Федеральное агентство по образованию

Государственное образовательное учреждение высшего профессионального образования

Вятский Государственный Гуманитарный Университет

Математический факультет

Кафедра алгебры и геометрии

Выпускная квалификационная работа

Восьмиэлементные ассоциативные кольца

Выполнил: студент V курса математического факультета Касьянов А.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Научный руководитель: д.ф.-м.н., профессор, зав. кафедры алгебры и геометрии

Вечтомов Е.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Рецензент: к.ф.-м.н., доцент

Чермных В.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Допущен к защите в ГАК

Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вечтомов Е.М.

« »\_\_\_\_\_\_\_\_\_\_\_

Декан факультета\_\_\_\_\_\_\_\_\_\_\_\_\_ Варанкина В.И.

« »\_\_\_\_\_\_\_\_\_\_\_

Киров 2005

Содержание

Введение. 3

§1. Абелевы группы по сложению. 5

§2. Кольца, образованные аддитивной группой . 8



§3. Кольца, образованные аддитивной группой . 10



Библиографический список. 11

Приложение 12

**Введение**

Понятие кольца появилось в математике в конце XIX века. Первыми примерами ассоциативных колец были числовые кольца, т.е. подкольца поля комплексных чисел и кольца классов вычетов целых чисел. Как самостоятельная область алгебры, теория ассоциативных колец оформилась к началу XX века. Из этой теории выделились в самостоятельные области алгебры теории коммутативных колец, тел, алгебр.

Дадим основные понятия, которыми мы будем пользоваться в дальнейшем.

Аддитивной абелевой группой  называется алгебра с бинарной операцией + (сложение), удовлетворяющей следующим аксиомам:

1. сложение ассоциативно, т.е. ;
2. в *G* существует нейтральный элемент 0 (ноль) такой, что ;
3. в *G* для любого элемента существует противоположный элемент, т.е. ;
4. Сложение коммутативно: ;

Мультипликативной полугруппой  называется алгебра с бинарной операцией  (умножение), удовлетворяющая закону ассоциативности: ; (в дальнейшем мы будем писать *ab*, и иметь ввиду произведение элементов *a* и *b* в *G*.

Кольцом называется алгебра  с двумя бинарными операциями сложения и умножения, которые удовлетворяют следующим условиям:

1. относительно сложения кольцо является абелевой группой;
2. относительно умножения – это полугруппа;
3. выполняются законы дистрибутивности:  и ;

Данная дипломная работа представляет собой изложение методики изучения определенных конечных алгебр. В настоящее время изучение конечных алгебр производится с помощью компьютера. Задача данной дипломной работы состоит в отыскании всех восьмиэлементных ассоциативных колец. Для этого мы сначала находим все абелевы группы из восьми элементов, а затем для каждой такой группы строим соответствующие полугруппы по умножению, и получаем искомые кольца.

Всего абелевых групп по сложению, с точностью до изоморфизма, будет три: , , и . Для группы , кольца находятся вручную. Для нахождения колец по остальным двум группам нужно использовать компьютер, так как количество всевозможных полугрупп по умножению будет велико и, соответственно, отбор восьмиэлементных колец без привлечения вычислительной машины займет очень много времени.



В результате получаем, что всего существует с точностью до изоморфизма 392 восьмиэлиментных кольца.

Для работы с абелевыми группами и полугруппами по умножению, а также для представления их в компьютере, будем пользоваться таблицами Кэли. Таблицей Кэли называется квадратная таблица произвольной алгебры, задаваемая для определённой бинарной операции. Заглавная строка таблицы заполняется в некотором порядке символами, обозначающими различные элементы, теми же символами и в том же порядке заполняется главный столбец. Если алгебра обладает нейтральным элементом, то этот элемент, как правило, помещается на первом месте. Если на i-м месте в заглавном столбце стоит символ ai, и на j-м месте в заглавной строке – символ aj, то на пересечении i-ой строки и j-го столбца записывается символ, обозначающий результат операции элементов ai и aj. В нашем случае мы не будем писать заглавные строки таблицы, подразумевая, что элементы пронумерованы числами от 0 до 7. Мы будем использовать таблицы Кэли без заглавных строки и столбца.

**§1. Абелевы группы по сложению**

Как уже было сказано выше, всего восьмиэлементных аддитивных абелевых групп с точностью до изоморфизма три: , , и .



Представим каждую из таких групп в виде таблиц Кэли. Для группы элементы представим числами от 0 до 7. Элементы для групп и обозначим следующим образом:



|  |  |
| --- | --- |
|  |  |
| 0≡ (0,0)  1≡ (1,0)  2≡ (2,0)  3≡ (3,0)  4≡ (0,1)  5≡ (1,1)  6≡ (2,1)  7≡ (3,1) | 0≡ (0,0,0)  1≡ (1,0,0)  2≡ (0,1,0)  3≡ (0,0,1)  4≡ (1,1,0)  5≡ (1,0,1)  6≡ (0,1,1)  7≡ (1,1,1) |

