МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №3.3**

по дисциплине: Дискретная математика

тема: «Фактормножества»

Выполнил: ст. группы ПВ-221

Лоёк Никита Викторович

Проверили:

Бондаренко Татьяна Владимировна

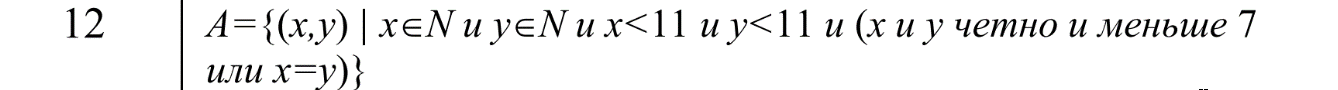
Рязанов Юрий Дмитриевич

Белгород 2023 г.

Л а б о р а т о р н а я р а б о т а № 3.3

**Цель работы:** научиться формировать фактормножесто для заданного отношения эквивалентности на ЭВМ.

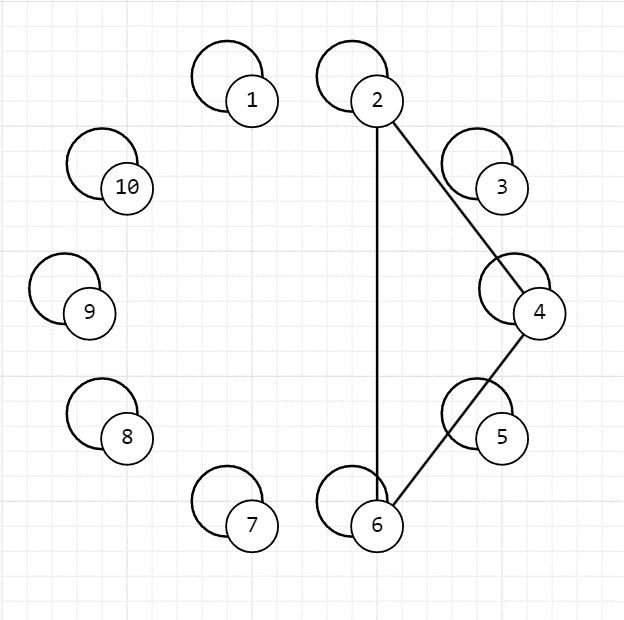
**Вариант 12**



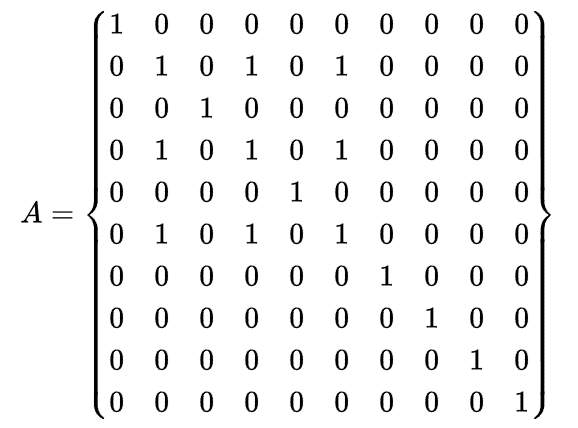
**Задания**

1. Отношение представить графом и характеристической функцией в матричной форме. Найти разбиение Ф, определяемое заданным отношением эквивалентности.

Граф:



Характеристическая функция в матричной форме:



