

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. ПЕТРА ВЕЛИКОГО

Институт прикладной математики и механики  
Кафедра Телематика  
Математика и компьютерные науки

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**  
**ГЕНЕРАЦИЯ МУЛЬТИМНОЖЕСТВА И РЕАЛИЗАЦИЯ ОПЕРАЦИЙ НАД НИМ**

2 курс, группа 23607/1

Преподаватель  
\_\_\_\_\_ А.В.Востров  
«\_\_\_\_\_» \_\_\_\_\_ 2017.

Выполнил  
Е.Г.Климов

Санкт-Петербург  
2017

## **Аннотация**

Данный отчёт описывает программную реализацию мультимножеств: создание универсума с помощью кода Грея, реализацию возможности выбора пользователем элементов из универсума, операции над мультимножествами.

## Содержание

<b>Введение</b> . . . . .	<b>4</b>
<b>1. Постановка задачи</b> . . . . .	<b>5</b>
<b>2. Математическая модель</b> . . . . .	<b>6</b>
2.1. Понятие мультимножества . . . . .	6
2.2. Операции над мультимножествами . . . . .	6
<b>3. Реализация</b> . . . . .	<b>7</b>
3.1. Структура мультимножества . . . . .	7
3.2. Операции над мультимножествами . . . . .	8
3.3. Бинарный код Грея . . . . .	10
3.4. Выбор множеств пользователем . . . . .	11
<b>4. Пример работы программы</b> . . . . .	<b>12</b>
4.1. Задание мощности и показателя для универсума . . . . .	12
4.2. Генерация универсума и работа с множествами . . . . .	13
<b>5. Заключение</b> . . . . .	<b>19</b>
<b>Список литературы</b> . . . . .	<b>20</b>

## Введение

Имеется широкий круг задач, отличительной особенностью которых является множественность и повторяемость данных, описывающих как сами рассматриваемые объекты, так и их свойства. С точки зрения математики такие многопризнаковые объекты можно представить как мультимножества или множества с повторяющимися элементами. Мультимножество можно рассматривать или как одну из частных форм множества (так обычно принято считать, например, в комбинаторной математике), или как самостоятельное понятие, более общее, чем множество.

Бинарный код Грея - способ генерация всех подмножеств  $n$ -элементного множества, использующий побитовое инвертирование, это делает его эффективнее других способов, использующих операции сложения элементов, так как побитовое инвертирование - незатратная операция.

## 1. Постановка задачи

1. Для введенного с клавиатуры  $n$  реализовать генерацию кода Грея (сформировать универсум).
2. На основе сформированного универсума реализовать мультимножество, где введенное пользователем  $k$  - максимальное значение показателя для элементов-носителей мультимножества (элементов из универсума).
3. Реализовать выбор множеств двумя способами: перечислением и случайным образом.
4. Выполнить операции над множествами

## 2. Математическая модель

### 2.1. Понятие мультимножества

Мультимножеством  $\mathbf{A}$ , порожденным основным (обычным) множеством  $U = \{x_1, x_2, \dots\}$ , все элементы  $x_i$  которого различны, называется совокупность групп одинаковых элементов

$$\mathbf{A} = \{k_{A1} \cdot x_1, k_{A2} \cdot x_2, \dots\}, \quad x_i \in U.$$

Группу одинаковых элементов  $k_{Ai} \cdot x_i$  будем называть компонентой мультимножества, одинаковые элементы  $x_i$ , входящие в компоненту  $k_{Ai} \cdot x_i$ , – экземплярами элементов мультимножества, а функцию  $k_A$ , значение которой  $k_A(x_i) = k_{Ai}$  определяет число вхождений элемента  $x_i \in U$  в мультимножество  $\mathbf{A}$  или «вес» элемента  $x_i$  в мультимножестве  $\mathbf{A}$  – функцией кратности или функцией числа экземпляров мультимножества  $\mathbf{A}$ . Таким образом, мультимножество – это множество, состоящее из различных групп одинаковых экземпляров элементов. Будем говорить, что элемент  $x$  принадлежит мультимножеству  $\mathbf{A}$  (обозначается  $x \in \mathbf{A}$ ) и в мультимножестве  $\mathbf{A}$  имеется ровно  $k$  экземпляров элемента  $x$  тогда и только тогда, когда кратность элемента  $x$  равна  $k_A(x) = k > 0$ . Когда кратность элемента  $x$  равна нулю  $k_A(x) = 0$ , тогда будем говорить, что элемент  $x$  не содержится в мультимножестве  $\mathbf{A}$  (обозначается  $x \notin \mathbf{A}$ )

