Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Дискретна математика

Лабораторна робота №2

«Бінарні відношення та їх основні властивості, операції над відношеннями»

Виконала:

студентка групи ІО-15

Кушнерик Є.О.

Залікова книжка № 1508

Перевірив Пономаренко А. М.

Київ

2022 р.

**Зміст:**

1. Тема лабораторної роботи.
2. Мета лабораторної роботи.
3. Загальне завдання.
4. Короткі теоретичні відомості, що були використані при написанні лабораторної роботи.
5. Варіант виразу відповідно до індивідуального завдання.
6. Код програми та результати.
7. Аналіз результатів та висновки.

1.**Тема:** «Бінарні відношення та їх основні властивості, операції над відношеннями».

2. **Мета:** вивчити основні властивості бінарних відношень та оволодіти операціями над бінарними відношеннями.

3. **Загальне завдання**

1. Написати в окремому модулі функцію для формування несуперечливих бінарних відношень.

2. Написати в окремому модулі функції виконання логічних операцій над бінарними відношеннями.

3. Пояснити правило формування несуперечливих відношень відповідно до Вашого варіанту.

4. **Короткі теоретичні відомості:**

**Упорядкована пара предметів** – це сукупність, що складається із двох

предметів, розташованих у деякому певному порядку.

**Бінарним** (або двомісним) відношенням R називають підмножину

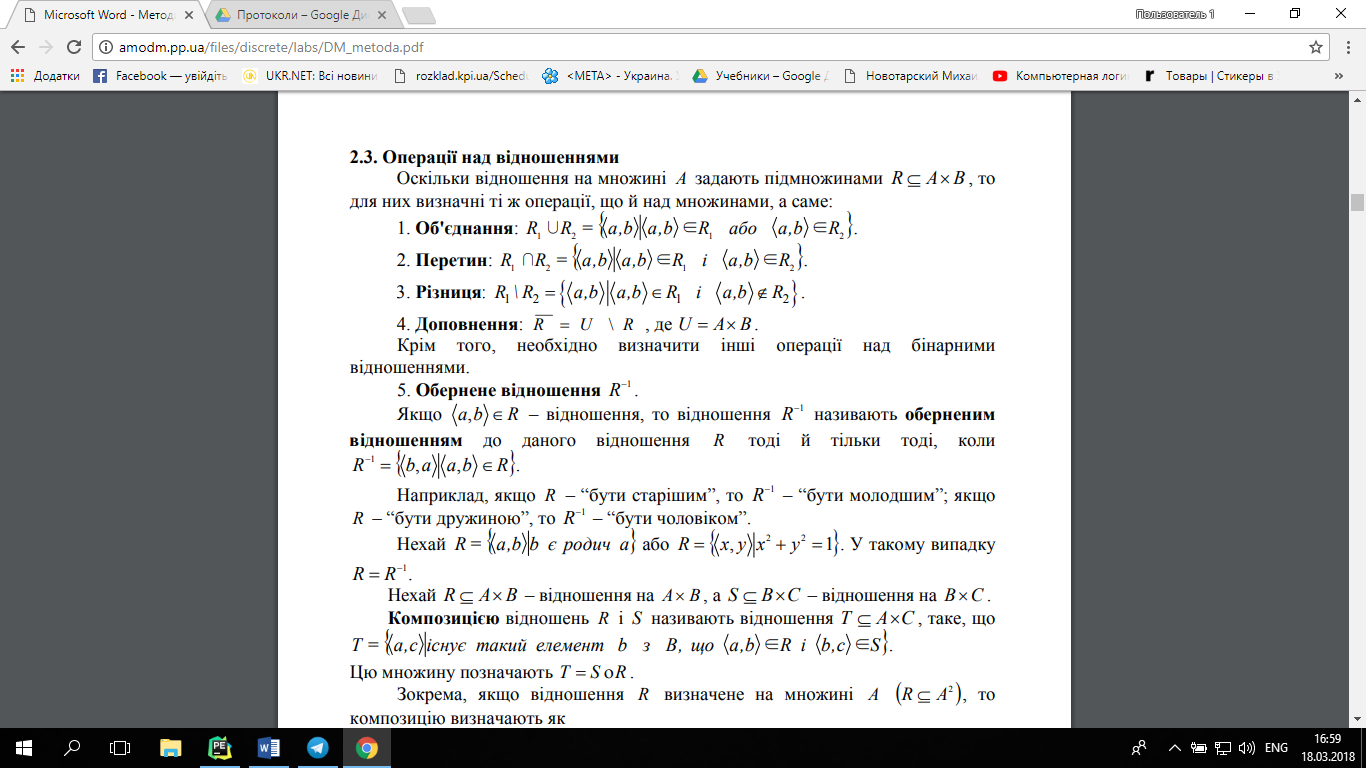
впорядкованих пар, тобто множину, кожен елемент якої є впорядкованою

парою.

**Декартовим добутком** X \*Y множин X і Y є множина {(x, y) | x є X, y єY}.

**Бінарним відношенням** R називають підмножину пар ( x, y) є R прямого добутку X \*Y. Усилу визначення бінарних відношень, як спосіб їх задавання можуть бути використані будь-які способи задавання множин. Відношення, визначені на скінченних множинах, зазвичай задають списком чи матрицею.

