Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №3

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему « Унарные и бинарные операции над графами»

Выполнил:

студент группы 20ВВ2

Кочергин В.П.

Принял:

к.т.н. доцент Юрова О.В.

д.т.н. профессор Митрохин М. А.

Пенза 2021

**Цель работы:** Изучить и написать алгоритм к унарным и бинарным операциям.

**Задание 1** Формирование матрицы и списка смежности.

Листинг:

#include <time.h>

#include <malloc.h>

#include <iostream>

#include <windows.h>

using namespace std;

struct Node

{

int items;

Node\* next;

};

typedef Node\* PNode;

PNode\* Head;

void InitializHead(int size)

{

int i;

Head = (PNode\*)malloc(size \* sizeof(PNode));

for (i = 0; i < size; i++) Head[i] = NULL;

}

PNode CreateNode(int items)

{

PNode NewNode = (Node\*)malloc(sizeof(Node));

NewNode->items = items;

NewNode->next = NULL;

return NewNode;

}

void AddFirst(PNode NewNode, int i)

{

NewNode->next = Head[i];

Head[i] = NewNode;

}

void AddAfter(PNode q, PNode NewNode)

{

NewNode->next = q->next;

q->next = NewNode;

}

void AddLast(PNode NewNode, int i)

{

PNode q = Head[i];

while (q->next) q = q->next;

AddAfter(q, NewNode);

}

void print(int i)

{

PNode q = Head[i];

while (q)

{

printf("%d ", q->items);

q = q->next;

}

printf("\n");

}

int main()

{

int size, i, j, ran, \*\*a;

PNode NewNode = NULL;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

printf("Укажите размер массива: ");

scanf\_s("%d", &size);

a = (int\*\*)malloc(size \* sizeof(int));

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

{

a[i] = (int\*)malloc(size \* sizeof(int));

}

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

{

for (j = 0; j < size; j++)

{

a[i][j] = 0;

}

}

srand(time(NULL));

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

{

for (j = 1 + i; j < size; j++)

{

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

}

}

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

{

for (j = 0; j < size; j++)

{

a[j][i] = a[i][j];

}

}

InitializHead(size);

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

{

NewNode = CreateNode(i);

AddFirst(NewNode, i);

}

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

{

for (j = 0; j < size; j++)

{

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

{

NewNode = CreateNode(j);

AddLast(NewNode, i);

}

}

}

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

{

for (j = 0; j < size; j++)

{

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

}

printf("\n");

}

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

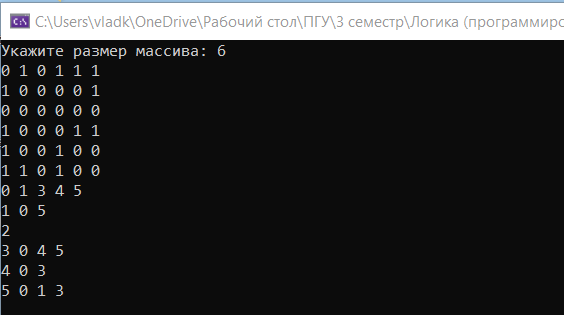
{

print(i);

}

}

Результаты работы программы:



**Задание 2** Унарные операции для матриц

Листинг:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>

void otozhd(int size, int \*\*a)

{

int i, j, max, min, \*\*g;

g = (int\*\*)malloc(size \* sizeof(int));

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

{

g[i] = (int\*)malloc(size \* sizeof(int));

}

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

{

for (j = 0; j < size; j++)

{

g[i][j] = 0;

}

}

printf("\n");

printf("Укажите max: ");

scanf\_s("%d", &max);

printf("Укажите min: ");

scanf\_s("%d", &min);

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

{

for (j = 0; j < size; j++)

{

if ((i < max) && (j < max)) g[i][j] = a[i][j];

if ((i < max) && (j == max)) g[i][min] += a[i][j];

if ((i > max) && (j == max)) g[i - 1][min] += a[i][j];

if ((i == max) && (j < max)) g[min][j] += a[i][j];

if ((i == max) && (j > max)) g[min][j - 1] += a[i][j];

if ((i < max) && (j > max)) g[i][j - 1] = a[i][j];

if ((j < max) && (i > max)) g[i - 1][j] = a[i][j];

