|  |
| --- |
| Министерство образования Российской Федерации  Пензенский государственный университет  Кафедра «Вычислительная техника» |
| Отчет  по лабораторной работе №3  по курсу «Л и ОА в ИЗ»  на тему «Унарные и бинарные операции над графами» |
|  |
|  |
| Выполнил:  студент группы 20ВВ4  Кривцов Н.А.  Слабинский Н.А.  Приняли:  Юрова О.В.  Акифьев И.В. |
| Пенза 2021 |

# Цель работы: выполнить над графами унарные и бинарные операции.

**Лабораторное задание:**

### **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:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "locale"

#include "stdio.h"

#include "conio.h"

#include <locale.h>

#include <stdlib.h>

#include <malloc.h>

void main()

{

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

int r1;

int r2;

int\*\* M1;

int\*\* M2;

int\*\* N3;

int\*\* N2;

int\*\* N;

int\*\* E;

printf("Введите размерность матрицы М1\n");

scanf("%d", &r1);

printf("Введите размерность матрицы М2\n");

scanf("%d", &r2);

int g = 0;

int l = 0, k = 0;

int vr = 0;

int q1, q2;

int w = r2 + 1;

M1 = (int\*\*)malloc(r1 \* sizeof(int\*));

M2 = (int\*\*)malloc(r2 \* sizeof(int\*));

N = (int\*\*)malloc(r2 \* sizeof(int\*));

N2 = (int\*\*)malloc(r2 \* sizeof(int\*));

N3 = (int\*\*)malloc(w \* sizeof(int\*));

E = (int\*\*)malloc(50 \* sizeof(int\*));

for (int i = 0; i < 50; i++)

{

E[i] = (int\*)malloc(50 \* sizeof(int));

}

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

{

N3[i] = (int\*)malloc(w \* sizeof(int));

}

for (int i = 0; i < r2; i++)

{

N[i] = (int\*)malloc(r2 \* sizeof(int));

}

for (int i = 0; i < r2; i++)

{

N2[i] = (int\*)malloc(r2 \* sizeof(int));

}

//----------------------------------------------------------------------

printf("Зад 1\n");

for (int i = 0; i < r1; i++) {

M1[i] = (int\*)malloc(r1 \* sizeof(int));

for (int j = 0; j < r1; j++) {

if (j > g) {

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

printf("%3d", M1[i][j]);

}

else if (j < g) {

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

printf("%3d", M1[i][j]);

}

else {

M1[i][j] = 0;

printf("%3d", M1[i][j]);

}

}

g++;

printf("\n");

}

g = 0;

printf("\n");

printf("\n");

for (int i = 0; i < r2; i++) {

M2[i] = (int\*)malloc(r2 \* sizeof(int));

for (int j = 0; j < r2; j++) {

if (j > g) {

M2[i][j] = rand() % 2;

printf("%3d", M2[i][j]);

}

else if (j < g) {

M2[i][j] = M2[j][i];

printf("%3d", M2[i][j]);

}

else {

M2[i][j] = 0;

printf("%3d", M2[i][j]);

}

}

g++;

printf("\n");

}

printf("Зад 2\n");

printf("Отождествление вершин\n");

printf("\nВершинки\n");

scanf("%d%d", &q1, &q2);

for (int j = 0; j < r2; j++) {

for (int i = 0; i < r2; i++)

{

N[i][j] = M2[i][j];

vr = M2[i][q1];

if ((N[i][q2] || vr) == 1) {

N[i][q2] = 1;

}

else

{

N[i][q2] = 0;

};

N[i][q1] = 0;

}

}

for (int i = 0; i < r2; i++) {

for (int j = 0; j < r2; j++)

{

vr = M2[q1][j];

if ((N[q2][j] || vr) == 1) {

N[q2][j] = 1;

}

else

{

N[q2][j] = 0;

};

N[q1][j] = 0;

}

}

for (int i = 0; i < r2; i++) {

k = 0;

if (i == q1) {

continue;

};

for (int j = 0; j < r2; j++)

{

if (j != q1) {

N2[l][k] = N[i][j];

k++;

};

}

l++;

}

for (int i = 0; i < r2 - 1; i++) {

for (int j = 0; j < r2 - 1; j++)

{

printf("%5d", N2[i][j]);

}

printf("\n");

}

//----------------------------------------------------------------------

printf("\nСтягивание ребра");

printf("\nВершинки\n");

scanf("%d%d", &q1, &q2);

vr = 0;

for (int j = 0; j < r2; j++) {

for (int i = 0; i < r2; i++)