Таким образом, группы будут иметь следующий вид:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 0 1 2 3 4 5 6 7  1 2 3 4 5 6 7 0  2 3 4 5 6 7 0 1  3 4 5 6 7 0 1 2  4 5 6 7 0 1 2 3  5 6 7 0 1 2 3 4  6 7 0 1 2 3 4 5  7 0 1 2 3 4 5 6 | 0 1 2 3 4 5 6 7  1 2 3 0 5 6 7 4  2 3 0 1 6 7 4 5  3 0 1 2 7 4 5 6  4 5 6 7 0 1 2 3  5 6 7 4 1 2 3 0  6 7 4 5 2 3 0 1  7 4 5 6 3 0 1 2 | 0 1 2 3 4 5 6 7  1 0 4 5 2 3 7 6  2 4 0 6 1 7 3 5  3 5 6 0 7 1 2 4  4 2 1 7 0 6 5 3  5 3 7 1 6 0 4 2  6 7 3 2 5 4 0 1  7 6 5 4 3 2 1 0 |

Затем, для каждой такой группы, мы будем строить полугруппы по умножению, пользуясь также таблицами Кэли. Для группы таких полугрупп будет всего 8, так как нам достаточно определить чему равно произведение 1•1. На место этого произведения мы можем поставить один из 8 элементов (от 0 до 7), а все остальные элементы будут определяться однозначно, согласно дистрибутивному закону. Ассоциативность умножения будет выполнятся, так как умножение сводится к сложению. Кроме того, умножение будет коммутативно. Таким образом, после вычеркивания изоморфных, мы получим 4 кольца с абелевой группой по сложению :



|  |  |  |  |
| --- | --- | --- | --- |
| 0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0  0 1 2 3 4 5 6 7  0 2 4 6 0 2 4 6  0 3 6 1 4 7 2 5  0 4 0 4 0 4 0 4  0 5 2 7 4 1 6 3  0 6 4 2 0 6 4 2  0 7 6 5 4 3 2 1 | 0 0 0 0 0 0 0 0  0 2 4 6 0 2 4 6  0 4 0 4 0 4 0 4  0 6 4 2 0 6 4 2  0 0 0 0 0 0 0 0  0 2 4 6 0 2 4 6  0 4 0 4 0 4 0 4  0 6 4 2 0 6 4 2 | 0 0 0 0 0 0 0 0  0 4 0 4 0 4 0 4  0 0 0 0 0 0 0 0  0 4 0 4 0 4 0 4  0 0 0 0 0 0 0 0  0 4 0 4 0 4 0 4  0 0 0 0 0 0 0 0  0 4 0 4 0 4 0 4 |

В случае с группой в полугруппе по умножению уже будет 4 независимых произведения, т.е. это такие элементы в таблице Кэли для полугруппы по умножению, на которые мы можем поставить любой из элементов от 0 до 7. Соответственно всего различных колец без учета ассоциативности и вычеркивания изоморфных будет 84. Проверка ассоциативности умножения и выделение изоморфных колец осуществляется программным способом. Чтобы найти кольцо, изоморфное данному, нужно сначала найти все автоморфизмы абелевой группы по сложению, а потом этими преобразованиями подействовать на полугруппы по умножению, и соответственно получившиеся одинаковые кольца вычеркнуть. Автоморфизмы группы будем искать вручную. Выделим в данной группе элементы 1, 3, 5, 7 – элементы четвертого порядка. Остальные элементы будут выражаться через них: 4=1+7, 6=1+5, 2=4+6 – это элементы второго порядка. Пары элементов 1≡(1,0), 3≡(3,0) и 5≡(1,1), 7≡(3,1) – противоположные друг другу элементы. Нам достаточно посмотреть, как будут вести себя элементы четвертого порядка при автоморфизмах. Это будут 6 взаимнооднозначных отображений, в том числе и тождественное, которые переводят данную группу в себя:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1→3  3→1  5→7  7→5  4→4  6→6  2→2 | 1→1  3→3  5→7  7→5  4→6  6→4  2→2 | 1→3  3→1  5→5  7→7  4→6  6→4  2→2 | 1→5  5→1  3→7  7→3  4→4  6→6  2→2 | 1→7  7→1  3→5  5→3  4→4  6→6  2→2 |

Колец с абелевой группой по сложению будет 89, так как независимых элементов в полугруппе по умножению будет 9. Нужные нам кольца, мы будем искать аналогично предыдущему случаю. Только автоморфизмы данной группы мы будем искать другим способом. Заметим, что группа является трехмерным векторным пространством над полем *Z2* и базисом группы мы назовем любой ее базис, как векторное пространство над *Z2.* Данная группа задается некоторыми тремя элементами – базисом группы, а остальные элементы выражаются через данный базис. К примеру начальный базис: (1,2,3). Соответственно: 4=1+2, 5=1+3, 6=2+3, 7=1+2+3. Таким образом, количество всех базисов – это количество всевозможных упорядоченных троек из данных семи элементов, за исключением таких троек, которые базис не образуют, на пример: (1,2,4), так как 4=1+2. Количество всех базисов – это и будет количество всех автоморфизмов данной группы. Сначала найдем всевозможные неупорядоченные тройки из 7 элементов, которые образуют базис. Получим 28 таких троек. Чтобы найти все базисы, нужно каждую найденную тройку, упорядочить, т.е. 28•6. Таким образом получаем 168 различных базисов.