### 2.2. Операции над мультимножествами

Обычно рассматриваются следующие операции над мультимножествами:

*объединение:*

$$\mathbf{A} \cup \mathbf{B} = \{k_{A \cup B}(x) \cdot x \mid k_{A \cup B}(x) = \max(k_A(x), k_B(x))\}.$$

*пересечение:*

$$\mathbf{A} \cap \mathbf{B} = \{k_{A \cap B}(x) \cdot x \mid k_{A \cap B}(x) = \min(k_A(x), k_B(x))\}.$$

*арифметическая сумма:*

$$\mathbf{A} \uplus \mathbf{B} = \{k_{A \uplus B}(x) \cdot x \mid k_{A \uplus B}(x) = k_A(x) + k_B(x)\}.^1$$

*арифметическая разность:*

$$\mathbf{A} \setminus \mathbf{B} = \{k_{A \setminus B}(x) \cdot x \mid k_{A \setminus B}(x) = \max(k_A(x) - k_B(x), 0)\}.$$

*симметрическая разность:*

$$\mathbf{A} \Delta \mathbf{B} = \{k_{A \Delta B}(x) \cdot x \mid k_{A \Delta B}(x) = |k_A(x) - k_B(x)|\}.$$

*дополнение:*

$$\mathbf{A}' = \{k_{A'}(x) \cdot x \mid k_{A'}(x) = k_U(x) - k_A(x)\}.$$

---

<sup>1</sup> $k_{A \uplus B}(x)$  не превышает  $k_U(x)$

## 3. Реализация

### 3.1. Структура мультимножества

Для хранения одного элемента мультимножества служит класс *MultisetElement*: он содержит поле, которое хранит само значение элемента (объект типа *Element* - это вспомогательный тип, сделанный для того, чтобы можно было обращаться напрямую к каждому биту типа *int*) и значение показателя для этого элемента.

В дальнейшем при создании мультимножества создается массив элементов типа *MultisetElement*.

Ограничение: максимальная разрядность - 8, это обусловлено размером типа *int*.

Листинг 1: Element.h

```
1  class Byte {
2  public:
3      size_t operator[](size_t index);
4
5  private:
6      unsigned char m_b0 : 1;
7      unsigned char m_b1 : 1;
8      unsigned char m_b2 : 1;
9      unsigned char m_b3 : 1;
10     unsigned char m_b4 : 1;
11     unsigned char m_b5 : 1;
12     unsigned char m_b6 : 1;
13     unsigned char m_b7 : 1;
14 };
15
16 union Element {
17 public:
18     int getValue() const { return m_value; }
19     void setValue(int value) { m_value = value; }
20
21     size_t operator[](size_t index);
22     bool operator< (const Element &other);
23
24 private:
25     int m_value;
26     Byte m_bytes[sizeof(int)];
27 };
28
29 class MultisetElement {
30 public:
31     int getElement() const { return m_element.getValue(); }
32     int getOccurrenceNo() const { return m_occurrenceNo; }
33
34     void setOccurrenceNo(int occurrenceNo) { m_occurrenceNo = occurrenceNo; }
35
36     bool operator< (const MultisetElement &other);
37     size_t operator[](size_t index);
38
39     friend void grayCodeGeneration(int, MultisetElement *, int, int);
40 private:
41     Element m_element;
42     int m_occurrenceNo;
43 };
44
45 void grayCodeGeneration(int n, MultisetElement *array, int depth, int occurrenceNo = 1);
```