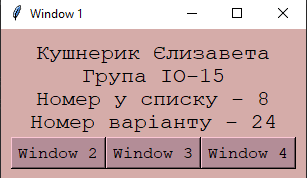
Операції над відношеннями:



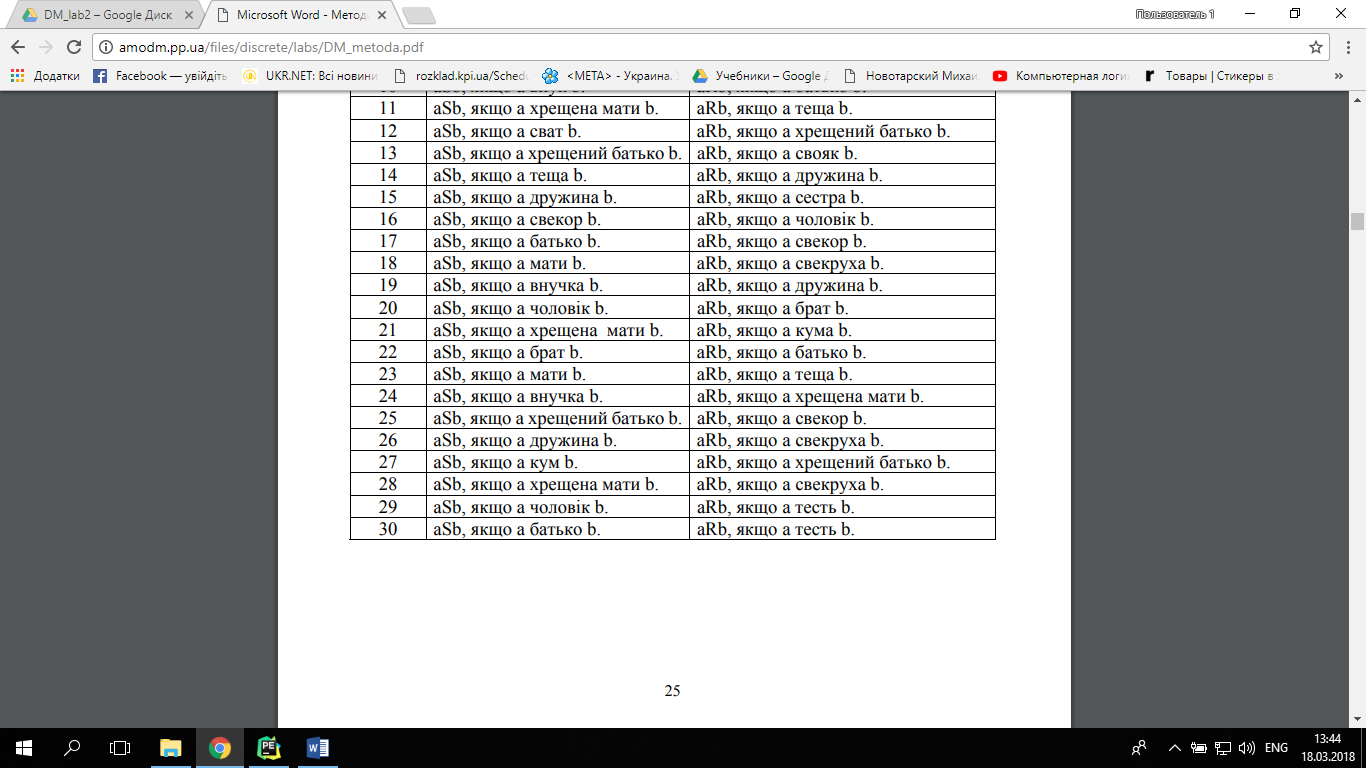
5. **Варіант виразу відповідно до індивідуального завдання:**

Визначення варіанту:

var = (8 + 15 % 60) % 30 + 1

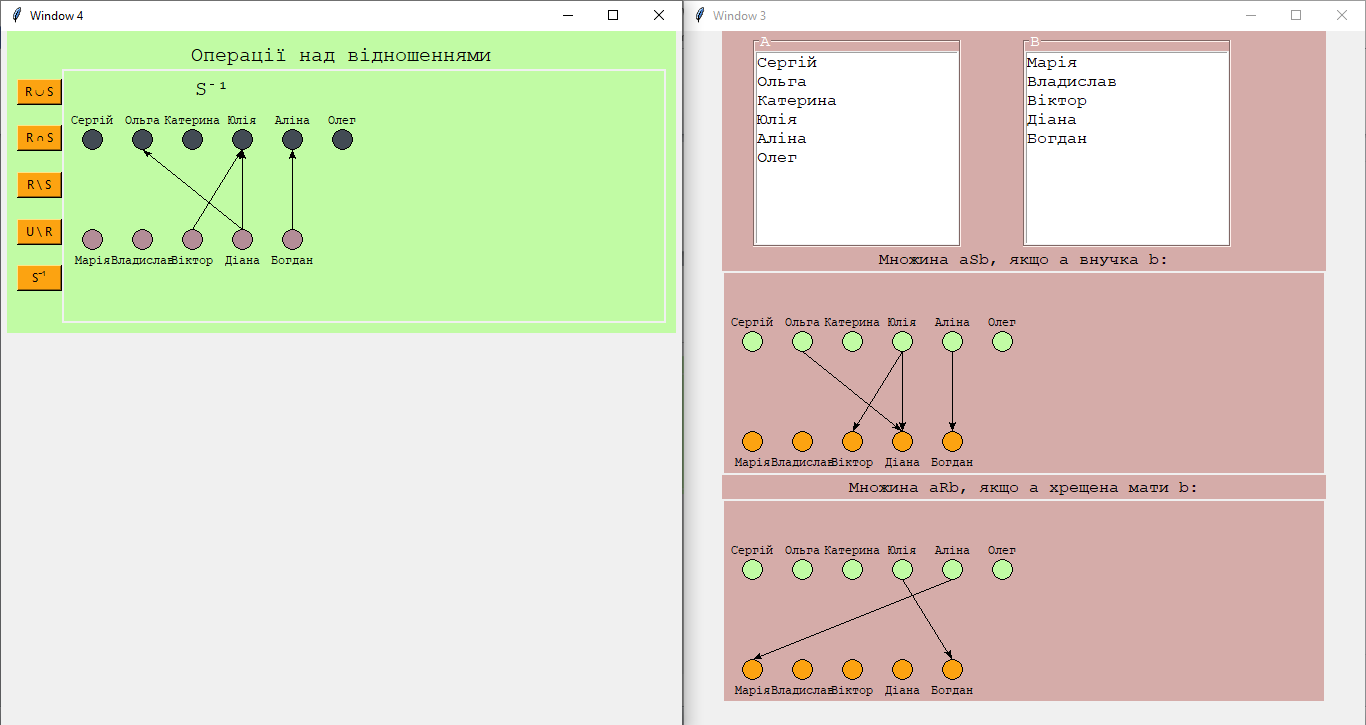
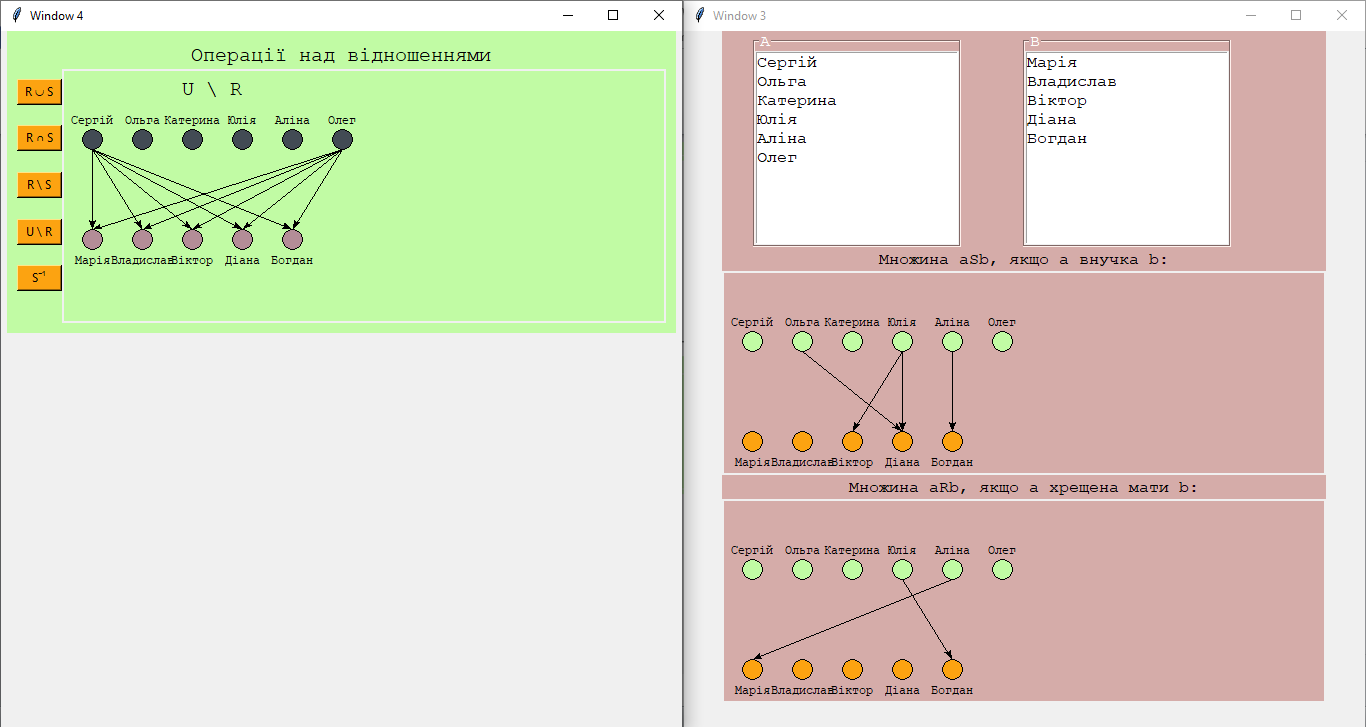
