Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Пенза 2021

**ОТЧЁТ**

по лабораторной работе №3

по курсу «Логика и основы алгоритмизации в инженерных задачах»

## на тему «Унарные и бинарные операции над графами»

Выполнили:

студентки группы 20ВВ4

Кожевникова А. В.

Мамелина Ю. В.

Приняла:

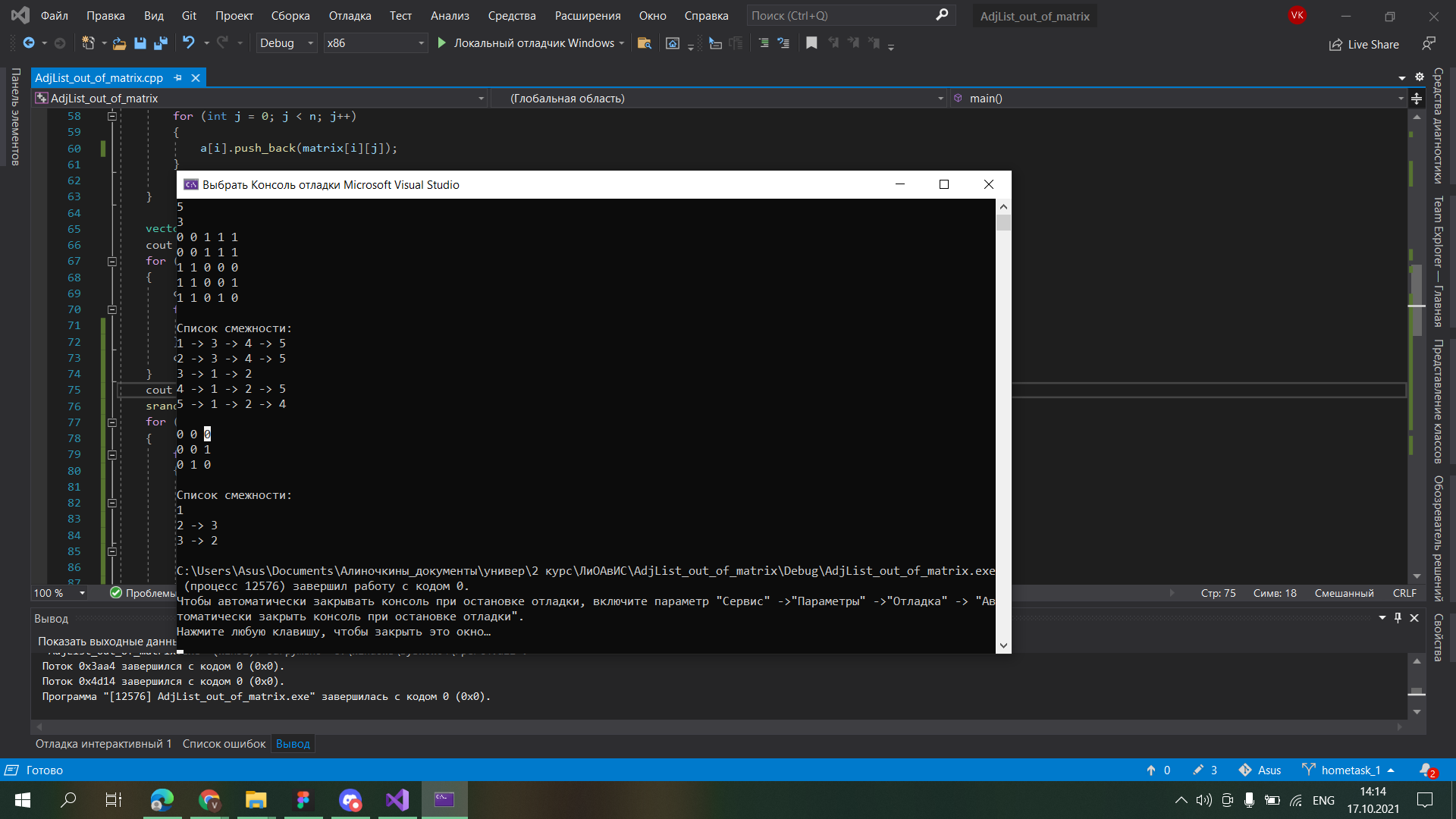
д.т.н Юрова О. В.

Пенза 2021

## Цель работы: изучить и научиться применять унарные и бинарные операции над графами.

**Задание 1:**

1. Сгенерировали (используя генератор случайных чисел) две матрицы *M*1*, М*2 смежности неориентированных помеченных графов *G*1, *G*2. Вывели сгенерированные матрицы на экран.
2. Для указанных графов преобразовали представление матриц смежности в списки смежности. Вывели полученные списки на экран.



Код программы – приложение 1.