### 3.2. Операции над мультимножествами

Все операции реализованы в полном соответствии с математической моделью, в качестве функции  $k_A(x_i)$  выступает поле `m_occurrenceNo` класса `MultisetElement`, которое равно количеству вхождений элемента. Для упрощения, множества, выбираемые пользователем, представляют собой кортежи длины  $2^n$ , значения элементов которых находятся в промежутке  $[0; occurrenceNo]$ , где `occurrenceNo` - число вхождений элемента в универсум.

Комментарий к листингу: `m_setA`, `m_setB`, `m_result` - упомянутые выше кортежи, `m_cardinality` - длина кода.

Листинг 2: Операции над мультимножествами

```
1 void Table::intersection()
2 {
3     size_t size = static_cast<size_t>(pow(2, this->m_cardinality));
4
5     for (size_t i = 0; i < size; ++i)
6     {
7         m_result[i] = std::min(m_setA[i], m_setB[i]);
8     }
9 }
10
11 void Table::union_()
12 {
13     size_t size = static_cast<size_t>(pow(2, this->m_cardinality));
14
15     for (size_t i = 0; i < size; ++i)
16     {
17         m_result[i] = std::max(m_setA[i], m_setB[i]);
18     }
19 }
20
21 void Table::sum()
22 {
23     size_t size = static_cast<size_t>(pow(2, this->m_cardinality));
24
25     for (size_t i = 0; i < size; ++i)
26     {
27         int N = m_setA[i] + m_setB[i];
28         m_result[i] = (N < m_occurrenceNo)? N : m_occurrenceNo;
29     }
30 }
31
32 void Table::differenceAB(){
33     size_t size = static_cast<size_t>(pow(2, this->m_cardinality));
34
35     for (size_t i = 0; i < size; ++i)
36     {
37         m_result[i] = (m_setA[i] - m_setB[i]) > 0 ? m_setA[i] - m_setB[i] : 0;
38     }
39 }
40
41 void Table::differenceBA()
42 {
43     size_t size = static_cast<size_t>(pow(2, this->m_cardinality));
44
45     for (size_t i = 0; i < size; ++i)
46     {
47         m_result[i] = (m_setB[i] - m_setA[i]) > 0 ? abs(m_setB[i] - m_setA[i]) : 0;
48     }
49 }
50 }
51
52 void Table::notA()
53 {
```



```

54     size_t size = static_cast<size_t>(pow(2, this->m_cardinality));
55
56     for (size_t i = 0; i < size; ++i)
57     {
58         m_result[i] = m_occurrenceNo - m_setA[i];
59     }
60 }
61
62 void Table::notB()
63 {
64     size_t size = static_cast<size_t>(pow(2, this->m_cardinality));
65
66     for (size_t i = 0; i < size; ++i)
67     {
68         m_result[i] = m_occurrenceNo - m_setB[i];
69     }
70 }
71
72 void Table::symmetricDifference()
73 {
74     size_t size = static_cast<size_t>(pow(2, this->m_cardinality));
75
76     for (size_t i = 0; i < size; ++i)
77     {
78         m_result[i] = abs(m_setA[i] - m_setB[i]);
79     }
80 }

```

Для того, чтобы применить операцию для множеств пользователю необходимо выбрать множества, затем выбрать требуемую операцию и нажать кнопку "Compute".

### 3.3. Бинарный код Грея

Бинарный код Грея реализован с помощью рекурсивной функции:

```
void grayCodeGeneration(int n, MultisetElement* array, int depth, int occurrenceNo);
```

- **n** - длина кода, соответственно  $2^n$  - мощность множества-носителя.
- **array** - массив типа MultisetElement, под который до вызова функции выделена память для хранения  $2^n$  элементов, после выполнения функции array - требуемый универсум.
- **depth** - параметр рекурсии.
- **occurrenceNo** - максимальное значение для показателя элемента (число вхождений элемента), вводится пользователем.