**§2. Кольца, образованные аддитивной группой**



Для нахождения колец, с абелевой группой по сложению мы будем использовать программу на языке Pascal (Приложение 1). Принцип работы программы следующий:



1) Находим все полугруппы по умножению, так, чтобы выполнялась дистрибутивность;

2) Проверяем каждую полугруппу на ассоциативность, и, соответственно, ненужные вычеркиваем;

3) Находим изоморфные кольца следующим образом: а) берем первую полугруппу по умножению и действуем на нее автоморфизмами для аддитивной абелевой группы; б) находим получившиеся полугруппы среди остальных, и вычеркиваем их из общего списка; в) затем берем следующую не вычеркнутую полугруппу и проделываем операции а)-б).

Поясним пункт а). Пусть *f* – автоморфизм группы . Берем мультипликативную полугруппу <A,•> соответствующего кольца, представленную таблицей Кэли. Тогда через <*f*A,\*> обозначим мультипликативную полугруппу на , полученную следующим образом. Для любых *a,b∈* полагаем *a*\**b=f(f –1(a)• f –1(b))*. При этом кольца <A,•,+> и <*f*A,\*,+> изоморфны.



Таким образом, мы получим 33 кольца с абелевой группой по сложению и мультипликативными полугруппами:



0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 1 1 1 1

0 0 0 0 1 1 1 1

0 0 0 0 1 1 1 1

0 0 0 0 1 1 1 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 0 0 0 2 2 2 2

0 0 0 0 2 2 2 2

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 4 4 4 4

0 0 0 0 4 4 4 4

0 0 0 0 4 4 4 4

0 0 0 0 4 4 4 4

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 5 5 5 5

0 0 0 0 5 5 5 5

0 0 0 0 5 5 5 5

0 0 0 0 5 5 5 5

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7

0 1 2 3 4 5 6 7

0 1 2 3 4 5 6 7

0 1 2 3 4 5 6 7

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 2 0 2 0 2 0 2

0 2 0 2 0 2 0 2

0 2 0 2 0 2 0 2

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 2 0 2 2 0 2 0

0 2 0 2 2 0 2 0

0 2 0 2 2 0 2 0

0 2 0 2 2 0 2 0

0 0 0 0 0 0 0 0

0 0 0 0 1 1 1 1

0 0 0 0 2 2 2 2

0 0 0 0 3 3 3 3

0 0 0 0 4 4 4 4

0 0 0 0 5 5 5 5

0 0 0 0 6 6 6 6

0 0 0 0 7 7 7 7

0 0 0 0 0 0 0 0

0 0 0 0 1 1 1 1

0 0 0 0 2 2 2 2

0 0 0 0 3 3 3 3

0 1 2 3 4 5 6 7

0 1 2 3 5 6 7 4

0 1 2 3 6 7 4 5

0 1 2 3 7 4 5 6

0 0 0 0 0 0 0 0

0 0 0 0 1 1 1 1

0 0 0 0 2 2 2 2

0 0 0 0 3 3 3 3

0 1 2 3 5 6 7 4

0 1 2 3 6 7 4 5

0 1 2 3 7 4 5 6

0 1 2 3 4 5 6 7

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 2 0 2 0 2 0 2

0 2 0 2 2 0 2 0

0 2 0 2 0 2 0 2

0 2 0 2 2 0 2 0

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 2 0 2 1 3 1 3

0 2 0 2 3 1 3 1

0 2 0 2 1 3 1 3

0 2 0 2 3 1 3 1

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 0 0 0 2 2 2 2

0 2 0 2 2 0 2 0

0 2 0 2 0 2 0 2

0 2 0 2 2 0 2 0

0 2 0 2 0 2 0 2

0 0 0 0 0 0 0 0

0 1 2 3 0 1 2 3

0 2 0 2 0 2 0 2

0 3 2 1 0 3 2 1

0 0 0 0 0 0 0 0

0 1 2 3 0 1 2 3

0 2 0 2 0 2 0 2

0 3 2 1 0 3 2 1

0 0 0 0 0 0 0 0

0 1 2 3 0 1 2 3

0 2 0 2 0 2 0 2

0 3 2 1 0 3 2 1

0 0 0 0 4 4 4 4

0 1 2 3 4 5 6 7

0 2 0 2 4 6 4 6

0 3 2 1 4 7 6 5

0 0 0 0 0 0 0 0

0 1 2 3 0 1 2 3

0 2 0 2 0 2 0 2

0 3 2 1 0 3 2 1

0 4 0 4 0 4 0 4

0 5 2 7 0 5 2 7

0 6 0 6 0 6 0 6

0 7 2 5 0 7 2 5

0 0 0 0 0 0 0 0

0 1 2 3 1 2 3 0

0 2 0 2 2 0 2 0

0 3 2 1 3 2 1 0

0 1 2 3 1 2 3 0

0 2 0 2 2 0 2 0

0 3 2 1 3 2 1 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 1 2 3 1 2 3 0

