Міністерство освіти і науки, молоді та спорту України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання лабораторної роботи № 1**

**«****Логіка висловлювань та предикатів. Комп’ютерне подання множин»**

(до тем «Математична логіка і доведення», «Теорія множин»)

**з дисципліни «Комп’ютерна дискретна математика »**

**Лектор:**

проф. кафедри ПЗ

Журавчак Л.М.

**Виконав:**

студ. групи ПЗ-14

Губик А. С.

**Прийняв:**

асистент кафедри ПЗ

Курапов П. P.

«\_\_\_» \_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2022

**ТЕМА РОБОТИ**: Логіка висловлювань та предикатів. Комп’ютерне подання множин

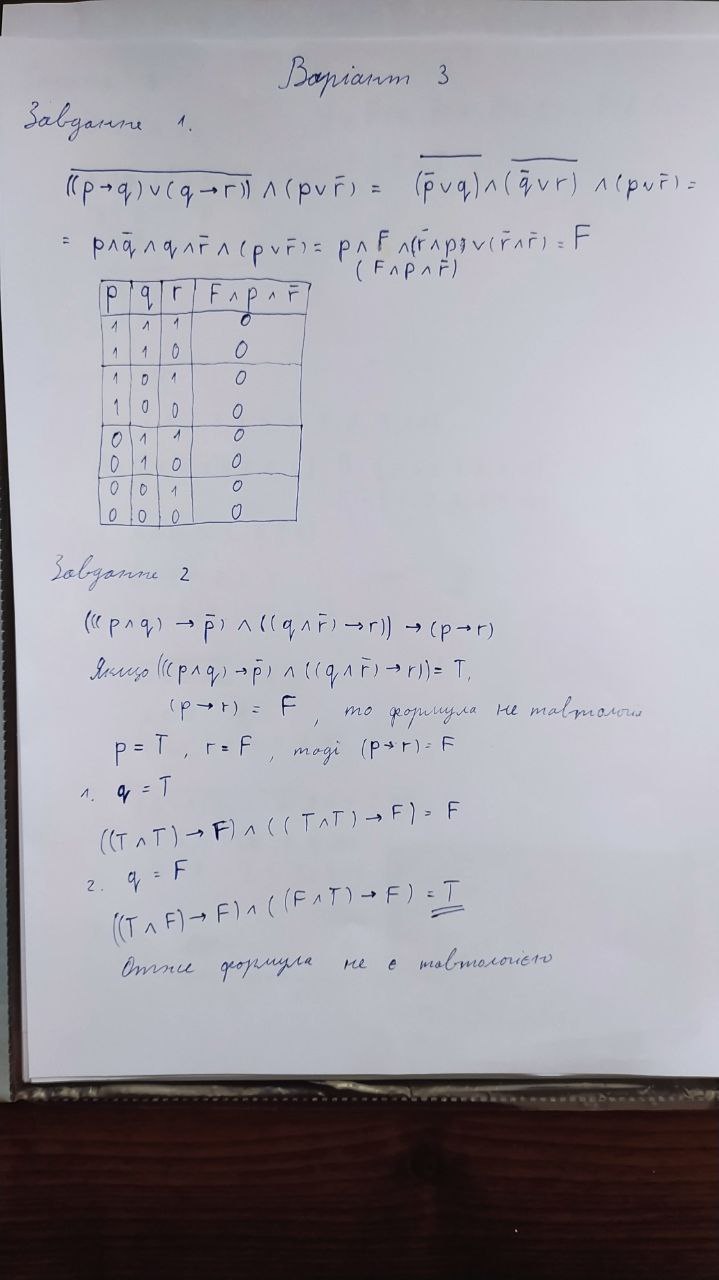
**МЕТА РОБОТИ**: Ознайомитись на практиці із основними поняттями математичної логіки і теорії множин, навчитись знаходити істинностні значення складних логічних висловлювань та засвоїти операції над множинами, використовуючи їхнє комп’ютерне подання.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ № 1

1.Реалізувати програмний код з використанням будь-якої відомої студентові мови програмування для знаходження істинністних значень логічних висловлювань при всіх можливих інтерпретаціях.

