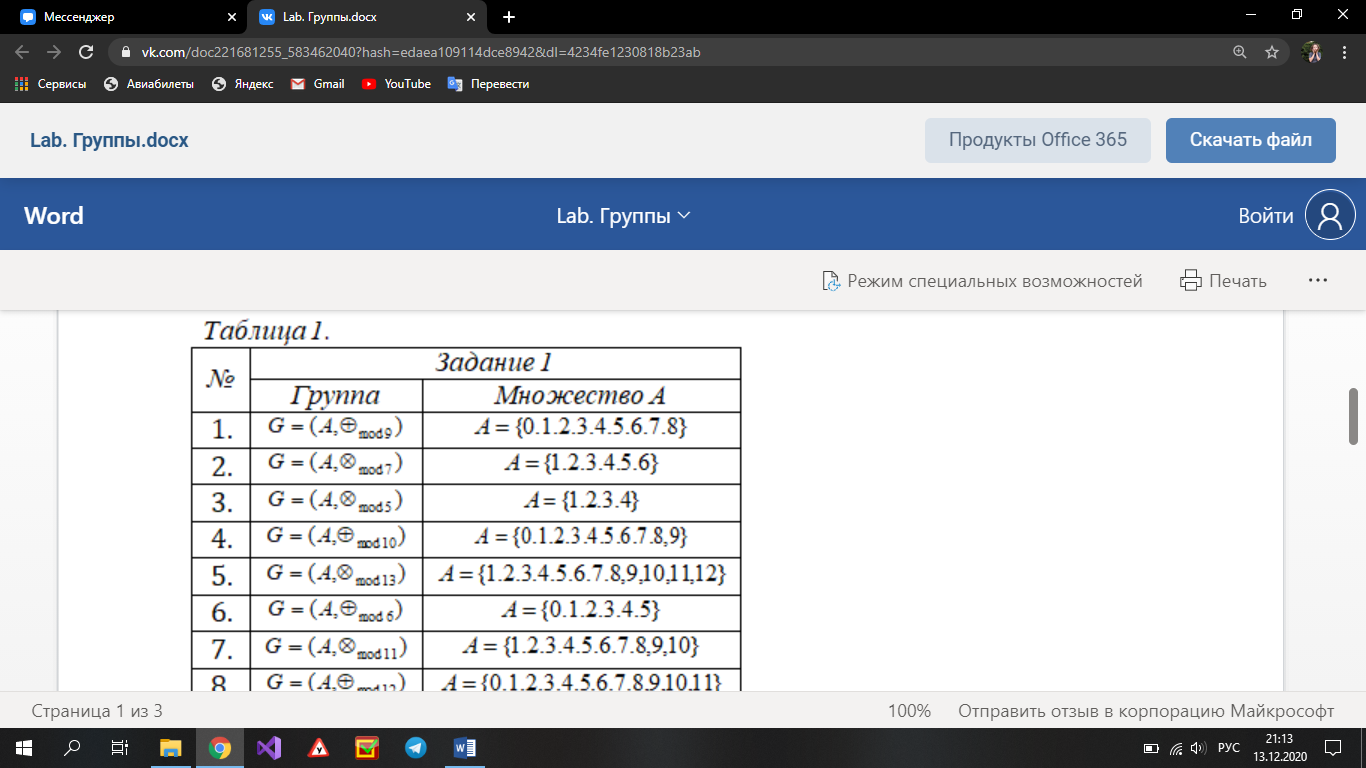
3. Определить является ли группа абелевой.

4. Указать все подгруппы заданной группы и их порядок. Определить порождающие элементы для подгрупп. Для одной из подгрупп построить таблицу Кэли.

5. Разложить группу на смежные классы по одной из подгрупп. Построить фактор множество для группы по данной подгруппе.

6. Построить таблицу Кэли для фактор-группы по одной из подгрупп.

Пункты 2, 3 выполнять вручную, остальные пункты – программная реализация.