[1] = {1}

[2] = {2, 4, 5}

[3] = {3}

[4] = {2, 4, 5}

[5] = {2, 4, 5}

[6] = {6}

[7] = {7}

[8] = {8}

[9] = {9}

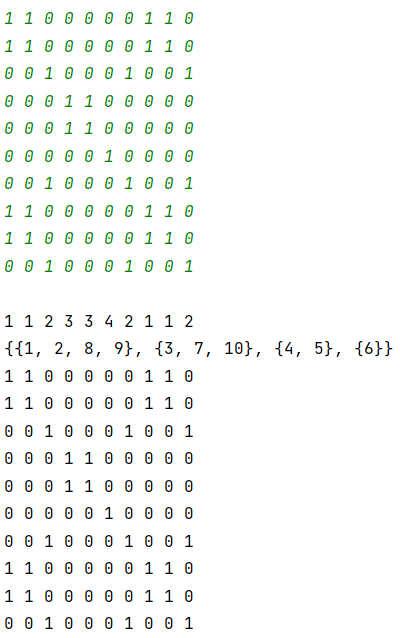
[9] = {10}

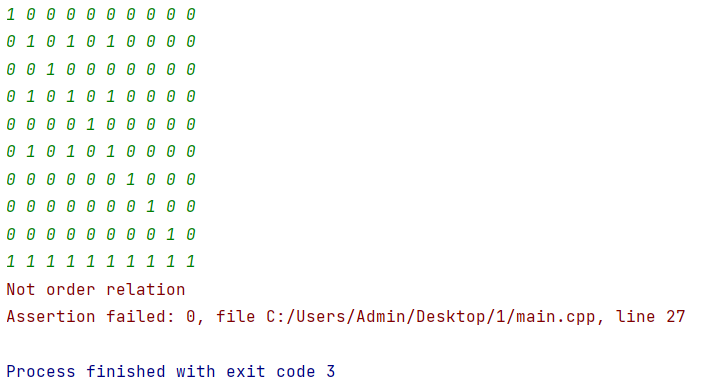
Разбиение

2. Написвть программу, которая формирует разбиение, определяемое заданным отношением эквивалентности.

#include <iostream>  
#include <vector>  
#include <cassert>  
  
  
using namespace std;  
  
  
vector<vector<bool>> inputMatrix(unsigned size) {  
 vector<vector<bool>> matrix(size, vector<bool>(size));  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 bool in;  
 cin >> in;  
 matrix[i][j] = in;  
 }  
 }  
  
 for (int i = 0; i < size; i++) {  
 if (matrix[i][i] != 1) {  
 cerr << "Not order relation" << endl;  
 **assert**(0);  
 }  
 for (int j = 0; j < size; j++) {  
 if (matrix[i][j] != matrix[j][i]) {  
 cerr << "Not order relation" << endl;  
 **assert**(0);  
 }  
 }  
 }  
  
 return matrix;  
}  
  
  
vector<unsigned> createPartition(vector<vector<bool>> &matrix) {  
 vector<unsigned> partition(matrix.size(), 0);  
 unsigned groupsCounter = 0;  
 for (int i = 0; i < partition.size(); i++) {  
 if (partition[i] == 0) {  
 groupsCounter++;  
 for (int j = 0; j < matrix.size(); j++) {  
 if (matrix[i][j]) {  
 partition[j] = groupsCounter;  
 }  
 }  
 }  
 }  
 return partition;  
}  
  
  
void outputArray(vector<unsigned> &array) {  
 for (int i = 0; i < array.size(); i++) {  
 cout << array[i] << ' ';  
 }  
}  
  
  
void outputMatrix(vector<vector<bool>> &matrix) {  
 for (int i = 0; i < matrix.size(); i++) {  
 for (int j = 0; j < matrix[0].size(); ++j) {  
 cout << matrix[i][j] << ' ';  
 }  
 cout << endl;  
 }  
  
}  
  
  
void outputPartition(vector<unsigned> &array) {  
 unsigned maxSize = 0;  
 for (int i = 0; i < array.size(); i++) {  
 if (maxSize < array[i]) {  
 maxSize = array[i];  
 }  
 }  
  
 cout << endl;  
 cout << "{";  
 bool flag1;  
 bool flag2 = false;  
 for (int i = 1; i < maxSize+1; i++) {  
 if (!flag2) {  
 cout << "{";  
 flag2 = true;  
 } else if (flag2) {  
 cout << ", {";  
 }  
  
 flag1 = false;  
 for (int j = 0; j < array.size(); j++) {  
 if (array[j] == i && !flag1) {  
 cout << j+1;  
 flag1 = true;  
 } else if (array[j] == i && flag1) {  
 cout << ", " << j+1;  
 }  
 }  
 cout << "}";  
 }  
 cout << "}" << endl;  
}  
  
  
vector<vector<bool>> f(vector<unsigned> &array) {  
 unsigned maxSize = 0;  
 for (int i = 0; i < array.size(); i++) {  
 if (maxSize < array[i]) {  
 maxSize = array[i];  
 }  
 }  
  
 vector<vector<bool>> matrix(array.size(),vector<bool>(array.size(), false));  
  
 for (int i = 0; i < maxSize; i++) {  
 vector<unsigned> arr;  
 for (int j = 0; j < array.size(); ++j) {  
 if (array[j] == (i + 1)) {  
 arr.push\_back(j+1);  
 }  
 }  
  
  
  
 for (int j = 0; j < arr.size(); ++j) {  
 for (int k = 0; k < arr.size(); ++k) {  
 matrix[arr[j]-1][arr[k]-1] = true;  
 }  
 }  
 }  
  
 return matrix;  
}  
  
  
int main() {  
 vector<vector<bool>> matrix = inputMatrix(10);  
 vector<unsigned> partition = createPartition(matrix);  
  
 cout << endl;  
 outputArray(partition);  
 outputPartition(partition);  
 vector<vector<bool>> m = f(partition);  
 outputMatrix(m);  
  
 return 0;  
}

Результат работы программы:





**Вывод**

**Вывод**: в ходе работы я научился формировать фактормножесто для заданного отношения эквивалентности на ЭВМ.