МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Лабораторна робота №2

з курсу “Дискретна математика ”

Виконала:  
ст.гр.  КН-110

Ямнюк Аліна

Викладач:

Мельникова Н.І.

Львів – 2018

**Тема:** ”Моделювання основних операцій для числових множин”

**Мета роботи:**

Ознайомитись на практиці із основними поняттями

теорії множин, навчитись будувати діаграми Ейлера-Венна операцій над

множинами, використовувати закони алгебри множин, освоїти принцип

включень-виключень для двох і трьох множин та комп’ютерне подання

множин.

**Теоретичні відомості:**

**2.1. Основні поняття теорії множин. Операції над множинами**

**Множина** – це сукупність об’єктів, які називають елементами.

Кажуть, що множина *А* є **підмножиною** множини *S* (цей факт

позначають *A* *S* , де – знак нестрогого включення), якщо кожен її

елемент автоматично є елементом множини *S*. Досить часто при цьому

кажуть, що множина *А* міститься в множині *S*.

Якщо *A* *S* і *S* *A*, то *A* називають **власною (строгою, істинною)**

**підмножиною** *S* (позначають *A**S* , де – знак строгого включення).

Дві множини *А* та *S* називаються **рівними***,* якщо вони складаються з

однакових елементів. У цьому випадку пишуть *А=S.*

Якщо розглядувані множини є підмножинами деякої множини, то її

називають **універсумом** або **універсальною множиною** і позначають

літерою *U* (зауважимо, що універсальна множина існує не у всіх випадках).

Множини як об’єкти можуть бути елементами інших множин, Множину,

елементами якої є множини, інколи називають **сімейством**.

Множину, елементами якої є всі підмножини множини *А* і тільки вони

(включно з порожньою множиною та самою множиною *А*), називають

**булеаном** або **множиною-степенем** множини *А* і позначають

*P(A).*

**Потужністю** скінченної множини *А* називають число її елементів,

позначають |*А*|.

Множина, яка не має жодного елемента, називається *порожньою* і

позначається ∅.

Вважається, що порожня множина є підмножиною будь-якої

множини, а також *A*⊂*A.*

Множина всіх підмножин множини *A* називається *булеаном* і

позначається *P*(*A*). Потужність скінченної множини дорівнює кількості її

елементів, позначається *A* . Потужність порожньої множини дорівнює 0.

**Варіант № 16**

**Завдання 1:**

1. Для даних скінчених множин A = {1,2,3,4,5,6,7}, B = {5,6,7,8,9,10} , C ={1,3,8,9,10} та універсума U ={1,2,3,4,5,6,7,8,9,10}

знайти множину, яку задано за допомогою операцій: а); б) (A\B)C.

а)

= {5,6,7} ; {0000111000}

= {1,2,3,4,8,9,10} ; {1111000111}

= {4} ; {0001000000}

б) (A\B)C

A\B = {1,2,3,4} ; {1111000000}

(A\B)C = {4,8,9,10} ; {0001000111}

2. На множинах задачі 1 побудувати булеан множини Знайти його потужність.

= {1,2,3,4}

= {4}

= {4,5,6,7,8,9,10}

= {8,9,10}

= 3

= = 8

= { ∅ ,{8},{9},{10},{8,9},{8,10},{9,10},{8,9,10} }

3. Нехай маємо множини: N ‒множина натуральних чисел, Z ‒

множина цілих чисел, Q ‒ множина раціональних чисел, R ‒ множина

дійсних чисел; А, В, С ‒ будь-які множини. Перевірити які твердження є

вірними (в останній задачі у випадку невірного твердження достатньо

навести контрприклад, якщо твердження вірне ‒ навести доведення):

а) {4} ⊂ {2,4,6,8};

б) Z ∩ R = R;

в) NQ ⊂ R ∩ Z;

г) N ∩ Q ⊂ Q \ Z;

д) якщо A ⊂ B C , то A ∩ ⊂ C.

