***Министерство образования Республики Беларусь***

***Учреждение образования***

***«Брестский государственный технический университет»***

***Кафедра ИИТ***

**Лабораторная работа №1**

**По дисциплине Дискретная математика за II семестр**

**Тема: «Множества»**

**Выполнил:**

Студент группы ИИ-15 (1)

1-го курса

Волк И. А.

**Проверил:**

Глущенко Т.А.

Брест 2017

***Задание 1 (вариант 7).***

На универсуме *U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11}* заданы множества *A, B, C*. Для указанных множеств:

1. Для заданного множества *A* построить булеан.
2. Алгоритмом «слияние» построить объединение множеств *A* и *B*.
3. Найти прямое произведение  и его мощность.
4. Вычислить выражение согласно варианту, проиллюстрировать результат диаграммой Эйлера-Венна.

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

using namespace std;

const int notName = system("color f0");

void print(int \*A, int n, int i)

{

if (n)

{

if (n & 1)

cout << A[i] << " ";

print(A, n >> 1, i + 1);

}

}

bool is\_num\_in\_arr(int num, int num\_arr[], int size)

{

for (int i = 0; i < size; i++)

{

if (num\_arr[i] == num) return true;

}

return false;

}

int main()

{

setlocale(LC\_ALL, "russian");

int U[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 };

int A[] = { 1, 2, 6, 9 };

int B[] = { 2, 3, 5, 6, 7 };

int C[] = { 2, 3, 5, 6, 8 };

/\*\* Boolean \*\*/

cout << "Boolean: \n";

int r, size\_A;

size\_A = sizeof(A) / sizeof(\*A);

r = 1 << size\_A;

for (int i = 0; i < r; i++)

{

print(A, i, 0);

cout << "\n";

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\* Union \*\*/

int size\_B = sizeof(B) / sizeof(\*B);

cout << "\nUnion A and B:\n";

bool A\_finish = false, B\_finish = false;

for (int i = 0, j = 0; !A\_finish || !B\_finish; )

{

if (A\_finish && !B\_finish)

{

cout << B[j] << " ";

if (j == size\_B - 1) { B\_finish = true; continue; }

j++;

}

else if (B\_finish && !A\_finish)

{

cout << A[i] << " ";

if (i == size\_A - 1) { A\_finish = true; continue; }

i++;

}

else

{

if (A[i] < B[j])

{

cout << A[i] << " ";

if (i == size\_A - 1) { A\_finish = true; continue; }

i++;

}

else if (B[j] < A[i])

{

cout << B[j] << " ";

if (j == size\_B - 1) { B\_finish = true; continue; }

j++;

}

else

{

cout << A[i] << " ";

if (i == size\_A - 1) { A\_finish = true; }

else

i++;

if (j == size\_B - 1) { B\_finish = true; }

else

j++;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\* Cartesian product and power \*\*/

int A\_B\_power = size\_A\*size\_B;

cout << "\nCartesian product A and B: \n{";

for (int i = 0; i < size\_A; i++)

{

for (int j = 0; j < size\_B; j++)

{

if (A[i] < B[j])

{

cout << "(" << A[i] << ", ";

cout << B[j];

cout << "), ";

}

else

{

cout << "(" << B[j] << ", ";

cout << A[i];

if (i == size\_A - 1 && j == size\_B - 1) { cout << ")}"; break; }

cout << "), ";

}

}

cout << endl;

}

cout << "\nPower = " << A\_B\_power << endl;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*Expression\*\*/

int size\_C = sizeof(C) / sizeof(\*C);

cout << "\n(A or B) \\ C: \n";

A\_finish = false, B\_finish = false;

for (int i = 0, j = 0; !A\_finish || !B\_finish; )

{

if (!A\_finish&&is\_num\_in\_arr(A[i], C, size\_C))

{

if (i == size\_A - 1) { A\_finish = true; continue; }

i++;

continue;

}

if (!B\_finish&&is\_num\_in\_arr(B[j], C, size\_C))

{

if (j == size\_B - 1) { B\_finish = true; continue; }

j++;

continue;

}

if (A\_finish && !B\_finish)

{

cout << B[j] << " ";

if (j == size\_B - 1) { B\_finish = true; continue; }

j++;

