БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра интеллектуальных информационных технологий

Лабораторная работа №4

ОПЕРАЦИИ НАД СООТВЕТСВИЯМИ

Выполнили: Xxx Y Z

Группа: xxxxxx

Проверила: Xxx Y Z

Минск, 2xxx

**Постановка задачи:**

Даны два соответствия. Выполнить операции над соответствиями: объединение, пересечение, разность, симметрическая разность, композиция, инверсия, образ, прообраз, сужение, продолжение.

**Уточнение постановки задачи:**

* Соответствие задаётся теоретическим способом (через задание области отправления, области прибытия и графика соответствия)
* Мощность множества области отправления – это натуральное число, которое задается пользователем и не превышает 10.
* Мощность множества области прибытия – это натуральное число, которое задается пользователем и не превышает 10.
* Мощность графика соответствия – это натуральное число, которое задаётся пользователем и не превышает 100.
* Элементы области отправления, области прибытия и графика соответствия задаются пользователем с помощью перечисления
* После задания соответствия пользователь выбирает операцию, выполняемую над соответствиями, либо выходит из программы
* После выполнения каждой операции результат выводится на экран
* Элементами графика соответствия являются упорядоченные пары (кортежи), элементами которых являются натуральные целые числа.
* Элементы множеств и графиков задаются перечислением.

**Определения:**

* **Соответствие между множествами X и Y** – тройка объектов Г = <Х, Y, G>, где X – область отправления соответствия, Y – область прибытия соответствия, G – график соответствия, причём G X × Y.



* **Область определения соответствия <X, Y, G>** – множество пр1G (проекция на первую ось (ось абсцисс) графика соответствия).
* **Область значений соответствия <X, Y, G>** – множество пр2G (проекция на вторую ось (ось ординат) графика соответствия).
* **Объединение соответствий Γ1 = <X, Y, F> и Γ2 = <W, Z, P>** –

Г1 ∪ Г2 = <X∪W, Y∪Z, F∪P>

* **Пересечение соответствий Γ1 = <X, Y, F> и Γ2 = <W, Z, P>** –

Г1 ⋂ Г2 = <X⋂W, Y⋂Z, F⋂P>

* **Разность соответствий Γ1 = <X, Y, F> и Γ2 = <W, Z, P>** –

Г1/Г2 = <X/W, Y/Z, F/P>

* **Симметрическая разность соответствий Γ1 = <X, Y, F> и Γ2 = <W, Z, P>** –

Г1ΔГ2 = <XΔW, YΔZ, FΔP>

* **Инверсия соответствия Γ = <X, Y, G>** – соответствие Г-1, такое, что множество Y является областью отправления соответствия Г-1; множество X является областью прибытия соответствия Г-1, а график соответствия G-1 является инверсией графика G соответствия Г.
* **Композиция (произведением) соответствий Γ1 = <X, Y, F> и Γ2 = <W, Z, P>** –соответствие Γ1·Γ2 = <X, Z, F·P>. Областью отправления конечного соответствия является область отправления Γ1, областью прибытия – область прибытия Γ2, а графиком – композиция графиков F и P.
* **Образ множества N при данном соответствии** – множество Г(N) = {у|(х,у)ϵ G и хϵN}.
* **Прообраз множества M при данном соответствии** – множество Г-1(M) = {x|(х,у)ϵ G и yϵM}.
* **Сужением соответствия Г­ = <X, Y, G> на S (S⊂X)** – соответствие Г­|S = <S, Y, F>, где F(x) = G(x) для всех x∈S. Для Г|S начальное соответствие Г является продолжением

**Описание алгоритма**

**1. Заполнение соответствий:**

**1.1.** Пользователь вводит мощность области отправления X соответствия А

**1.2.** Пользователь вводит элементы множества X.

**1.3.** Пользователь вводит мощность области прибытия Y соответствия А.

**1.4.** Пользователь вводит элементы множества Y.

**1.5.** Пользователь вводит мощность графика F соответствия А.

**1.6.** Пользователь вводит элементы графика F.

**1.7.** Область отправления X выводятся на экран.

**1.8.** Область прибытия Y выводятся на экран.

**1.9.** График F выводятся на экран.

**1.10.** Пользователь вводит мощность области отправления W соответствия B.

**1.11.** Пользователь вводит элементы множества W.

**1.12.** Пользователь вводит мощность области прибытия Z соответствия B.

**1.13.** Пользователь вводит элементы множества Z.

**1.14.** Пользователь вводит мощность графика P соответствия B.

**1.15.** Пользователь вводит элементы графика P.

**1.16.** Область отправления W выводятся на экран.

**1.17.** Область прибытия Z выводятся на экран.

**1.18.** График P выводятся на экран.

**2. Выбор операции:**

**2. 1** Пользователь выбирает одну из предложенных операций:

* Объединение
* Пересечение
* Разность А/В
* Разность В/А
* Симметрическая разность
* Композиция А&В
* Композиция В&А
* Инверсия А
* Инверсия В
* Образ А(С)
* Образ В(С)
* Прообраз А-1(D)
* Прообраз В-1(D)
* Сужение А на S
* Сужение В на S
* Продолжение А
* Продолжение В
* Выход из программы

