**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ПЕНЗЕНСКИЙ ГОСУДРАСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ**

Факультет: «**Вычислительная техника**»

Кафедра: «**Математическое обеспечение и применение ЭВМ**»

Направление подготовки: **09.03.04 «Программная инженерия»**

**Матрицы бинарных отношений  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#### По дисциплине «Дискретная математика»

#### ОТЧЕТ

**По лабораторной работе №6**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
| Выполнили: | Угроватов Д. Лялин Н. |
| Группа | 16ВП1 |
| Принял: | доц. Горюнов Ю.Ю. |

Пенза 2018

**Матрицы бинарных отношений**

**Задание:**

1. Сформулировать определение матрицы бинарного отношения.
2. Создать программу для представления бинарных отношений матрицами.

**Требования к программе:**

В тексте программы в виде списков задать множества:

В тексте программы в виде списка списков задать бинарные отношения: , графически их изобразить.

Вывести на экран матрицы:

Программно проверить выполнение равенства

**Ход работы:**

Рассмотрим отношение заданное на множестве действительных чисел . Каждому бинарному отношению можно поставить в соответствие матрицу бинарного отношения, элементы которой определяются следующим правилом:

**Код файла labа6.py:**

A = [1, 2, 3, 4, 5];

B = ['x', 'y', 'z', 'q', 'g'];

delAB = [[1, 'x'], [2, 'y'], [3, 'z'], [3, 'g'], [4, 'z'], [5, 'q'], [5, 'g']] #   AB

roAB = [[2, 'x'], [3, 'y'], [3, 'z'], [3, 'g'], [4, 'x'], [5, 'x'], [1, 'q']] #   AB

# нахождение матрицы бинарного отношения:

def matr(rAB, A, B): # матрица rAB  AB

M = [[0 for x in range(len(B))] for y in range(len(A))]

for row in range(len(A)):

for col in range(len(B)):

M[row][col]=int([A[row], B[col]] in rAB)

return M

def outM(M, cRows): # построчный вывод матриц,

for i in range(cRows):

print(M[i])

Mdel = matr(delAB, A,B)

print("Mdel=");

outM(Mdel,5);

Mro = matr(roAB, A,B)

print("Mro="); outM(Mro,5)

def matrT(M,row,col):

mT=[[0 for x in range(row)] for y in range(col)]

for i in range(row):

for j in range(col):

mT[j][i]=M[i][j]

return mT

def Obr(M,row,col):

obrM=[[0 for x in range(row)] for y in range(col)]

for i in range(row):

for j in range(col):

if M[i][j]==0:obrM[i][j]=1

else:obrM[i][j]=0

return obrM

def Pr(M1,M2,col,row):

M=[[0 for x in range(row)] for y in range(col)]

for i in range(row):

for j in range(col):

if M1[i][j]\*M2[j][i]==1: M[i][j]=1

else:M[i][j]=0

return M

def mUnInterRel(Ms,Mr,row,col,what):

M=[[0 for x in range(col)] for y in range(row)]

for i in range(row):

for j in range(col):

if what==1: M[i][j]=int(Ms[i][j] or Mr[i][j])

else:M[i][j]=int(Ms[i][j] and Mr[i][j])

return M

def MdelAndMro(m1,m2):

s=0

t=[]

m3=[]

if len(m2)!=len(m1[0]):

print("Матрицы не могут быть перемножены")

else:

r1=len(m1)

c1=len(m1[0])

c2=len(m2[0])

for z in range(0,r1):

for j in range(0,c2):

for i in range(0,c1):

s=s+m1[z][i]\*m2[i][j]

if s>=1: t.append(1)

else:t.append(0)

s=0

m3.append(t)

t=[]

return m3

MdAndMr =MdelAndMro(Mdel, Mro)

print("MdAndMr="); outM(MdAndMr,5)

print('Обратная матрица MdelObr=')

MdelObr=Obr(Mdel,5,5)

outM(MdelObr,5)

MdelT=matrT(Mdel,5,5)

print('Транспонированная матрица MdelT=')

outM(MdelT,5)

print('Матрица пересечения Mdel и MdelObr=')

Ma=mUnInterRel(Mdel,MdelObr,5,5,0)