/\ (p \/ )

ОПИС ВИКОНАННЯ РОБОТИ



ТЕКСТ ПРОГРАМИ

#include <stdio.h>

void DrawLine(int n){

for(int i = 0; i < n; i++){

printf("-");

}

printf("\n");

}

int SolveEquasion(int id, int p, int q, int r){

if (id == 1){

return !((p <= q) || (q <= r)) && (p || !r);

}else{

return (((p && q) <= !p) && ((q && !r) <= r)) <= (p <= r);

}

}

int main(){

static int ArrLength = 8,

LineLength = 22;

static int P[] = {1, 1, 1, 1, 0, 0, 0, 0},

Q[] = {1, 1, 0, 0, 1, 1, 0, 0},

R[] = {1, 0, 1, 0, 1, 0, 1, 0};

int res[ArrLength];

int id;

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \n");

printf("#1: ((P -> Q) v (Q -> R)) ^ (P v R)\n");

printf(" \_ \_ \n");

printf("#2: (((P ^ Q) -> P) ^ ((Q ^ R) -> R)) -> (P -> R)\n\n");

printf("Enter number of equasion: ");

scanf("%d", &id);

DrawLine(LineLength);

printf("| P | Q | R | Result |\n");

DrawLine(LineLength);

for (int i = 0; i < ArrLength; i++){

res[i] = SolveEquasion(id, P[i], Q[i], R[i]);

printf("| %d | %d | %d | %d |\n", P[i], Q[i], R[i], res[i]);

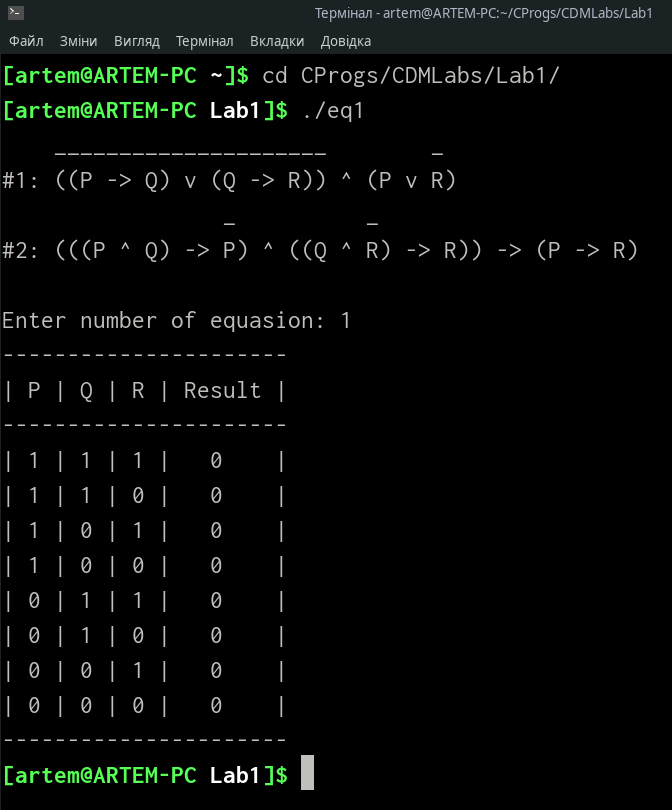
}

DrawLine(LineLength);

return 0;