{

N[i][j] = M2[i][j];

vr = M2[i][q1];

if ((N[i][q2] || vr) == 1) {

N[i][q2] = 1;

}

else

{

N[i][q2] = 0;

};

N[i][q1] = 0;

}

}

for (int i = 0; i < r2; i++) {

for (int j = 0; j < r2; j++)

{

vr = M2[q1][j];

if ((N[q2][j] || vr) == 1) {

N[q2][j] = 1;

}

else

{

N[q2][j] = 0;

};

N[q1][j] = 0;

}

}

for (int i = 0; i < r2; i++) {

for (int j = 0; j < r2; j++)

{

if (N[q2][q2] == 1) {

N[q2][q2] = 0;

};

}

}

l = 0;

for (int i = 0; i < r2; i++) {

k = 0;

if (i == q1) {

continue;

};

for (int j = 0; j < r2; j++)

{

if (j != q1) {

N2[l][k] = N[i][j];

k++;

};

}

l++;

}

for (int i = 0; i < r2 - 1; i++) {

for (int j = 0; j < r2 - 1; j++)

{

printf("%5d", N2[i][j]);

}

printf("\n");

}

printf("\n");

//----------------------------------------------------------------------

int vver;

int nov = 0;

printf("\nРасщепление вершины");

printf("\nВершинка\n");

scanf("%d", &vver);

for (int j = 0; j < r2; j++) {

for (int i = 0; i < r2; i++)

{

N3[i][j] = M2[i][j];

if (i == r2) {

N3[i][j] = 0;

}

}

}

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

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

{

if (j == r2) {

N3[i][j] = 0;

}

}

}

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

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

{

if (i == r2) {

N3[i][j] = 0;

}

}

}

for (int j = 0; j < r2; j++) {

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

if (j % 2 == 0 && N3[vver][j] == 1)

{

N3[vver][j] = 0;

N3[r2][j] = 1;

}

else {

continue;

}

}

}

for (int i = 0; i < r2; i++) {

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

if (i % 2 == 0 && N3[i][vver] == 1)

{

N3[i][vver] = 0;

N3[i][r2] = 1;

}

else {

continue;

}

}

}

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

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

{

if (i == vver)

{

N3[i][r2] = 1;

}

if (i == r2)

{

N3[i][vver] = 1;

}

}

}

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

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

printf("%5d,", N3[i][j]);

}

printf("\n");

}

//----------------------------------------------------------------------

printf("\n\nЗад 3 Объединение графов \n");

int r;

if (r1 >= r2)

{

w = r2;

r = r1;

}

else {

w = r1;

r = r2;

}

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

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

{

if (r2 >= r1)

{

E[i][j] = M2[i][j];

}

else {

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

}

}

}

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

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

if ((M1[i][j] || M2[i][j]) == 1) {

E[i][j] = 1;

}

else

{

E[i][j] = 0;

}

}

}

printf("\n\n");

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

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

printf("%5d,", E[i][j]);

}

printf("\n");

}

//----------------------------------------------------------------------

printf("\nПересечение графов \n");

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

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

{

E[i][j] = 0;

}

}

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

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

if ((M1[i][j] && M2[i][j]) == 1) {

E[i][j] = 1;

}

else

{

E[i][j] = 0;

}

}

printf("\n");

}

printf("\n\n");

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

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

printf("%5d,", E[i][j]);

}

printf("\n");

}

//----------------------------------------------------------------------

printf("\nКольцевая сумма графов\n");

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

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

{

if (r2 <= r1)

{

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

}

else {

E[i][j] = M2[i][j];

}

}

}

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

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

if ((M1[i][j] ^ M2[i][j]) == 1) {

E[i][j] = 1;

}

else

{

E[i][j] = 0;

}

}

printf("\n");

}

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

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

printf("%5d,", E[i][j]);

}

printf("\n");

}

for (int i = 0; i < r2; i++)

free(M2[i]);

free(M2);

getchar(); getchar();

for (int i = 0; i < r2; i++)

free(N2[i]);

free(N2);

getchar(); getchar();

for (int i = 0; i < r2; i++)

free(N[i]);

free(N);

getchar(); getchar();

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

free(N3[i]);

free(N3);

getchar(); getchar();