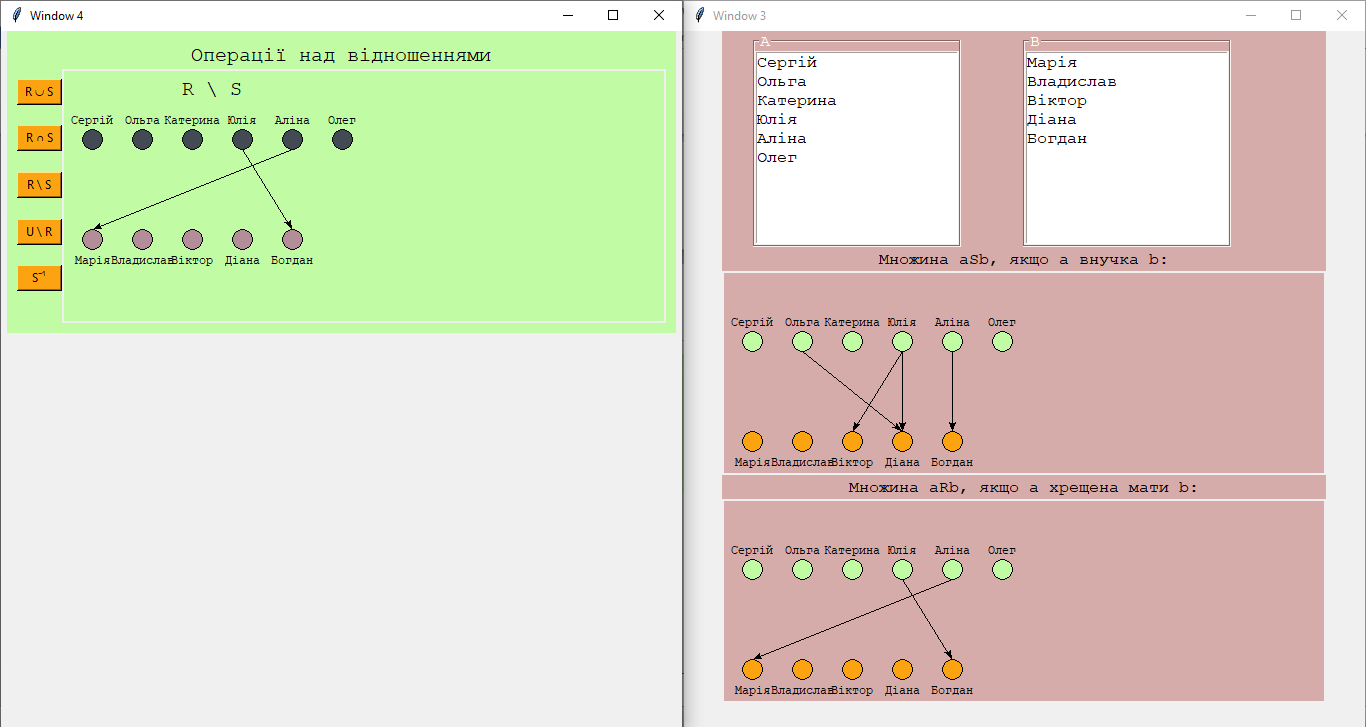
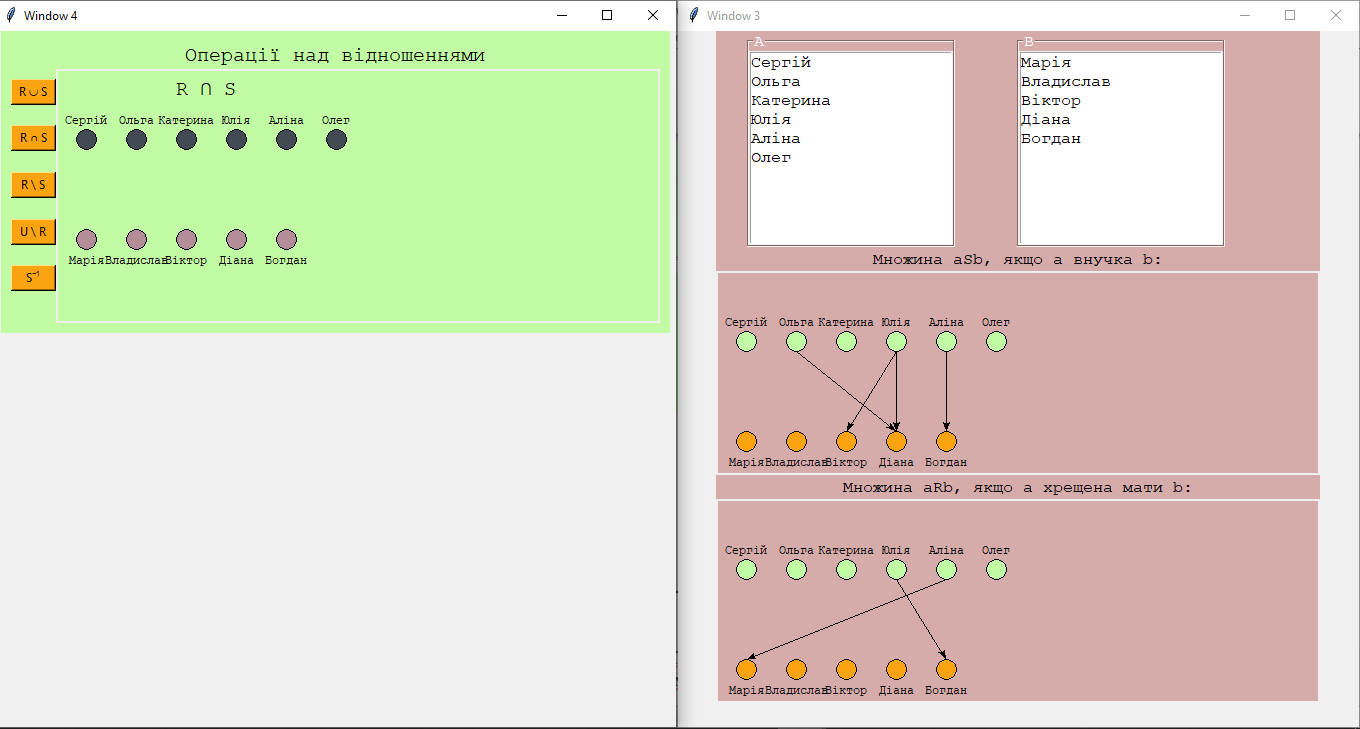