}

РЕЗУЛЬТАТИ

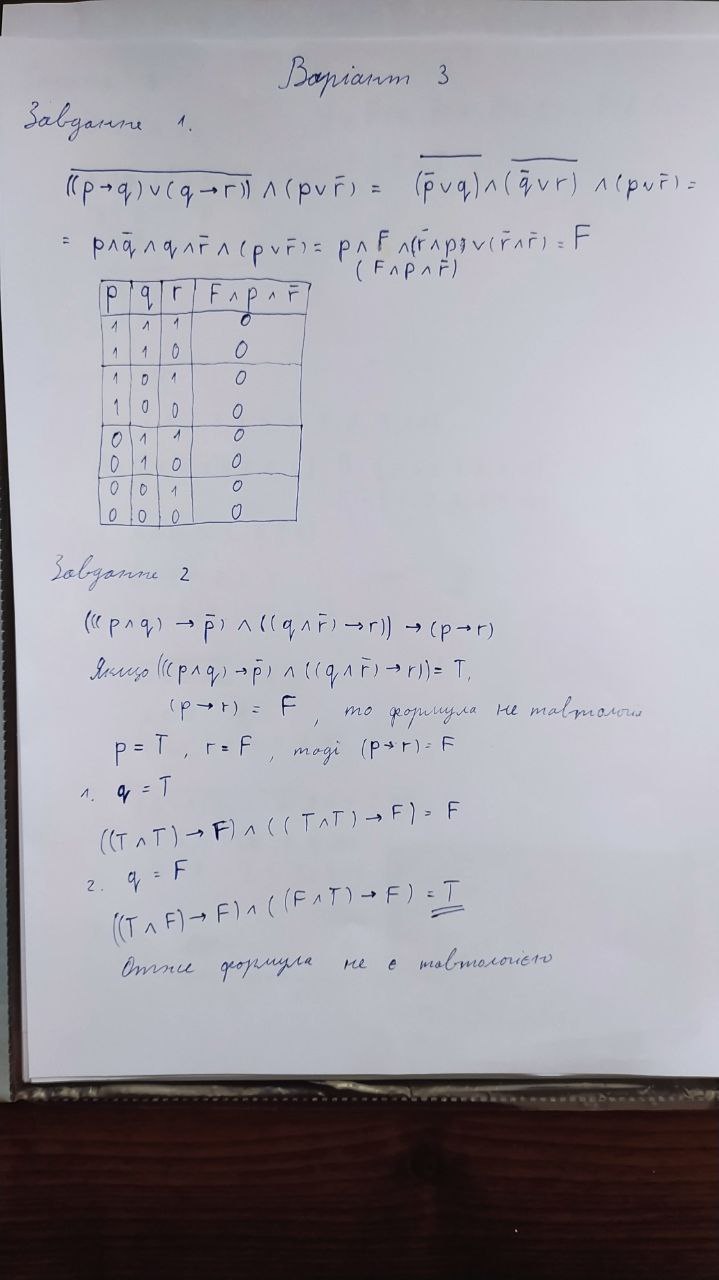


ІНДИВІДУАЛЬНЕ ЗАВДАННЯ № 2

Методом відшукання контрприкладу перевірити, чи є тавтологією висловлювання

((( p ^ q) ) ^ ((q ^ r) )) ( p r). Протестувати розробленою для завдання № 1 програмою.

ОПИС ВИКОНАННЯ РОБОТИ



ТЕКСТ ПРОГРАМИ

#include <stdio.h>

void DrawLine(int n){

for(int i = 0; i < n; i++){

printf("-");

}

printf("\n");

}

int SolveEquasion(int id, int p, int q, int r){

if (id == 1){

return !((p <= q) || (q <= r)) && (p || !r);

}else{

return (((p && q) <= !p) && ((q && !r) <= r)) <= (p <= r);

}

}

int main(){

static int ArrLength = 8,

LineLength = 22;

static int P[] = {1, 1, 1, 1, 0, 0, 0, 0},

Q[] = {1, 1, 0, 0, 1, 1, 0, 0},

R[] = {1, 0, 1, 0, 1, 0, 1, 0};

int res[ArrLength];

int id;

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \n");

printf("#1: ((P -> Q) v (Q -> R)) ^ (P v R)\n");

printf(" \_ \_ \n");

printf("#2: (((P ^ Q) -> P) ^ ((Q ^ R) -> R)) -> (P -> R)\n\n");

printf("Enter number of equasion: ");

scanf("%d", &id);

DrawLine(LineLength);

printf("| P | Q | R | Result |\n");

DrawLine(LineLength);

for (int i = 0; i < ArrLength; i++){

res[i] = SolveEquasion(id, P[i], Q[i], R[i]);

printf("| %d | %d | %d | %d |\n", P[i], Q[i], R[i], res[i]);