}

else if (B\_finish && !A\_finish)

{

cout << A[i] << " ";

if (i == size\_A - 1) { A\_finish = true; continue; }

i++;

}

else

{

if (A[i] < B[j])

{

cout << A[i] << " ";

if (i == size\_A - 1) { A\_finish = true; continue; }

i++;

}

else if (B[j] < A[i])

{

cout << B[j] << " ";

if (j == size\_B - 1) { B\_finish = true; continue; }

j++;

}

else

{

cout << A[i] << " ";

if (i == size\_A - 1) { A\_finish = true; }

else

i++;

if (j == size\_B - 1) { B\_finish = true; }

else

j++;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

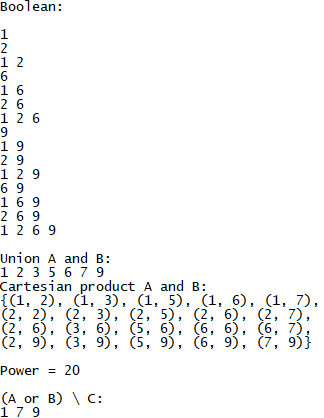
cout << endl;

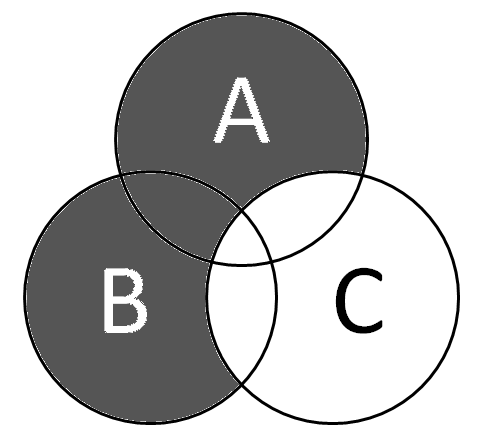
system("pause");

return 0;

}

Вывод:





5. Указать примеры покрытий и разбиения множества *A = {1, 2, 6, 9}*.

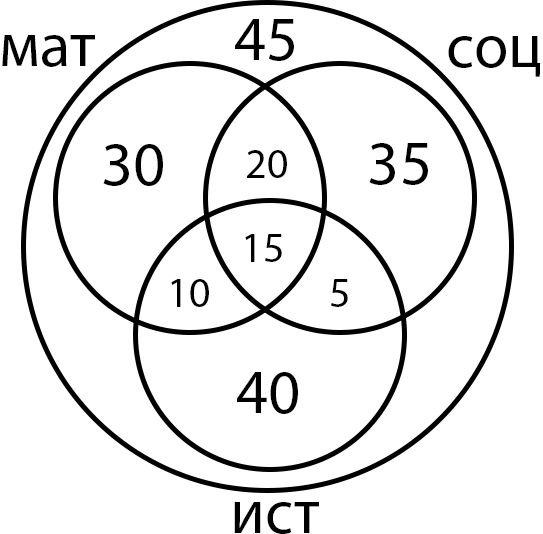
**{**{1, 6}, {9}, {2}**}**; **{**{1, 9}, {2, 6}**}** – разбиения;

**{**{2, 6}, {1, 2, 9}**}**; **{**{1, 2}, {6, 9}, {1, 6}, {9}**}** – покрытия.

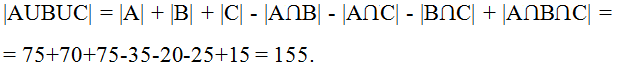
***Задание 2 (вариант 1).***

Решить задачу на применение формулы включений и исключений.

В группе из 200 студентов 75 изучают математику, 70 - историю, 75 - социологию, 35 изучают математику и социологию, 20 - историю и социологию, 25 изучают математику и историю, 15 студентов - все три предмета.



а) Сколько студентов изучают хотя бы один из трех предметов?



б) Сколько студентов изучают только один из трех предметов?

30+35+40 = 105.

в) Сколько студентов изучают историю или математику, но не изучают социологию?

30+40+10= 80.

г) Сколько студентов не изучают ровно два из трех предметов?

45+30+35+40+15 = 165.

д) Сколько студентов не выбрали историю или математику?

45+20+15+5+35=120.