**2.2** Если пользователь выбрал «Объединение», переходим к пункту 3.

**2.3** Если пользователь выбрал «Пересечение», переходим к пункту 4.

**2.4** Если пользователь выбрал «Разность А/В», переходим к пункту 5.

**2.5** Если пользователь выбрал «Разность B/A», переходим к пункту 6.

**2.6** Если пользователь выбрал «Симметрическая разность», переходим к пункту 7.

**2.7** Если пользователь выбрал «Композиция А&В», переходим к пункту 8.

**2.8** Если пользователь выбрал «Композиция В&А», переходим к пункту 9.

**2.9** Если пользователь выбрал «Инверсия А», переходим к пункту 10.

**2.10** Если пользователь выбрал «Инверсия В», переходим к пункту 11.

**2.11** Если пользователь выбрал «Образ А(С)», переходим к пункту 12.

**2.12** Если пользователь выбрал «Образ В(С)», переходим к пункту 13.

**2.13** Если пользователь выбрал «Прообраз А-1(D)», переходим к пункту 14.

**2.14** Если пользователь выбрал «Прообраз B-1(D)», переходим к пункту 15.

**2.15** Если пользователь выбрал «Сужение А», переходим к пункту 16.

**2.16** Если пользователь выбрал «Сужение В», переходим к пункту 17.

**2.17** Если пользователь выбрал «Продолжение А», переходим к пункту 18.

**2.18** Если пользователь выбрал «Продолжение В», переходим к пункту 19.

**2.19** Если пользователь выбрал «Выход из программы», переходим к пункту 20.

**3. Операция объединения:**

**3.1** Объединение графиков F и P соответствий A и B:

**3.1.1** Программа создает пустой график С.

**3.1.2** Выбирается первый элемент графика А.

**3.1.3** Записываем выбранный элемент графика А в график С.

**3.1.4** Если выбранный элемент графика А – последний, переходим к пункту 3.1.7.

**3.1.5** Выбираем следующий элемент графика А.

**3.1.6** Переход к пункту 3.1.3.

**3.1.7** Выбираем первый элемент графика В.

**3.1.8** Выбираем первый элемент графика А.

**3.1.9** Если первая компонента выбранного элемента графика В не равна первой компоненте выбранного элемента графика А:

**3.1.9.1** Выбранный элемент графика B записывается в график C:

**3.1.9.1.1** Создаем кортеж c длины 2.

**3.1.9.1.2** Записываем первую компоненту выбранного элемента графика B на место первой компоненты кортежа c.

**3.1.9.1.3** Записываем вторую компоненту выбранного элемента графика B на место второй компоненты кортежа c.

**3.1.9.1.4** Кортеж c записывается в график С.

**3.1.9.2** Если выбранный элемент графика В является последним.

**3.1.9.2.1** Переход к пункту 3.1.2.

**3.1.9.3** Выбираем следующий элемент графика В.

**3.1.9.4** Переход к пункту 3.1.8

**3.1.10** Если вторая компонента выбранного элемента графика В не равна второй компоненте выбранного элемента графика А:

**3.1.10.1** Переход к пункту 3.1.9.1

**3.1.11** Если выбранный элемент графика В не является последним.

**3.1.11.1** Выбираем следующий элемент графика В.

**3.1.11.2** Переход к пункту 3.1.8

**3.2** Объединение множеств X и W соответствий A и B

**3.2.1** Программа создает пустое множество С1.

**3.2.2** Выбирается первый элемент множества X.

**3.2.3** Выбранный элемент множества X записывается во множество C1.

**3.2.4** Если выбранный элемент множества X – последний, переходим к пункту 3.2.7.

**3.2.5** Выбираем следующий элемент множества X.

**3.2.6** Переход к пункту 3.2.3.

**3.2.7** Выбираем первый элемент множества W.

**3.2.8** Выбираем первый элемент множества X.

**3.2.9** Если выбранный элемент множества X не равен выбранному элементу множества W:

**3.2.9.1** Если выбранный элемент множества X является последним:

**3.2.9.1.1** Переходим к пункту 3.2.11.

**3.2.9.2** Выбираем следующий элемент множества X.

**3.2.9.3** Переход к пункту 3.2.9.

**3.2.10** Если выбранный элемент множества X равен выбранному элементу множества W:

**3.2.10.1** Если выбранный элемент множества W является последним:

**3.2.10.1.1** Переходим к пункту 3.3.

**3.2.10.2** Выбираем следующий элемент множества W.

**3.2.10.3** Переход к пункту 3.2.9.

**3.2.11** Записываем выбранный элемент множества W во множество С1.

**3.2.12** Если выбранный элемент множества W не является последним:

**3.2.12.1** Выбираем следующий элемент множества W.

**3.2.12.2** Переход пункту 3.2.8.

**3.3** Объединение множеств Y и Z соответствий A и B

**3.3.1** Программа создает пустое множество С2.

**3.3.2** Выбирается первый элемент множества Y.

**3.3.3** Выбранный элемент множества Y записывается во множество C2.

**3.3.4** Если выбранный элемент множества Y – последний, переходим к пункту 3.2.7.

**3.3.5** Выбираем следующий элемент множества Y.

**3.3.6** Переход к пункту 3.3.3.

**3.3.7** Выбираем первый элемент множества Z.

**3.3.8** Выбираем первый элемент множества Y.

**3.3.9** Если выбранный элемент множества Y не равен выбранному элементу множества Z:

**3.3.9.1** Если выбранный элемент множества Y является последним:

**3.3.9.1.1** Переходим к пункту 3.3.11.

**3.3.9.2** Выбираем следующий элемент множества Y.

**3.3.9.3** Переход к пункту 3.3.9.

**3.3.10** Если выбранный элемент множества Y равен выбранному элементу множества Z:

**3.3.10.1** Если выбранный элемент множества Z является последним:

**3.3.10.1.1** Переходим к пункту 3.4.

**3.3.10.2** Выбираем следующий элемент множества Z.

**3.3.10.3** Переход к пункту 3.3.9.

**3.3.11** Записываем выбранный элемент множества Z во множество С2.

**3.3.12** Если выбранный элемент множества Z не является последним:

**3.3.12.1** Выбираем следующий элемент множества Z.

**3.3.12.2** Переход пункту 3.3.8.

**3.4** Выводим множества C1 на экран.

**3.5** Выводим множества C2 на экран.

**3.6** Выводим график C на экран.

**3.7** Переход к пункту 2.1.

**4. Операция пересечения:**

**4.1** Пересечение графиков F и P соответствий A и B:

**4.1.1** Программа создает пустой график D.

**4.1.2** Выбираем первый элемент графика P.

**4.1.3** Выбираем первый элемент графика F.

**4.1.4** Если первая компонента выбранного элемента графика P равна первой компоненте выбранного элемента графика F

**4.1.4.1** Если вторая компонента выбранного элемента графика P равна второй компоненте выбранного элемента графика F:

**4.1.4.1.1** Выбранный элемент графика P записывается в график D.

**4.1.5** Если выбранный элемент из графика P является последним

**4.1.5.1** Переход к пункту 4.1.8.

**4.1.6** Выбираем следующий элемент графика P.

**4.1.7** Переход к пункту 4.1.3.

**4.1.8** Если выбранный элемент графика F не является последним

**4.1.8.1** Выбираем следующий элемент графика F.

**4.1.8.2** Переход к пункту 4.1.4.

**4.2** Пересечение множеств X и W соответствий A и B:

**4.2.1** Создаём пустое множество D1.

**4.2.2** Выбираем первый элемент множества X.

**4.2.3** Выбираем первый элемент множества W.