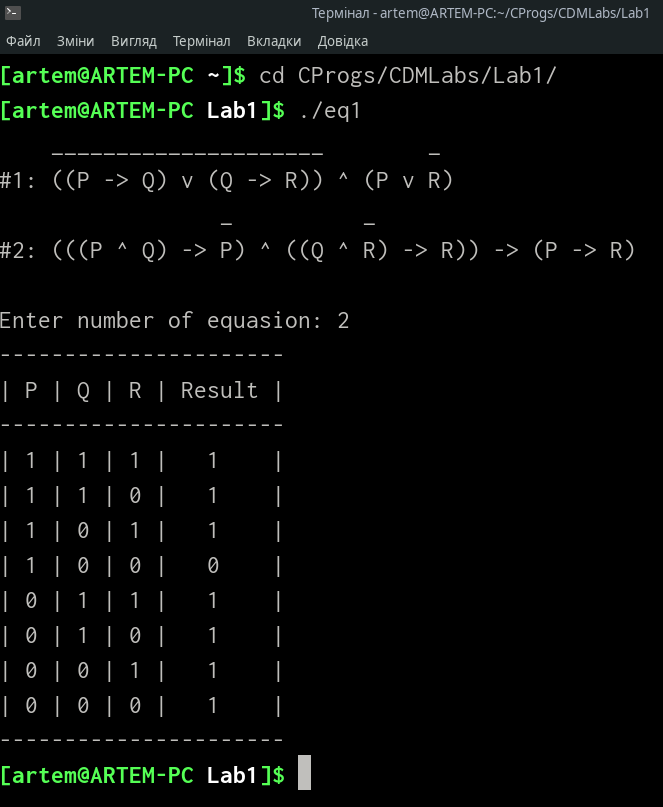
}

DrawLine(LineLength);

return 0;

}

**РЕЗУЛЬТАТИ**



ІНДИВІДУАЛЬНЕ ЗАВДАННЯ № 3

Задано предикати P(x,y) та Q(x,y) у предметній області D{a,b} таблицею:

x a a b b

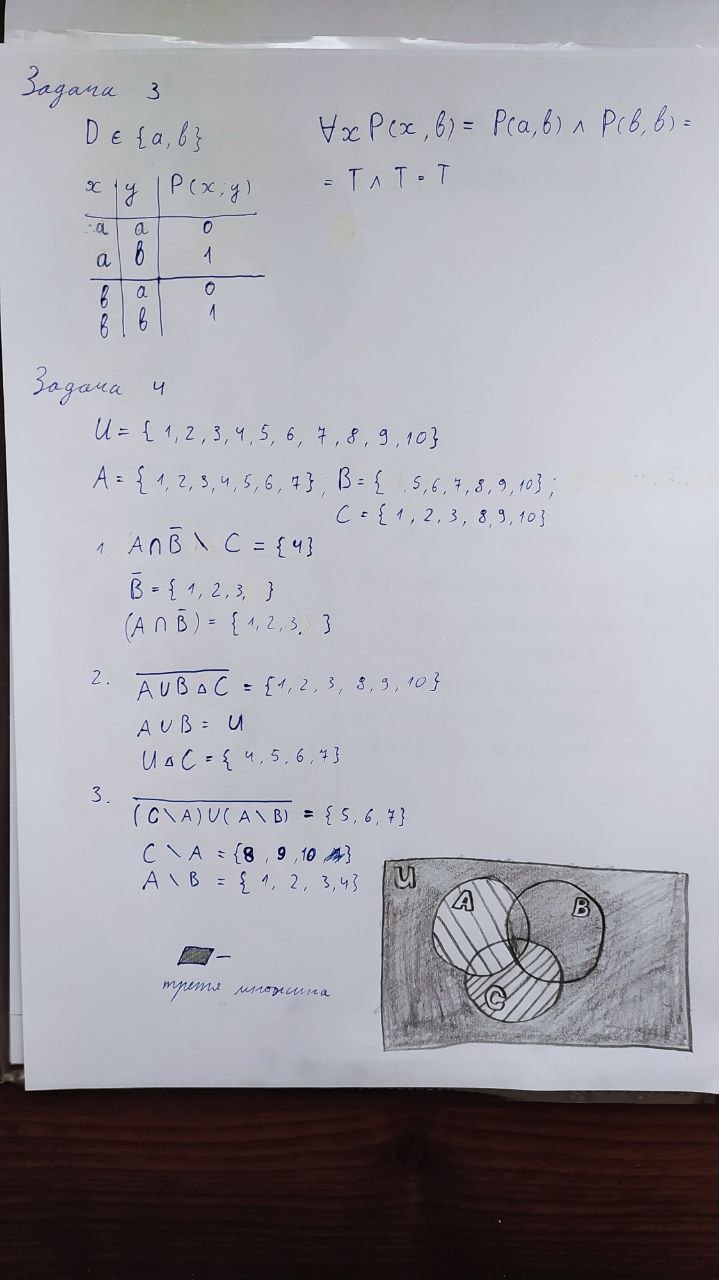
y a b a b

P 0 1 0 1

Q 1 0 0 1

Реалізувати програмний код з використанням будь-якої відомої студентові мови програмування для визначення, чи є формула з табл. 1.3 істинною (хибною).