Код программы:

#include <iostream>

#include <iomanip>

#include <vector>

#include <iterator>

#include <algorithm>

using namespace std;

const int N = 10;

void task1(vector<int>);

void task2();

int mult(int, int); //умножение по модулю 11

int neutral(vector<int>); //функция поиска нейтрального элемента

bool abel(vector<int>); //функция, проверяющая на коммутативность

bool comp(vector<int>, vector<int>); //сравнение двух групп

int main()

{

setlocale(0, "");

vector<int> group = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }; //множество, на котором задана операция

task1(group);

task2();

return 0;

}

void task1(vector<int> group)

{

vector<vector<int>> Keli(N, vector<int>(N)); //таблица Кэли

vector<vector<int>> ord(N);

vector<vector<int>> subgroup; //подгруппы

cout << "Таблица Кэли для группы: " << endl;

cout << " ";

for (int i = 0; i < N; i++)

cout << setw(4) << group[i];

cout << endl;

for (int i = 0; i < N; i++)

{

cout << setw(4) << group[i];

for (int j = 0; j < N; j++)

{

Keli[i][j] = mult(group[i], group[j]);

cout << setw(4) << Keli[i][j];

}

cout << endl;

}

int n = neutral(group); //поиск нейтрального элемента

if (n == -1) cout << "Нейтрального элемента нет." << endl;

else

{

cout << "Нейтральный элемент: " << n << endl;

cout << "Обратные элементы: " << endl;

for (int i = 0; i < N; i++)

{

cout << group[i] << " -> ";

for (int j = 0; j < N; j++)

{

if (Keli[i][j] == n)

{

cout << group[j] << endl;

break;

}

}

}

cout << "Порядки элементов: " << endl;

for (int i = 1; i < N; i++)

{

int tmp = 1;

do

{

tmp = mult(tmp, group[i]);

ord[i].push\_back(tmp);

} while (tmp != n);

cout << "<" << group[i] << "> = " << ord[i].size() << endl;

sort(ord[i].begin(), ord[i].end());

}

}

for (int i = 1; i < N; i++)

if (ord[i] == group) //если есть порождающий элемент, то группы циклическая

{

cout << "Группа циклическая. " << endl;

break;

}

if (abel(group)) cout << "Группа абелева." << endl; //проверка на коммутативность

else cout << "Группа не абелева." << endl;

vector<bool> vis(N, false);

int p = 1;

for (int i = 1; i < N; i++)

{

if ((ord[i] != group) && (vis[i] == false))

{

subgroup.push\_back(ord[i]);

cout << "H" << p++ << " = { ";

copy(ord[i].begin(), ord[i].end(), ostream\_iterator<int>(cout, " "));

cout << "} ";

for (int j = 0; j < N; j++)

if (ord[i] == ord[j])

{

cout << "= <" << group[j] << "> ";

vis[j] = true;

}

cout << endl;

}

}

cout << "Таблица Кэли для подгруппы H1:" << endl;

cout << " ";

for (int i = 0; i < subgroup[0].size(); i++)

cout << setw(4) << subgroup[0][i];

cout << endl;

for (int i = 0; i < subgroup[0].size(); i++)

{

cout << setw(4) << subgroup[0][i];

for (int j = 0; j < subgroup[0].size(); j++)

{

cout << setw(4) << mult(subgroup[0][i], subgroup[0][j]);

}

cout << endl;

}

cout << "Левые смежные классы: " << endl;

vector<vector<int>> c; //смежные классы

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

{

c.clear();

for (int j = 0; j < N; j++)

{

vector<int> tmp;

bool flag = false;

for (int k = 0; k < subgroup[i].size(); k++)

tmp.push\_back(mult(group[j], subgroup[i][k]));

for (int k = 0; k < c.size(); k++)

if (comp(tmp, c[k]))

{

flag = true;

break;

}

if (!flag)

{

if (j == 0) cout << "H" << i + 1 << " = { ";

else cout << j + 1 << "H" << i + 1 << " = { ";

copy(tmp.begin(), tmp.end(), ostream\_iterator<int>(cout, " "));

cout << "}" << endl;

c.push\_back(tmp);

}

}

cout << "G/H" << i + 1 << " = { ";

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

{

if (c[j][0] == 1) cout << "H" << i + 1 << " ";

else cout << c[j][0] << "H" << i + 1 << " ";

}

cout << "}" << endl;

}

cout << "Таблица Кэли для G/H" << subgroup.size() << ":" << endl;

cout << setw(15) << "H" << subgroup.size();

for (int i = 1; i < c.size(); i++)

cout << setw(6) << c[i][0] << "H" << subgroup.size();

cout << endl;