}

}

for (i = max + 1; i < size; i++)

{

for (j = max + 1; j < size; j++)

{

g[i - 1][j - 1] = a[i][j];

}

}

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

{

for (j = 0; j < size; j++)

{

if (g[i][j] > 1) g[i][j] = 1;

}

}

for (i = 0; i < size - 1; i++)

{

for (j = 0; j < size - 1; j++)

{

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

}

printf("\n");

}

}

void styag(int size, int\*\* a)

{

int razn, i, j, min, max, \*\*g;

g = (int\*\*)malloc(size \* sizeof(int));

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

{

g[i] = (int\*)malloc(size \* sizeof(int));

}

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

{

for (j = 0; j < size; j++)

{

g[i][j] = 0;

}

}

i = 0; j = 0;

while (a[i][j] == 0)

{

printf("Укажите адрес строки: ");

scanf\_s("%d", &i);

printf("Укажите адрес столбца: ");

scanf\_s("%d", &j);

printf("\n");

if (a[i][j] == 0) printf("Некоректный ввод\n");

}

if (i > j)

{

max = i;

min = j;

}

else

{

max = j;

min = i;

}

razn = max - min;

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

{

for (j = 0; j < size; j++)

{

if ((i < max) && (j < max)) g[i][j] = a[i][j];

if ((i < min) && (j == max)) g[i][min] += a[i][j];

if (((i > max) && (j == max)) && (i != min)) g[i - 1][min] += a[i][j];

if ((i == max) && (j < min)) g[min][j] += a[i][j];

if (((i == max) && (j > max)) && (j != min)) g[min][j - 1] += a[i][j];

if ((i < max) && (j > max)) g[i][j - 1] = a[i][j];

if ((j < max) && (i > max)) g[i - 1][j] = a[i][j];

}

}

for (i = max + 1; i < size; i++)

{

for (j = max + 1; j < size; j++)

{

g[i - 1][j - 1] = a[i][j];

}

}

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

{

for (j = 0; j < size; j++)

{

if (g[i][j] > 1) g[i][j] = 1;

}

}

for (i = 0; i < size - 1; i++)

{

for (j = 0; j < size - 1; j++)

{

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

}

printf("\n");

}

}

void raschep(int size, int\*\* a)

{

int razn, i, j, num, \*\*g, sizeg;

sizeg = size + 1;

g = (int\*\*)malloc(sizeg \* sizeof(int));

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

{

g[i] = (int\*)malloc(sizeg \* sizeof(int));

}

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

{

for (j = 0; j < sizeg; j++)

{

g[i][j] = 0;

}

}

printf("\n");

printf("Укажите столбец, который хотите расщепить: ");

scanf\_s("%d", &num);

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

{

for (j = 0; j < size; j++)

{

if ((i <= num) && (j <= num)) g[i][j] = a[i][j];

if ((i >= num) && (j <= num)) g[i + 1][j] = a[i][j];

if ((j >= num) && (i <= num)) g[i][j + 1] = a[i][j];

if (((i >= num) && (j <= num)) && ((i != num) && (j != num))) g[i + 1][j] = a[i][j];

if (((j >= num) && (i <= num)) && ((i != num) && (j != num))) g[i][j + 1] = a[i][j];

if (((i == num + 1) && (j == num)) || ((i == num) && (j == num + 1))) g[i][j] = 1;

}

}

for (i = num; i < size; i++)

{

for (j = num; j < size; j++)

{

g[i + 1][j + 1] = a[i][j];

}

}

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

{

for (j = 0; j < sizeg; j++)

{

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

}

printf("\n");

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setvbuf(stdin, NULL, \_IONBF, 0);

setvbuf(stdout, NULL, \_IONBF, 0);

int i, j, \*\*a, max, \*\*g, size, min, type;

printf("Укажите размер матрицы: ");

scanf\_s("%d", &size);

a = (int\*\*)malloc(size \* sizeof(int));

//srand(time(NULL));