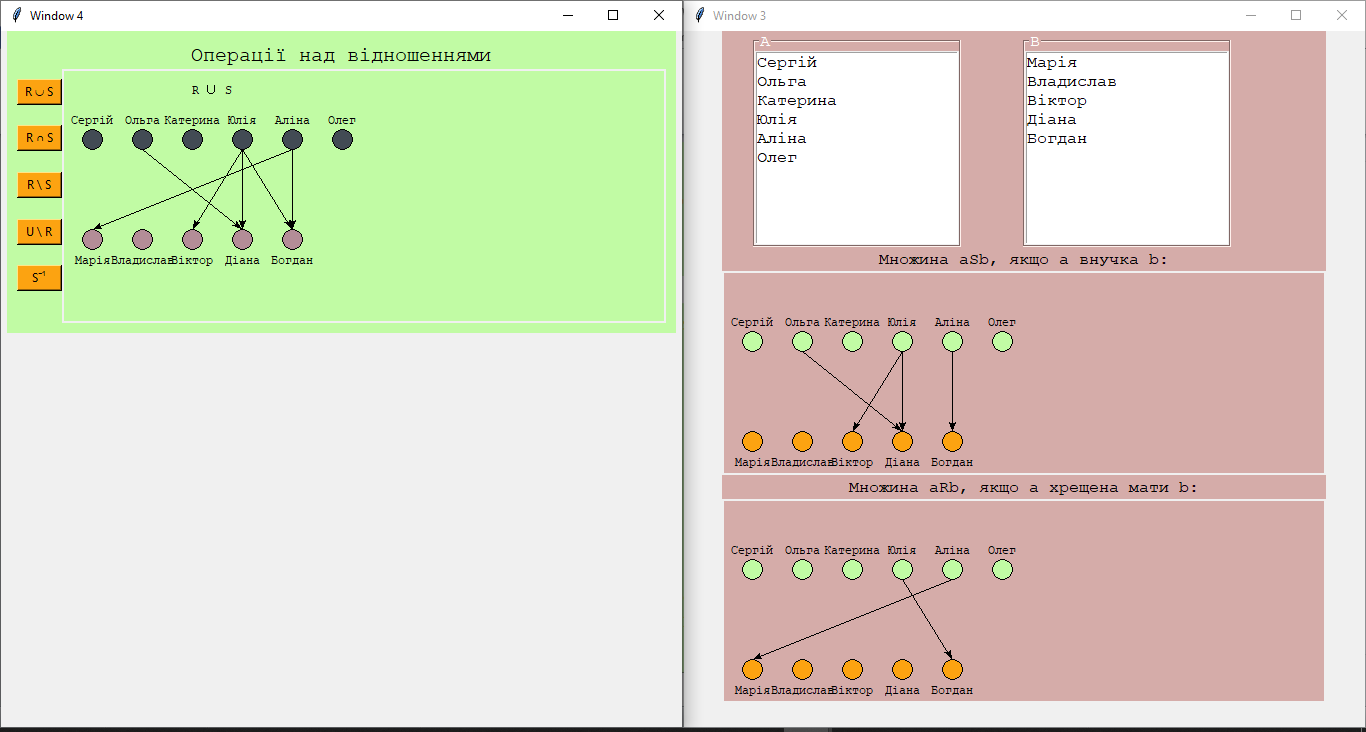
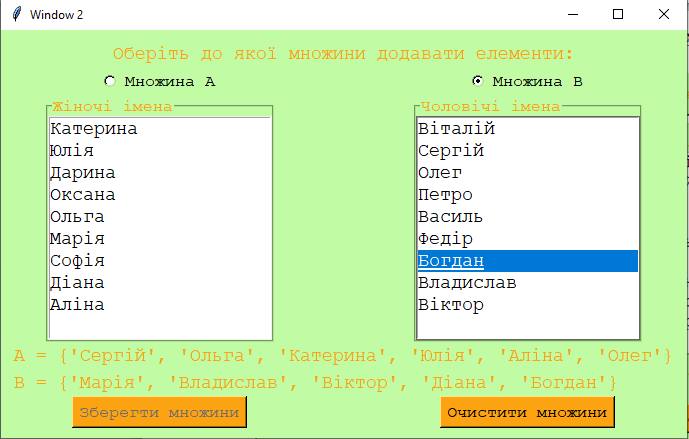
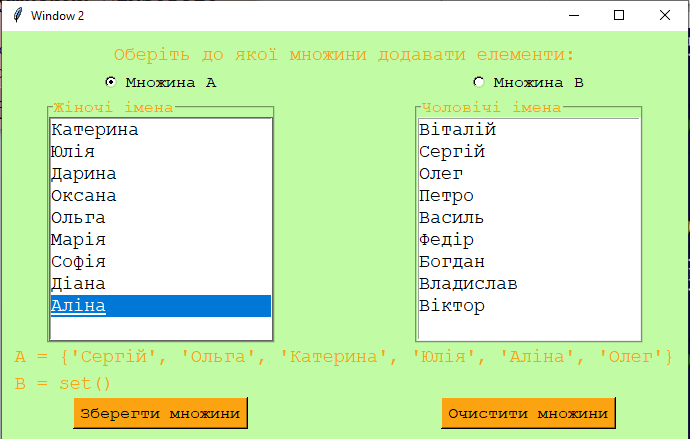


