**МІНІСТЕРСТО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**



**Дискретна математика**

Лабораторна робота №1

Операції над множинами

**Виконав:**

Студент групи КІ-107

Дигдало Т.Л.

**Перевірив:**

Попович Б.Р.

Львів – 2023

**Тема**. Операції над множинами.

**Мета**. Засвоїти способи задання множин та операції над множинами.

**Теоретичні відомості**

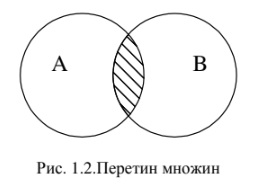
Операції над множинами

Розглянемо дві множини А і В та введемо низку операцій над ними. Для графічної ілюстрації використовують діаграми (кола) Ейлера. Для зображення множини на площині креслять замкнену лінію із заштрихованою внутрішньою областю (найчастіше – це коло, звідси й назва відповідного інструмента, що широко застосовується в теорії множин).

Об’єднання А і В – множина, що складається з усіх елементів множини А,

всіх елементів множини В і не містить ніяких інших елементів (рис. 1.1), тобто

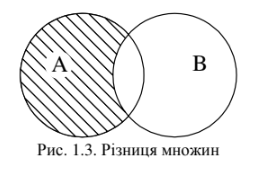
А ∪ В = {x | x ∈ А або x ∈ В}.



Перетин А і В – множина, що складається з тих і тільки з тих елементів,

які належать одночасно множині А та множині В (рис. 1.2), тобто

А ∩ В = {x | x ∈ А і x ∈ В}.



Різниця А і В (відносне доповнення) – множина, що складається з тих і

тільки тих елементів, які належать множині А й не належать множині В

(рис. 1.3), тобто А \ В = {x | x ∈ А і x ∉ В}.



Диз’юнктивна сума А і В (симетрична різниця) – множина, що

складається усіх елементів А, які не належать множині В, й усіх елементів В,

які не належать множині А, та яка не містить ніяких інших елементів (рис. 1.4),

тобто А ⊕ В = {x | (x ∈ А і x ∉ В) або (x ∈ В і x ∉ А)}.

Зручно сукупність допустимих об’єктів зафіксувати явно та вважати, що множини, які

розглядаються, складаються з елементів цієї сукупності. Її називають основною

множиною (універсумом) і позначають U. Універсум U арифметики – числа,

універсум U зоології – тварини і т.д. Будь-яку множину розглядатимемо у

зв’язку з універсумом, який на діаграмах Ейлера асоціюватимемо з

прямокутником на площині, всередині якого зображатимемо множини (рис. 1.5).



Доповнення множини А – це множина, що містить усі елементи

універсуму, за винятком елементів А (рис. 1.6),

тобто A = {x | x ∈U, x ∉ A} .

Множина А називається підмножиною множини В, якщо кожен елемент А є елементом В.

Для позначення цього факту вводиться

знак ⊂ - символ

строгого включення (або ⊆ - символ нестрогого включення) (рис. 1.7)

Якщо необхідно підкреслити, що множина В містить також інші елементи, крім

елементів множини А, то використовують символ строгого включення А ⊂ В.

Дві множини рівні, якщо вони складаються з одних і тих самих

елементів. Справджується таке: А = В тоді і тільки тоді, коли А ⊆ В і В ⊆ А.

1.2. Порядок виконання роботи

1. Складіть програму, яка як вхідні дані одержує дві множини і визначає, чи рівні ці множини, чи є одна з них підмножиною іншої.

Код програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <string>

#include <algorithm>

#include <locale.h>

using namespace std;

void BringOutMas(int n, int\* m)

{

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

cout << m[i] << ' ';

cout << endl;

}

void EnterMas(int n, int\* m)

{

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

{

cout << "\nEnter " << i << " the element:";

cin >> m[i];

}

}

bool SubSet(int n1, int\* a, int n2, int\* b)

{

for (int i = 0; i < n2; i++)

{

bool notfound = true;

for (int j = 0; j < n1; j++)

{

if (b[i] == a[j])

{

notfound = false;

break;

}

}

if (notfound)return false;

}

return true;

}

int main()