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

{

a[i] = (int\*)malloc(size \* sizeof(int));

}

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

{

for (j = 0; j < size; j++)

{

a[i][j] = 0;

}

}

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

{

for (j = 1 + i; j < size; j++)

{

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

}

}

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

{

for (j = 0; j < size; j++)

{

a[j][i] = a[i][j];

}

}

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

{

for (j = 0; j < size; j++)

{

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

}

printf("\n");

}

printf("\n");

printf("Выберите тип унарной операции \n1 - отождествление \n2 - стягивание \n3 - расщепление\n");

scanf\_s("%d", &type);

printf("\n");

switch (type)

{

case 1: otozhd(size, a);

break;

case 2: styag(size, a);

break;

case 3: raschep(size, a);

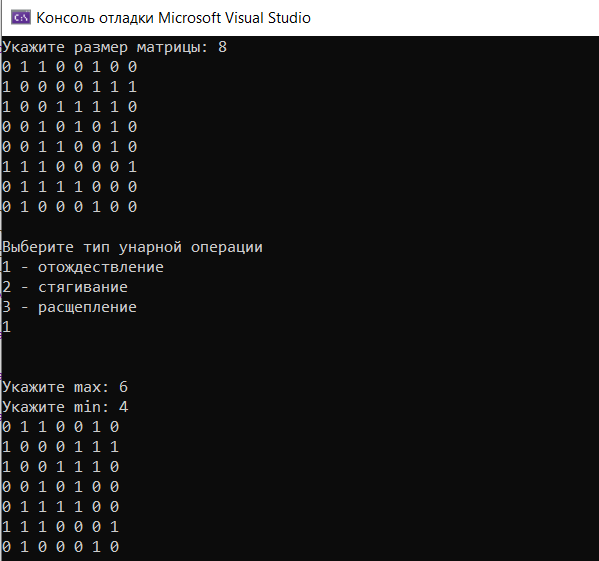
break;

}

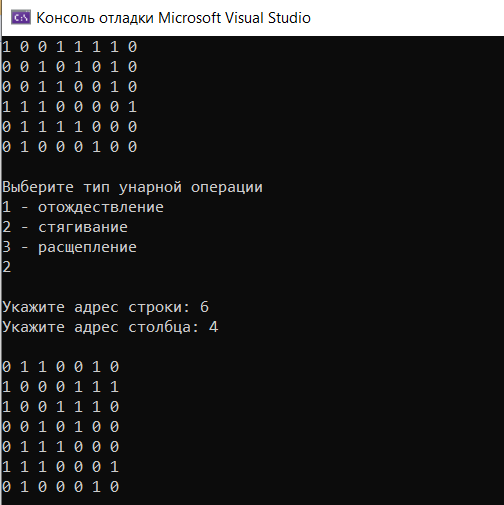
}

Результат работы программы

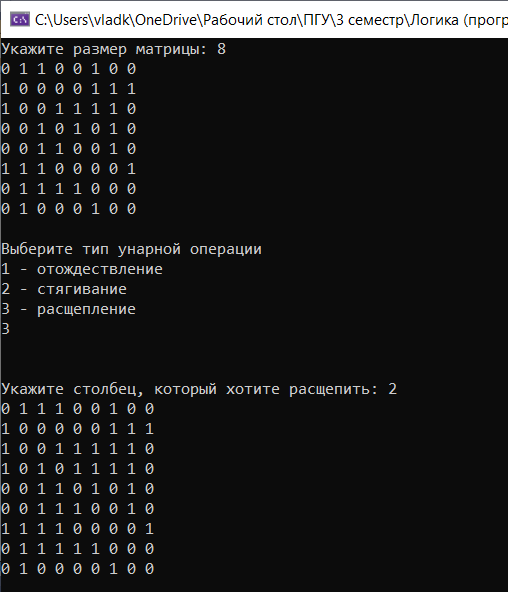
1) Отождествление



2) Стягивание



3) Расщепление



**Задание 3** Бинарные операции для матриц

Листинг:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>

void objed(int size1, int size2, int size, int\*\* g, int\*\* a, int\*\* b)

