МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Вычислительной техники»

**ОТЧЁТ**

по лабораторной работе №6

по курсу «Л и ОА в ИЗ»

на тему «Унарные и бинарные операции над графами»

Выполнили:

Студенты группы 22ВВС1

Лосяков Г.С.

Осмаев М-А.Х.

Приняли:

Акифьев И.В.

Юрова О.В.

Пенза 2023

**Лабораторноезадание*:***

### Задание 1

1. Сгенерируйте (используя генератор случайных чисел) две матрицы *M*1*, М*2 смежности неориентированных помеченных графов *G*1, *G*2. Выведите сгенерированные матрицы на экран.

### Задание 2

1. Для матричной формы представления графов выполните операцию:

а) отождествления вершин

б) стягивания ребра

в) расщепления вершины

Номера выбираемых для выполнения операции вершин ввести с клавиатуры.

Результат выполнения операции выведите на экран.

**Задание 3**

1. Для матричной формы представления графов выполните 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 выведите на экран.

**Листинг:**

Задание 1

#include <iostream>

using namespace std;

int main()

{

const int N = 4;

int M[N][N];

int M1[N][N];

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

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

M[i][j] = 0;

}

}

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

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

M[i][j] = rand() % 2;

}

}

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

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

M[i][j] = M[j][i];

}

}

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

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

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

}

cout << "\n";

}

cout << "\n";

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

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

M1[i][j] = 0;

}

}

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

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

M1[i][j] = rand() % 2;

}

}

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

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

M1[i][j] = M1[j][i];

}

}

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

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

cout << M1[i][j] << "\t";

}

cout << "\n";

}

}

Задание 2

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

const int N = 6;

int M[N][N];

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

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

M[i][j] = 0;

}

}

/\*for (int i = 0; i < N; i++) {

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

M[i][j] = rand() % 2;

}

}\*/

M[0][1] = 1;

M[0][4] = 1;

M[1][0] = 1;

M[1][2] = 1;

M[1][4] = 1;

M[2][1] = 1;

M[2][3] = 1;

M[2][4] = 1;

M[3][2] = 1;

M[3][4] = 1;

M[4][0] = 1;

M[4][1] = 1;

M[4][2] = 1;

M[4][3] = 1;

/\*for (int i = 1; i < N; i++) {

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

M[i][j] = M[j][i];

}

}\*/

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

for (int j = 0; j < N - 1; j++) {

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

}

cout << "\n";

}

cout << "\n";

cout << "Выберите какую оперцию вы хотите выполнить: " << "\n" << "1 - операцию отождествления вершин" << "\n" << "2 - операцию стягивания ребра";

cout << "\n" << "3 - операцию расщепления вершины" << "\n";

int vibor;

cin >> vibor;

if (vibor == 1) {

int v1 = -1, v2 = -1;

while (1) {

cout << "Введите вершины, которые необходимо замкнуть" << "\n";

cin >> v1 >> v2;

if ((v1 < 0) && (v1 >= N) && (v2 < 0) && (v2 >= N) || (v1 == v2) && (M[v1][v2] == 0)) {

cout << "Введены неправильные значения" << "\n";

}

else {

break;

}

}

if (v2 < v1) {

int t;

t = v1;

v1 = v2;

v2 = t;

}

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

M[v1][i] = M[v1][i] + M[v2][i];

if (M[v1][i] > 1) { M[v1][i] = 1; }

}

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

M[i][v1] = M[i][v1] + M[i][v2];

if (M[i][v1] > 1) { M[i][v1] = 1; }

}

for (int i = 0; i < N - 1; i++) { M[v2][i] = 0; }

for (int i = 0; i < N; i++) { M[i][v2] = 0; }

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

for (int j = 0; j < N - 1; j++) {

if ((i != v2) && (j != v2)) {

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

}

}

cout << "\n";

}

}

if (vibor == 2) {

int v1 = -1, v2 = -1;

int r;

cout << "Введите номер ребра, которое необходимо стянуть" << "\n";

cin >> r;

/\*while (1) {

cout << "Введите номер ребра, которое необходимо стянуть" << "\n";

cin >> v1;

if ((v1 < 0) && (v1 >= N) && (v2 < 0) && (v2 >= N) || (v1 == v2) && (M[v1][v2] == 0)) {

cout << "Введены неправильные значения" << "\n";

}

else {

break;

}

}\*/

int schet = 0;

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

for (int j = i + 1; j < N - 1; j++) {

if (M[i][j] == 1) { schet++; }

if (schet == r) {

v1 = i;

v2 = j;

break;