### **Задание 2:**

1. Для матричной формы представления графов выполнили операции:

а) отождествления вершин;

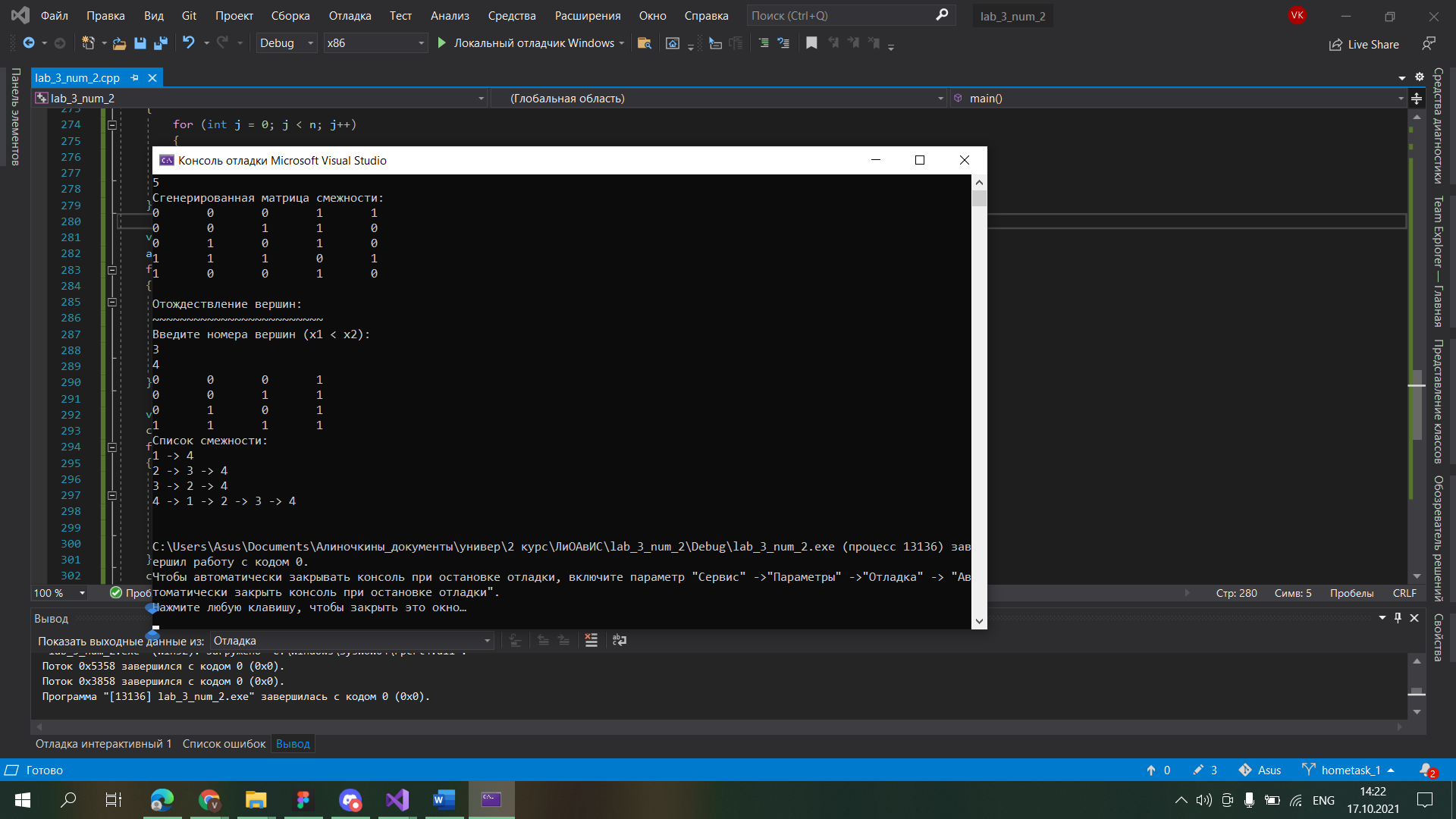
б) стягивание вершины;

в) расщепления вершины;

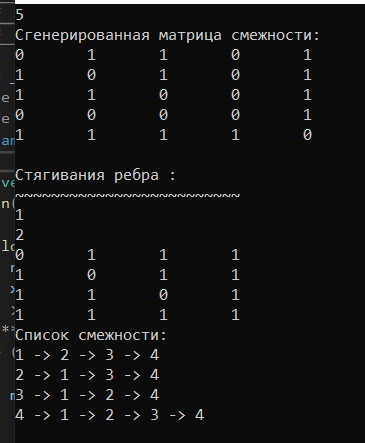
Номера выбираемых для выполнения операции вершин ввели с клавиатуры. Результат выполнения операции вывели на экран.