{

int i, j;

for (i = 0; i < size1; i++)

{

for (j = 0; j < size1; j++)

{

g[i][j] = g[i][j] + a[i][j];

}

}for (i = 0; i < size2; i++)

{

for (j = 0; j < size2; j++)

{

g[i][j] = g[i][j] + b[i][j];

}

}

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

{

for (j = 0; j < size; j++)

{

if (g[i][j] > 1) g[i][j] = 1;

}

}

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

{

for (j = 0; j < size; j++)

{

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

}

printf("\n");

}

}

void peres(int minsize, int size, int\*\* g, int\*\* a, int\*\* b)

{

int i, j;

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

{

for (j = 0; j < minsize; j++)

{

g[i][j] = a[i][j] \* b[i][j];

}

}

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

{

for (j = 0; j < size; j++)

{

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

}

printf("\n");

}

}

void sum(int minsize, int size, int size1, int size2, int\*\* g, int\*\* a, int\*\* b)

{

int i, j;

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

{

for (j = 0; j < minsize; j++)

{

if ((a[i][j] == 1) && (b[i][j] == 1)) g[i][j] = 0;

if ((a[i][j] == 1) && (b[i][j] == 0)) g[i][j] = 1;

if ((a[i][j] == 0) && (b[i][j] == 1)) g[i][j] = 1;

}

}

if (size1 > size2)

{

for (i = 0; i < size1; i++)

{

for (j = 0; j < size1; j++)

{

if (j >= size2) if ((a[i][j] == 1)) g[i][j] = 1;

if (i >= size2) if ((a[i][j] == 1)) g[i][j] = 1;

}

}

}

else

{

for (i = 0; i < size2; i++)

{

for (j = 0; j < size2; j++)

{

if (j >= size1) if ((b[i][j] == 1)) g[i][j] = 1;

if (i >= size1) if ((b[i][j] == 1)) g[i][j] = 1;

}

}

}

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

{

for (j = 0; j < size; j++)

{

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

}

printf("\n");

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setvbuf(stdin, NULL, \_IONBF, 0);

setvbuf(stdout, NULL, \_IONBF, 0);

int i, j, \*\* a, \*\*b, max, \*\* g, size1, size2, size, minsize, min, razn, type;

printf("Укажите размер 1-ой матрицы: ");

scanf\_s("%d", &size1);

printf("Укажите размер 2-ой матрицы: ");

scanf\_s("%d", &size2);

a = (int\*\*)malloc(size1 \* sizeof(int));

b = (int\*\*)malloc(size2 \* sizeof(int));

if (size1 > size2)

{

size = size1;

minsize = size2;

}

else

{

size = size2;

minsize = size1;

}

g = (int\*\*)malloc(size \* sizeof(int));

//srand(time(NULL));

for (i = 0; i < size1; i++)

{

a[i] = (int\*)malloc(size1 \* sizeof(int));

}

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

{

g[i] = (int\*)malloc(size \* sizeof(int));

}

for (i = 0; i < size1; i++)

{

for (j = 0; j < size1; j++)

{

a[i][j] = 0;

}

}

for (i = 0; i < size1; i++)

{

for (j = 1 + i; j < size1; j++)

{

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

}

}

for (i = 0; i < size1; i++)

{

for (j = 0; j < size1; j++)

{

a[j][i] = a[i][j];

}

}

for (i = 0; i < size1; i++)

{

for (j = 0; j < size1; j++)

{

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

}

printf("\n");

}

printf("\n");

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

{

for (j = 0; j < size; j++)

{

g[i][j] = 0;

}

}

//srand(time(NULL));

for (i = 0; i < size2; i++)

{

b[i] = (int\*)malloc(size2 \* sizeof(int));

}

for (i = 0; i < size2; i++)

{

for (j = 0; j < size2; j++)

{

b[i][j] = 0;

}

}

for (i = 0; i < size2; i++)

{

for (j = 1 + i; j < size2; j++)

{

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

}

}

for (i = 0; i < size2; i++)

{

for (j = 0; j < size2; j++)

{

b[j][i] = b[i][j];

}

}

for (i = 0; i < size2; i++)