а) {4} ⊂ {2,4,6,8} – твердження вірне, бо множина {4} є власною підмножиною {2,4,6,8};

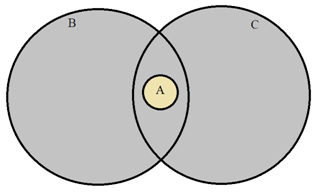
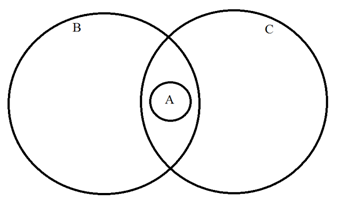
б) Z ∩ R = R – твердження невірне, бо перетин множин Z і R є цілими числами, які є підмножиною дійсних чисел, однак не дорівнюють їм.

в) NQ ⊂ R ∩ Z – твердження невірне, бо об’єднання множин N і Q чисел не є множиною цілих чисел, тобто перетином множин R і Z.

г) N ∩ Q ⊂ Q \ Z – твердження невірне, бо перетин множин N і Q чисел є множиною натуральних чисел, а Q \ Z є множиною раціональних нецілих чисел.

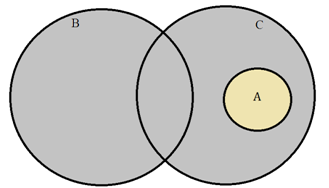
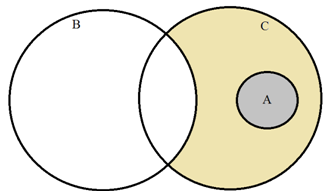
д) якщо A ⊂ B C , то A ∩ ⊂ C. Розглянемо декілька варіантів:

1.

  
 A ⊂ B C A і не перетинаються, тому не може

належати С. Отже, твердження невірне.

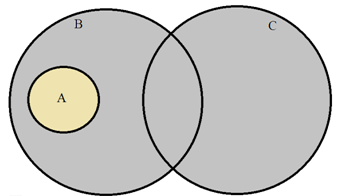
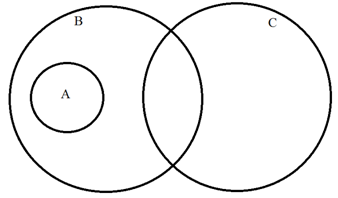
2.

A ⊂ B C A∩ ⊂ С

Отже твердження вірне.

3.



A ⊂ B C A і не перетинаються, тому не може

належати С. Отже, твердження невірне

Якщо за визначенням , отже A ∩ , , тільки за визначенням строгого включення

4. Логічним методом довести тотожність:

Для доведення скористаємось законами алгебри множин.

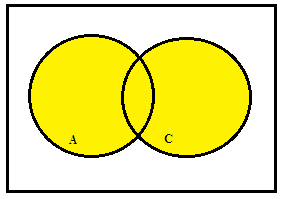
= = = =

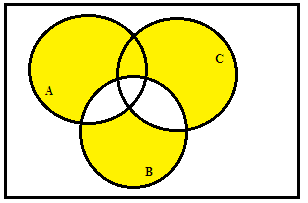
= = = =

Тотожність доведено.

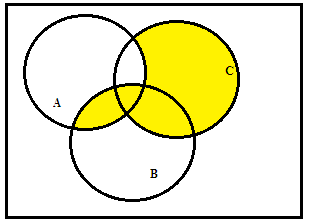
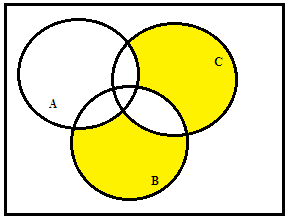
5. Зобразити на діаграмі Ейлера-Венна множину (((АС)В)\А) В

АС: (АС)В:

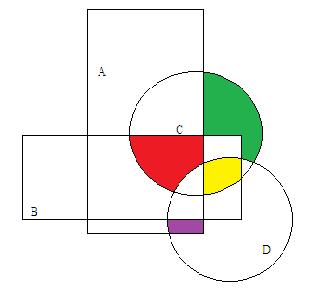




(АС)В)\А: **(((АС)В)\А) В** :



6. Множину зображено на діаграмі. Записати її за допомогою операцій.