}

}

}

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

M[v1][i] = M[v1][i] + M[v2][i];

if (M[v1][i] > 1) { M[v1][i] = 1; }

}

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

M[i][v1] = M[i][v1] + M[i][v2];

if (M[i][v1] > 1) { M[i][v1] = 1; }

}

for (int i = 0; i < N - 1; i++) { M[v2][i] = 0; }

for (int i = 0; i < N - 1; i++) { M[i][v2] = 0; }

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

for (int j = 0; j < N - 1; j++) {

if ((i != v2) && (j != v2)) {

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

}

}

cout << "\n";

}

}

if (vibor == 3) {

int v1 = -1, v2 = -1;

int r;

cout << "Введите номер вершины, которую необходимо расщепить" << "\n";

cin >> r;

/\*while (1) {

cout << "Введите номер ребра, которое необходимо стянуть" << "\n";

cin >> v1;

if ((v1 < 0) && (v1 >= N) && (v2 < 0) && (v2 >= N) || (v1 == v2) && (M[v1][v2] == 0)) {

cout << "Введены неправильные значения" << "\n";

}

else {

break;

}

}\*/

/\*for (int i = 0; i < N; i++) {

M[v1][i] = M[v1][i] + M[v2][i];

if (M[v1][i] > 1) { M[v1][i] = 1; }

}

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

M[i][v1] = M[i][v1] + M[i][v2];

if (M[i][v1] > 1) { M[i][v1] = 1; }

}

for (int i = 0; i < N; i++) { M[v2][i] = 0; }

for (int i = 0; i < N; i++) { M[i][v2] = 0; }

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

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

if ((i != v2) && (j != v2)) {

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

}

}

cout << "\n";

}\*/

M[N - 1][r] = 1;

M[r][N - 1] = 1;

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

M[N - 1][i] = M[r][i];

}

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

M[i][N - 1] = M[i][r];

}

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

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

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

}

cout << "\n";

}

}

}

Задание 3

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

const int N = 4;

int M[N][N];

int M1[N][N];

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

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

M[i][j] = 0;

}

}

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

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

M[i][j] = rand() % 2;

}

}

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

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

M[i][j] = M[j][i];

}

}

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

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

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

}

cout << "\n";

}

cout << "\n";

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

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

M1[i][j] = 0;

}

}

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

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

M1[i][j] = rand() % 2;

}

}

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

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

M1[i][j] = M1[j][i];

}

}

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

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

cout << M1[i][j] << "\t";

}

cout << "\n";

}

cout << "\n";

cout << "Выберите какую оперцию вы хотите выполнить: " << "\n" << "1 - Операцию объединения матриц" << "\n" << "2 - операцию пересечения матриц";

cout << "\n" << "3 - операцию кольцевой суммы" << "\n";

int vibor;

cin >> vibor;

if (vibor == 1) {

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

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

if (M[i][j] + M1[i][j] != 2) {

M[i][j] = M[i][j] + M1[i][j];

}

}

}

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

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

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

}

cout << "\n";

}

}

if (vibor == 2) {

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

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

if ((M[i][j] == 1) && (M1[i][j] == 1)) {

M[i][j] = 1;

}

else {

M[i][j] = 0;

}

}

}

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

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

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

}

cout << "\n";

}

}

if (vibor == 3) {

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

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

if (M[i][j] != M1[i][j]) {

M[i][j] = 1;

}

else {

M[i][j] = 0;

}

}

}

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

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

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

}

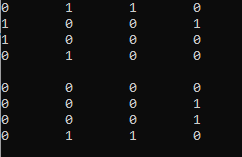
cout << "\n";

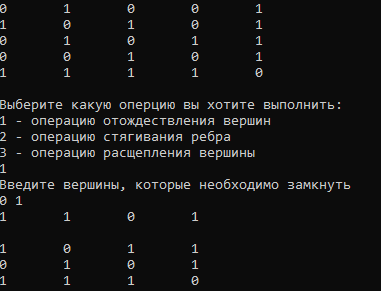
}

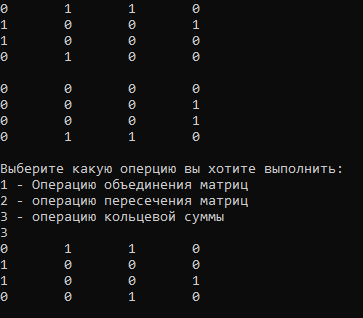
}

}

**Результат работы программы**

****

****

****

**Вывод**

В ходе выполнения лабораторной работы научились выполнять унарные и бинарные операции над неориентированными графами.