ОПИС ВИКОНАННЯ РОБОТИ



ТЕКСТ ПРОГРАМИ

#include <stdio.h>

int P(char x, char c){

if (x == 'a' && c == 'a')

return 0;

if (x == 'a' && c == 'b')

return 1;

if (x == 'b' && c == 'a')

return 0;

if (x == 'b' && c == 'b')

return 1;

return 0;

}

int main(){

int res = 1;

for(int x = 'a'; x <= 'b'; ++x)

res = res && P(x, 'b');

printf("For all x is P(x, b).\n");

if (res){

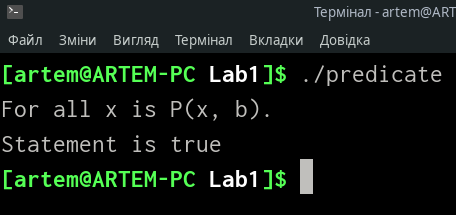
printf("Statement is true\n");

}else{

printf("Statement is false\n");

}return 0;

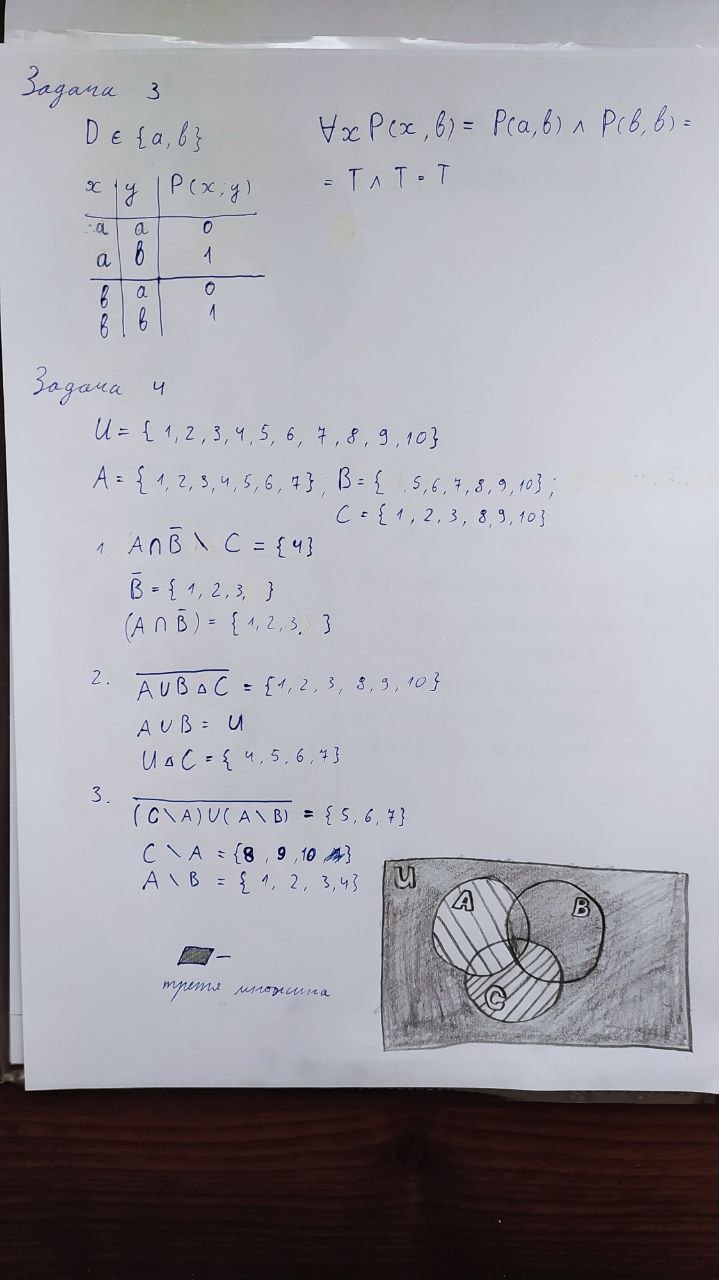
**РЕЗУЛЬТАТИ**



ІНДИВІДУАЛЬНЕ ЗАВДАННЯ № 4

Використовуючи комп’ютерне подання множин, реалізувати програмний код на будь-якій відомій студентові мові програмування для знаходження трьох множин, заданих за допомогою операцій (табл. 1.4) та побудови булеану і знаходження потужності третьої множини. Універсум U = {1,2,3,4,5,6,7,8,9,10}. Зобразити на діаграмі Ейлера-Венна третю множину.