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

{

if (i == 0) cout << setw(7) << "H" << subgroup.size();

else cout << setw(6) << c[i][0] << "H" << subgroup.size();

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

{

int tmp = mult(c[i][0], c[j][0]);

vector<int> temp;

for (int k = 0; k < subgroup.back().size(); k++)

temp.push\_back(mult(tmp, subgroup.back()[k]));

sort(temp.begin(), temp.end());

if (temp[0] == 1) cout << setw(7) << "H" << subgroup.size();

else cout << setw(6) << temp[0] << "H" << subgroup.size();

}

cout << endl;

}

}

void task2()

{

vector<int> a = { 0,1,2,3 }; //множество А

vector<int> b = { 1,3,9,27 }; //множество B

vector<int> tmp(a.size(), 0);

vector<int> tmp2(b.size(), 0);

bool flag = true;

cout << "A = { ";

copy(a.begin(), a.end(), ostream\_iterator<int>(cout, " "));

cout << "}\nB = { ";

copy(b.begin(), b.end(), ostream\_iterator<int>(cout, " "));

cout << "}" << endl;

cout << "Соответсвие: " << endl;

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

{

if (b[i] != pow(3, a[i])) flag = false; //проверка на биекцию

cout << a[i] << " <--> " << pow(3, a[i]) << endl;

}

if (flag) cout << "Изоморфизм." << endl;

else

{

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

{

int temp = pow(3, a[i]);

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

if (temp == b[j])

{

tmp[i]++; //проверка на сюръективность

tmp2[j]++; //проверка на инъективность

}

}

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

if (tmp[i] != 1) flag = true;

if (!flag) cout << "Эпиморфизм." << endl;

else

{

for (int i = 0; i < tmp2.size(); i++)

if (tmp2[i] != 1) flag = false;

if (flag) cout << "Мономорфизм." << endl;

}

}

}

int mult(int a, int b)

{

return (a \* b) % 11; //умножение по модулю 11

}

int neutral(vector<int> group)

{

int neutral;

for (int i = 0; i < N; i++)

{

neutral = group[i];

for (int j = 0; j < N; j++)

{

if ((mult(group[i], group[j]) != mult(group[j], group[i])) || (mult(group[i], group[j]) != group[j]))

{

neutral = neutral && 0;

break;

}

}

if (neutral) return neutral;

}

return -1;

}

bool abel(vector<int> group)

{

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

if (mult(group[i], group[j]) != mult(group[j], group[i])) return false; //проверка на коммутативность

}

return true;

}

bool comp(vector<int> a, vector<int> b)