**4.2.4** Если выбранный элемент из множества X равен выбранному элементу из множества W:

**4.2.4.1** Выбранный элемент из множества W записываем в множество D1.

**4.2.5** Если выбранный элемент из множества W – последний

**4.2.5.1** Переход к пункту 4.2.8.

**4.2.6** Выбираем следующий элемент множества W.

**4.2.7** Переход к пункту 4.2.4.

**4.2.8** Если выбранный элемент множества X – последний:

**4.2.8.1** Переход к пункту 4.3.

**4.2.9** Выбираем следующий элемент множества X.

**4.2.10** Переход к пункту 4.2.4.

**4.2** Пересечение множеств Y и Z соответствий A и B:

**4.3.1** Создаём пустое множество D2.

**4.3.2** Выбираем первый элемент множества Y.

**4.3.3** Выбираем первый элемент множества Z.

**4.3.4** Если выбранный элемент из множества Y равен выбранному элементу из множества Z:

**4.3.4.1** Выбранный элемент из множества Z записываем в множество D2.

**4.3.5** Если выбранный элемент из множества Z – последний

**4.3.5.1** Переход к пункту 4.3.8.

**4.3.6** Выбираем следующий элемент множества Z.

**4.3.7** Переход к пункту 4.3.4.

**4.3.8** Если выбранный элемент множества Y – последний:

**4.3.8.1** Переход к пункту 4.4.

**4.3.9** Выбираем следующий элемент множества Y.

**4.3.10** Переход к пункту 4.3.4.

**3.4** Выводим множества D1 на экран.

**3.5** Выводим множества D2 на экран.

**3.6** Выводим график D на экран.

**3.7** Переход к пункту 2.1.

**5. Операция разности А/В:**

**5.1** Разность графиков F и P соответствий A и B:

**5.1.1** Программа создает пустой график Е.

**5.1.2** Выбираем первый элемент графика F.

**5.1.3** Выбираем первый элемент графика P.

**5.1.4** Если первая компонента выбранного элемента графика F не равна первой компоненте выбранного элемента графика P:

**5.1.4.1** Если выбранный элемент из графика P является последним

**5.1.4.1.1** Выбранный элемент графика F записывается в график E.

**5.1.4.1.2** Переход к пункту 5.1.6.

**5.1.4.2** Если выбранный элемент графика P не является последним

**5.1.4.2.1** Выбираем следующий элемент графика P.

**5.1.4.2.2** Переход к пункту 5.1.4.

**5.1.5** Если вторая компонента выбранного элемента графика P не равна второй компоненте выбранного элемента графика F:

**5.1.5.1** Переход к пункту 5.1.4.1.

**5.1.6** Если выбранный элемент графика F не является последним

**5.1.6.1** Выбираем следующий элемент график F.

**5.1.6.2** Переход к пункту 5.1.3.

**5.2** Разность множеств X и W соответствий A и B:

**5.2.1** Создаём пустое множество Е1.

**5.2.2** Выбираем первый элемент множества X.

**5.2.3** Выбираем первый элемент множества W.

**5.2.4** Если выбранный элемент множества X не равен выбранному элементу множества W:

**5.2.4.1** Если выбранный элемент из множества W является последним

**5.2.4.1.1** Выбранный элемент множества X записывается в множество Е1.

**5.2.4.1.2** Переход к пункту 5.2.5.

**5.2.4.3** Выбираем следующий элемент множества W.

**5.2.4.4** Переход к пункту 5.2.4.

**5.2.5** Если выбранный элемент множества X не является последним

**5.2.5.1** Выбираем следующий элемент множества X.

**5.2.5.2** Переход к пункту 5.2.3.

**5.3** Разность множеств Y и Z соответствий A и B:

**5.3.1** Создаём пустое множество Е2.

**5.3.2** Выбираем первый элемент множества Y.

**5.3.3** Выбираем первый элемент множества Z.

**5.3.4** Если выбранный элемент множества Y не равен выбранному элементу множества Z:

**5.3.4.1** Если выбранный элемент из множества Z является последним

**5.3.4.1.1** Выбранный элемент множества Y записывается в множество Е2.

**5.3.4.1.2** Переход к пункту 5.3.5.

**5.3.4.3** Выбираем следующий элемент множества Z.

**5.3.4.4** Переход к пункту 5.3.4.

**5.3.5** Если выбранный элемент множества Y не является последним

**5.3.5.1** Выбираем следующий элемент множества Y.

**5.3.5.2** Переход к пункту 5.3.3.

**5.4** Выводим множества E1 на экран.

**5.5** Выводим множества E2 на экран.

**5.6** Выводим график E на экран.

**5.7** Переход к пункту 2.1.

**6. Операция разности B/A:**

**6.1** Разность графиков P и F соответствий B и A:

**6.1.1** Программа создает пустой график H.

**6.1.2** Выбираем первый элемент графика P.

