

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту



ЗВІТ

Про виконання лабораторної роботи № 2

З дисципліни

«Дискретна математика»

Студентки групи ІТ-11

Проців Роксолани Василівни

Прийняв викладач

Юринець Р.В.

Множина. Кортеж. Декартів добуток

Мета роботи: Вивчення множин, операцій над множинами, комп'ютерного подання множин.

Хід роботи:

Варіант №20

Завдання 1: Задана універсальна множина $U=\{1,2,3,4,\dots,25\}$ та три її підмножини A, B, C. Знайти F

$A = \{1,8,14,18,19,22,23,24,25\}$; $B = \{1,4,5,10,11,17,18,23,25\}$; $C = \{1,10,11,12,15,19,20,24,25\}$

$$F = A \cup B \setminus A \cap C$$

```
#include <iostream>
#include <cmath>
#include<algorithm>

using namespace std;

int UniversalPluralSize = 25;
int UniversalPlural[] = {1,2,3,4,5,6,7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25};

//input
int A[9] = { 1,8,14,18,19,22,23,24,25 };
int B[9] = { 1,4,5,10,11,17,18,23,25 };
int C[9] = { 1,10,11,12,15,19,20,24,25 };

const int SizeA = 9;
const int SizeB = 9;
const int SizeC = 9;

void PrintPluar(const int* z, const int size)
{
    cout << "{";
    for (int i = 0; i < size; i++)
    {
        cout << z[i] << " ";
    }
    cout << "}" << endl;
}

int* Add(const int* a, const int size_a, const int* b, const int size_b, int& size)
{
    size = size_a + size_b;
    int* res = new int[size];
    int total = 0;

    for (int i = 0; i < size_a; i)
    {
        res[i] = a[i];
        i++;
        total++;
    }
}
```

```

        for (int i = 0; i < size_b && total < size; i++, total++)
        {
            res[total] = b[i];
        }

        return res;
    }
    int* Subtraction(const int* a, const int size_a, const int* b, const int size_b, int&
size)
    {
        int index = 0;
        int* tmp = new int[size_a];
        for (int i = 0; i < size_a; i++)
        {
            for (int j = 0; j < size_b; j++)
            {
                if (a[i] == b[j])
                {
                    tmp[index] = i;
                    index++;
                }
            }
        }

        size = size_a - index;
        int* res = new int[size];

        for (int i = 0, count = 0; i < size_a; i++)
        {
            if (i == tmp[count])
            {
                count++;
            }
            else
            {
                res[i - count] = a[i];
            }
        }

        delete[] tmp;
        tmp = nullptr;

        return res;
    }
    int* Unification(const int* a, const int size_a, const int* b, const int size_b, int&
size)
    {
        int* tmp = new int[size_b];
        int indexTmpArray = 0;

        for (int i = 0; i < size_a; i++)
        {
            for (int j = 0; j < size_b; j++)
            {
                if (a[i] == b[j])
                {
                    tmp[indexTmpArray] = a[i];
                    indexTmpArray++;
                }
            }
        }

        int* tmp1 = new int[indexTmpArray];
        for (int i = 0; i < indexTmpArray; i++)
        {

```

```

        tmp1[i] = tmp[i];
    }

    delete[] tmp;
    tmp = nullptr;
    int s;
    int* tmp2 = Subtraction(b, size_b, tmp1, indexTmpArray, s);

    delete[] tmp1;

    int* res = Add(tmp2, s, a, size_a, size);

    return res;
}

int* Intersection(const int* a, const int size_a, const int* b, const int size_b, int&
size)
{
    int s = 0;
    for (int i = 0; i < size_a; i++)
    {
        for (int j = 0; j < size_b; j++)
        {
            if (a[i] == b[j])
            {
                s++;
            }
        }
    }

    size = s;
    int* res = new int[size];
    for (int i = 0, s = 0; i < size_a; i++)
    {
        for (int j = 0; j < size_b; j++)
        {
            if (a[i] == b[j])
            {
                res[s] = a[i];
                s++;
            }
        }
    }

    return res;
}

// (A U B) \ (A^B)
int main()
{
    cout << "[U]"; PrintPluar(UniversalPlural, UniversalPluralSize);
    cout << "[A]"; PrintPluar(A, SizeA);
    cout << "[B]"; PrintPluar(B, SizeB);
    cout << "[C]"; PrintPluar(C, SizeC);
    cout << endl;

    // A u B
    int size_z1;
    int* z1;
    z1 = Unification(A, SizeA, B, SizeB, size_z1);
    cout << "[A U B]"; PrintPluar(z1, size_z1);

    // A^C
    int size_z2;
    int* z2;

```

```

z2 = Intersection(A, SizeA, C, SizeC, size_z2);
cout << "[A ^^ C]"; PrintPluar(z2, size_z2);

// (A u B) \ A^^C
int size_z3;
int* z3;
z3 = Subtraction(z1, size_z1, z2, size_z2, size_z3);
cout << "[(A u B) \ (A^^C)]"; PrintPluar(z3, size_z3);

return 0;
}

```

```

Microsoft Visual Studio Debug Console

[U]{1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 }
[A]{1 8 14 18 19 22 23 24 25 }
[B]{1 4 5 10 11 17 18 23 25 }
[C]{1 10 11 12 15 19 20 24 25 }

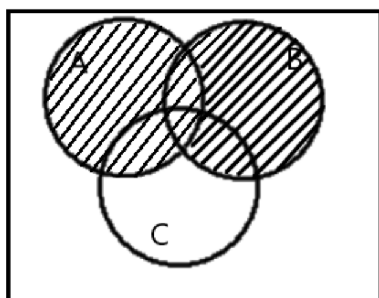
[A u B]{4 5 10 11 17 1 8 14 18 19 22 23 24 25 }
[A ^^ C]{1 19 24 25 }
[(A u B) \ (A^^C)]{4 5 10 11 17 8 14 18 22 23 }

C:\Users\User\source\repos\дискретна\Dyskretna 2\Debug\Dyskretna 2.exe (process 14500) exited with code 0.
Press any key to close this window . . .

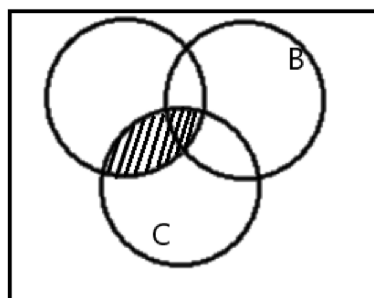
```

Завдання 2: Для формули побудувати діаграми Венна за допомогою будь-якого графічного редактора.

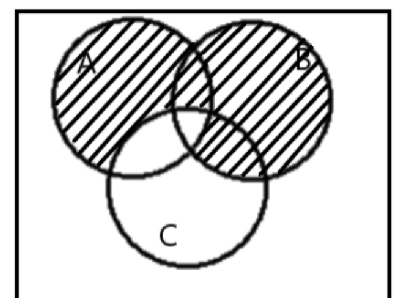
$$A \cup B \setminus A \cap C$$



$A \cup B$



$A \cap C$



$A \cup B \setminus A \cap C$

Завдання 3: Задано множини A та B. Знайти декартовий добуток $A \times B$ та A^2

$$A = \{1,2\}; B = \{1,2,3,4,8\}$$

$$A \times B = \{(1;1), (1;2), (1;3), (1;4), (1;8), (2;1), (2;2), (2;3), (2;4), (2;8)\}$$

$$A^2 = A \times A = \{(1;1), (1;2), (2;1), (2;2)\}$$

Висновок: при виконанні лабораторної роботи я вивчила множини, операції над ними та комп'ютерне подання множин.