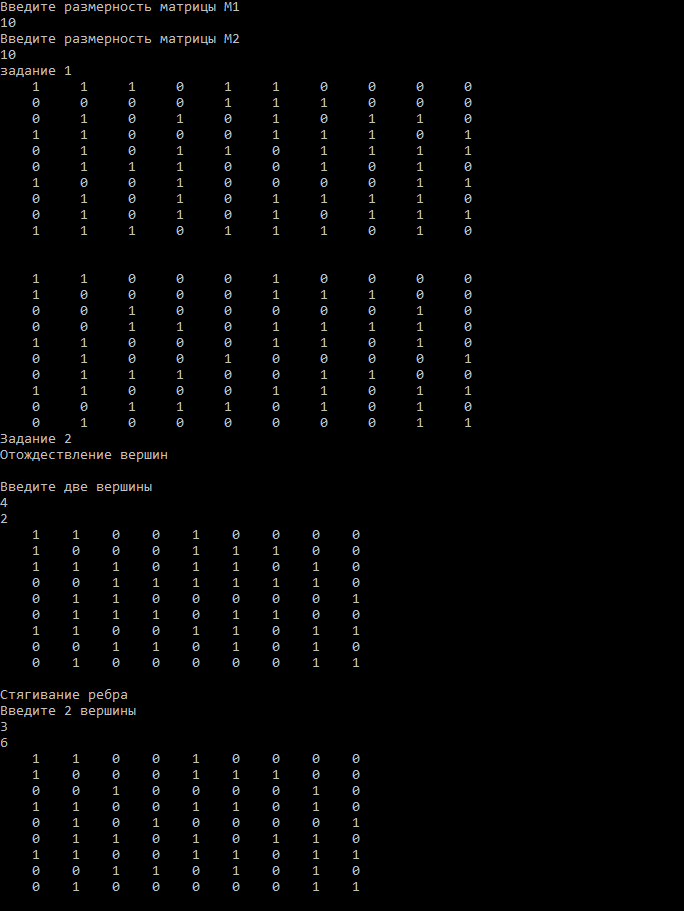
for (int i = 0; i < r; i++)

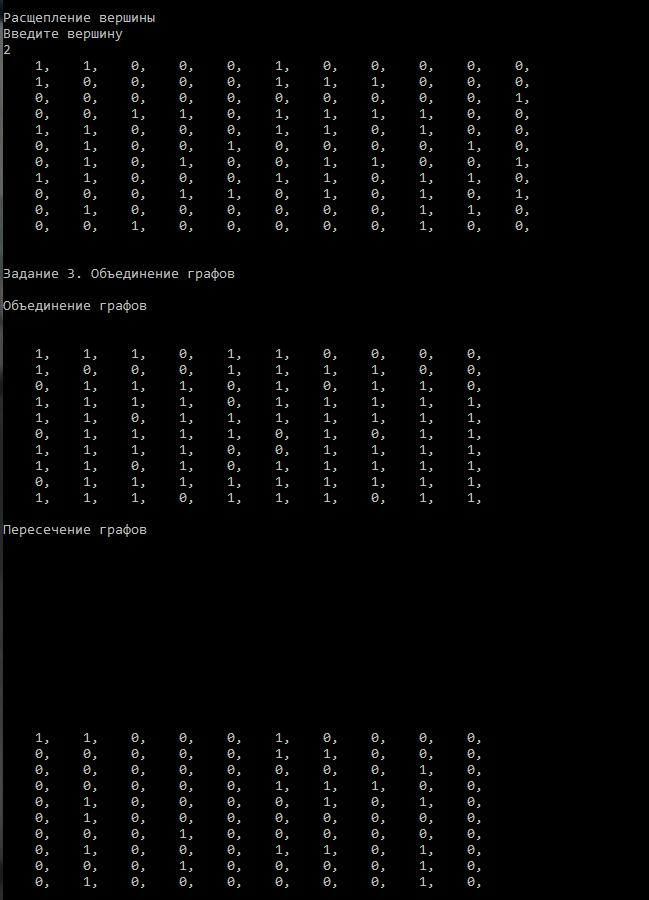
free(E[i]);

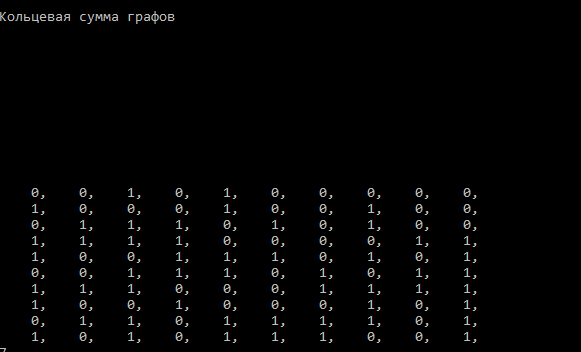
free(E);

getchar(); getchar();

}







**Вывод:** Выполнили унарные и бинарные операции над графами.