Листинг 3: Gray code

```
1 void grayCodeGeneration(int n, MultisetElement* array, int depth, int occurrenceNo)
2 {
3     int i, t = (1 << (depth - 1));
4
5     if (depth == 0)
6     {
7         array[0].m_element.setValue(0);
8         array[0].m_occurrenceNo = occurrenceNo;
9     }
10    else
11    {
12        for (i = 0; i < t; ++i)
13        {
14            array[t + i].m_element.setValue(array[t - i - 1].m_element.getValue() +
15                                            (1 << (depth - 1)));
16            array[t + i].m_occurrenceNo = occurrenceNo;
17        }
18    }
19    if (depth != n)
20        grayCodeGeneration(n, array, depth + 1, occurrenceNo);
21 }
```

### 3.4. Выбор множеств пользователем

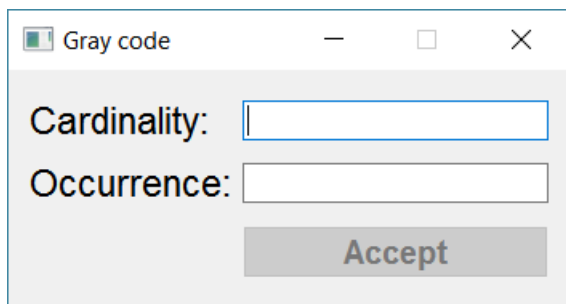
Интерфейс приложения создан с помощью фреймворка Qt.

Первоначально, пользователю следует ввести в окно значение длины кода  $n$ , которому будет соответствовать мощность множества  $2^n$ , и значение максимального возможного количества вхождения элементов. Значение  $n$  не может быть меньше нуля и не может превышать восьми, что обусловлено размером, которое занимает тип `int` в памяти. Значение количества вхождений не может быть меньше нуля, а сверху ограничено значением 999 999 999. Для выбора количества вхождений элемента в пользовательские множества используются виджеты `QSpinBox`, с диапазоном значений  $[0; occurrenceNo]$ , где `occurrenceNo` - число вхождений элемента в универсум, введенное до этого. Помимо этого, пользователь может задать множества случайно, нажав на кнопку "Make Sets" и может очистить введенные ранее значения с помощью кнопки "Reset". Пример будет представлен в разделе "Пример работы программы" данного отчета.

## 4. Пример работы программы

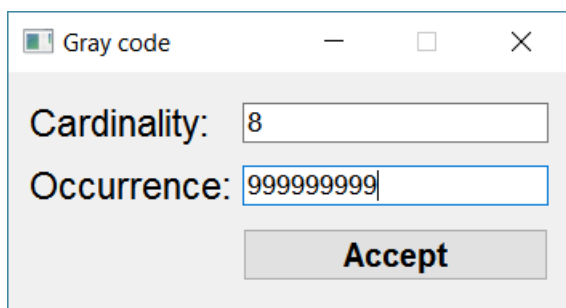
### 4.1. Задание мощности и показателя для универсума

При запуске приложения пользователю доступно только окно выбора мощности и показателя



The screenshot shows a window titled "Gray code" with standard Windows window controls (minimize, maximize, close). Inside the window, there are two labels: "Cardinality:" and "Occurrence:". Each label is followed by an empty text input field. Below these fields is a gray button with the text "Accept" in bold.

Рис. 1: Начальное состояние окна



The screenshot shows the same "Gray code" window. The "Cardinality:" input field now contains the number "8". The "Occurrence:" input field now contains the string "999999999". The "Accept" button remains at the bottom.

Рис. 2: Состояние после ввода значений

После нажатия кнопки "Асепт"это окно закрывается и пользователю становится доступным новое окно, необходимое для работы с множествами.