**6.1.3** Выбираем первый элемент графика F.

**6.1.4** Если первая компонента выбранного элемента графика P не равна первой компоненте выбранного элемента графика F:

**6.1.4.1** Если выбранный элемент из графика F является последним

**6.1.4.1.1** Выбранный элемент графика P записывается в график H.

**6.1.4.1.2** Переход к пункту 6.1.6.

**6.1.4.2** Если выбранный элемент графика F не является последним

**6.1.4.2.1** Выбираем следующий элемент графика F.

**6.1.4.2.2** Переход к пункту 6.1.4.

**6.1.5** Если вторая компонента выбранного элемента графика F не равна второй компоненте выбранного элемента графика P:

**6.1.5.1** Переход к пункту 6.1.4.1.

**6.1.6** Если выбранный элемент графика P не является последним

**6.1.6.1** Выбираем следующий элемент график P.

**6.1.6.2** Переход к пункту 6.1.3.

**6.2** Разность множеств W и X соответствий B и A:

**6.2.1** Создаём пустое множество H1.

**6.2.2** Выбираем первый элемент множества W.

**6.2.3** Выбираем первый элемент множества X.

**6.2.4** Если выбранный элемент множества W не равен выбранному элементу множества X:

**6.2.4.1** Если выбранный элемент из множества X является последним

**6.2.4.1.1** Выбранный элемент множества W записывается в множество H1.

**6.2.4.1.2** Переход к пункту 6.2.5.

**6.2.4.3** Выбираем следующий элемент множества X.

**6.2.4.4** Переход к пункту 6.2.4.

**6.2.5** Если выбранный элемент множества W не является последним

**6.2.5.1** Выбираем следующий элемент множества W.

**6.2.5.2** Переход к пункту 6.2.3.

**6.3** Разность множеств Z и Y соответствий B и A:

**6.3.1** Создаём пустое множество H2.

**6.3.2** Выбираем первый элемент множества Z.

**6.3.3** Выбираем первый элемент множества Y.

**6.3.4** Если выбранный элемент множества Z не равен выбранному элементу множества Y:

**6.3.4.1** Если выбранный элемент из множества Y является последним

**6.3.4.1.1** Выбранный элемент множества Z записывается в множество H2.

**6.3.4.1.2** Переход к пункту 6.3.5.

**6.3.4.3** Выбираем следующий элемент множества Y.

**6.3.4.4** Переход к пункту 6.3.4.

**6.3.5** Если выбранный элемент множества Z не является последним

**6.3.5.1** Выбираем следующий элемент множества Z.

**6.3.5.2** Переход к пункту 6.3.3.

**6.4** Выводим множества H1 на экран.

**6.5** Выводим множества H2 на экран.

**6.6** Выводим график H на экран.

**6.7** Переход к пункту 2.1.

**7. Операция симметрической разности:**

**7.1** Симметрическая разность графиков F и P соответствий A и B:

**7.1.1** Программа создает пустой график K.

**7.1.2** Выбираем первый элемент графика F.

**7.1.3** Выбираем первый элемент графика P.

**7.1.4** Если первая компонента выбранного элемента графика F не равна первой компоненте выбранного элемента графика P:

**7.1.4.1** Если выбранный элемент из графика P является последним

**7.1.4.1.1** Выбранный элемент графика F записывается в график K.

**7.1.4.1.2** Переход к пункту 5.1.6.

**7.1.4.2** Если выбранный элемент графика P не является последним

**7.1.4.2.1** Выбираем следующий элемент графика P.

**7.1.4.2.2** Переход к пункту 5.1.4.

**7.1.5** Если вторая компонента выбранного элемента графика P не равна второй компоненте выбранного элемента графика F:

**7.1.5.1** Переход к пункту 5.1.4.1.

**7.1.6** Если выбранный элемент графика F не является последним

**7.1.6.1** Выбираем следующий элемент график F.

**7.1.6.2** Переход к пункту 5.1.3.

**7.1.7** Выбираем первый элемент графика P.

**7.1.8** Выбираем первый элемент графика F.

**7.1.9** Если первая компонента выбранного элемента графика P не равна первой компоненте выбранного элемента графика F:

**7.1.9.1** Если выбранный элемент из графика F является последним

**7.1.9.1.1** Выбранный элемент графика P записывается в график K.

**7.1.9.1.2** Переход к пункту 7.1.11.

**7.1.9.2** Если выбранный элемент графика F не является последним

**7.1.9.2.1** Выбираем следующий элемент графика F.

**7.1.9.2.2** Переход к пункту 7.1.9

**7.1.10** Если вторая компонента выбранного элемента графика F не равна второй компоненте выбранного элемента графика P:

**7.1.10.1** Переход к пункту 7.1.9.1.

**7.1.11** Если выбранный элемент графика P не является последним

**7.1.11.1** Выбираем следующий элемент график P.

**7.1.11.2** Переход к пункту 7.1.9.

**7.2** Симметрическая разность множеств X и W соответствий A и B

**7.2.1** Создаём пустое множество K­1.

**7.2.2** Выбираем первый элемент множества X.

**7.2.3** Выбираем первый элемент множества W.

**7.2.4** Если выбранный элемент множества X не равен выбранному элементу множества W:

**7.24.1** Если выбранный элемент из множества W является последним

**7.2.4.1.1** Выбранный элемент множества X записывается в множество K1.

**7.2.4.1.2** Переход к пункту 7.2.5.

**7.2.4.3** Выбираем следующий элемент множества W.

**7.2.4.4** Переход к пункту 7.2.4.

**7.2.5** Если выбранный элемент множества X не является последним

**7.2.5.1** Выбираем следующий элемент множества X.

**7.2.5.2** Переход к пункту 7.2.3.

**7.2.6** Выбираем первый элемент множества W.

**7.2.7** Выбираем первый элемент множества X.

**7.2.8** Если выбранный элемент множества W не равен выбранному элементу множества X:

**7.2.8.1** Если выбранный элемент из множества X является последним

**7.2.8.1.1** Выбранный элемент множества W записывается в множество K1.

**7.2.8.1.2** Переход к пункту 7.2.9.

**7.2.8.3** Выбираем следующий элемент множества X.

**7.2.8.4** Переход к пункту 7.2.8.

**7.2.9** Если выбранный элемент множества W не является последним

**7.2.9.1** Выбираем следующий элемент множества W.

**7.2.9.2** Переход к пункту 7.2.7.

**7.3** Симметрическая разность множеств X и W соответствий A и B

**7.3.1** Создаём пустое множество K2.

**7.3.2** Выбираем первый элемент множества Y.

**7.3.3** Выбираем первый элемент множества Z.

**7.3.4** Если выбранный элемент множества Y не равен выбранному элементу множества Z:

**7.3.4.1** Если выбранный элемент из множества Z является последним

**7.3.4.1.1** Выбранный элемент множества Y записывается в множество K2.

**7.3.4.1.2** Переход к пункту 7.3.5.

**7.3.4.3** Выбираем следующий элемент множества Z.

**7.3.4.4** Переход к пункту 7.3.4.

**7.3.5** Если выбранный элемент множества Y не является последним

**7.3.5.1** Выбираем следующий элемент множества Y.

**7.3.5.2** Переход к пункту 7.3.3.

**7.3.6** Выбираем первый элемент множества Z.

**7.3.7** Выбираем первый элемент множества Y.

**7.3.8** Если выбранный элемент множества Z не равен выбранному элементу множества Y:

**7.3.8.1** Если выбранный элемент из множества Y является последним

**7.3.8.1.1** Выбранный элемент множества Z записывается в множество K2.

**7.3.8.1.2** Переход к пункту 7.3.9.

**7.3.8.3** Выбираем следующий элемент множества Y.

**7.3.8.4** Переход к пункту 7.3.8.

**7.3.9** Если выбранный элемент множества Z не является последним

**7.3.9.1** Выбираем следующий элемент множества Z.

**7.3.9.2** Переход к пункту 7.3.7.

**7.4** Выводим множества K1 на экран.

**7.5** Выводим множества K2 на экран.

**7.6** Выводим график K на экран.

**7.7** Переход к пункту 2.1.

**8. Операция композиции над А&В:**

**8.1** Программа создает пустой график R.

**8.2** Выбираем первый элемент графика F.

**8.3** Выбираем первый элемент графика P.

**8.4** Если вторая компонента выбранного элемента графика F равна первой компоненте выбранного элемента графика P:

**8.4.1** Создаем кортеж r.

**8.4.2** Записываем первую компоненту выбранного элемента графика F на место первой компоненты кортежа r.

**8.4.3** Записываем вторую компоненту выбранного элемента графика P на место второй компоненты кортежа r.

**8.4.4** Проверка нет ли такого же кортежа в графике R:

**8.4.4.1** Выбираем первый элемент графика R.

**8.4.4.2** Если первая компонента выбранного элемента графика R не равна первой компоненте кортежа r:

**8.4.4.2.1** Если выбранный элемент графика R не является последним:

**8.4.4.2.1.1** Выбираем следующий элемент графика R.

**8.4.4.2.2.2** Переходим к пункту 8.4.4.2.

**8.4.4.2.1** Переходим к пункту 8.4.5.

**8.4.4.3** Если вторая компонента выбранного элемента графика R не равна второй компоненте кортежа r:

**8.4.4.2.2** Переходим к пункту 8.4.4.2

**8.4.4.4** Переход к пункту 8.4.6

**8.4.5** Записываем кортеж r графика R.

**8.4.6** Если элемент графика P не является последним

**8.4.6.1** Выбираем следующий элемент графика P

**8.4.6.2** Переход к пункту 8.4

**8.5** Если выбранный элемент графика F не является последним:

**8.5.1** Выбираем следующий элемент графика F.

**8.5.2** Переход к пункту 8.3

**8.6** Выводим множества Xна экран.

**8.7** Выводим множества Z на экран.

**8.8** Выводим график R на экран.

**8.9** Переход к пункту 2.1.

**9. Операция композиции над В&A:**

**9.1** Программа создает пустой график I.

**9.2** Выбираем первый элемент графика P.

**9.3** Выбираем первый элемент графика F.

**9.4** Если вторая компонента выбранного элемента графика P равна первой компоненте выбранного элемента графика F:

**9.4.1** Создаем кортеж i.

**9.4.2** Записываем первую компоненту выбранного элемента графика P на место первой компоненты кортежа i.

