МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №2

По дисциплине «Дискретная математика»

За 3 семестр

Тема: «Отношения. Функции»

Выполнила:

студентка 2 курса

группы АС-56

Карпенко М.В.

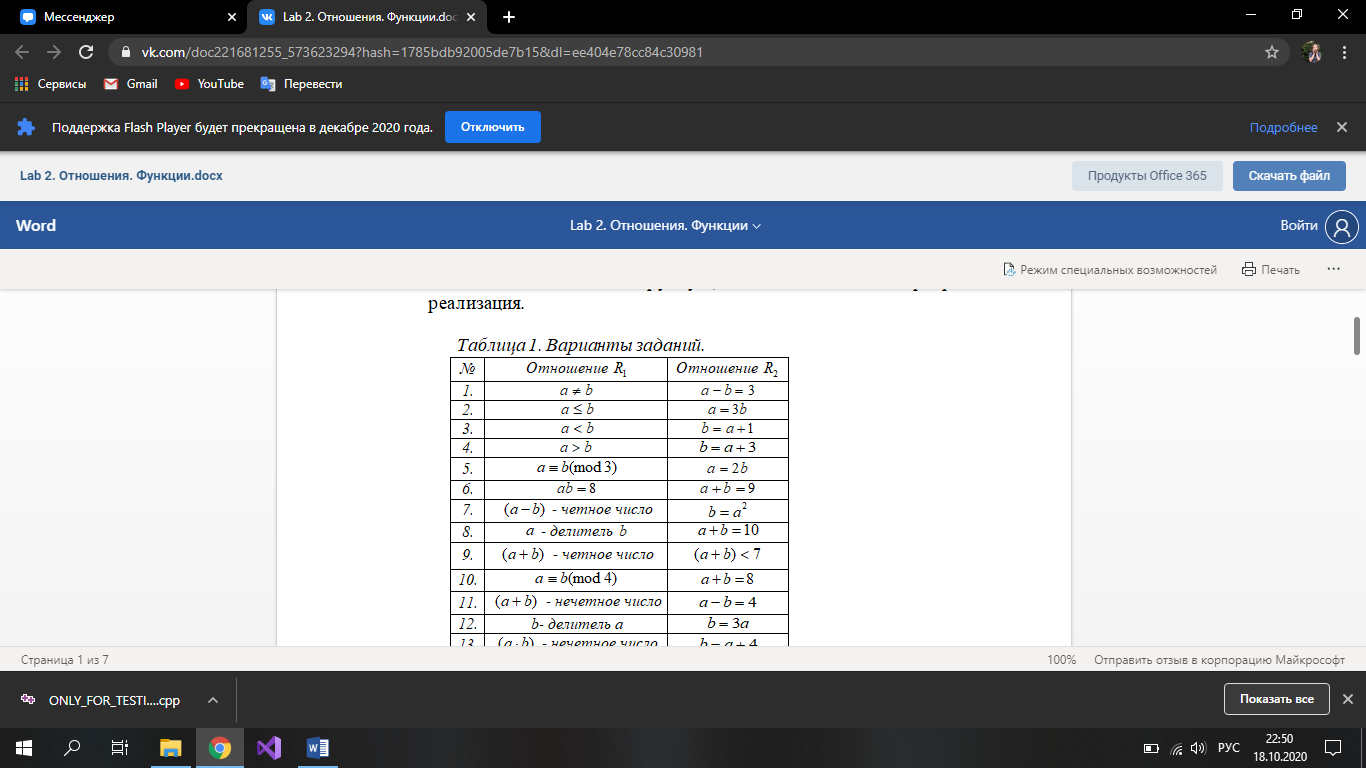
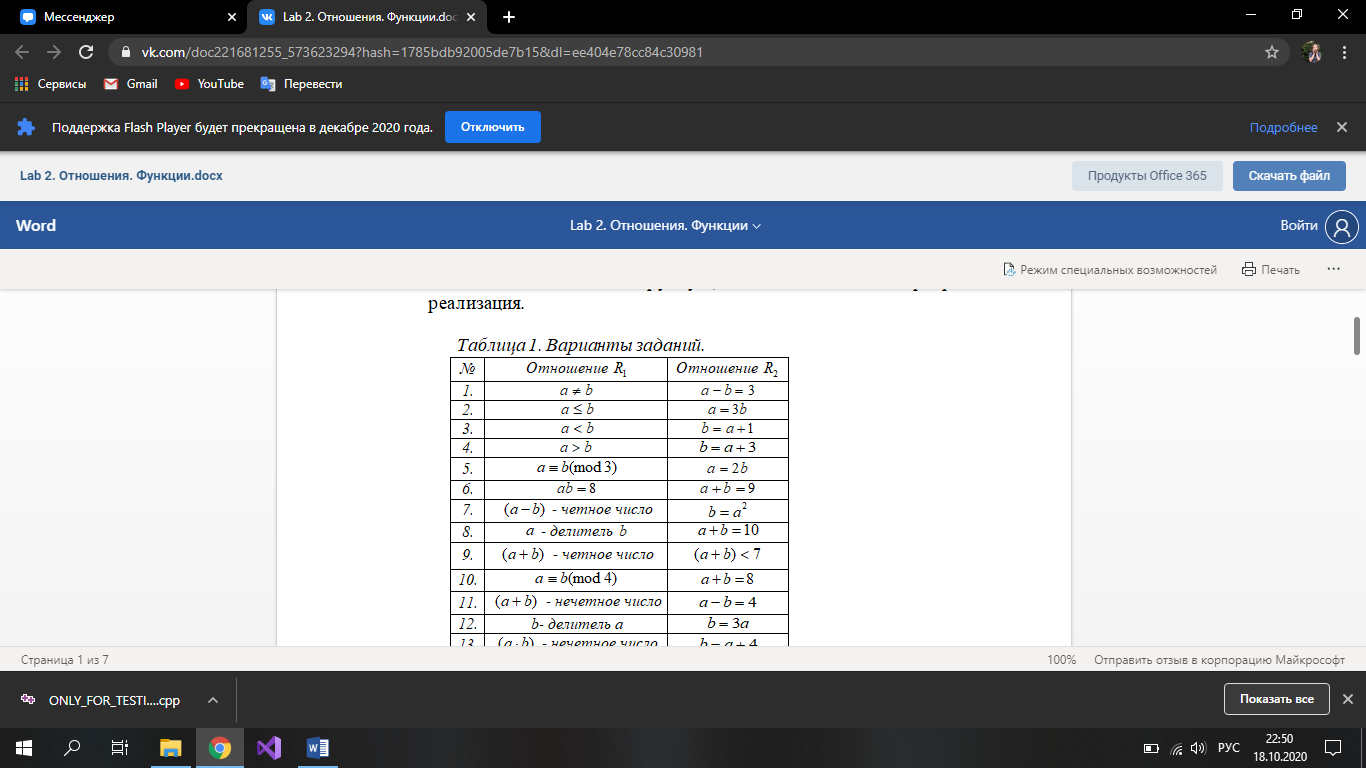
Проверил:

Глущенко Т.А.

Брест 2020

*Отношения. Задание 1 (вариант 7)*

На множестве A = {1,2,3,4,5,6,7,8,9,10} заданы отношения и .  
1. Для заданных отношений составить матрицы отношений. Построить орграфы отношений.  
2. Найти обратные отношения и дополнения отношений.  
3. Указать свойства отношений.  
4. Для отношения, не обладающего свойством транзитивности, построить транзитивное замыкание алгоритмом Флойда-Уоршолла.  
5. Найти композицию отношений, указать обладает ли операция композиции отношений свойством коммутативности.

*Код программы:*

#include <iostream>

using namespace std;

static int i = 1;

void matr(int arr[10][10], string arrname);

void transpon(int arr[10][10], string arrname);

void composition(int R1[10][10], int R2[10][10]);

void oshelfloid(int R1[10][10]);

int main()

{

setlocale(0, " ");

int A[10] = { 1,2,3,4,5,6,7,8,9,10 };

int R1[10][10];

int R2[10][10];

int r1[10][10];

int r2[10][10];

//R1=(a-b)-четное число

//R2= (b=a\*a)

for (int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

{

if ((A[j] - A[i]) % 2 == 0)

R1[i][j] = 1;

else R1[i][j] = 0;

cout << R1[i][j] << " ";

}

cout << endl;