1. Для представления графов в виде списков смежности выполнили операцию:

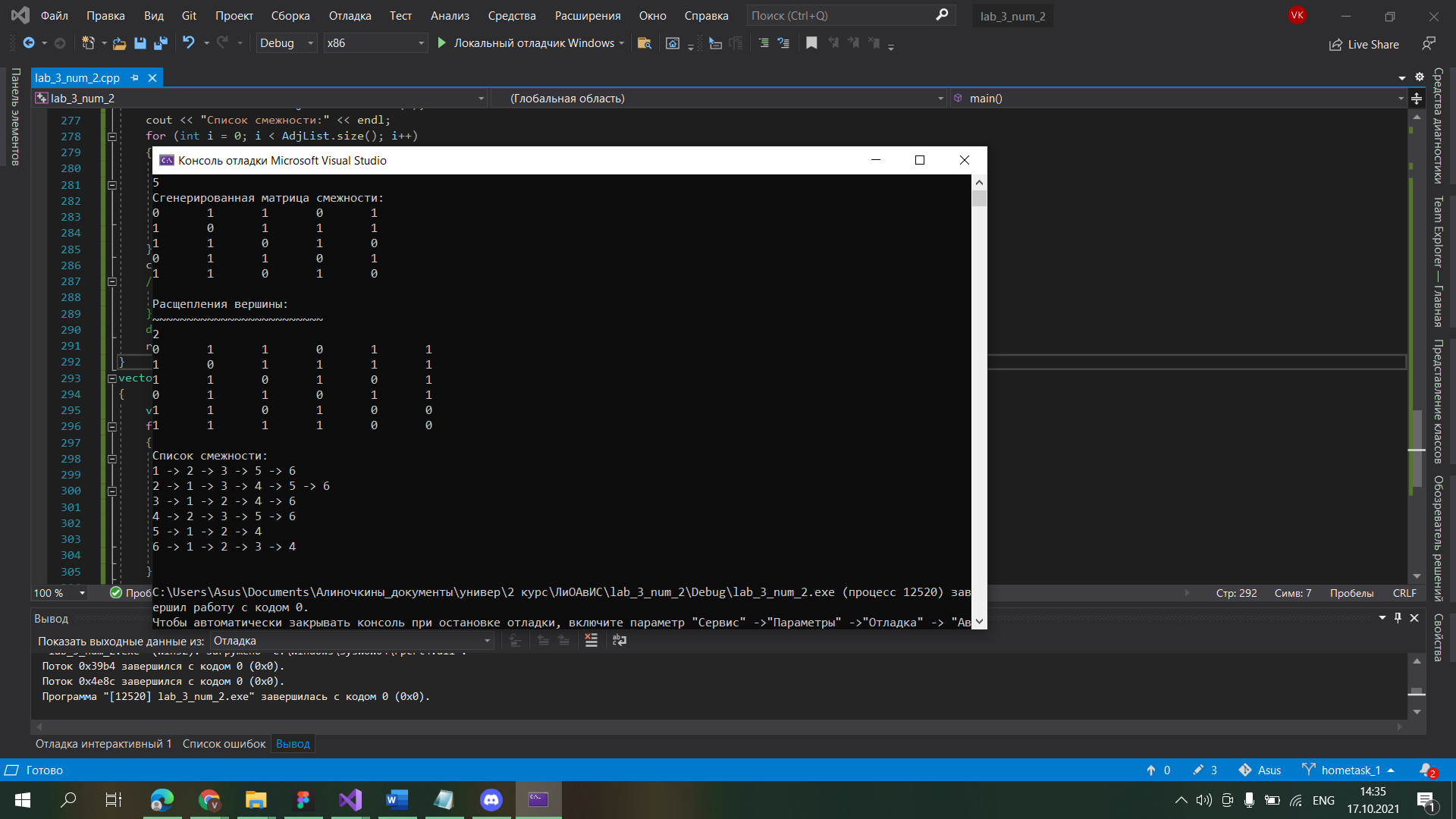
а) отождествления вершин;



б) стягивания ребра;