Завдання варіанту:



6. **Код програми та результати**

from tkinter import \*  
import random, copy  
from tkinter import \*  
import random, copy  
  
class Main:  
 def \_\_init\_\_(self, main):  
 self.main = main  
 main.title('Window 1')  
 main.geometry('305x150')  
 var = (8 + 15 % 60) % 30 + 1  
 self.frame\_main = Frame(main, bg='#D5ACA9', bd=10)  
 self.frame\_main.pack()  
 self.name = Label(self.frame\_main, text="Кушнерик Єлизавета\nГрупа ІО-15\nНомер у списку - 8\nНомер варіанту - {}".format(var),  
 bg='#D5ACA9', font=('Courier New', 16), fg='black')  
 self.name.grid(row=0, columnspan=3)  
  
 self.btn\_win2 = Button(self.frame\_main, text='Window 2', bg='#B38D97', font=('Courier New', 12), fg='black',  
 command=self.window2)  
 self.btn\_win2.grid(row=1, column=0)  
 self.btn\_win3 = Button(self.frame\_main, text='Window 3', bg='#B38D97', font=('Courier New', 12), fg='black',  
 command=self.window3)  
 self.btn\_win3.grid(row=1, column=1)  
 self.btn\_win4 = Button(self.frame\_main, text='Window 4', bg='#B38D97', font=('Courier New', 12), fg='black',  
 command=self.window4)  
 self.btn\_win4.grid(row=1, column=2)  
  
 self.set\_A = ['Катерина', 'Юлія', 'Дарина', 'Оксана', 'Ольга', 'Марія', 'Софія', 'Діана', 'Аліна']  
 self.set\_B = ['Віталій', 'Сергій', 'Олег', 'Петро', 'Василь', 'Федір', 'Богдан', 'Владислав', 'Віктор']  
  
 def window2(self):  
 self.win2 = Toplevel(self.main)  
 self.win2.title('Window 2')  
 self.var = IntVar()  
 self.var.set(0)  
 self.a = set()  
 self.b = set()  
  
 def add\_women(event):  
 if self.var.get() == 0:  
 self.a.add(self.set\_A[event.widget.curselection()[0]])  
 if self.var.get() == 1:  
 self.b.add(self.set\_A[event.widget.curselection()[0]])  
 self.lbl\_a['text'] = 'A = {}'.format(self.a)  
 self.lbl\_b['text'] = 'B = {}'.format(self.b)  
  
 def add\_men(event):  
 if self.var.get() == 0:  
 self.a.add(self.set\_B[event.widget.curselection()[0]])  
 if self.var.get() == 1:  
 self.b.add(self.set\_B[event.widget.curselection()[0]])  
 self.lbl\_b['text'] = 'B = {}'.format(self.b)  
 self.lbl\_a['text'] = 'A = {}'.format(self.a)  
  
 def save():  
 with open('result.txt', 'a') as f:  
 f.write(str(self.a))  
 f.write('\n')  
 f.write(str(self.b))  
 f.write('\n')  
 self.save\_btn.config(state=DISABLED)  
  
 def del\_set():  
 self.a = set()  
 self.b = set()  
 self.lbl\_a['text'] = 'A = {}'.format(self.a)  
 self.lbl\_b['text'] = 'B = {}'.format(self.b)  
 self.save\_btn.config(state=NORMAL)  
 with open('result.txt', 'w') as f:  
 f.write('')  
  
 self.frm\_win2 = Frame(self.win2, bg='#C1FBA4', bd=10)  
 self.frm\_win2.pack()  
 self.choose\_set = Label(self.frm\_win2, text='Оберіть до якої множини додавати елементи:', bg='#C1FBA4',  
 font=('Courier New', 14), fg='#FCA311')  
 self.choose\_set.grid(row=0, columnspan=3)  
 self.radiobtn\_A = Radiobutton(self.frm\_win2, text='Множина А', font=('Courier New', 12), bg='#C1FBA4',  
 activebackground='#C1FBA4', variable=self.var, value=0)  
 self.radiobtn\_A.grid(row=1, column=0)  
 self.radiobtn\_B = Radiobutton(self.frm\_win2, text='Множина B', font=('Courier New', 12), bg='#C1FBA4',  
 activebackground='#C1FBA4', variable=self.var, value=1)  
 self.radiobtn\_B.grid(row=1, column=2)  
 self.lbl\_fr1 = LabelFrame(self.frm\_win2, text='Жіночі імена', font=('Courier New', 12), bg='#C1FBA4', fg='#FCA311')  
 self.lbl\_fr1.grid(row=2, column=0)  
 self.lst1 = Listbox(self.lbl\_fr1, font=('Courier New', 14), selectmode=EXTENDED)  
 self.lst1.bind("<<ListboxSelect>>", add\_women)  
 self.lst1.grid(row=2, column=0)  
 for i in self.set\_A:  
 self.lst1.insert(END, i)  
  