**9.4.3** Записываем вторую компоненту выбранного элемента графика F на место второй компоненты кортежа i.

**9.4.4** Проверка нет ли такого же кортежа в графике I:

**9.4.4.1** Выбираем первый элемент графика I.

**9.4.4.2** Если первая компонента выбранного элемента графика I не равна первой компоненте кортежа i

**9.4.4.2.1** Если выбранный элемент графика I не является последним

**9.4.4.2.1.1** Выбираем следующий элемент графика I

**9.4.4.2.2.2** Переходим к пункту 9.4.4.2

**9.4.4.2.1** Переходим к пункту 9.4.5

**9.4.4.3** Если вторая компонента выбранного элемента графика I не равна второй компоненте кортежа i:

**9.4.4.2.2** Переходим к пункту 9.4.4.2

**9.4.4.4** Переход к пункту 9.4.6

**9.4.5** Записываем кортеж i графика I.

**9.4.6** Если элемент графика F не является последним

**9.4.6.1** Выбираем следующий элемент графика F.

**9.4.6.2** Переход к пункту 9.4

**9.5** Если выбранный элемент графика B не является последним:

**9.5.1** Выбираем следующий элемент графика P.

**9.5.2** Переход к пункту 9.3

**9.6** Выводим множества Wна экран.

**9.7** Выводим множества Y на экран.

**9.8** Выводим график I на экран.

**9.9** Переход к пункту 2.1.

**10. Операция инверсии соответствия A:**

**10.1** Программа создает пустой график J.

**10.2** Выбираем первый элемент графика F.

**10.3** Создаем кортеж j длины 2.

**10.4** Записываем первую компоненту выбранного элемента графика F на место второй компоненты кортежа j.

**10.5** Записываем вторую компоненту выбранного элемента графика F на место первой компоненты кортежа j.

**10.6** Записываем кортеж j в график J.

**10.7** Если выбранный элемент графика F не является последним

**10.7.1** Выбираем следующий элемент графика F.

**10.7.2** Переход к пункту 10.3.

**10.8** Выводим множества Yна экран.

**10.9** Выводим множества X на экран.

**10.10** Выводим график J на экран.

**10.11** Переходим к пункту 2.1.

**11. Операция инверсии соответствия B:**

**11.1** Программа создает пустой график Q.

**11.2** Выбираем первый элемент графика P.

**11.3** Создаем кортеж q длины 2.

**11.4** Записываем первую компоненту выбранного элемента графика P на место второй компоненты кортежа q.

**11.5** Записываем вторую компоненту выбранного элемента графика P на место первой компоненты кортежа q.

**11.6** Записываем кортеж q в график Q.

**11.7** Если выбранный элемент графика P не является последним

**11.7.1** Выбираем следующий элемент графика P.

**11.7.2** Переход к пункту 10.3.

**11.8** Выводим множества Zна экран.

**11.9** Выводим множества W на экран.

**11.10** Выводим график Q на экран.

**11.11** Переходим к пункту 2.1.

**12. Операция образа множества N при соответствии A:**

**12.1** Пользователь вводит мощность множества N.

**12.2** Пользователь вводит элементы множества N.

**12.3** Создаём пустое множество N1

**12.4** Выбираем первый элемент графика F.

**12.5** Выбираем первый элемент множества N.

**12.6** Если первая компонента выбранного элемента графика F равна выбранному элементу множества N:

**12.6.1** Записываем вторую компоненту выбранного элемента графика F в множество N1

**12.6.2** Переходим к пункту 12.8.

**12.7** Если выбранный элемент множества N не последний:

**12.7.1** Выбираем следующий элемент множества N.

**12.7.2** Переход к пункту 12.6.

**12.8** Если выбранный элемент множества F не последний:

**12.8.1** Выбираем следующий элемент множества F.

**12.8.2** Переход к пункту 12.5.

**12.9** Множество N1 выводится на экран (множество N1 является образом множества N при соответствии А).

**12.10** Переходим к пункту 2.1.

**13. Операция образа множества N при соответствии B:**

**13.1** Пользователь вводит мощность множества N.

**13.2** Пользователь вводит элементы множества N.

**13.3** Создаём пустое множество N1

**13.4** Выбираем первый элемент графика P.

**13.5** Выбираем первый элемент множества N.

**13.6** Если первая компонента выбранного элемента графика P равна выбранному элементу множества N:

**13.6.1** Записываем вторую компоненту выбранного элемента графика P в множество N1

**13.6.2** Переходим к пункту 13.8.

**13.7** Если выбранный элемент множества N не последний:

**13.7.1** Выбираем следующий элемент множества N.

**13.7.2** Переход к пункту 13.6.

**13.8** Если выбранный элемент множества P не последний:

**13.8.1** Выбираем следующий элемент множества P.

**13.8.2** Переход к пункту 13.5.

**13.9** Множество N1 выводится на экран (множество N1 является образом множества N при соответствии B).

**13.10** Переходим к пункту 2.1.

**14. Операция прообраза множества M при соответствии A:**

**14.1** Пользователь вводит мощность множества M.

**14.2** Пользователь вводит элементы множества M.

**14.3** Создаём пустое множество M1

**14.4** Выбираем первый элемент графика F.

**14.5** Выбираем первый элемент множества M.

**14.6** Если вторая компонента выбранного элемента графика F равна выбранному элементу множества M, то:

**14.6.1** Записываем первую компоненту выбранного элемента графика F в множество M1

**14.6.2** Переходим к пункту 14.8.