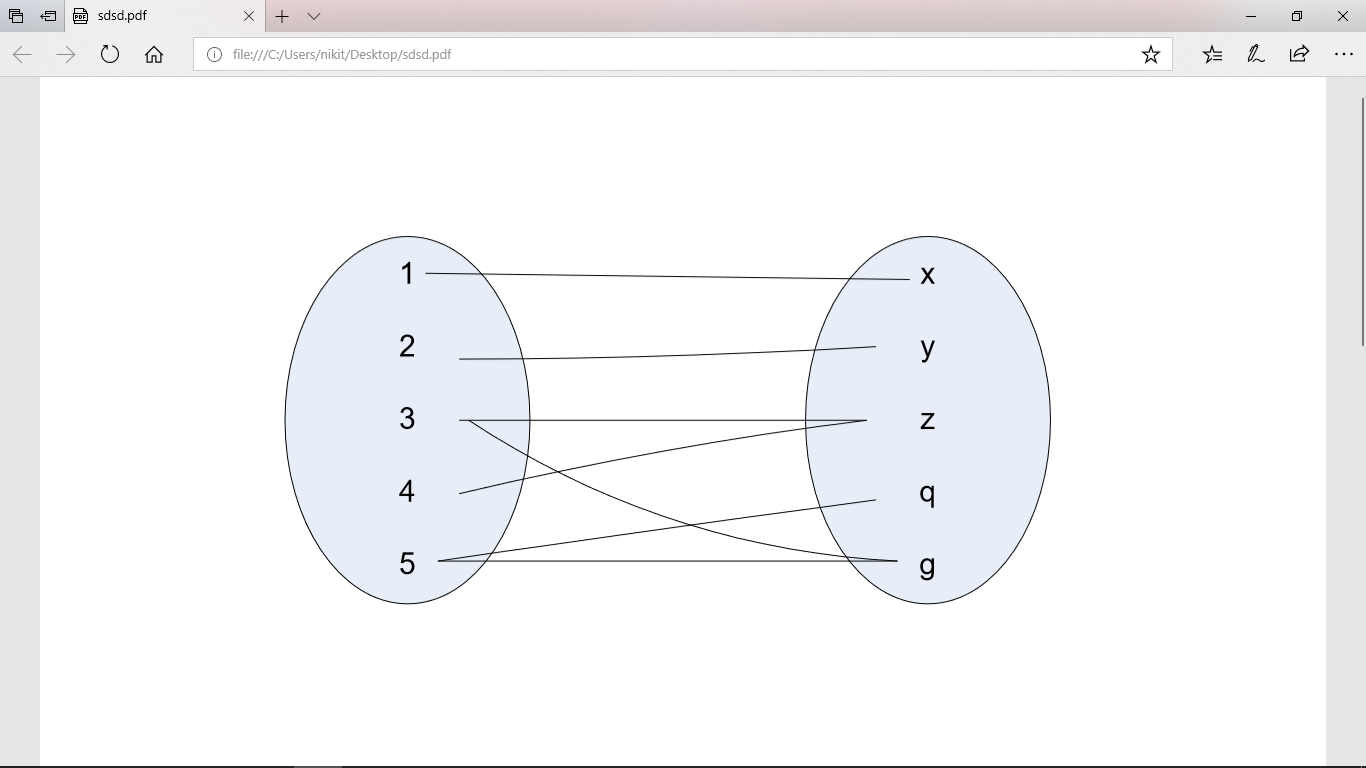
outM(Ma,5)