0 2 0 2 2 0 2 0

0 3 2 1 3 2 1 0

0 1 2 3 4 5 6 7

0 2 0 2 5 7 5 7

0 3 2 1 6 5 4 7

0 0 0 0 7 7 7 7

0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7

0 2 0 2 0 2 0 2

0 3 2 1 4 7 6 5

0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7

0 2 0 2 0 2 0 2

0 3 2 1 4 7 6 5

0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7

0 2 0 2 0 2 0 2

0 3 2 1 4 7 6 5

0 4 0 4 0 4 0 4

0 5 2 7 4 1 6 3

0 6 0 6 0 6 0 6

0 7 2 5 4 3 6 1

0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7

0 2 0 2 0 2 0 2

0 3 2 1 4 7 6 5

0 4 0 4 2 6 2 6

0 5 2 7 6 3 4 1

0 6 0 6 2 4 2 4

0 7 2 5 6 1 4 3

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 0 0 0 4 4 4 4

0 2 0 2 4 6 4 6

0 0 0 0 4 4 4 4

0 2 0 2 4 6 4 6

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 0 0 0 0 0 0 0

0 2 0 2 0 2 0 2

0 2 0 2 2 0 2 0

0 0 0 0 2 2 2 2

0 2 0 2 2 0 2 0

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 2 0 2 1 3 1 3

0 0 0 0 2 2 2 2

0 2 0 2 3 1 3 1

0 1 2 3 4 5 6 7

0 3 2 1 5 4 7 6

0 1 2 3 6 7 4 5

0 3 2 1 7 6 5 4

0 0 0 0 0 0 0 0

0 2 0 2 2 0 2 0

0 0 0 0 0 0 0 0

0 2 0 2 2 0 2 0

0 2 0 2 0 2 0 2

0 0 0 0 2 2 2 2

0 2 0 2 0 2 0 2

0 0 0 0 2 2 2 2

0 0 0 0 0 0 0 0

0 2 0 2 2 0 2 0

0 0 0 0 0 0 0 0

0 2 0 2 2 0 2 0

0 2 0 2 5 7 5 7

0 0 0 0 7 7 7 7

0 2 0 2 5 7 5 7

0 0 0 0 7 7 7 7

0 0 0 0 0 0 0 0

0 2 0 2 3 1 3 1

0 0 0 0 2 2 2 2

0 2 0 2 1 3 1 3

0 3 2 1 5 4 7 6

0 1 2 3 4 5 6 7

0 3 2 1 7 6 5 4

0 1 2 3 6 7 4 5

0 0 0 0 0 0 0 0

0 4 0 4 0 4 0 4

0 0 0 0 0 0 0 0

0 4 0 4 0 4 0 4

0 0 0 0 0 0 0 0

0 4 0 4 0 4 0 4

0 0 0 0 0 0 0 0

0 4 0 4 0 4 0 4

0 0 0 0 0 0 0 0

0 4 0 4 2 6 2 6

0 0 0 0 0 0 0 0

0 4 0 4 2 6 2 6

0 2 0 2 0 2 0 2

0 6 0 6 2 4 2 4

0 2 0 2 0 2 0 2

0 6 0 6 2 4 2 4

0 0 0 0 0 0 0 0

0 4 0 4 6 2 6 2

0 0 0 0 0 0 0 0

0 4 0 4 6 2 6 2

0 6 0 6 6 0 6 0

0 2 0 2 0 2 0 2

0 6 0 6 6 0 6 0

0 2 0 2 0 2 0 2

**§3. Кольца, образованные аддитивной группой**



Для нахождения колец с данной группой по сложению использовалась программа на языке Pascal (Приложение 2). Принцип действия данной программы, аналогичен принципу, описанному в предыдущем параграфе. Добавляется только пункт по нахождению всех базисов данной аддитивной группы.

Всего колец с аддитивной группой будет 355. Выпишем некоторые из них:



0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 0 1 0 1 1 1

0 0 0 1 0 1 1 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 3 0 3 3 3

0 0 0 0 0 0 0 0

0 0 0 3 0 3 3 3

0 0 0 3 0 3 3 3

0 0 0 3 0 3 3 3

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 1 0 1 0 1 1

0 0 0 0 0 0 0 0

0 0 1 0 1 0 1 1

0 0 1 0 1 0 1 1

0 0 1 0 1 0 1 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 1 1 1 1 0 0

0 0 0 0 0 0 0 0

0 0 1 1 1 1 0 0

0 0 1 1 1 1 0 0

0 0 1 1 1 1 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 2 3 2 3 6 6

0 0 0 0 0 0 0 0

0 0 2 3 2 3 6 6

0 0 2 3 2 3 6 6

0 0 2 3 2 3 6 6

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 2 5 2 5 7 7

0 0 0 0 0 0 0 0

0 0 2 5 2 5 7 7

0 0 2 5 2 5 7 7

0 0 2 5 2 5 7 7

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 1 1 3 0 5 5 3

0 0 0 0 0 0 0 0

0 1 1 3 0 5 5 3

0 1 1 3 0 5 5 3

0 1 1 3 0 5 5 3

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 1 1 7 0 6 6 7

0 0 0 0 0 0 0 0

0 1 1 7 0 6 6 7

0 1 1 7 0 6 6 7

0 1 1 7 0 6 6 7

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7

0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7

0 1 2 3 4 5 6 7

0 1 2 3 4 5 6 7

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 4 0 3 4 7 3 7

0 0 0 0 0 0 0 0

0 4 0 3 4 7 3 7

0 4 0 3 4 7 3 7

0 4 0 3 4 7 3 7

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 4 0 6 4 5 6 5

0 0 0 0 0 0 0 0

0 4 0 6 4 5 6 5

0 4 0 6 4 5 6 5

0 4 0 6 4 5 6 5

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 0 1 0 1 1 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 0 1 0 1 1 1