## 4.2. Генерация универсума и работа с множествами

Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	0	0	0
00000001	99999999	0	0	0
00000010	99999999	0	0	0
00000011	99999999	0	0	0
00000100	99999999	0	0	0
00000101	99999999	0	0	0
00000110	99999999	0	0	0
00000111	99999999	0	0	0
00001000	99999999	0	0	0
00001001	99999999	0	0	0
00001010	99999999	0	0	0
00001011	99999999	0	0	0
00001100	99999999	0	0	0
00001101	99999999	0	0	0
00001110	99999999	0	0	0
00001111	99999999	0	0	0
00010000	99999999	0	0	0
00010001	99999999	0	0	0
00010010	99999999	0	0	0
00010011	99999999	0	0	0

Operation:

☐  $A \cup B$

☐  $A \oplus B$

☐  $A \cap B$

☐  $A \setminus B$

☐  $B \setminus A$

☐  $A'$

☐  $B'$

☐  $A \Delta B$

Compute

Make Sets

Reset

Рис. 3: Начальное состояние окна

Для выбора множеств можно нажать кнопку "Make Sets" или сделать это вручную

Gray code					
Universum	Occurrence in Universum	Set A		Set B	
00000000	999999999	41	▲▼	26362	▲▼ 0
00000001	999999999	18467	▲▼	4886	▲▼ 0
00000010	999999999	6334	▲▼	18875	▲▼ 0
00000011	999999999	26500	▲▼	28433	▲▼ 0
00000100	999999999	19169	▲▼	29869	▲▼ 0
00000101	999999999	15724	▲▼	20142	▲▼ 0
00000110	999999999	11478	▲▼	23844	▲▼ 0
00000111	999999999	29358	▲▼	1416	▲▼ 0
00001000	999999999	26962	▲▼	21881	▲▼ 0
00001001	999999999	24464	▲▼	31998	▲▼ 0
00001010	999999999	5705	▲▼	10322	▲▼ 0
00001011	999999999	28145	▲▼	18651	▲▼ 0
00001100	999999999	23281	▲▼	10021	▲▼ 0
00001101	999999999	16827	▲▼	5699	▲▼ 0
00001110	999999999	9961	▲▼	3557	▲▼ 0
00001111	999999999	491	▲▼	28476	▲▼ 0
00010000	999999999	2995	▲▼	27892	▲▼ 0
00010001	999999999	11942	▲▼	24389	▲▼ 0
00010010	999999999	4827	▲▼	5075	▲▼ 0
00010011	999999999	5436	▲▼	10712	▲▼ 0

Operation:

☐  $A \cup B$

☐  $A \cup B$

☐  $A \cap B$

☐  $A \setminus B$

☐  $B \setminus A$

☐  $A'$

☐  $B'$

☐  $A \Delta B$

Compute

Make Sets

Reset

Рис. 4: Таблица после нажатия "Make Sets"

После этого пользователь может выбрать одну операцию из списка, их реализация и математическое описание было разобрано в соответствующих частях отчета. Для получения результата пользователю необходимо нажать на кнопку "Compute"

Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	41	26362	26362
00000001	99999999	18467	4886	18467
00000010	99999999	6334	18875	18875
00000011	99999999	26500	28433	28433
00000100	99999999	19169	29869	29869
00000101	99999999	15724	20142	20142
00000110	99999999	11478	23844	23844
00000111	99999999	29358	1416	29358
00001000	99999999	26962	21881	26962
00001001	99999999	24464	31998	31998
00001010	99999999	5705	10322	10322
00001011	99999999	28145	18651	28145
00001100	99999999	23281	10021	23281
00001101	99999999	16827	5699	16827
00001110	99999999	9961	3557	9961
00001111	99999999	491	28476	28476
00010000	99999999	2995	27892	27892
00010001	99999999	11942	24389	24389
00010010	99999999	4827	5075	5075
00010011	99999999	5436	10712	10712