{

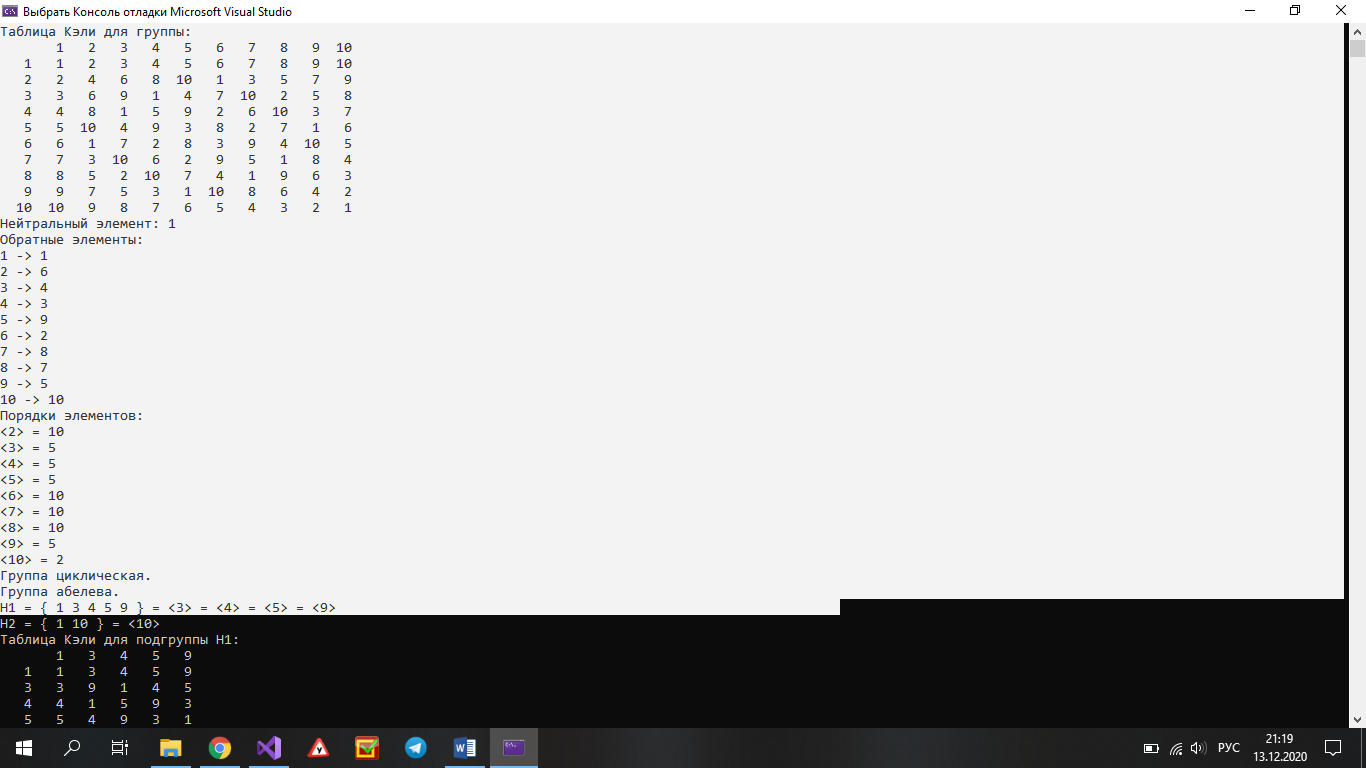
sort(a.begin(), a.end());