0 0 0 1 0 1 1 1

0 0 0 1 0 1 1 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 1 0 1 0 1 1

0 0 0 1 0 1 1 1

0 0 1 0 1 0 1 1

0 0 1 1 1 1 0 0

0 0 1 1 1 1 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 1 1 1 1 0 0

0 0 0 1 0 1 1 1

0 0 1 1 1 1 0 0

0 0 1 0 1 0 1 1

0 0 1 0 1 0 1 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 1 0 1 1 1

0 0 1 2 1 2 4 4

0 0 0 1 0 1 1 1

0 0 1 2 1 2 4 4

0 0 1 4 1 4 2 2

0 0 1 4 1 4 2 2

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 2 0 2 2 2

0 0 0 3 0 3 3 3

0 0 0 2 0 2 2 2

0 0 0 3 0 3 3 3

0 0 0 6 0 6 6 6

0 0 0 6 0 6 6 6

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 2 0 2 2 2

0 0 0 5 0 5 5 5

0 0 0 2 0 2 2 2

0 0 0 5 0 5 5 5

0 0 0 7 0 7 7 7

0 0 0 7 0 7 7 7

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 2 0 2 2 2

0 0 2 3 2 3 6 6

0 0 0 2 0 2 2 2

0 0 2 3 2 3 6 6

0 0 2 6 2 6 3 3

0 0 2 6 2 6 3 3

**Библиографический список.**

1. Кострикин А.И. Введение в алгебру. – М.: Наука, 1997.
2. Скорняков Л.А. Элементы алгебры. – М.: Наука, 1986.

**Приложение**

**1. Текст программы поиска колец, имеющих аддитивную группу .**



Программа 1. Формирование мультипликативных полугрупп:

program form;

uses crt;

type mas=array [0..7,0..7] of integer;

dmas=array [1..7,1..7] of integer;

var a:mas;

b:dmas;

f:text;

procedure sc(var a:mas);

var i,j:integer;

begin

assign(f,'a:\mat.txt');

reset(f);

for i:=0 to 7 do begin

for j:=0 to 7 do read(f,a[i,j]);

readln(f);

end;

close(f);

end;

procedure zap1(k1,k2,k3,k4:integer;a:mas; var b:dmas);

var t2,t3,t5,t6,t7:integer;

begin

b[1,1]:=k1; b[1,4]:=k2; b[4,1]:=k3; b[4,4]:=k4;

b[1,2]:=a[b[1,1],b[1,1]]; b[1,3]:=a[b[1,1],b[1,2]];

b[1,5]:=a[b[1,1],b[1,4]]; b[1,6]:=a[b[1,1],b[1,5]];

b[1,7]:=a[b[1,1],b[1,6]];

for t2:=1 to 7 do b[2,t2]:=a[b[1,t2],b[1,t2]];

for t3:=1 to 7 do b[3,t3]:=a[b[1,t3],b[2,t3]];

b[4,2]:=a[b[4,1],b[4,1]]; b[4,3]:=a[b[4,1],b[4,2]];

b[4,5]:=a[b[4,1],b[4,4]]; b[4,6]:=a[b[4,1],b[4,5]];

b[4,7]:=a[b[4,1],b[4,6]];

for t5:=1 to 7 do b[5,t5]:=a[b[1,t5],b[4,t5]];

for t6:=1 to 7 do b[6,t6]:=a[b[1,t6],b[5,t6]];

for t7:=1 to 7 do b[7,t7]:=a[b[1,t7],b[6,t7]];

end;

procedure zap(a:mas);

var f1:text;

i,j,k1,k2,k3,k4,r:integer;

begin

assign(f1,'a:\a2.txt');

rewrite(f1);

r:=0;

for k1:=0 to 7 do

for k2:=0 to 7 do

for k3:=0 to 7 do

for k4:=0 to 7 do begin

zap1(k1,k2,k3,k4,a,b);

for i:=1 to 7 do begin

for j:=1 to 7 do write(f1,b[i,j]);

writeln(f1);

end;

writeln(f1); inc(r);

end;

close(f1);

writeln(r);

end;

begin

clrscr;

sc(a);

zap(a);

writeln('ok');

readln;

end.

Программа 2. Проверка на ассоциативность:

program assoc;

uses crt;

type mas=array [0..7,0..7] of integer;

var a,b:mas;

i,j,k1,k2,k3,n,i1,i2,r,q:integer;

f,f1:text;

begin

clrscr;

q:=0;

assign(f,'a:\w03.txt');

reset(f);

assign(f1,'a:\aw03.txt');

rewrite(f1);

n:=1; r:=0;

while n<=2048 do begin

for i:=0 to 7 do begin

for j:=0 to 7 do read(f,a[i,j]);

readln(f);

end;

for k1:=1 to 7 do

for k2:=1 to 7 do

for k3:=1 to 7 do

if a[k1,a[k2,k3]]<>a[a[k1,k2],k3] then q:=1;

if q<>1 then begin

for i1:=0 to 7 do begin

for i2:=0 to 7 do write(f1,a[i1,i2],' ');

writeln(f1);

end;

inc(r);

writeln(f1);

end;

readln(f);inc(n);q:=0;

end;

close(f);

close(f1);

writeln(r);

writeln('ok');

readln;

end.