 self.lbl\_fr2 = LabelFrame(self.frm\_win2, text='Чоловічі імена', font=('Courier New', 12), bg='#C1FBA4', fg='#FCA311')  
 self.lbl\_fr2.grid(row=2, column=2)  
 self.lst2 = Listbox(self.lbl\_fr2, font=('Courier New', 14), selectmode=EXTENDED)  
 self.lst2.bind("<<ListboxSelect>>", add\_men)  
 self.lst2.grid(row=2, column=2)  
 for i in self.set\_B:  
 self.lst2.insert(END, i)  
  
 self.lbl\_a = Label(self.frm\_win2, text='A =', bg='#C1FBA4', font=('Courier New', 14), fg='#FCA311')  
 self.lbl\_a.grid(row=3, columnspan=3, sticky='w')  
 self.lbl\_b = Label(self.frm\_win2, text='B =', bg='#C1FBA4', font=('Courier New', 14), fg='#FCA311')  
 self.lbl\_b.grid(row=4, columnspan=3, sticky='w')  
  
 self.save\_btn = Button(self.frm\_win2, text='Зберегти множини', font=('Courier New', 12), bg= '#FCA311', command=save)  
 self.save\_btn.grid(row=5, column=0)  
 self.del\_btn = Button(self.frm\_win2, text='Очистити множини', font=('Courier New', 12), bg = '#FCA311', command=del\_set)  
 self.del\_btn.grid(row=5, column=2)  
  
 def window3(self):  
 self.win3 = Toplevel(self.main)  
 self.win3.title('Window 3')  
  
 self.frm\_win3 = Frame(self.win3, bg='#D5ACA9')  
 self.frm\_win3.pack()  
 self.lbl\_fr3 = LabelFrame(self.frm\_win3, bg='#D5ACA9', text='A', font=('Courier New', 12), fg='white')  
 self.lbl\_fr3.grid(row=0, column=0)  
 self.lbl\_fr4 = LabelFrame(self.frm\_win3, bg='#D5ACA9', text='B', font=('Courier New', 12), fg='white')  
 self.lbl\_fr4.grid(row=0, column=1)  
 self.list\_a = Listbox(self.lbl\_fr3, font=('Courier New', 12))  
 self.list\_a.grid()  
 for i in self.a:  
 self.list\_a.insert(END, i)  
 self.list\_b = Listbox(self.lbl\_fr4, font=('Courier New', 12))  
 self.list\_b.grid()  
 for i in self.b:  
 self.list\_b.insert(END, i)  
 self.lbl\_aSb = Label(self.frm\_win3, text='Множина aSb, якщо a внучка b:', font=('Courier New', 12), bg='#D5ACA9',  
 fg='black')  
 self.lbl\_aSb.grid(row=1, columnspan=3)  
 self.lbl\_aRb = Label(self.frm\_win3, text='Множина aRb, якщо a хрещена мати b:', font=('Courier New', 12),  
 bg='#D5ACA9', fg='black')  
 self.lbl\_aRb.grid(row=4, columnspan=3)  
  
 def a\_onychka\_b():  
 A = set()  
 for i in self.a:  
 if i in self.set\_A:  
 A.add(i)  
 B = self.b  
 S = []  
 for i in range(min(len(A), len(B))):  
 p = random.choice(list(A))  
 q = random.choice(list(B))  
 if p != q:  
 S.append([p, q])  
 return S  
  
 def a\_khrechshena\_b():  
 A = set()  
 for i in self.a:  
 if i in self.set\_A:  
 A.add(i)  
 B = self.b  
 R = []  
 for i in range(min(len(A), len(B))):  
 p = random.choice(list(A))  
 q = random.choice(list(B))  
 if p != q:  
 if [p, q] not in self.S:  
 R.append([p, q])  
 return R  
  
 self.S = a\_onychka\_b()  
 self.R = a\_khrechshena\_b()  
  
 aSb = Canvas(self.frm\_win3, width=600, height=200, bg='#D5ACA9')  
 dict\_SA = {}  
 dict\_SB = {}  
 for i in range(len(self.a)):  
 aSb.create\_text(30 + i \* 50, 50, text=list(self.a)[i], font=('Courier New', 9))  
 aSb.create\_oval([20 + i \* 50, 60], [40 + i \* 50, 80], fill="#C1FBA4")  
 dict\_SA.update({list(self.a)[i]: [30 + i \* 50, 80]})  
 for j in range(len(self.b)):  
 aSb.create\_text(30 + j \* 50, 190, text=list(self.b)[j], font=('Courier New', 9))  
 aSb.create\_oval([20 + j \* 50, 160], [40 + j \* 50, 180], fill="#FCA311")  
 dict\_SB.update({list(self.b)[j]: [30 + j \* 50, 160]})  
 for k in self.S:  
 aSb.create\_line(dict\_SA[k[0]], dict\_SB[k[1]], arrow=LAST)  
 aSb.grid(row=2, column=0, columnspan=3, rowspan=2)  
  