в) расщепления вершины

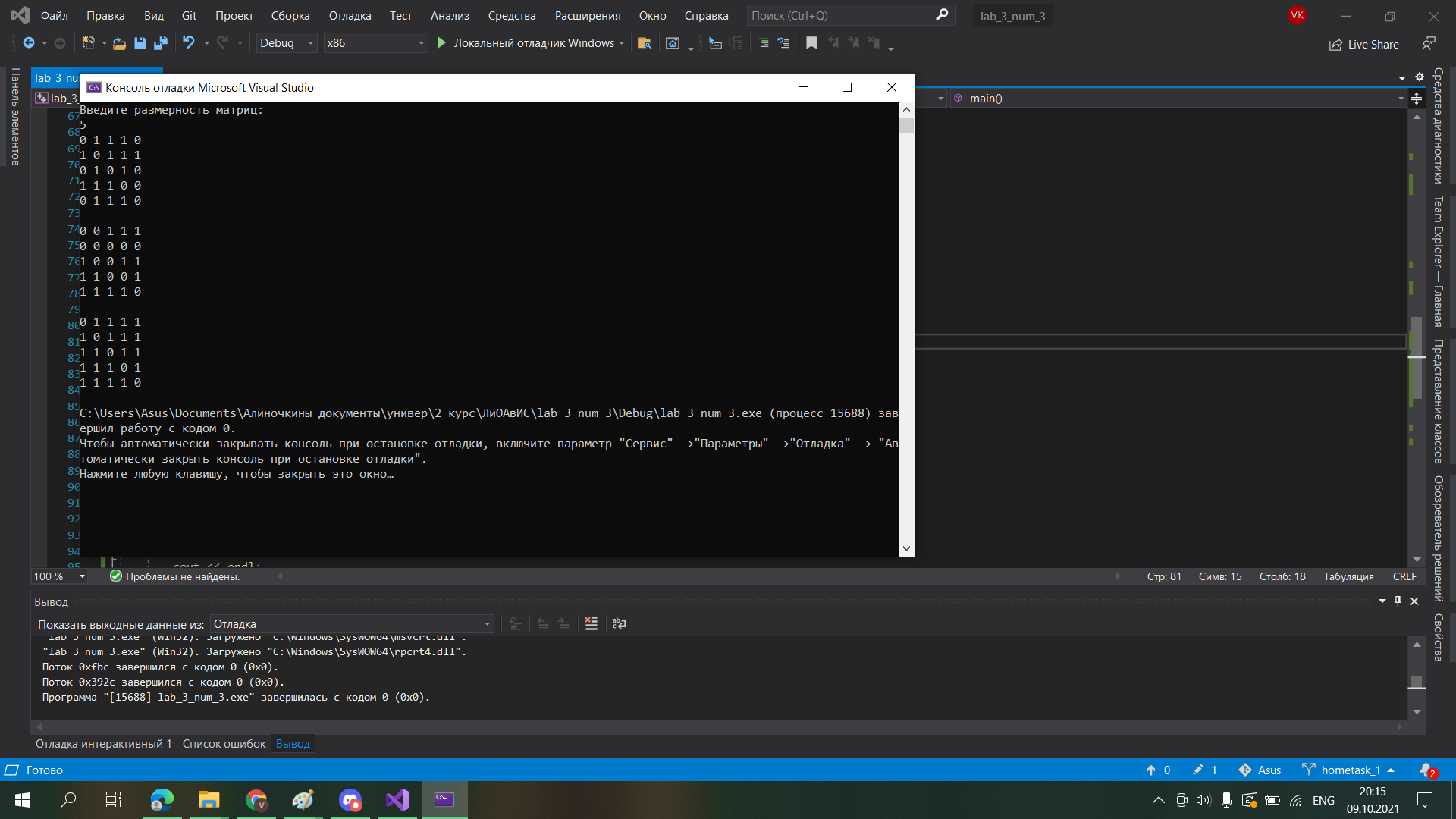


Результат выполнения операции вывели на экран, списки смежности включены в приложения 2, 3, 4.