{

setlocale(LC\_ALL, "UKRAINIAN");

cout << "\nEnter the number of elements of the first array:";

int nx; cin >> nx;

int\* x = new int[nx];

EnterMas(nx, x);

cout << "\nEnter the number of elements of the second array:";

int ny; cin >> ny;

int\* y = new int[ny];

EnterMas(ny, y);

if ((SubSet(nx, x, ny, y)) && (SubSet(ny, y, nx, x))) {

cout << "\nthey are equal\n";

}

else

{

if

(SubSet(nx, x, ny, y)) {

cout << "\ny belongs to x\n";

}

else

{

cout << "\nno set belongs to another\n";

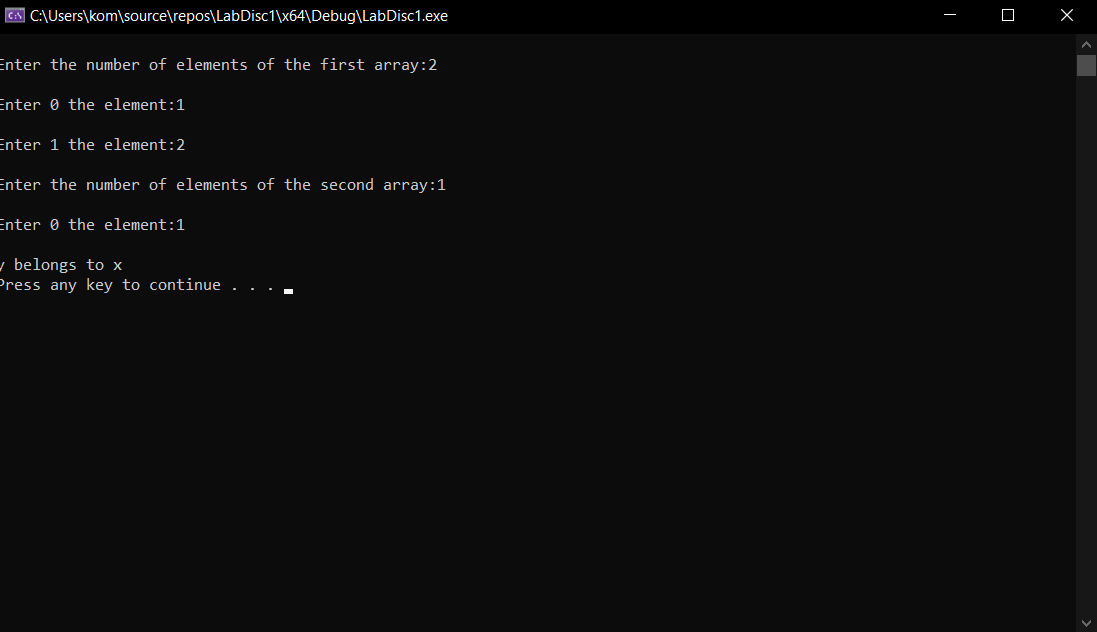
}

}

system("pause");

return 0;

}







2. Складіть програму, яка як вхідні дані одержує множину і утворює список всіх можливих підмножин даної множини.

Код програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <string>

#include <algorithm>

#include <locale.h>

using namespace std;

int main()

{

int x;

cout << "Enter the size:";

cin >> x;

int\* mas = new int;

mas = new int[x];

cout << "Enter elements:\n";

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

{

cin >> mas[j];

}

int i, mx, j;

for (i = 0, mx = 1; i < x; i++)

{

mx \*= 2;

}

for (i = 0; i < mx; i++)

{

for (j = 0; j < x; j++)