{

for (j = 0; j < size2; j++)

{

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

}

printf("\n");

}

printf("\n");

printf("Выберите тип бинарной операции \n1 - объединение \n2 - пересечение \n3 - кольцевая сумма\n");

scanf\_s("%d", &type);

printf("\n");

switch (type)

{

case 1: objed(size1, size2, size, g, a, b);

break;

case 2: peres(minsize, size, g, a, b);

break;

case 3: sum(minsize, size, size1, size2, g, a, b);

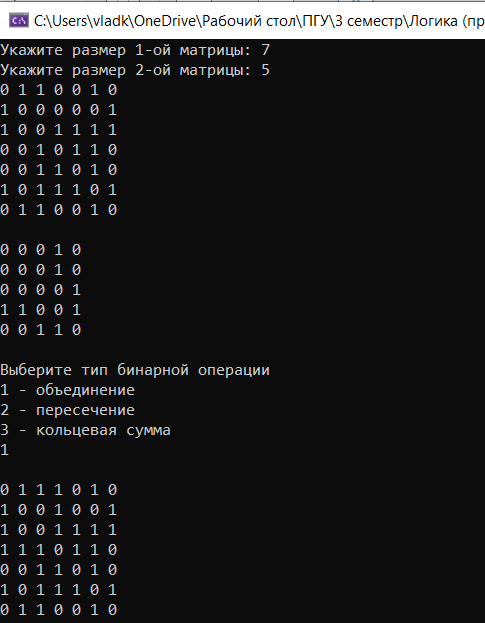
break;

}

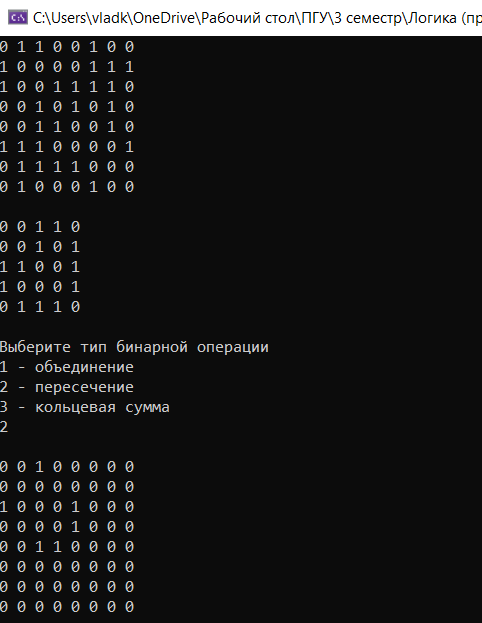
}

Результат работы программы

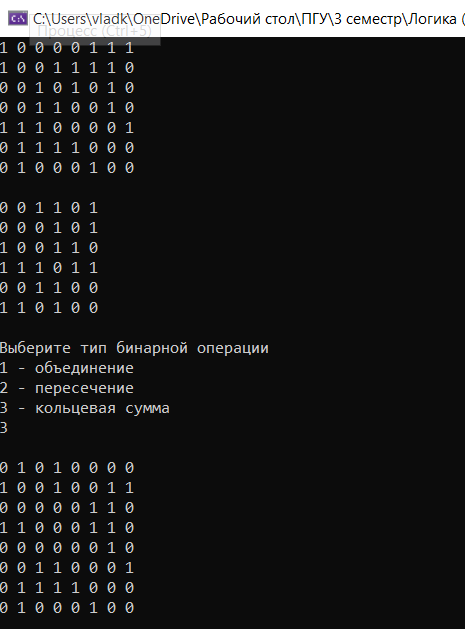
1) Объединение



2) Пересечение



3) Кольцевая сумма



**Задание 4** Декартово произведение.

Листинг:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>

using namespace std;

int main()

{

int i, j, size, \*\* a, \*\* g, \*\*A, m, n, ran;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setvbuf(stdin, NULL, \_IONBF, 0);

setvbuf(stdout, NULL, \_IONBF, 0);

printf("Введите размеры массивов ");

scanf\_s("%d", &size);

printf("Матрица смежности:\n");

a = (int\*\*)malloc(size \* sizeof(int));

g = (int\*\*)malloc(size \* sizeof(int));