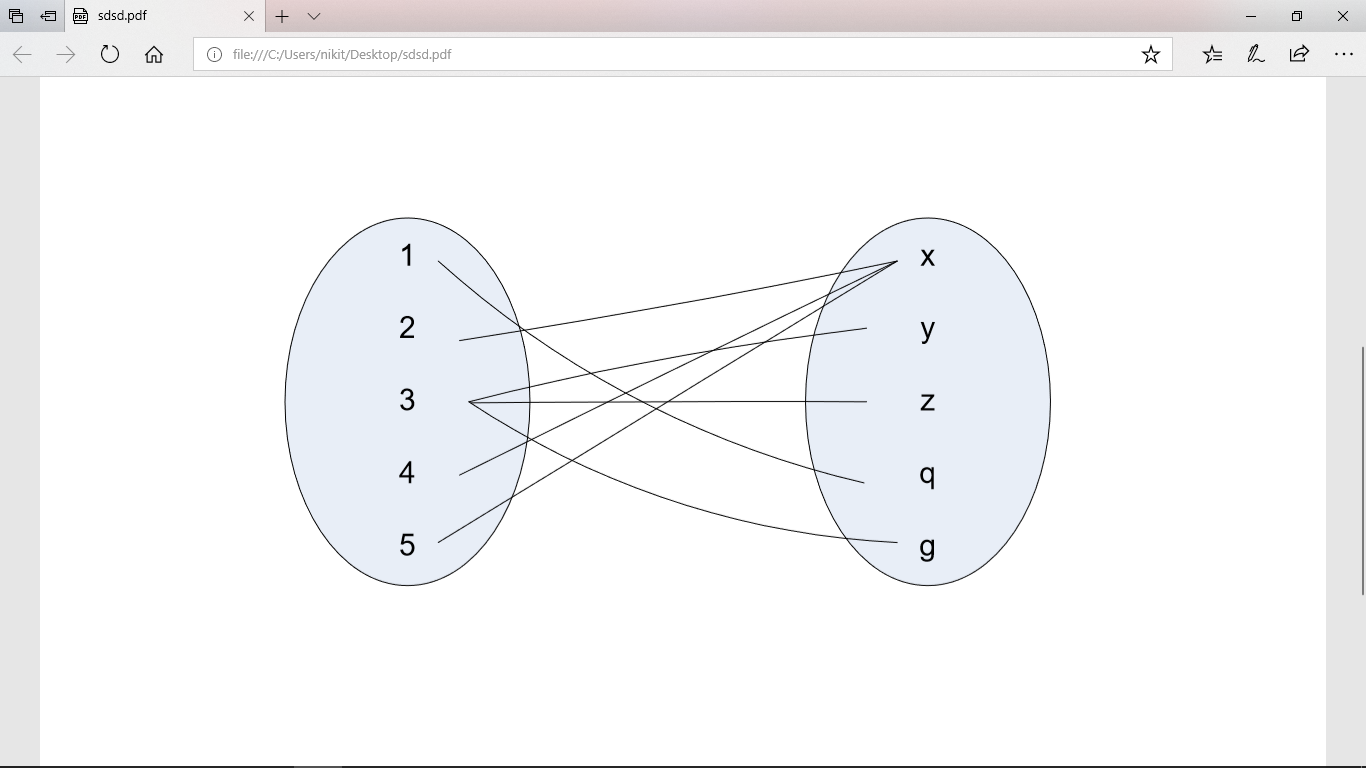
print('Произведение матриц Mdel и NdelT')

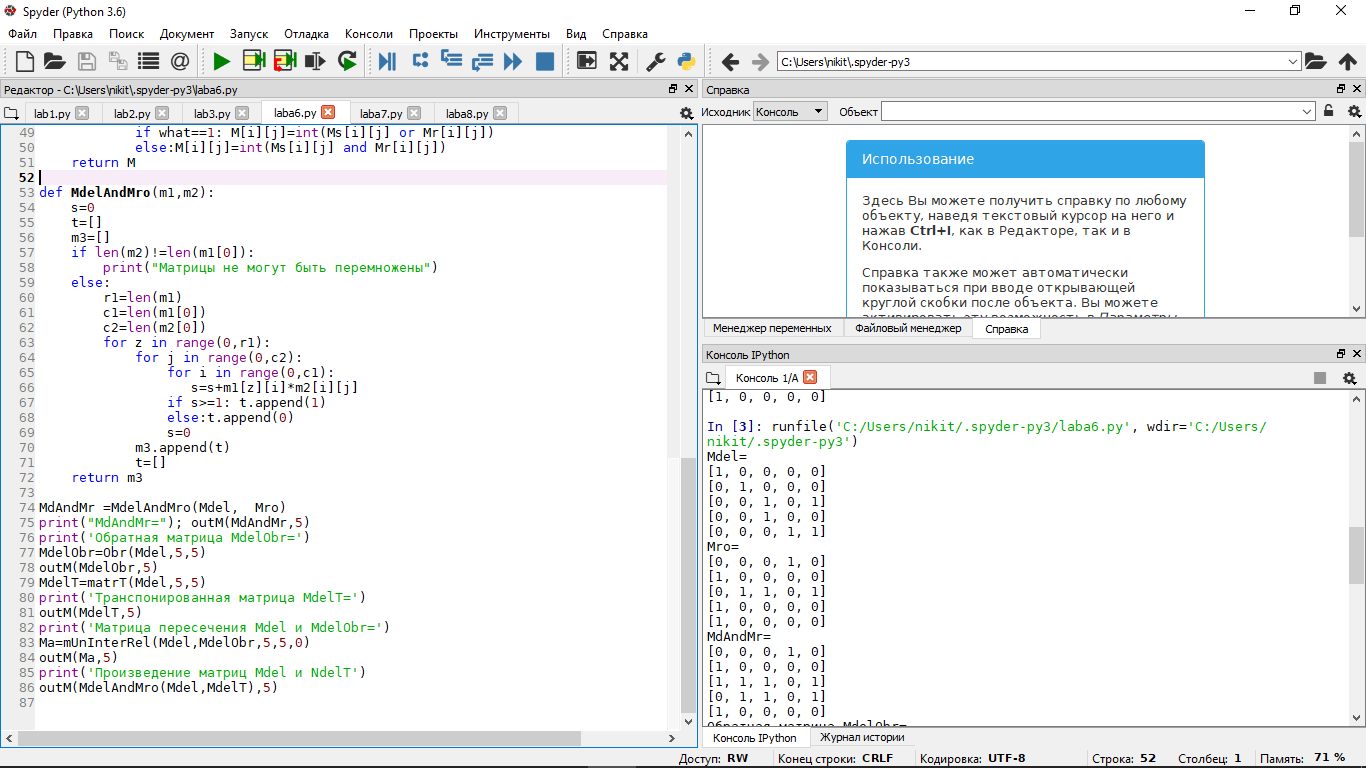
outM(MdelAndMro(Mdel,MdelT),5)