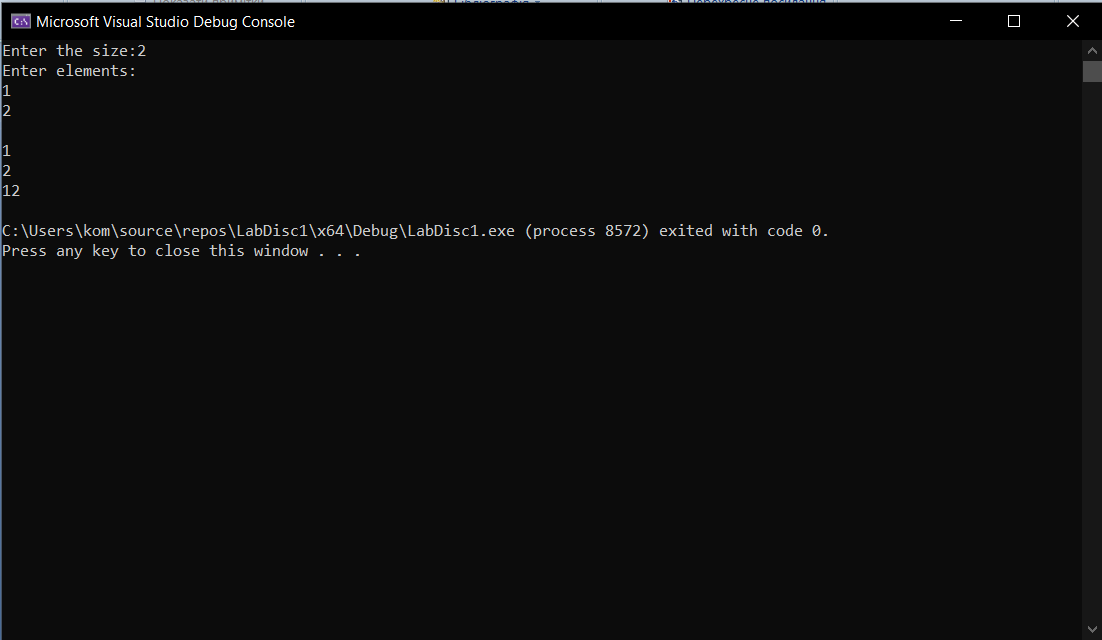
if ((i >> j) & 1) std::cout << mas[j];

std::cout << ::endl;

}

return 0;

}





3. Складіть програму, яка як вхідні дані одержує дві множини A, B і утворює декартові добутки A×B та B×A.

Код програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <string>

#include <algorithm>

#include <locale.h>

using namespace std;

int main()

{

unsigned int A;

cout << "Enter Size A =" << " ";

cin >> A;

int\* ArrayA = new int[A];

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

cout << "ary[" << i << "]:";

cin >> ArrayA[i];

}

unsigned int B;

cout << "Enter Size B =" << " ";

cin >> B;

int\* ArrayB = new int[B];

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

cout << "ary[" << i << "]:";

cin >> ArrayB[i];

}

std::cout << "AxB" << std::endl;

for (int j = 0; j < A; j++) {

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

std::cout << "(" << ArrayA[j] << "," << ArrayB[n] << ")" << std::endl;

}

}

std::cout << "BxA" << std::endl;

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

for (int j = 0; j < A; j++) {

std::cout << "(" << ArrayB[n] << "," << ArrayA[j] << ")" << std::endl;

}

}

}





4. Складіть програму, яка моделює операції A ∪ B , A ∩ B , A \ B , A ⊕ B , A над множинами у графічному режимі.

Код програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <string>

#include <algorithm>

#include <locale.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "UKRAINIAN");

int choose\_number;

printf("1 - set of sets \n");

printf("2 - intersection of sets \n");

printf("3 - set difference \n");

printf("4 - the disjunctive sum of sets \n");

printf("5 - complement set \n");

printf("\n Select the set operation from the list:");

scanf("%d", &choose\_number);

switch (choose\_number) {

case 1:

{

unsigned int A;

printf("\nEnter the size of the set A: ");

scanf("%d", &A);

int ArrayA[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayA[i]);

}

unsigned int B;

printf("\nEnter the size of the set B: ");

scanf("%d", &B);

int ArrayB[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayB[i]);

}

printf("\nCombining sets A and B: ");

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

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

}

for (int j = 0; j < B; j++) {

printf("%d, ", ArrayB[j]);

}

printf("\n");

return 0;

}

case 2:

{

unsigned int A;

printf("\nEnter the size of the set A: ");

scanf("%d", &A);

int ArrayA[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayA[i]);

}

unsigned int B;

printf("\nEnter the size of the set B: ");

scanf("%d", &B);

int ArrayB[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayB[i]);

}

printf("\nIntersection of sets A and B: ");