Программа 3. Нахождение изоморфных колец:

program izomorf;

uses crt;

const p=1;

type mas=array [0..7,0..7] of integer;

mas1=array [1..8,1..8] of integer;

var a,b:mas;

y:mas1;

i,j,n,i1,i2,r,w,h,g,d:integer;

f,f1,f2:text;

label 1;

procedure alfa(a:mas; var b:mas);

var k1,k2,j1,j2,z1,z2,z3,z4,q3,w3:integer;

begin

for k1:=0 to 7 do begin

b[0,k1]:=0;

b[1,k1]:=a[7,k1];

b[2,k1]:=a[2,k1];

b[3,k1]:=a[5,k1];

b[4,k1]:=a[4,k1];

b[5,k1]:=a[3,k1];

b[6,k1]:=a[6,k1];

b[7,k1]:=a[1,k1];

end;

for k2:=0 to 7 do begin

z1:=b[k2,3];z2:=b[k2,5];

b[k2,5]:=z1;b[k2,3]:=z2;

z3:=b[k2,1];z4:=b[k2,7];

b[k2,7]:=z3;b[k2,1]:=z4;

end;

for j1:=0 to 7 do

for j2:=0 to 7 do begin

if b[j1,j2]=1 then b[j1,j2]:=-1;

if b[j1,j2]=7 then b[j1,j2]:=-7;

end;

for j1:=0 to 7 do

for j2:=0 to 7 do begin

if b[j1,j2]=-1 then b[j1,j2]:=7;

if b[j1,j2]=-7 then b[j1,j2]:=1;

end;

for j1:=0 to 7 do

for j2:=0 to 7 do begin

if b[j1,j2]=5 then b[j1,j2]:=-5;

if b[j1,j2]=3 then b[j1,j2]:=-3;

end;

for j1:=0 to 7 do

for j2:=0 to 7 do begin

if b[j1,j2]=-5 then b[j1,j2]:=3;

if b[j1,j2]=-3 then b[j1,j2]:=5;

end;

for q3:=0 to 7 do begin

for w3:=0 to 7 do write(b[q3,w3],' ');

writeln;

end;

end;

procedure prov(b:mas);

var q,i3,j3,t,m,x,i4,j4:integer;

a1:mas;

begin

t:=1;

reset(f);

rewrite(f1);

for m:=1 to 9 do readln(f);

while t<=p do begin

for i3:=0 to 7 do begin

for j3:=0 to 7 do read(f,a1[i3,j3]);

readln(f);

end;

q:=0;

for i3:=0 to 7 do

for j3:=0 to 7 do begin

if a1[i3,j3]=b[i3,j3] then inc(q); end;

if q<>64 then

{for x:=1 to 9 do readln(f) else}

begin

for i4:=0 to 7 do begin

for j4:=0 to 7 do write(f1,a1[i4,j4],' ');

writeln(f1);

end;

writeln(f1);

end;

readln(f);

inc(t);

end;

end;

begin

clrscr;

assign(f1,'a:\pr.txt');

assign(f2,'a:\iaw.txt');

rewrite(f2);

assign(f,'a:\aw03.txt');

n:=1;

reset(f);

rewrite(f1);

for i:=0 to 7 do begin

for j:=0 to 7 do read(f,a[i,j]);

readln(f);

end;

{if a[i,j]=8 then begin n:=8; goto 1; end;}

alfa(a,b);

readln(f);

w:=0;

for h:=0 to 7 do

for g:=0 to 7 do

if a[h,g]=b[h,g] then inc(w);

if w<>64 then prov(b) else begin

r:=1;

while r<=p do begin

for i1:=1 to 8 do begin

for i2:=1 to 8 do read(f,y[i1,i2]);

readln(f);

end;

for i1:=1 to 8 do begin

for i2:=1 to 8 do write(f1,y[i1,i2],' ');

writeln(f1);

end;

writeln(f1);

readln(f);

inc(r);

end;

end;

for i1:=0 to 7 do begin

for i2:=0 to 7 do write(f2,a[i1,i2],' ');

writeln(f2);

end;

writeln(f2);

reset(f1);

rewrite(f);

d:=1;

while d<=p do begin

for i1:=1 to 8 do begin

for i2:=1 to 8 do read(f1,y[i1,i2]);

readln(f1);

end;

for i1:=1 to 8 do begin

for i2:=1 to 8 do write(f,y[i1,i2],' ');

writeln(f);

end;

writeln(f);

readln(f1);

inc(d);

end;

{readln(f1);

writeln(f);

inc(n);

end;}

close(f);

close(f1);

close(f2);

writeln('ok');

readln;

end.