Графические изображения бинарных отношений

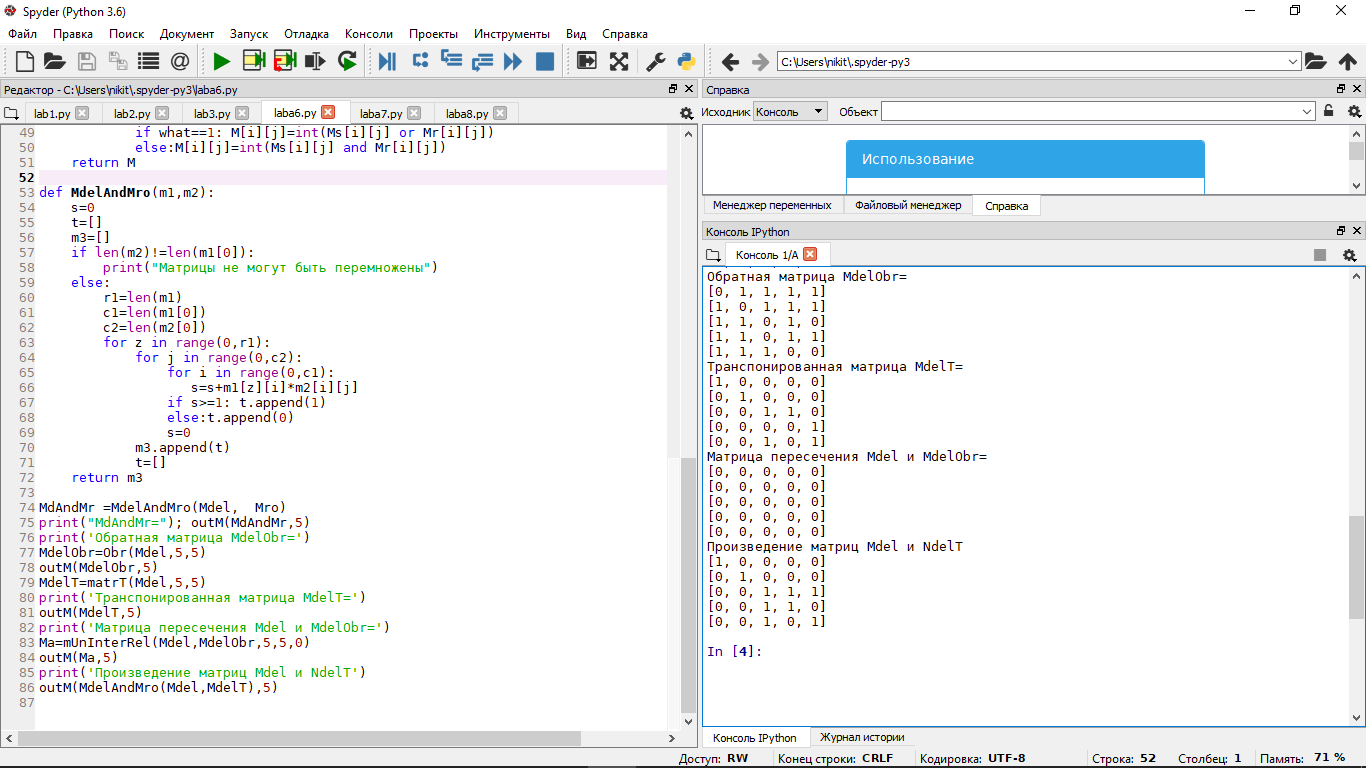
:







Проверка выполнения равенства



**Вывод**

В ходе выполнения лабораторной работы №6 мы научились работать с матрицами бинарных отношений.