ОПИС ВИКОНАННЯ РОБОТИ



ТЕКСТ ПРОГРАМИ

#include <stdio.h>

#include <stdlib.h>

enum{

UniversumSize = 10,

ASize = 7,

BSize = 6,

CSize = 6,

};

int SolveEquasion(int id, int a[], int b[], int c[], int i){

if (id == 1)

return a[i] && !b[i] && !c[i];

if (id == 2)

return !((a[i] || b[i]) != c[i]);

if (id == 3)

return !(c[i] && !a[i]) && !(a[i] && !b[i]);

}

int CalculateArraySize(int id, int a[], int b[], int c[]){

int XSize = 0;

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

XSize += SolveEquasion(id, a, b, c, i);

return XSize;

}

void ConstructArray(int id, int X[],

int a[], int b[], int c[], const int U[]){

int j = 0;

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

if(SolveEquasion(id, a, b, c, i)){

X[j] = U[i];

j++;

}

}

void ConstructBitSet(const int U[], const int X[], int XSize, int x[]){

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

for(int j = 0; j < XSize; ++j)

if (U[i] == X[j]){

x[i] = 1;

break;

}

}

void PrintArray(int X[], int l){

printf("{");

for(int i = 0; i < l - 1; ++i)

printf("%d, ", X[i]);

printf("%d}", X[l - 1]);

}

void PrintBoolean(int X[], int XSize){

int \*tmp;

printf("\n{{},\n");

tmp = malloc(sizeof(int));

for(int i = 0; i < XSize; ++i){

\*tmp = X[i];

PrintArray(tmp, 1);

printf(", ");

}

printf("\n");

tmp = malloc(2 \* sizeof(int));

for(int i = 0; i < XSize - 1; ++i){

tmp[0] = X[i];

for(int j = 1; j < XSize; ++j){

if (i == j)

continue;

tmp[1] = X[j];

PrintArray(tmp, 2);

printf(", ");

}

}

free(tmp);

printf("\n");

PrintArray(X, XSize);

printf("}\n");

}

int main(){

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

A[ASize] = {1, 2, 3, 4, 5, 6, 7},

B[BSize] = {5, 6, 7, 8, 9, 10},

C[CSize] = {1, 2, 3, 8, 9, 10};

int a[UniversumSize] = {0},

b[UniversumSize] = {0},

c[UniversumSize] = {0};

int S1Size, S2Size, S3Size;

int \*S1, \*S2, \*S3;

ConstructBitSet(U, A, ASize, a);

ConstructBitSet(U, B, BSize, b);

ConstructBitSet(U, C, CSize, c);

S1Size = CalculateArraySize(1, a, b, c);

S2Size = CalculateArraySize(2, a, b, c);

S3Size = CalculateArraySize(3, a, b, c);

S1 = malloc(S1Size \* sizeof(int));

S2 = malloc(S2Size \* sizeof(int));

S3 = malloc(S3Size \* sizeof(int));

ConstructArray(1, S1, a, b, c, U);

ConstructArray(2, S2, a, b, c, U);

ConstructArray(3, S3, a, b, c, U);

printf("First set: ");

PrintArray(S1, S1Size);

printf("\n");

printf("Second set: ");

PrintArray(S2, S2Size);

printf("\n");

printf("Third set: ");

PrintArray(S3, S3Size);

printf("\n");

printf("Third set power: %d\n", S3Size);

printf("Third set boolean: ");

PrintBoolean(S3, S3Size);

free(S1);

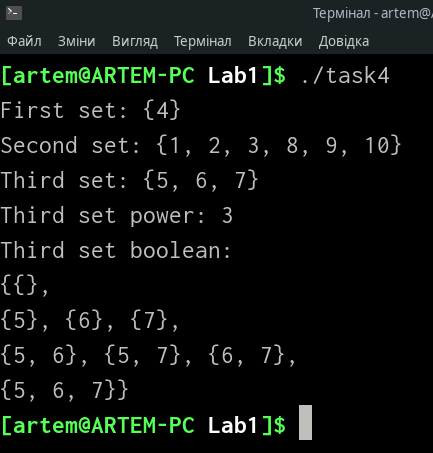
free(S2);

free(S3);

return 0;