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

for (int j = 0; j < B; j++) {

if (ArrayA[i] == ArrayB[j]) {

int coincidence = ArrayA[i];

printf("%d ", coincidence);

}

}

}

return 0;

}

case 3:

{

unsigned int A;

printf("\nEnter the size of the set A: ");

scanf("%d", &A);

int ArrayA[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayA[i]);

}

unsigned int B;

printf("\nEnter the size of the set B: ");

scanf("%d", &B);

int ArrayB[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayB[i]);

}

printf("\nThe difference between the sets A and B: ");

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

int j = 0;

while (j < B && ArrayB[j] != ArrayA[i]) {

j++;

if (j == B) {

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

}

}

}

return 0;

}

case 4:

{

unsigned int A;

printf("\nEnter the size of the set A: ");

scanf("%d", &A);

int ArrayA[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayA[i]);

}

unsigned int B;

printf("\nEnter the size of the set B: ");

scanf("%d", &B);

int ArrayB[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayB[i]);

}

int a = 0;

printf("\nDisjunctive sum of sets A and B: ");

do {

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

int j = 0;

while (j < B && ArrayB[j] != ArrayA[i]) {

j++;

if (j == B) {

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

}

}

}

for (int j = 0; j < B; j++) {

int i = 0;

while (i < A && ArrayA[i] != ArrayB[j]) {

i++;

if (i == A) {

printf("%d ", ArrayB[j]);

}

}

}

} while (a = 0);

return 0;

}

case 5:

{

unsigned int A;

printf("\nEnter the size of the set A: ");

scanf("%d", &A);

int ArrayA[100];

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

printf("array[%d]: ", i);

scanf("%d", &ArrayA[i]);

}

printf("\nAddition of the set A: any number of the universe, except: \n");

for (int j = 0; j < A; j++) {

printf("%d ", ArrayA[j]);

}

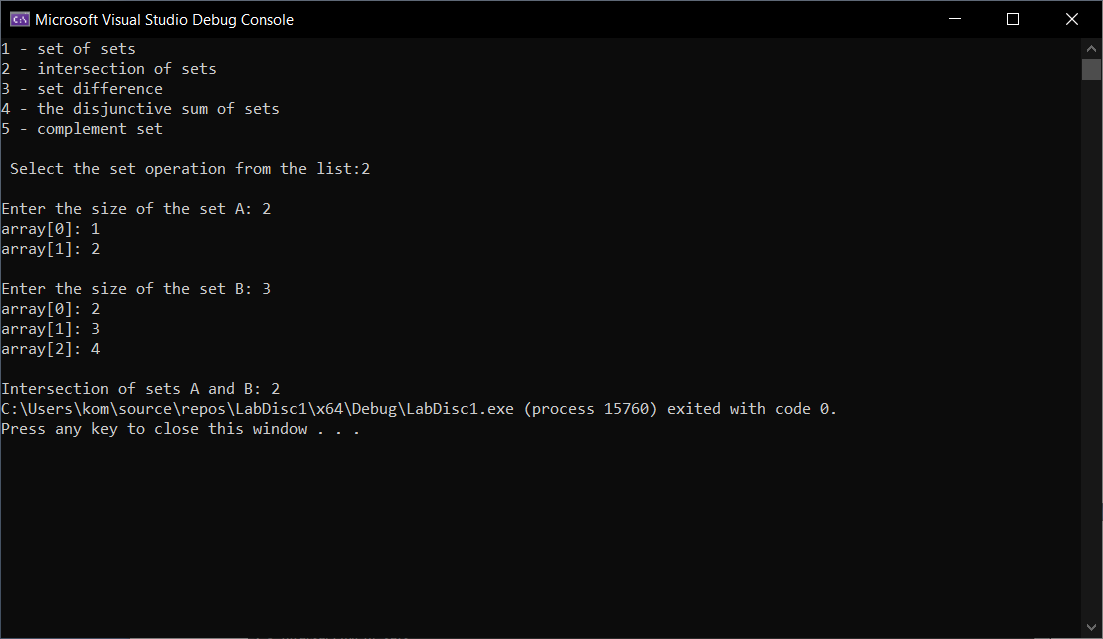
return 0;

}

}

}











**Висновок:** я засвоїв способи задання множин та операції над ними.