}

cout << "\n\n";

for (int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

{

if ((A[i] \* A[i]) == A[j])

R2[i][j] = 1;

else R2[i][j] = 0;

cout << R2[i][j] << " ";

}

cout << endl;

}

cout << "\n\n";

matr(R1, "R1");

matr(R2, "R2");

transpon(R1, "R1");

transpon(R2, "R2");

composition(R1, R2);

composition(R2, R1);

oshelfloid(R2);

}

void oshelfloid(int R1[10][10])

{

int FW[10][10];

for (int k = 0; k < 10; k++) {

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

FW[i][j] = R1[i][j] || (R1[i][k] && R1[k][j]);

}

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

cout << " " << FW[i][j] << " ";

}

cout << endl;

}

}

void matr(int arr[10][10], string arrname)

{

cout << '\t' << arrname << '\n';

for (int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

{

if (arr[i][j])

cout << "0 ";

else

cout << "1 ";

}

cout << "\n";

}

cout << "\n\n";

}

void transpon(int arr[10][10], string arrname)

{

cout << '\t' << arrname << '\n';

for (int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

{

cout << arr[j][i] << " ";

}

cout << "\n";

}

cout << "\n\n";

}

void composition(int R1[10][10], int R2[10][10])

{

cout << "Composition:\n";

cout << "R" << (i == 1 ? '1' : '2');

cout << 'o' << 'R' << (i++ == 1 ? '2' : '1');

cout << endl;

int line[10];

for (int m = 0; m < 10; m++)

{

for (int n = 0; n < 10; n++)

{

for (int j = 0, k = 0; j < 10; j++)

{

if (i == 1)

{

if (R1[m][j] && R2[j][n])

line[k++] = 1;

else

line[k++] = 0;

}

else if (R2[m][j] && R1[j][n]) line[k++] = 1;

else line[k++] = 0;

}

if (line[0] || line[1] || line[2] || line[3] || line[4] || line[5] || line[6] || line[7] || line[8] || line[9]) {

cout << "1 ";

}

else cout << "0 ";

}

cout << endl;

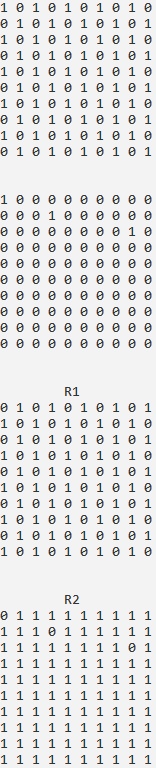
}

cout << endl;

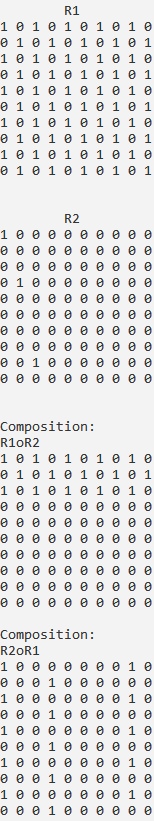
}

*Вывод результата:*

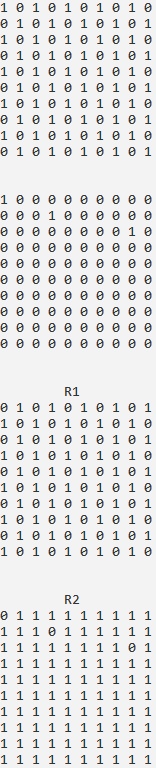
1. Вывод матриц отношений.



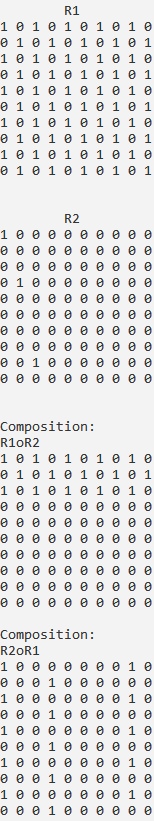
1. Вывод обратных матриц.



1. Вывод дополнительных матриц.