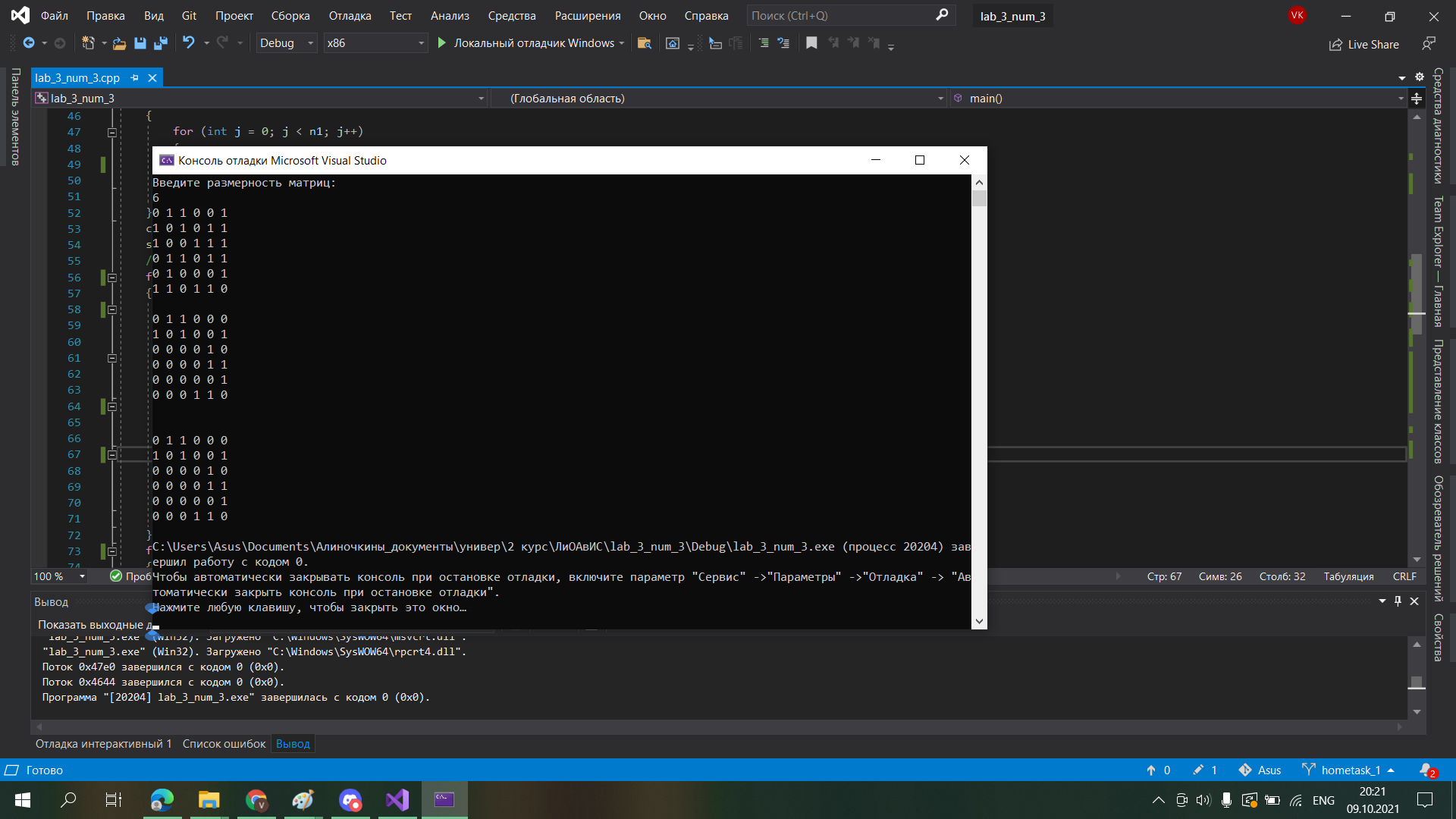
**Задание 3**

1. Для матричной формы представления графов выполнили операцию:

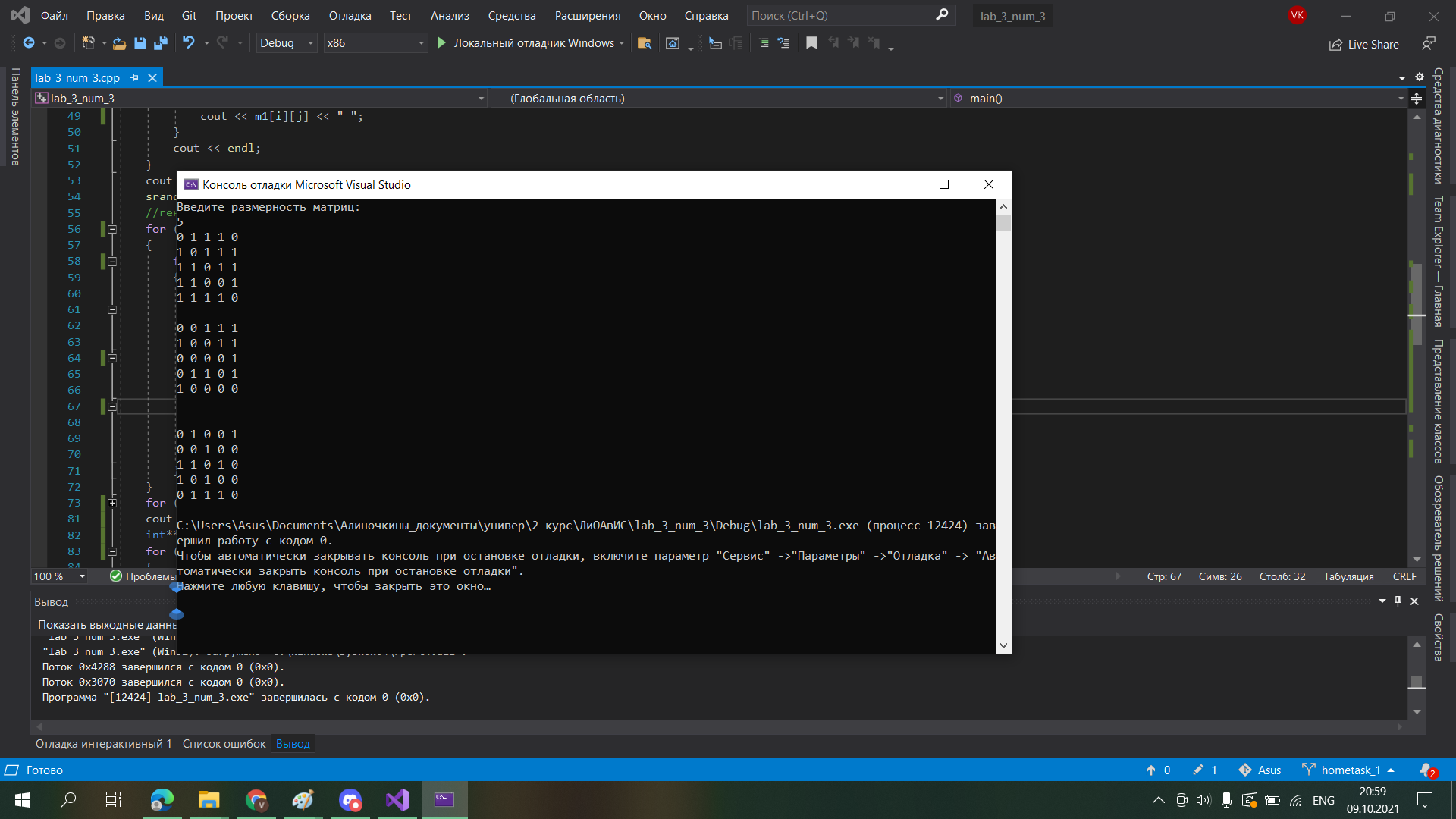
а) объединения *G* = *G*1  *G*2



б) пересечения *G* = *G*1  *G*2



в) кольцевой суммы *G* = *G*1  *G*2



Результат выполнения операции вывели на экран.

Код программы – приложение 5, 6, 7.

**Задание 4**

1. Для матричной формы представления графов выполнили операцию декартова произведения графов *G = G*1X *G*2.

Результат выполнения операции вывели на экран.

## Вывод: изучили и научились применять унарные и бинарные операции над неориентированными графами.

**Приложения**

Приложение 1:

#include <iostream>

#include <vector>

using namespace std;

// конвертируем матрицу смежности в список смежности

vector<vector<int>> convert(vector<vector<int>> a);

int main()

{

setlocale(LC\_ALL, "rus");

int n;

cin >> n;

int\*\* matrix = new int\* [n];

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

{

matrix[i] = new int[n];

}

int n1;

cin >> n1;

int\*\* matrix1 = new int\* [n1];

for (int i = 0; i < n1; ++i)

{

matrix1[i] = new int[n1];

}

srand(time(NULL));

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

{

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

{

matrix[i][j] = rand() % 100;

if (i == j) {

matrix[i][j] = 0;

}

if (matrix[i][j] >= 0 && matrix[i][j] <= 30) {

matrix[i][j] = 0;

}

else if (matrix[i][j] >= 31 && matrix[i][j] < 100)

{

matrix[i][j] = 1;

}

}

}

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

{

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

{

matrix[i][j] = matrix[j][i];

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

}

cout << endl;

}

cout << endl;

vector<vector<int>> a;

a.resize(n);

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

{

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

{

a[i].push\_back(matrix[i][j]);

}

}

vector<vector<int>> AdjList = convert(a);

cout << "Список смежности:" << endl;

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

{

cout << i + 1;

for (int j = 0; j < AdjList[i].size(); j++) {

cout << " -> " << AdjList[i][j];

}

cout << endl;

}

cout << endl;

srand(time(NULL));

for (int i = 0; i < n1; i++)

{

for (int j = 0; j < n1; j++)

{

matrix1[i][j] = rand() % 100;

if (i == j) {

matrix1[i][j] = 0;

}

if (matrix1[i][j] >= 0 && matrix1[i][j] <= 30) {

matrix1[i][j] = 0;

}

else if (matrix1[i][j] >= 31 && matrix1[i][j] < 100)

{

matrix1[i][j] = 1;

}

}

}

for (int i = 0; i < n1; i++)

{

for (int j = 0; j < n1; j++)

{

matrix1[i][j] = matrix1[j][i];

cout << matrix1[i][j] << " ";

}

cout << endl;

}

cout << endl;

vector<vector<int>> a1;

a1.resize(n1);

for (int i = 0; i < n1; i++)

{

for (int j = 0; j < n1; j++)

{

a1[i].push\_back(matrix1[i][j]);

}

}

vector<vector<int>> AdjList1 = convert(a1);

cout << "Список смежности:" << endl;

for (int i = 0; i < AdjList1.size(); i++)

{

cout << i + 1;

for (int j = 0; j < AdjList1[i].size(); j++) {

cout << " -> " << AdjList1[i][j];

}

cout << endl;

}

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

{

delete[] matrix[i];

}

delete[] matrix;

for (int i = 0; i < n1; ++i)

{

delete[] matrix1[i];

}

delete[] matrix1;

return 0;

}

vector<vector<int>> convert(vector<vector<int>> a)

{

vector<vector<int>> adjList(a.size());

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

{

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

{

if (a[i][j] == 1)

{

adjList[i].push\_back(j + 1);

}

}

}

return adjList;

}

Приложение 2:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <vector>

using namespace std;

// конвертируем матрицу смежности в список смежности

vector<vector<int>> convert(vector<vector<int>> a);

int main()