A = (int\*\*)malloc((size \* size) \* sizeof(int));

srand(time(NULL));

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

{

a[i] = (int\*)malloc(size \* sizeof(int));

g[i] = (int\*)malloc(size \* sizeof(int));

}

for (i = 0; i < (size \* size); i++)

{

A[i] = (int\*)malloc((size \* size) \* sizeof(int));

}

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

{

for (j = 0; j < size; j++)

{

a[i][j] = 0;

g[i][j] = 0;

}

}

for (i = 0; i < (size \* size); i++)

{

for (j = 0; j < (size \* size); j++)

{

A[i][j] = 0;

}

}

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

{

for (j = 1 + i; j < size; j++)

{

ran = rand() % 101;

if (ran > 50) a[i][j] = 1;

else a[i][j] = 0;

}

}

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

{

for (j = 1 + i; j < size; j++)

{

ran = rand() % 101;

if (ran > 50) g[i][j] = 1;

else g[i][j] = 0;

}

}

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

{

for (j = 0; j < size; j++)

{

a[j][i] = a[i][j];

g[j][i] = g[i][j];

}

}

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

{

for (j = 0; j < size; j++)

{

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

}

printf("\n");

}

printf("\n");

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

{

for (j = 0; j < size; j++)

{

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

}

printf("\n");

}

printf("\n");

for (m = 0; m < size; m++)

{

for (i = size\*m; i < size + size\*m; i++)//строка

{

for (n = 0; n < size; n++)

{

for (j = size\*n; j < size + size\*n; j++)//столбец

{

if ((m == n) && (i == j))

A[i][j] = 0;

if ((m == n) && (i != j))

A[i][j] = g[i - size \* m][j - size \* n];

if ((m != n) && (i - size \* m == j - size \* n))

A[i][j] = a[m][n];

if ((m != n) && (i - size \* m != j - size \* n))

A[i][j] = 0;

}

}

}

}

for (i = 0; i < (size \* size); i++)

{

for (j = 0; j < (size \* size); j++)

{

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

}

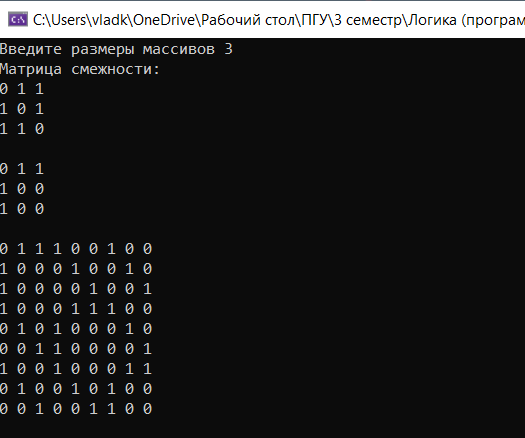
printf("\n");

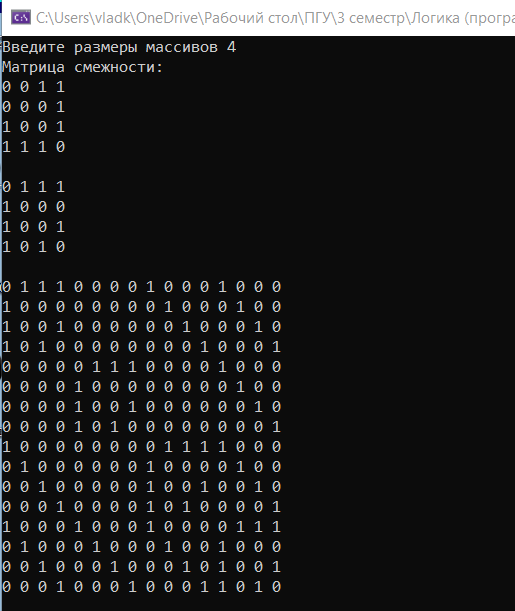
}

printf("\n");

}

Результаты работы программы:





Вывод: В конечном счёте удалось разобраться с данными операциями и написать к ним алгоритм.