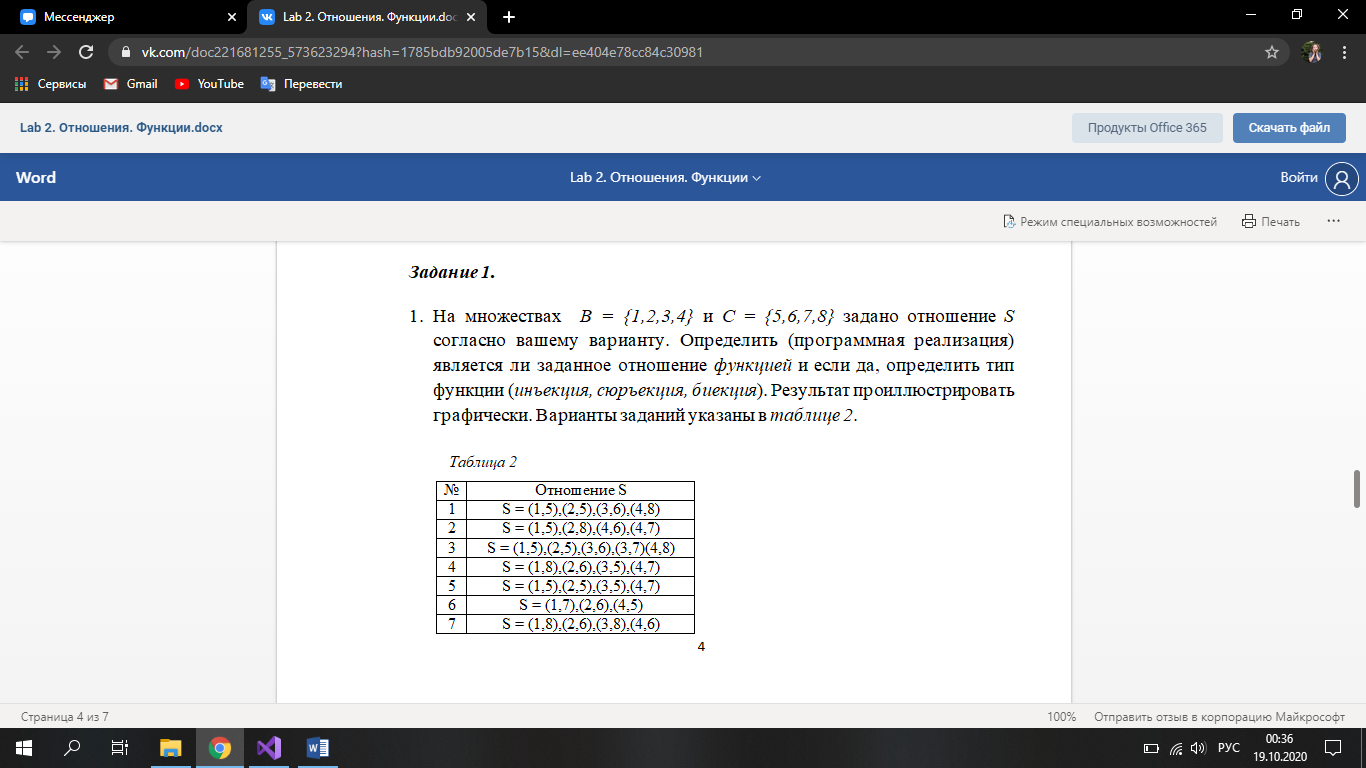


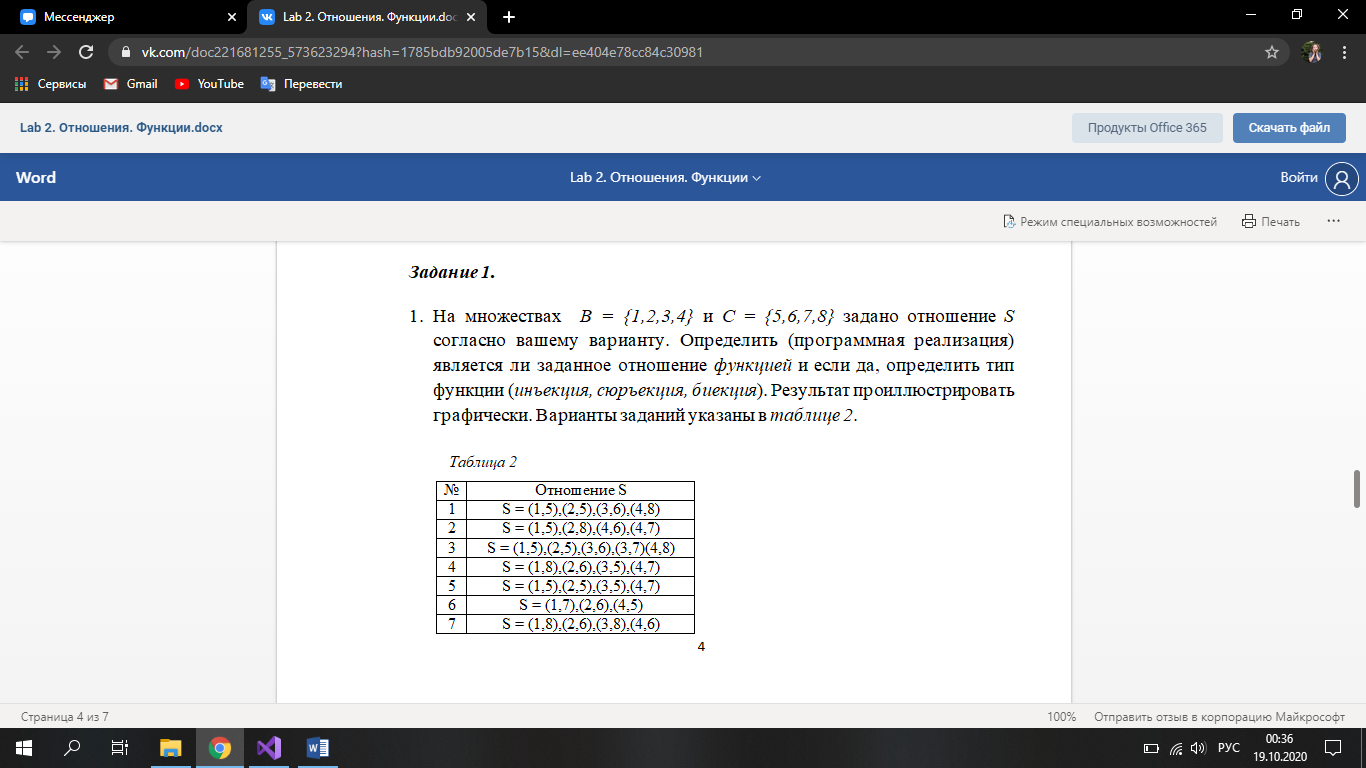
1. Композиция отношений.



*Функции. Задание 1 (вариант 7)*

На множествах B = {1,2,3,4} и C = {5,6,7,8} задано отношение S. Определить является ли заданное отношение функцией и если да, определить тип функции (инъекция, сюръекция, биекция). Результат проиллюстрировать графически.





*Код программы:*

#include <iostream>

using namespace std;

void func(int R[4][4]);

void task(int S[4][2]);

int main()

{

int S[4][2] = { {1,8 }, { 2,6 }, { 3,8 }, { 4,6 } };

task(S);

system("pause");

return 0;

}

void func(int R[4][4])

{

int b = 0, check = 0, inj = 1, surj = 1;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

if (!check) {

if (R[i][j]) {

check++;

b++;

}

}

else if (R[i][j]) {

cout << "It's not function" << endl;

b = 0;

break;

}

}

check = 0;

}

if (b) {

cout << "It's function" << endl;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++)

check += R[i][j];

if (!check) surj = 0;

else inj = 0;

check = 0;

}

if (surj) cout << "It's not surjection" << endl;

else cout << "It's surjection" << endl;

if (inj) cout << "It's injection" << endl;

else cout << "It's not injection" << endl;

if (surj && inj) cout << "It's bijection" << endl;

}

}

void task(int S[4][2])

{

cout << "Task:\n";

int U[] = { 1,2,3,4,5,6,7,8 };

int R[4][4];

for (int k = 0; k < 4; k++)

for (int i = 4; i < 8; i++) {

for (int j = 0; j < 4; j++) {

if ((S[j][0] == U[k] && S[j][1] == U[i]) || R[k][i - 4] == 1)

R[k][i - 4] = 1;

else R[k][i - 4] = 0;

}

}

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) cout << R[i][j] << ' ';

cout << endl;

}

func(R);

}

*Вывод результата:*