**2.** **Текст программы поиска колец, имеющих аддитивную группу .**



Программа 1. Нахождение всех базисов данной аддитивной группы:

program basis;

uses crt;

type mas=array [1..3] of integer;

var a,b,c,d,e,f,g:char;

i,j,n,k:integer;

x:mas;

f1,f2:text;

begin

assign(f2,'a:\28.txt');

reset(f2);

assign(f1,'a:\new168.txt');

rewrite(f1);

n:=1;

while n<=28 do begin

for i:=1 to 3 do read(f2,x[i]);

write(f1,x[1],' ',x[2],' ',x[3]);

writeln(f1);

write(f1,x[1],' ',x[3],' ',x[2]);

writeln(f1);

write(f1,x[3],' ',x[1],' ',x[3]);

writeln(f1);

write(f1,x[3],' ',x[2],' ',x[1]);

writeln(f1);

write(f1,x[2],' ',x[3],' ',x[1]);

writeln(f1);

write(f1,x[2],' ',x[1],' ',x[3]);

writeln(f1);

readln(f2);

writeln(f1);

inc(n);

end;

close(f2);

close(f1);

writeln('ok');

readln;

end.

Программа 2. Формирование полугрупп, с учетом ассоциативности:

program assZ2;

uses crt;

type mas=array [0..7,0..7] of integer;

var a,a1:mas;

i1,i2,i3,i4,i5,i6,i7,i8,i9,k1,k2,k3,r:integer;

n:real;

f,f1:text;

procedure sc(var a1:mas);

var i,j:integer;

begin

assign(f1,'a:\z2summ.txt');

reset(f1);

for i:=0 to 7 do begin

for j:=0 to 7 do read(f1,a1[i,j]);

readln(f1);

end;

close(f1);

end;

procedure zap(i1,i2,i3,i4,i5,i6,i7,i8,i9:integer;a1:mas; var r:integer);

var i,j,t1,t2,t3,t4,q:integer;

begin

q:=0;

for t1:=0 to 7 do a[0,t1]:=0;

for t2:=1 to 7 do a[t2,0]:=0;

a[1,1]:=i1; a[1,2]:=i2; a[1,3]:=i3;

a[2,1]:=i4; a[2,2]:=i5; a[2,3]:=i6;

a[3,1]:=i7; a[3,2]:=i8; a[3,3]:=i9;

for t3:=1 to 3 do begin

a[t3,4]:=a1[a[t3,1],a[t3,2]];

a[t3,5]:=a1[a[t3,1],a[t3,3]];

a[t3,6]:=a1[a[t3,2],a[t3,3]];

a[t3,7]:=a1[a[t3,1],a[t3,6]];

end;

for t4:=1 to 7 do begin

a[4,t4]:=a1[a[1,t4],a[2,t4]];

a[5,t4]:=a1[a[1,t4],a[3,t4]];

a[6,t4]:=a1[a[2,t4],a[3,t4]];

a[7,t4]:=a1[a[1,t4],a[6,t4]];

end;

{for i:=0 to 7 do begin

for j:=0 to 7 do write(a[i,j],' ');

writeln;

end;}

for k1:=0 to 7 do

for k2:=0 to 7 do

for k3:=0 to 7 do

if a[k1,a[k2,k3]]<>a[a[k1,k2],k3] then q:=1;

{writeln(q);}

if q=0 then begin

inc(r);

{for i:=0 to 2 do begin

for j:=0 to 2 do write(f,a[i,j],' ');

writeln(f);

end;

writeln(f);

end;

q:=0;}

end;

end;

begin

clrscr;

r:=0;

sc(a1);

assign(f,'a:\z2.txt');

rewrite(f);

{ i1:=6;i2:=6;i3:=6;i4:=6;i5:=6;i6:=6;i7:=6;i8:=6;i9:=6;}

for i1:=0 to 7 do

for i2:=0 to 7 do

for i3:=0 to 7 do

for i4:=0 to 7 do

for i5:=0 to 7 do

for i6:=0 to 7 do

for i7:=0 to 7 do

for i8:=0 to 7 do

for i9:=0 to 7 do begin

zap(i1,i2,i3,i4,i5,i6,i7,i8,i9,a1,r);

writeln(r);

end;

writeln(' ',r);

close(f);

writeln('ok');

readln;

end.

Программа 3. Нахождение изоморфных колец:

program izomorf;

uses crt;

const p=1;

type mas=array [0..7,0..7] of integer;

mas1=array [1..3] of integer;

var a,b,a1:mas;

y:mas1;

n,j1,i2,t,h,g,d:integer;

f,f1,f2,f3,f8:text;

procedure basis(var y:mas1);

var i:integer;

begin

for i:=1 to 3 do read(f2,y[i]);

readln(f2);

end;

procedure sc(var a:mas);

var i,j:integer;

begin

reset(f1);

for i:=0 to 7 do begin

for j:=0 to 7 do read(f1,a[i,j]);

readln(f1);

end;

end;

{procedure glav(b:mas);

var q,i,j,x,k:integer;

begin

x:=0;k:=1;

reset(f1);

rewrite(f8);

while n<=t do begin

q:=0;

for i:=0 to 7 do begin

for j:=0 to 7 do read(f1,a1[i,j]);

readln(f1);

end;

for i:=0 to 7 do

for j:=0 to 7 do

if a1[i,j]=b[i,j] then inc(q);

if q=64 then begin inc(h);

x:=1;

end

else

for i:=0 to 7 do begin

for j:=0 to 7 do write(f8,a1[i,j],' ');

writeln(f8);

end;

inc(n);

readln(f1);

writeln(f8);

end;

if x=1 then t:=t-2 else t:=t-1;

reset(f8);

rewrite(f1);

while k<=t do begin

for i:=0 to 7 do begin

for j:=0 to 7 do read(f8,a1[i,j]);