{

setlocale(LC\_ALL, "rus");

int n;

int x1 = 0, x2 = 0;

cin >> n;

int\*\* matrix = new int\* [n];

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

{

matrix[i] = new int[n];

}

std::cout << "Сгенерированная матрица смежности:" << endl;

srand(time(NULL));

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

{

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

{

matrix[i][j] = rand() % 100;

if (i == j)

{

matrix[i][j] = 0;

}

if (0 <= matrix[i][j] && matrix[i][j] <= 30) {

matrix[i][j] = 0;

}

else if (matrix[i][j] >= 31 && matrix[i][j] <= 100)

{

matrix[i][j] = 1;

}

}

}

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

{

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

{

matrix[i][j] = matrix[j][i];

std::cout << matrix[i][j] << "\t";

}

std::cout << endl;

}

cout << endl;

cout << "Отождествление вершин: " << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "Введите номера вершин (x1 < x2): " << endl;

cin >> x1 >> x2;

for (x1; x1 < n; x1++)

{

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

{

matrix[x1][j] = matrix[x1][j] | matrix[x2][j];

}

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

{

matrix[i][x1] = matrix[i][x1] | matrix[i][x2];

}

}

//удаление вершины

n--;

for (x2; x2 < n; x2++) {

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

matrix[i][x2] = matrix[i][x2 + 1];

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

matrix[x2][j] = matrix[x2 + 1][j];

}

}

}

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

{

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

{

cout << matrix[i][j] << "\t";

}

cout << endl;

}

vector<vector<int>> a;

a.resize(n);

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

{

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

{

a[i].push\_back(matrix[i][j]);

}

}

vector<vector<int>> AdjList = convert(a);

cout << "Список смежности:" << endl;

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

{

cout << i + 1;

for (int j = 0; j < AdjList[i].size(); j++) {

cout << " -> " << AdjList[i][j];

}

cout << endl;

}

cout << endl;

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

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

vector<vector<int>> convert(vector<vector<int>> a)

{

vector<vector<int>> adjList(a.size());

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

{

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

{

if (a[i][j] == 1)

{

adjList[i].push\_back(j + 1);

}

}

}

return adjList;

}

Приложение 3:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <vector>

using namespace std;

vector<vector<int>> convert(vector<vector<int>> a);

int main()

{

setlocale(LC\_ALL, "rus");

int n;

int x1 = 0, x2 = 0;

cin >> n;

int\*\* matrix = new int\* [n];

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

{

matrix[i] = new int[n];

}

std::cout << "Сгенерированная матрица смежности:" << endl;

srand(time(NULL));

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

{

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

{

matrix[i][j] = rand() % 100;

if (i == j)

{

matrix[i][j] = 0;

}

if (0 <= matrix[i][j] && matrix[i][j] <= 30) {

matrix[i][j] = 0;

}

else if (matrix[i][j] >= 31 && matrix[i][j] <= 100)

{

matrix[i][j] = 1;

}

}

}

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

{

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

{

matrix[i][j] = matrix[j][i];

std::cout << matrix[i][j] << "\t";

}

std::cout << endl;

}

std::cout << endl;

std::cout << "Стягивания ребра : " << endl;

std::cout << "~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cin >> x1 >> x2;

if (matrix[x1][x2] == 1 || matrix[x2][x1] == 1)

{

for (x1; x1 < n; x1++)

{

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

{

matrix[x1][j] = matrix[x1][j] | matrix[x2][j];

}

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

{

matrix[i][x1] = matrix[i][x1] | matrix[i][x2];

}

matrix[x1][x1] = 0;

}

n--;

for (x2; x2 < n; x2++) {

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

matrix[i][x2] = matrix[i][x2 + 1];

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

matrix[x2][j] = matrix[x2 + 1][j];

}

}

}

}

else {

cout << "Вершины не смежны!" << endl;

}

//вывод

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

{

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

{

std::cout << matrix[i][j] << "\t";

}

std::cout << endl;

}

vector<vector<int>> a;

a.resize(n);

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

{

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

{

a[i].push\_back(matrix[i][j]);

}

}

vector<vector<int>> AdjList = convert(a);

cout << "Список смежности:" << endl;

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

{

cout << i + 1;

for (int j = 0; j < AdjList[i].size(); j++) {

cout << " -> " << AdjList[i][j];

}

cout << endl;

}

cout << endl;

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

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

vector<vector<int>> convert(vector<vector<int>> a)

{

vector<vector<int>> adjList(a.size());

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

{

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

{

if (a[i][j] == 1)

{

adjList[i].push\_back(j + 1);

}

}

}

return adjList;

}

Приложение 4:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <vector>

using namespace std;

vector<vector<int>> convert(vector<vector<int>> a);

int main()

{

setlocale(LC\_ALL, "rus");

int n;

int x1 = 0, x2 = 0;

cin >> n;

int\*\* matrix = new int\* [n + 1];

for (int i = 0; i < n + 1; ++i)