**14.7** Если выбранный элемент множества M не последний:

**14.7.1** Выбираем следующий элемент множества M.

**14.7.2** Переход к пункту 14.6.

**14.8** Если выбранный элемент графика F не последний:

**14.8.1** Выбираем следующий элемент множества F.

**14.8.2** Переход к пункту 14.5.

**14.9** Множество M1 выводится на экран (множество M1 является прообразом множества M при соответствии A).

**14.10** Переходим к пункту 2.1.

**15. Операция прообраза множества M при соответствии B:**

**15.1** Пользователь вводит мощность множества M.

**15.2** Пользователь вводит элементы множества M.

**15.3** Создаём пустое множество M1

**15.4** Выбираем первый элемент графика P.

**15.5** Выбираем первый элемент множества M.

**15.6** Если вторая компонента выбранного элемента графика P равна выбранному элементу множества M, то:

**15.6.1** Записываем первую компоненту выбранного элемента графика P в множество M1

**15.6.2** Переходим к пункту 15.8.

**15.7** Если выбранный элемент множества M не последний:

**15.7.1** Выбираем следующий элемент множества M.

**15.7.2** Переход к пункту 15.6.

**15.8** Если выбранный элемент графика P не последний:

**15.8.1** Выбираем следующий элемент множества P.

**15.8.2** Переход к пункту 15.5.

**15.9** Множество M1 выводится на экран (множество M1 является прообразом множества M при соответствии B).

**15.10** Переходим к пункту 2.1.

**16. Операция сужения соответствия A на S:**

**16.1** Пользователь вводит мощность множества S.

**16.2** Пользователь вводит элементы множества S.

**16.3** Создаём пустой график O.

**16.4** Выбираем первый элемент графика F.

**16.5** Выбираем первый элемент множества S.

**16.6** Если первая компонента выбранного элемента графика F равна выбранному элементу множества S:

**16.6.1** Выбранный элемент графика F записывается в график O:

**16.7** Если выбранный элемент множества S не последний,

**16.7.1** Выбираем следующий элемент множества S.

**16.7.2** Переходим к пункту 16.6.

**16.8** Если выбранный элемент графика F не последний:

**16.8.1** Выбираем следующий элемент графика F.

**16.8.1** Переходим к пункту 16.5.

**16.9** Выводим множества Sна экран.

**16.10** Выводим множества Y на экран.

**16.11** Выводим график O на экран.

**16.12** Переходим к пункту 2.1.

**17. Операция сужения соответствия B на S:**

**17.1** Пользователь вводит мощность множества S.

**17.2** Пользователь вводит элементы множества S.

**17.3** Создаём пустой график T.

**17.4** Выбираем первый элемент графика P.

**17.5** Выбираем первый элемент множества S.

**17.6** Если первая компонента выбранного элемента графика P равна выбранному элементу множества S:

**17.6.1** Выбранный элемент графика P записывается в график T:

**17.7** Если выбранный элемент множества S не последний,

**17.7.1** Выбираем следующий элемент множества S.

**17.7.2** Переходим к пункту 17.6.

**17.8** Если выбранный элемент графика P не последний:

**17.8.1** Выбираем следующий элемент графика P.

**17.8.1** Переходим к пункту 17.5.

**17.9** Выводим множества Sна экран.

**17.10** Выводим множества Z на экран.

**17.11** Выводим график T на экран.

**17.12** Переходим к пункту 2.1.

**18. Продолжение соответствия A:**

**18.1** Создаем пустой график L.

**18.2** Выбираем первый элемент множества X.

**18.3** Выбираем первый элемент множества Y.

**18.4** Создаем кортеж l длины 2.

**18.5** Записываем выбранный элемента множества X на место первой компоненты кортежа l.

**18.6** Записываем выбранный элемента множества Y на место второй компоненты кортежа l.

**18.7** Кортеж l записывается в график L.

**18.8** Если выбранный элемент Y не последний:

**18.8.1** Выбираем следующий элемент множества Y.

**18.8.2** Переход к пункту 18.4.

**18.9** Если выбранный элемент X не последний:

**18.9.1** Выбираем следующий элемент множества X.

**18.9.2** Переход к пункту 18.3.

**18.10** Выводим множества Xна экран.

**18.11** Выводим множества Y на экран.

**18.12** Выводим график L на экран.

**18.13** Переходим к пункту 2.1.

**19. Продолжение соответствия B:**

**19.1** Создаем пустой график U.

**19.2** Выбираем первый элемент множества W.

**19.3** Выбираем первый элемент множества Z.

**19.4** Создаем кортеж u длины 2.

**19.5** Записываем выбранный элемента множества W на место первой компоненты кортежа u.

**19.6** Записываем выбранный элемента множества Z на место второй компоненты кортежа u.

**19.7** Кортеж u записывается в график U.

**19.8** Если выбранный элемент Z не последний:

**19.8.1** Выбираем следующий элемент множества Z.

**19.8.2** Переход к пункту 18.4.

**19.9** Если выбранный элемент W не последний:

**19.9.1** Выбираем следующий элемент множества W.

**19.9.2** Переход к пункту 18.3.

**19.10** Выводим множества Wна экран.

**19.11** Выводим множества Z на экран.

**19.12** Выводим график U на экран.

**19.13** Переходим к пункту 2.1.

**20. Выход из программы**