readln(f8);

end;

for i:=0 to 7 do begin

for j:=0 to 7 do write(f1,a1[i,j],' ');

writeln(f1);

end;

inc(k);

end;

end;}

procedure newb(y:mas1;a:mas; var b:mas);

var i,j,z1,z2,z3:integer;

begin

for i:=0 to 7 do begin

z1:=a[1,i]; a[1,i]:=a[y[1],i]; a[y[1],i]:=z1;

z2:=a[2,i]; a[2,i]:=a[y[2],i]; a[y[2],i]:=z2;

z3:=a[3,i]; a[3,i]:=a[y[3],i]; a[y[3],i]:=z3;

end;

for j:=0 to 7 do begin

z1:=a[j,1]; a[j,1]:=a[j,y[1]]; a[j,y[1]]:=z1;

z2:=a[j,2]; a[j,2]:=a[j,y[2]]; a[j,y[2]]:=z2;

z3:=a[j,3]; a[j,3]:=a[j,y[3]]; a[j,y[3]]:=z3;

end;

for i:=0 to 7 do

for j:=0 to 7 do begin

if a[i,j]=1 then a[i,j]:=-1;

if a[i,j]=y[1] then a[i,j]:=-y[1];

end;

for i:=0 to 7 do

for j:=0 to 7 do begin

if a[i,j]=-1 then a[i,j]:=y[1];

if a[i,j]=-y[1] then a[i,j]:=1;

end;

for i:=0 to 7 do

for j:=0 to 7 do begin

if a[i,j]=2 then a[i,j]:=-2;

if a[i,j]=y[2] then a[i,j]:=-y[2];

end;

for i:=0 to 7 do

for j:=0 to 7 do begin

if a[i,j]=-2 then a[i,j]:=y[2];

if a[i,j]=-y[2] then a[i,j]:=2;

end;

for i:=0 to 7 do

for j:=0 to 7 do begin

if a[i,j]=3 then a[i,j]:=-3;

if a[i,j]=y[3] then a[i,j]:=-y[3];

end;

for i:=0 to 7 do

for j:=0 to 7 do begin

if a[i,j]=-3 then a[i,j]:=y[3];

if a[i,j]=-y[3] then a[i,j]:=3;

end;

for i:=0 to 7 do

for j:=0 to 7 do b[i,j]:=a[i,j];

{glav(b);}

for i:=0 to 7 do begin

for j:=0 to 7 do write(f,b[i,j],' ');

writeln(f);

end;

writeln(f);

end;

procedure new5(b,a:mas);

var i,j,z1,z2,i1:integer;

begin

for i:=1 to 5 do begin

if i mod 2<>0 then begin

for j:=0 to 7 do begin

z1:=b[2,j]; b[2,j]:=b[3,j]; b[3,j]:=z1;

end;

for j:=0 to 7 do begin

z2:=b[j,2]; b[j,2]:=b[j,3]; b[j,3]:=z2;

end;

for i1:=0 to 7 do

for j:=0 to 7 do begin

if b[i1,j]=3 then b[i1,j]:=-3;

if b[i1,j]=2 then b[i1,j]:=-2;

end;

for i1:=0 to 7 do

for j:=0 to 7 do begin

if b[i1,j]=-3 then b[i1,j]:=2;

if b[i1,j]=-2 then b[i1,j]:=3;

end;

{glav(b);}

for i1:=0 to 7 do begin

for j:=0 to 7 do write(f,b[i1,j],' ');

writeln(f);

end;

writeln(f);

end;

if i mod 2=0 then begin

for j:=0 to 7 do begin

z1:=b[1,j]; b[1,j]:=b[2,j]; b[2,j]:=z1;

end;

for j:=0 to 7 do begin

z2:=b[j,1]; b[j,1]:=b[j,2]; b[j,2]:=z2;

end;

for i1:=0 to 7 do

for j:=0 to 7 do begin

if b[i1,j]=1 then b[i1,j]:=-1;

if b[i1,j]=2 then b[i1,j]:=-2;

end;

for i1:=0 to 7 do

for j:=0 to 7 do begin

if b[i1,j]=-1 then b[i1,j]:=2;

if b[i1,j]=-2 then b[i1,j]:=1;

end;

{glav(b);}

for i1:=0 to 7 do begin

for j:=0 to 7 do write(f,b[i1,j],' ');

writeln(f);

end;

writeln(f);

end;

end;

end;

begin

clrscr;

assign(f1,'a:\vse\_z21.txt');

reset(f1);

assign(f2,'a:\28.txt');

reset(f2);

assign(f,'a:\xx.txt');

rewrite(f);

{assign(f3,'a:\ist.txt');

rewrite(f3);

assign(f8,'a:\pom.txt');

rewrite(f8);}sc(a);

for g:=1 to 28 do begin

basis(y);{d:=1;g:=0;}

{while d<=t do begin}

{inc(g);writeln(g);}

{for j1:=0 to 7 do begin

for i2:=0 to 7 do write(f3,a[j1,i2]);

writeln(f3);

end;

writeln(f3);}

newb(y,a,b);

{n:=1;}

new5(b,a);

{inc(d);}

end;

close(f);

close(f1);

close(f2);

{close(f3);}

{close(f8);}

{ writeln(h); }

writeln('ok');

readln;

end.