**Цель занятия**: научиться решать теоретико-множественные уравнения с применением ЭВМ.

Вариант 15

(A ∩ X ∪ B) ∩ (C – X) = ∪ (A – X) ∩ C

U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

A = {1, 2, 5, 8, 9}

B = {1, 2, 6, 7, 9, 10}

C = {2, 4, 6, 9, 10}

X – ?

1. Преобразовать исходное уравнение в уравнение с пустой правой частью.

(A ∩ X ∪ B) ∩ (C – X) = ∪ (A – X) ∩ C

(A ∩ X ∪ B) ∩ (C – X) Δ ∪ (A – X) ∩ C = 

2. Преобразовать левую часть уравнения к виду , ∩ ∪ X ∩ U используя разложение Шеннона по неизвестному множеству X.

3. Написать программу, вычисляющую значения множеств  и U при заданных исходных множествах.

Текст программы для реализации и программную реализацию вычислений смотреть в «Приложение 1».

4. Вычислить значения множеств  и U и сделать вывод о существовании решения уравнения. Если решения уравнения не существует, то выполнить п.п. 1—4 для следующего (предыдущего) варианта.

5. Определить мощность общего решения, найти некоторые (или все) частные решения, в том числе частные решения наименьшей и наибольшей мощности.

6. Написать программу для проверки найденных решений.

Текст программы для реализации и программную реализацию вычислений смотреть в «Приложение 2».

Приложение 1

**program** z\_4;

**uses** or\_arr;

**var** a, b, c, x, p, d, e, f, g: t\_arr;

n, m, k, j, l, y, z, w, h: byte;

**begin**

writeln ('Введите количество элементов множества A');

read (n);

writeln ('Введите элементы множества А');

input (a, n);

writeln ('Введите количество элементов множества B');

read (m);

writeln ('Введите элементы множества B');

input (b, m);

writeln ('Введите количество элементов множества C');

read (k);

writeln ('Введите элементы множества C');

input (c, k);

z := complement (b, m, e);

w := complement (c, k, f);

y := intersection (a, c, n, k, d);

h := unification (e, f, z, w, g);

z := unification (g, d, h, y, e);

y := complement (p, l, d);

write ('𝛗∅ = { ');

output (e, z);

write ('}');

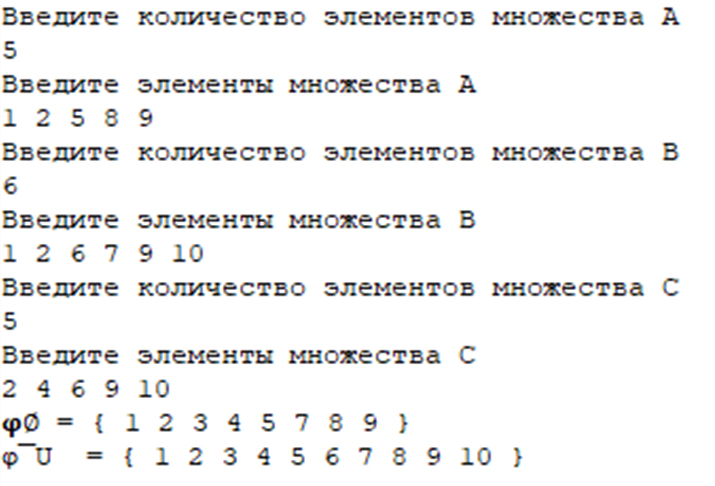
writeln;

write ('φ̅U = { ');

output (d, y);

write ('}');

**end**.



Приложение 2

**program** z\_6;

**uses** or\_arr;

**var** a, b, c, x, d, e, f, g: t\_arr;

n, m, k, j, y, z, w, h: byte;

**begin**

writeln ('Введите количество элементов множества A');

read (n);

writeln ('Введите элементы множества А');

input (a, n);

writeln ('Введите количество элементов множества B');

read (m);

writeln ('Введите элементы множества B');

input (b, m);

writeln ('Введите количество элементов множества C');

read (k);

writeln ('Введите элементы множества C');

input (c, k);

writeln ('Введите количество элементов множества X');

read (j);

writeln ('Введите элементы множества X');

input (x, j);

y := intersection (a, x, n, j, d);

z := unification (d, b, y, m, e);

y := difference (c, x, k, j, d);

w := intersection (e, d, z, y, f);

y := difference (a, x, n, j, d);

z := intersection (d, c, y, k, e);

h := complement (x, j, g);

y := unification (g, e, h, z, d);

**if** equality(f, d, w, y) **then**

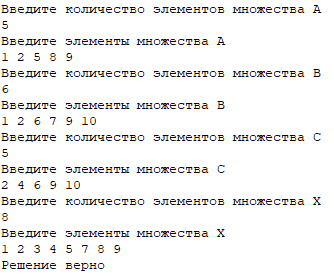
writeln('Решение верно')

**else**

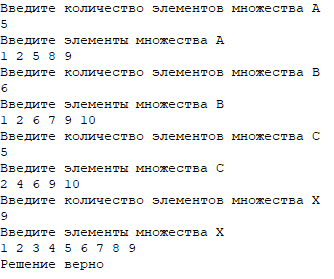
writeln('Решение не верно');

**end**.

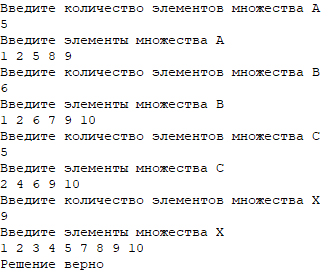
Решение 1.



Решение 2.



Решение 3.



Решение 4.