 - ((С \ D) \ B) \ A



- ((A ∩ B) ∩ C) \ D



- ((C ∩ D) ∩ B) \ A



 - (A ∩ D) \ B

7. Спростити вигляд множини, яка задана за допомогою операцій, застосовуючи закони алгебри множин (у відповідь множини можуть входити не більше одного разу): (C \ ( A ∩ B )) B.

(C \ ( A ∩ B )) B = ( C ∩ ())B = (C ∩ ()) B = ((C ∩) (C∩)) B = (B (C ∩)) (C ∩) = ((B C) ∩ (B )) (C ∩) =

= ((B C) ∩ U) (C ∩ ) = (B C) (C ∩) = (C (C ∩)) B =

= (C ( ∩ C)) B = C B

8. У групі 15 студентів добре знають математику, 11 – програмування і 8 – і математику, і програмування. Скільки студентів на курсі добре знають лише один предмет?

= 15

= 11

= 8

- ?

= -

= 15+11-8 = 18

= 18-8 = 10

Відповідь: 10 студентів на курсі добре знають лише один предмет.

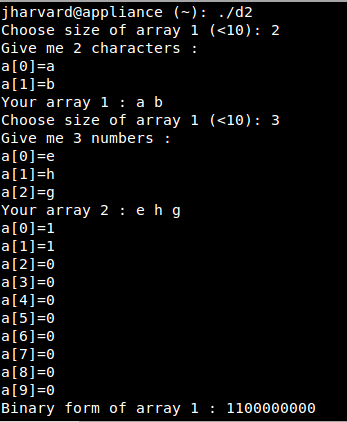
**Завдання 2:**

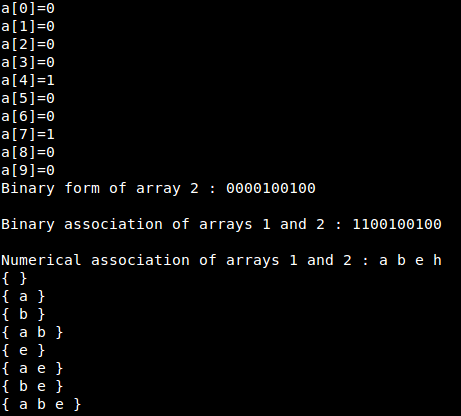
**Програма:**

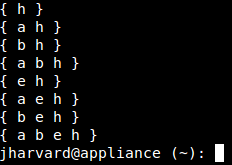
Ввести з клавіатури дві множини символьних даних. Реалізувати операцію об’єднання над цими множинами. Вивести на екран новоутворену множину. Програмно реалізувати побудову булеану цієї множини..

[#include](tg://search_hashtag?hashtag=include) <stdio.h>  
[#include](tg://search_hashtag?hashtag=include) <cs50.h>  
[#include](tg://search_hashtag?hashtag=include) <string.h>  
[#include](tg://search_hashtag?hashtag=include) <math.h>  
  