}

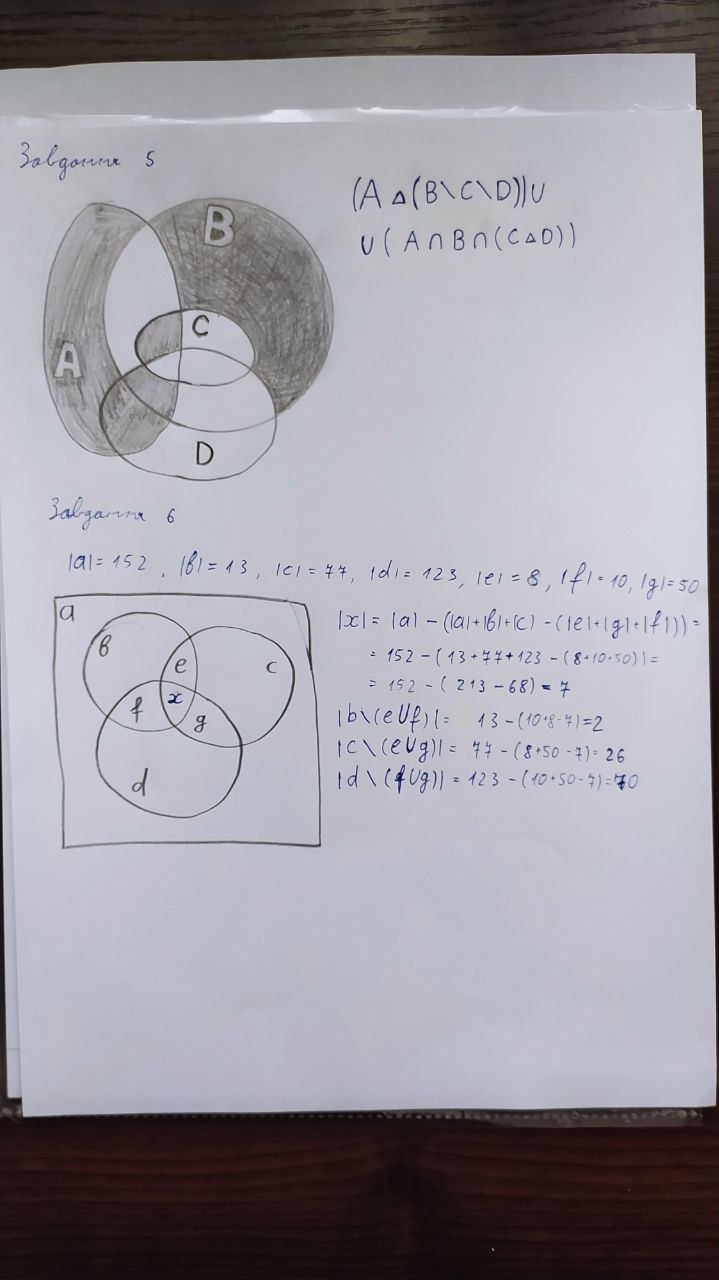
**РЕЗУЛЬТАТИ**



ІНДИВІДУАЛЬНЕ ЗАВДАННЯ № 1

1.Реалізувати програмний код з використанням будь-якої відомої студентові мови програмування для знаходження істинністних значень логічних висловлювань (табл. 1.1) при всіх можливих інтерпретаціях.