{

matrix[i] = new int[n + 1];

}

std::cout << "Сгенерированная матрица смежности:" << endl;

srand(time(NULL));

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

{

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

{

matrix[i][j] = rand() % 100;

if (i == j)

{

matrix[i][j] = 0;

}

if (0 <= matrix[i][j] && matrix[i][j] <= 30) {

matrix[i][j] = 0;

}

else if (matrix[i][j] >= 31 && matrix[i][j] <= 100)

{

matrix[i][j] = 1;

}

}

}

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

{

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

{

matrix[i][j] = matrix[j][i];

std::cout << matrix[i][j] << "\t";

}

std::cout << endl;

}

std::cout << endl;

std::cout << "Расщепления вершины: " << endl;

std::cout << "~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cin >> x1;

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

matrix[i][n] = matrix[i][x1];

matrix[n][i] = matrix[x1][i];

matrix[n][n] = 0;

matrix[x1][n] = 1;

matrix[n][x1] = 1;

}

n++;

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

{

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

{

cout << matrix[i][j] << "\t";

}

cout << endl;

}

std::cout << endl;

vector<vector<int>> a;

a.resize(n);

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

{

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

{

a[i].push\_back(matrix[i][j]);

}

}

vector<vector<int>> AdjList = convert(a);

cout << "Список смежности:" << endl;

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

{

cout << i + 1;

for (int j = 0; j < AdjList[i].size(); j++) {

cout << " -> " << AdjList[i][j];

}

cout << endl;

}

cout << endl;

for (int i = 0; i < n + 1; ++i) {

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

vector<vector<int>> convert(vector<vector<int>> a)

{

vector<vector<int>> adjList(a.size());

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

{

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

{

if (a[i][j] == 1)

{

adjList[i].push\_back(j + 1);

}

}

}

return adjList;

}

Приложение 5, 6, 7.

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

int n1;

cout << "Введите размерность матриц:" << endl;

cin >> n1;

//выделение памяти под матрицы

int\*\* m1 = new int\* [n1];

for (int i = 0; i < n1; ++i){

m1[i] = new int[n1];

}

int\*\* m2 = new int\* [n1];

for (int i = 0; i < n1; ++i)

{

m2[i] = new int[n1];

}

//генерация неориентированного графа

srand(time(NULL));

for (int i = 0; i < n1; i++)

{

for (int j = 0; j < n1; j++)

{

m1[i][j] = rand() % 100;

if (i == j) {

m1[i][j] = 0;

}

if (m1[i][j] >= 0 && m1[i][j] <= 30) {

m1[i][j] = 0;

}

else if (m1[i][j] >= 31 && m1[i][j] < 100)

{

m1[i][j] = 1;

}

}

}

for (int i = 0; i < n1; i++){

for (int j = 0; j < n1; j++)

{

cout << m1[i][j] << " ";

}

cout << endl;

}

cout << endl;

srand(time(NULL));

//генерация неориентированного графа

for (int i = 0; i < n1; i++)

{

for (int j = 0; j < n1; j++)

{

m2[i][j] = rand() % 100;

if (i == j) {

m2[i][j] = 0;

}

if (m2[i][j] >= 0 && m2[i][j] <= 30) {

m2[i][j] = 0;

}

else if (m2[i][j] >= 31 && m2[i][j] < 100)

{

m2[i][j] = 1;

}

}

}

for (int i = 0; i < n1; i++){

for (int j = 0; j < n1; j++)

{

cout << m2[i][j] << " ";

}

cout << endl;

}

cout << endl;

int\*\* resM = new int\* [n1];

for (int i = 0; i < n1; ++i)

{

resM[i] = new int[n1];

}

//объединение матриц:

cout << "Объединение матриц:" << endl;

for (int i = 0; i < n1; i++)

{

for (int j = 0; j < n1; j++)

{

resM[i][j] = m1[i][j] | m2[i][j];

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

}

cout << endl;

}

cout << endl;

//пересечение матриц:

cout << "Пересечение матриц:" << endl;

for (int i = 0; i < n1; i++)

{

for (int j = 0; j < n1; j++)

{

resM[i][j] = m1[i][j] & m2[i][j];

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

}

cout << endl;

}

//кольцевая сумма:

cout << "Кольцевая сумма:" << endl;

for (int i = 0; i < n1; i++)

{

for (int j = 0; j < n1; j++)

{

resM[i][j] = 0;

if (m1[i][j] != m2[i][j])

{

resM[i][j] = m1[i][j] | m2[i][j];

}

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

}

cout << endl;

}

//освобождение памяти

for (int i = 0; i < n1; ++i)

{

delete[] m1[i];

}

delete[] m1;

for (int i = 0; i < n1; ++i)

{

delete[] m2[i];

}

delete[] m2;

for (int i = 0; i < n1; ++i)

{

delete[] resM[i];

}

delete[] resM;

return 0;

}