int main ()  
{  
char U[10]= "abcdefghij"; //універсум  
//задання розміру масиву  
int n;  
do  
{  
printf ("Choose size of array 1 (<10): ");  
n=GetInt();  
if (n<0 || n>10)  
printf ("size must be <10\n");  
}  
while (n<0 || n>10);  
int a[n];  
//введення елементів масиву  
printf ("Give me %d characters :\n",n);  
for(int i=0; i<n; i++)  
{   
do  
{  
printf ("a[%d]=", i);  
a[i]=GetChar();  
if ((a[i]<='`' || a[i]>='k')) printf ("try again\n");  
}  
while (a[i]<='`' || a[i]>='k');  
}  
//вивід масиву на екран  
printf ("Your array 1 : ");  
for( int i=0; i<n; i++)  
{  
printf ("%c ",a[i]);  
}  
//задання розміру масиву   
int m;  
do  
{  
printf ("\nChoose size of array 1 (<10): ");  
m=GetInt();  
if (m<0 || m>10)  
printf ("size must be <10\n");  
}  
while (m<0 || m>10);  
int b[m];  
//введення елементів масиву  
printf ("Give me %d numbers :\n",m);  
for(int i=0; i<m; i++)  
{  
do  
{  
printf ("a[%d]=", i);  
b[i]=GetChar();  
if ((b[i]<='`' || b[i]>='k')) printf ("try again\n");  
}  
while (b[i]<='`' || b[i]>='k');  
}  
//вивід масиву на екран  
printf ("Your array 2 : ");  
for( int i=0; i<m; i++)  
{  
printf ("%c ",b[i]);  
}  
int c[10];  
int k=0;  
//перетворення масиву в бінарну форму  
for (int i=0; i<10; i++)  
{  
if (a[k]==U[i])  
{  
c[i]=1;  
k++;  
}  
else if (a[k]!=U[i])  
{  
c[i]=0;  
}  
}  
//вивід елеметнів у бінарній формі  
for (int i=0; i<10; i++)  
{  
printf ("\na[%d]=%d",i,c[i]);  
}  
printf ("\nBinary form of array 1 : ");  
for (int i=0; i<10; i++)  
{  
printf ("%d" , c[i]);  
}  
printf ("\n");  
  
int d[10];  
int j=0;  
//перетворення масиву в бінарну форму  
for (int i=0; i<10; i++)  
{  
if (b[j]==U[i])  
{  
d[i]=1;  
j++;  
}  
else if (b[j]!=U[i])  
{  
d[i]=0;  
}   
}  
//вивід елеметнів у бінарній формі  
for (int i=0; i<10; i++)  
{  
printf ("\na[%d]=%d",i,d[i]);  
}  
printf ("\nBinary form of array 2 : ");  
for (int i=0; i<10; i++)  
{  
printf ("%d" , d[i]);  
}  
printf ("\n");  
  
int w[10];  
//об'єднання множин у бінарній формі  
for (int i=0; i<10; i++)  
{  
if (c[i]==1 && d[i]==0)  
{  
w[i]=1;  
}  
else if (c[i]==0 && d[i]==1)  
w[i]=1;  
else if (c[i]==1 && d[i]==1)  
w[i]=1;  
else if (c[i]==0 && d[i]==0)  
w[i]=0;  
}  
//вивід на екран бінарного об'єднання множин  
printf ("\nBinary association of arrays 1 and 2 : ");  
for (int i=0; i<10; i++)  
{  
printf ("%d",w[i]);  
}  
printf ("\n");  
  
  
int p=0;  
int s[p];  
//перетворення бінарної форми в нормальну  
for (int i=0; i<10; i++)  
{  
if (w[i]==1)  
{  
s[p]=U[i];  
p++;  
}  
}  
//вивід нормальної форми  
printf ("\nNumerical association of arrays 1 and 2 : ");  
for (int i=0; i<p; i++)  
{  
printf ("%c ",s[i]);  
}  
printf ("\n");  
int l = pow(2, p);  
for (int i = 0; i < l; i++)  
{  
printf ("{ ");  
for (int k = 0; k < p; k++)  
{  
if (i & (1 << k))   
{  
printf ("%c ", s[k]);  
}  
}  
printf("}\n");  
}   
return 0;  
}

**Результат роботи програми:**

****

****

****

**Висновки:**

Я ознайомилася на практиці із основними поняттями

теорії множин, навчилася будувати діаграми Ейлера-Венна для операцій над

множинами, використовувати закони алгебри множин, освоїла принцип

включень-виключень для двох і трьох множин та комп’ютерне подання

множин.