Рис. 5: Результат объединения

Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	41	26362	26403
00000001	99999999	18467	4886	23353
00000010	99999999	6334	18875	25209
00000011	99999999	26500	28433	54933
00000100	99999999	19169	29869	49038
00000101	99999999	15724	20142	35866
00000110	99999999	11478	23844	35322
00000111	99999999	29358	1416	30774
00001000	99999999	26962	21881	48843
00001001	99999999	24464	31998	56462
00001010	99999999	5705	10322	16027
00001011	99999999	28145	18651	46796
00001100	99999999	23281	10021	33302
00001101	99999999	16827	5699	22526
00001110	99999999	9961	3557	13518
00001111	99999999	491	28476	28967
00010000	99999999	2995	27892	30887
00010001	99999999	11942	24389	36331
00010010	99999999	4827	5075	9902
00010011	99999999	5436	10712	16148

Рис. 6: Результат арифметической суммы

Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	41	26362	41
00000001	99999999	18467	4886	4886
00000010	99999999	6334	18875	6334
00000011	99999999	26500	28433	26500
00000100	99999999	19169	29869	19169
00000101	99999999	15724	20142	15724
00000110	99999999	11478	23844	11478
00000111	99999999	29358	1416	1416
00001000	99999999	26962	21881	21881
00001001	99999999	24464	31998	24464
00001010	99999999	5705	10322	5705
00001011	99999999	28145	18651	18651
00001100	99999999	23281	10021	10021
00001101	99999999	16827	5699	5699
00001110	99999999	9961	3557	3557
00001111	99999999	491	28476	491
00010000	99999999	2995	27892	2995
00010001	99999999	11942	24389	11942
00010010	99999999	4827	5075	4827
00010011	99999999	5436	10712	5436

Рис. 7: Результат пересечения

Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	41	26362	0
00000001	99999999	18467	4886	13581
00000010	99999999	6334	18875	0
00000011	99999999	26500	28433	0
00000100	99999999	19169	29869	0
00000101	99999999	15724	20142	0
00000110	99999999	11478	23844	0
00000111	99999999	29358	1416	27942
00001000	99999999	26962	21881	5081
00001001	99999999	24464	31998	0
00001010	99999999	5705	10322	0
00001011	99999999	28145	18651	9494
00001100	99999999	23281	10021	13260
00001101	99999999	16827	5699	11128
00001110	99999999	9961	3557	6404
00001111	99999999	491	28476	0
00010000	99999999	2995	27892	0
00010001	99999999	11942	24389	0
00010010	99999999	4827	5075	0
00010011	99999999	5436	10712	0

Рис. 8: Результат арифметической разности  $A \setminus B$



Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	41	26362	26321
00000001	99999999	18467	4886	0
00000010	99999999	6334	18875	12541
00000011	99999999	26500	28433	1933
00000100	99999999	19169	29869	10700
00000101	99999999	15724	20142	4418
00000110	99999999	11478	23844	12366
00000111	99999999	29358	1416	0
00001000	99999999	26962	21881	0
00001001	99999999	24464	31998	7534
00001010	99999999	5705	10322	4617
00001011	99999999	28145	18651	0
00001100	99999999	23281	10021	0
00001101	99999999	16827	5699	0
00001110	99999999	9961	3557	0
00001111	99999999	491	28476	27985
00010000	99999999	2995	27892	24897
00010001	99999999	11942	24389	12447
00010010	99999999	4827	5075	248
00010011	99999999	5436	10712	5276

Operation:

- ☐  $A \cup B$   
☐  $A \oplus B$   
☐  $A \cap B$   
☐  $A \setminus B$   
☒  $B \setminus A$   
☐  $A'$   
☐  $B'$   
☐  $A \Delta B$