**ОПИС ВИКОНАННЯ РОБОТИ**



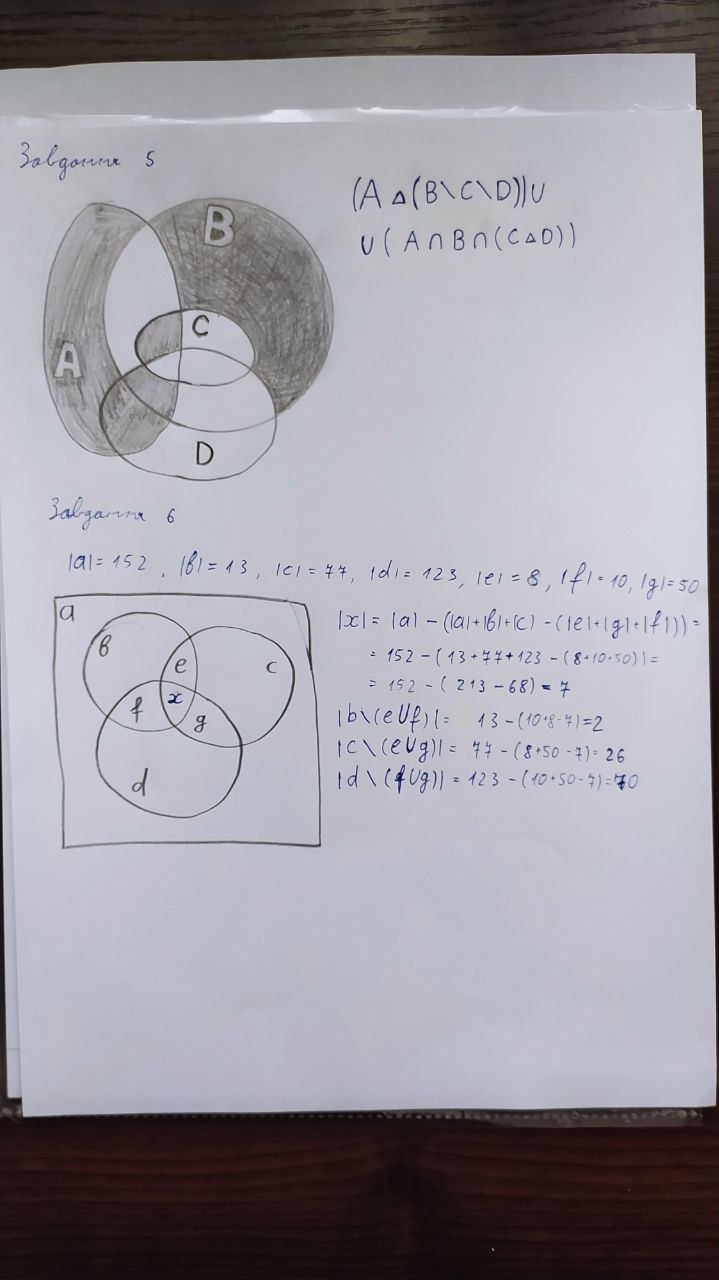
ІНДИВІДУАЛЬНЕ ЗАВДАННЯ № 5

ЕОМ має a мікропроцесорів, з них b мікропроцесорів обробляють текстову інформацію, c – графічну, d – символьну, e мікропроцесорів одночасно обробляють графічну та текстову, f – текстову та символьну, g – графічну та символьну, а частина мікропроцесорів одночасно обробляють графічну, текстову та символьну інформацію (табл. 1.6). Написати на будь-якій відомій студентові мові програмування програмний код для визначення, чи можлива така реалізація ЕОМ. Програмно обчислити, скільки мікропроцесорів обробляють лише текстову інформацію, скільки лише графічну, скільки лише символьну.

№ вар. a b c d e f g

3 152 13 77 123 8 10 50

ОПИС ВИКОНАННЯ РОБОТИ



ТЕКСТ ПРОГРАМИ

#include <stdio.h>

enum{

a = 152, b = 13, c = 77, d = 123,

e = 8, f = 10, g = 50

};

int main(){

int x, text, graphics, symbols;

x = a - (b + c + d - (e + f + g));

if (x < 0){

printf("Such a system doesn\'t exist\n");

return 0;

}

text = b - (e + f - x);

graphics = c - (e + g - x);

symbols = d - (f + g - x);

printf("Only text: %d \n", text);

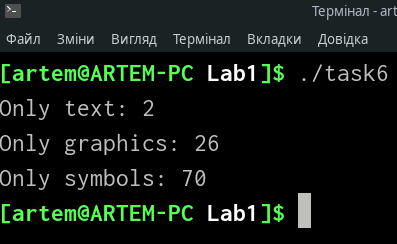
printf("Only graphics: %d \n", graphics);

printf("Only symbols: %d \n", symbols);

return 0;

}

**РЕЗУЛЬТАТИ**



ВИСНОВКИ

Подаючи логічні висловлювання, як логічні операції над числами, а множини, як масиви елементів, можна написати програму для обчислення довільних логічних виразів та операцій над множинами.