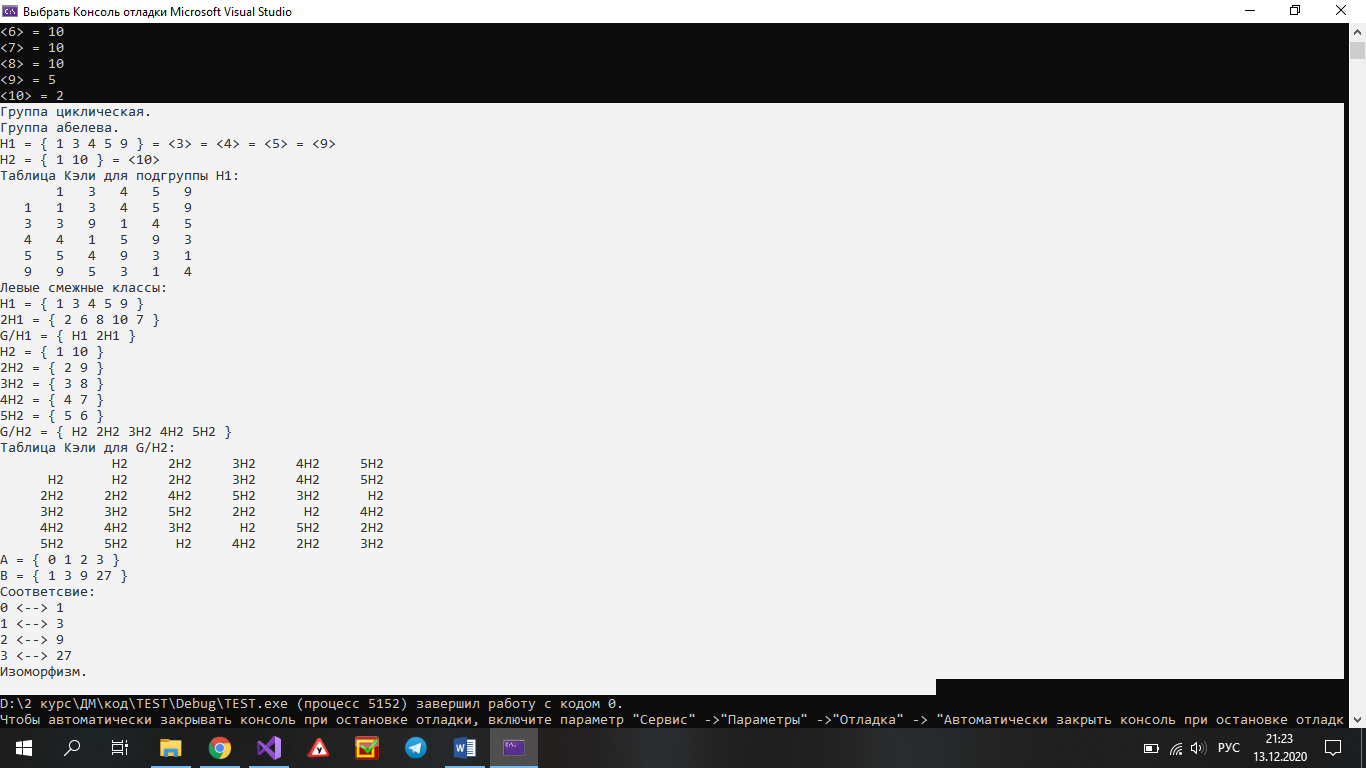
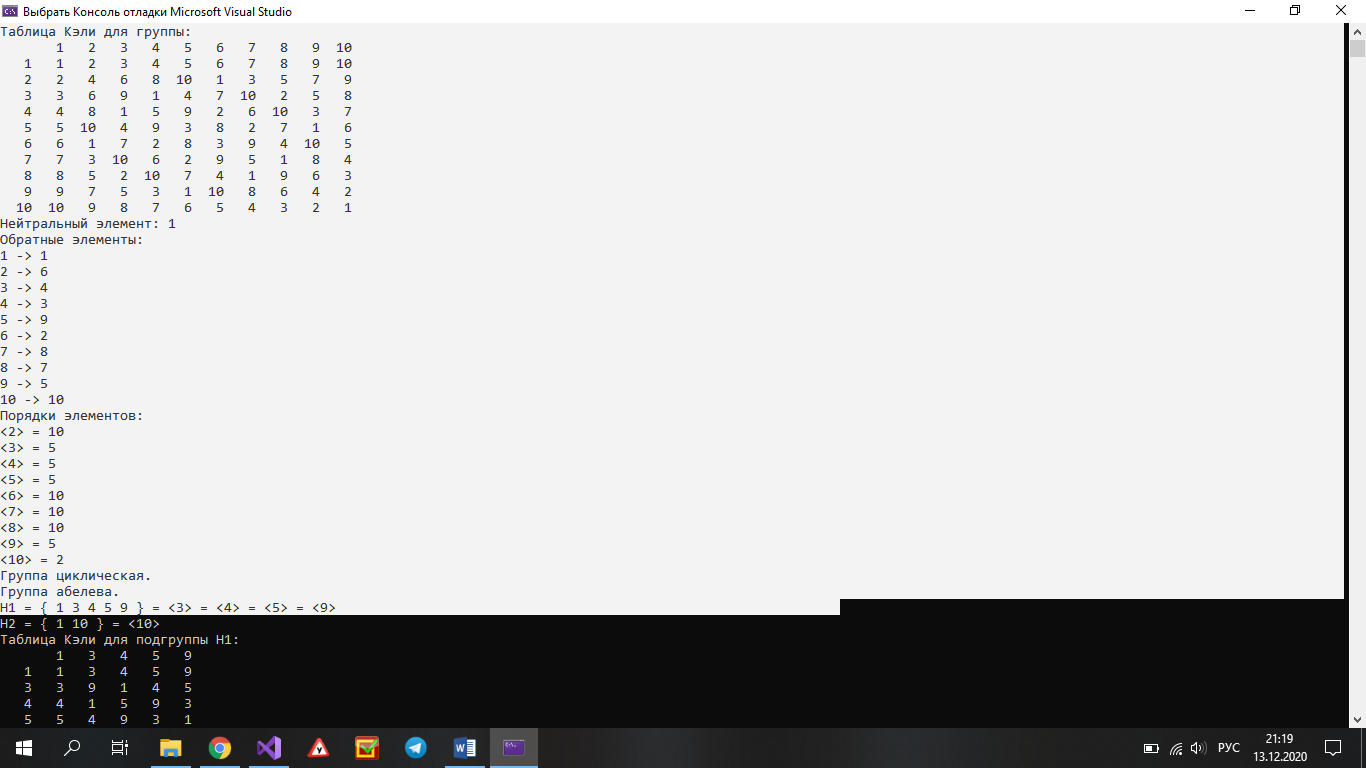
sort(b.begin(), b.end());