Compute

Make Sets

Reset

Рис. 9: Результат арифметической разности  $B \setminus A$

Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	41	26362	999999958
00000001	99999999	18467	4886	999981532
00000010	99999999	6334	18875	999993665
00000011	99999999	26500	28433	999973499
00000100	99999999	19169	29869	999980830
00000101	99999999	15724	20142	999984275
00000110	99999999	11478	23844	999988521
00000111	99999999	29358	1416	999970641
00001000	99999999	26962	21881	999973037
00001001	99999999	24464	31998	999975535
00001010	99999999	5705	10322	999994294
00001011	99999999	28145	18651	999971854
00001100	99999999	23281	10021	999976718
00001101	99999999	16827	5699	999983172
00001110	99999999	9961	3557	999990038
00001111	99999999	491	28476	999999508
00010000	99999999	2995	27892	999997004
00010001	99999999	11942	24389	999988057
00010010	99999999	4827	5075	999995172
00010011	99999999	5436	10712	999994563

Operation:

- ☐  $A \cup B$   
☐  $A \oplus B$   
☐  $A \cap B$   
☐  $A \setminus B$   
☐  $B \setminus A$   
☒  $A'$   
☐  $B'$   
☐  $A \Delta B$

Compute

Make Sets

Reset

Рис. 10: Результат дополнения  $A$

Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	41	26362	999973637
00000001	99999999	18467	4886	999995113
00000010	99999999	6334	18875	999981124
00000011	99999999	26500	28433	999971566
00000100	99999999	19169	29869	999970130
00000101	99999999	15724	20142	999979857
00000110	99999999	11478	23844	999976155
00000111	99999999	29358	1416	999998583
00001000	99999999	26962	21881	999978118
00001001	99999999	24464	31998	999968001
00001010	99999999	5705	10322	999989677
00001011	99999999	28145	18651	999981348
00001100	99999999	23281	10021	999989978
00001101	99999999	16827	5699	999994300
00001110	99999999	9961	3557	999996442
00001111	99999999	491	28476	999971523
00010000	99999999	2995	27892	999972107
00010001	99999999	11942	24389	999975610
00010010	99999999	4827	5075	999994924
00010011	99999999	5436	10712	999989287

Рис. 11: Результат дополнения  $B$

Universum	Occurrence in Universum	Set A	Set B	Result
00000000	99999999	41	26362	26321
00000001	99999999	18467	4886	13581
00000010	99999999	6334	18875	12541
00000011	99999999	26500	28433	1933
00000100	99999999	19169	29869	10700
00000101	99999999	15724	20142	4418
00000110	99999999	11478	23844	12366
00000111	99999999	29358	1416	27942
00001000	99999999	26962	21881	5081
00001001	99999999	24464	31998	7534
00001010	99999999	5705	10322	4617
00001011	99999999	28145	18651	9494
00001100	99999999	23281	10021	13260
00001101	99999999	16827	5699	11128
00001110	99999999	9961	3557	6404
00001111	99999999	491	28476	27985
00010000	99999999	2995	27892	24897
00010001	99999999	11942	24389	12447
00010010	99999999	4827	5075	248
00010011	99999999	5436	10712	5276

Рис. 12: Результат симметрической разности  $A \Delta B$

## 5. Заключение

В ходе работы был реализован алгоритм работы кода Грея. Это один из способов генерации универсума, его отличие в том, что все элементы отличаются только на один разряд и получается рекурсивно. Минусом этого способа является то, что универсум получается неупорядоченным.

Помимо этого, были созданы мультимножества на основе сгенерированного универсума:

1. Реализована возможность выбирать элементы из универсума для пользовательских множеств.
2. Реализованы операции над мультимножествами:
  - Объединение
  - Пересечение
  - Арифметическая сумма
  - Арифметическая разность
  - Симметрическая разность
  - Дополнение
3. В разделе "Пример работы программы" продемонстрированы все операции.

Мощность универсума ограничена из-за особенности реализации. Все операции реализованы в полном соответствии с математической моделью и не используют никаких дополнительных алгоритмов.

Возможная доработка:

- Убрать ограничение на мощность универсума, изменив представление множества в программе.
- Реализовать действия для нескольких множеств.

## Список литературы

- [1] А.Б.Петровский. "Пространства множеств и мультимножеств"
- [2] Ф.А.Новиков. "Дискретная математика для программистов"