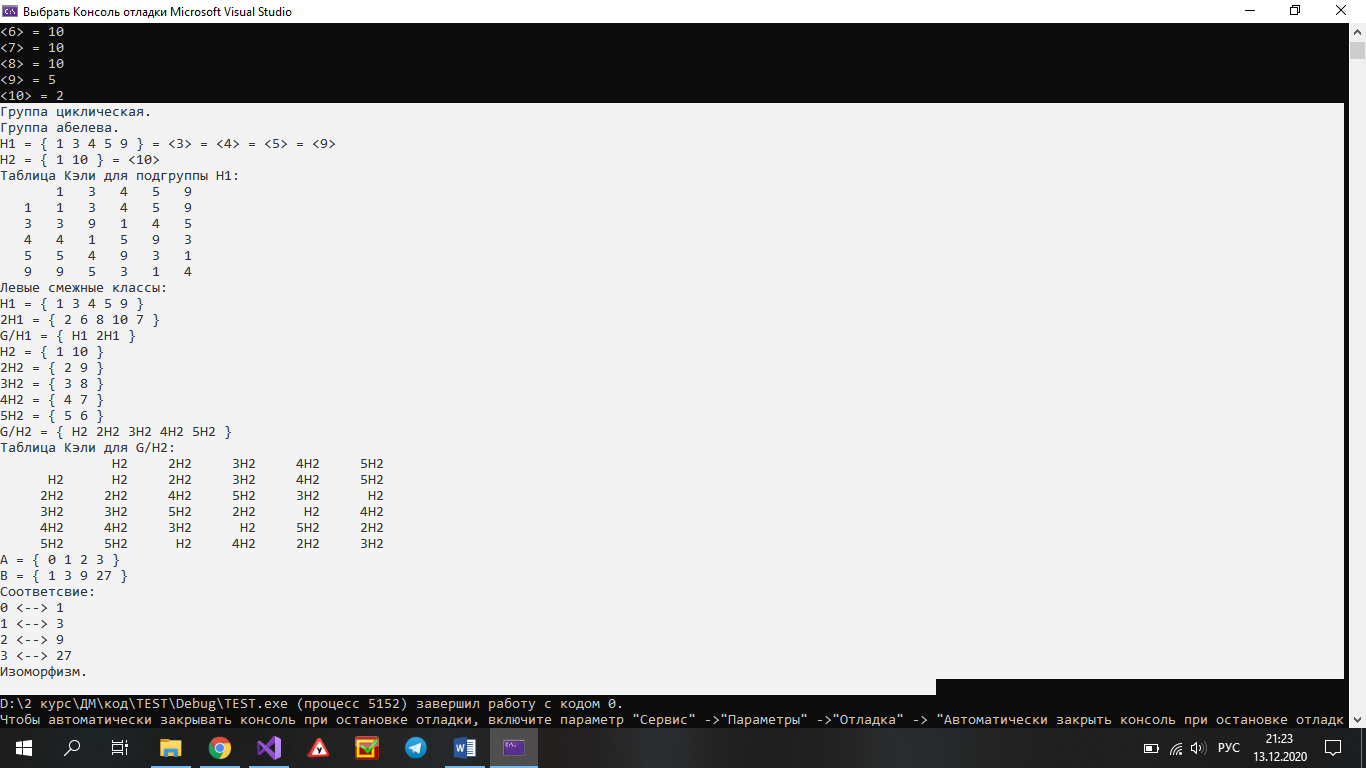
return (a == b);

}

Результаты работы программы:





Задание 2.

1. Представить заданную подстановку произведением независимых циклов. Варианты заданий указаны в таблице 2. Подстановка из 6 элементов задана в таблице 2 второй строкой подстановки. Например: подстановка записана в таблице как .

2. Определить порядок p заданной подстановки.

3. Составить таблицу Кэли для циклической группы, порожденной вашей постановкой. Группа будет иметь вид: . Для вашей подстановки найти обратную подстановку в полученной группе.

4. Определить четность подстановки: – через число транспозиций в разложении подстановки в произведение транспозиций; – через число инверсий в подстановке.

5. Определить тип подстановки.

6. Определить является ли ваша группа абелевой.