 aRb = Canvas(self.frm\_win3, width=600, height=200, bg='#D5ACA9')  
 dict\_RA = {}  
 dict\_RB = {}  
 for i in range(len(self.a)):  
 aRb.create\_text(30 + i \* 50, 50, text=list(self.a)[i], font=('Courier New', 9))  
 aRb.create\_oval([20 + i \* 50, 60], [40 + i \* 50, 80], fill="#C1FBA4")  
 dict\_RA.update({list(self.a)[i]: [30 + i \* 50, 80]})  
 for j in range(len(self.b)):  
 aRb.create\_text(30 + j \* 50, 190, text=list(self.b)[j], font=('Courier New', 9))  
 aRb.create\_oval([20 + j \* 50, 160], [40 + j \* 50, 180], fill="#FCA311")  
 dict\_RB.update({list(self.b)[j]: [30 + j \* 50, 160]})  
 for k in self.R:  
 aRb.create\_line(dict\_RA[k[0]], dict\_RB[k[1]], arrow=LAST)  
 aRb.grid(row=5, column=0, columnspan=3, rowspan=2)  
  
 def window4(self):  
 self.win4 = Toplevel(self.main)  
 self.win4.title('Window 4')  
  
 self.frm\_win4 = Frame(self.win4, bg='#C1FBA4', bd=10)  
 self.frm\_win4.pack()  
 self.lbl\_oper = Label(self.frm\_win4, text='Операції над відношеннями', font=('Courier New', 15), bg='#C1FBA4')  
 self.lbl\_oper.grid(row=0, columnspan=4)  
  
 def btn1():  
 self.canv.delete("all")  
 self.canv.create\_text(150, 20, text='R \u222A S', font=('Courier New', 9))  
 dict\_b1 = {}  
 dict\_b2 = {}  
 V = self.R+self.S  
 for i in range(len(self.a)):  
 self.canv.create\_text(30+i\*50, 50, text=list(self.a)[i], font=('Courier New', 9))  
 self.canv.create\_oval([20+i\*50, 60], [40+i\*50, 80], fill="#424B54")  
 dict\_b1.update({list(self.a)[i]: [30+i\*50, 80]})  
 for j in range(len(self.b)):  
 self.canv.create\_text(30+j\*50, 190, text=list(self.b)[j], font=('Courier New', 9))  
 self.canv.create\_oval([20+j\*50, 160], [40+j\*50, 180], fill="#B38D97")  
 dict\_b2.update({list(self.b)[j]: [30+j\*50, 160]})  
 for k in V:  
 self.canv.create\_line(dict\_b1[k[0]], dict\_b2[k[1]], arrow=LAST)  
  
 def btn2():  
 self.canv.delete("all")  
 self.canv.create\_text(150, 20, text='R \u2229 S', font=('Courier New', 15))  
 dict\_b1 = {}  
 dict\_b2 = {}  
 V = []  
 for i in self.R:  
 if i in self.S:  
 V.append(i)  
  
 for i in range(len(self.a)):  
 self.canv.create\_text(30+i\*50, 50, text=list(self.a)[i], font=('Courier New', 9))  
 self.canv.create\_oval([20+i\*50, 60], [40+i\*50, 80], fill="#424B54")  
 dict\_b1.update({list(self.a)[i]: [30+i\*50, 80]})  
 for j in range(len(self.b)):  
 self.canv.create\_text(30+j\*50, 190, text=list(self.b)[j], font=('Courier New', 9))  
 self.canv.create\_oval([20+j\*50, 160], [40+j\*50, 180], fill="#B38D97")  
 dict\_b2.update({list(self.b)[j]: [30+j\*50, 160]})  
  
 for k in V:  
 if len(V) != 0:  
 self.canv.create\_line(dict\_b1[k[0]], dict\_b2[k[1]], arrow=LAST)  
  
 def btn3():  
 self.canv.delete("all")  
 self.canv.create\_text(150, 20, text='R \ S', font=('Courier New', 15))  
 dict\_b1 = {}  
 dict\_b2 = {}  
 V = copy.deepcopy(self.R)  
 for i in V:  
 if i in self.S:  
 V.remove(i)  
  
 for i in range(len(self.a)):  
 self.canv.create\_text(30+i\*50, 50, text=list(self.a)[i], font=('Courier New', 9))  
 self.canv.create\_oval([20+i\*50, 60], [40+i\*50, 80], fill="#424B54")  
 dict\_b1.update({list(self.a)[i]: [30+i\*50, 80]})  
 for j in range(len(self.b)):  
 self.canv.create\_text(30+j\*50, 190, text=list(self.b)[j], font=('Courier New', 9))  
 self.canv.create\_oval([20+j\*50, 160], [40+j\*50, 180], fill="#B38D97")  
 dict\_b2.update({list(self.b)[j]: [30+j\*50, 160]})  
 for k in V:  
 self.canv.create\_line(dict\_b1[k[0]], dict\_b2[k[1]], arrow=LAST)  
  
 def btn4():  
 self.canv.delete("all")  
 self.canv.create\_text(150, 20, text='U \ R', font=('Courier New', 15))  
 dict\_b1 = {}  
 dict\_b2 = {}  
  
 V = []  
 a = set()  
 for i in self.a:  
 if i in self.set\_B:  
 a.add(i)  
 b = copy.deepcopy(self.b)  
 for i in a:  
 for j in b:  
 V.append([i,j])  
  
 for i in V:  
 if i in self.R:  
 V.remove(i)  
  
 for i in range(len(self.a)):  
 self.canv.create\_text(30+i\*50, 50, text=list(self.a)[i], font=('Courier New', 9))  
 self.canv.create\_oval([20+i\*50, 60], [40+i\*50, 80], fill="#424B54")  
 dict\_b1.update({list(self.a)[i]: [30+i\*50, 80]})  
 for j in range(len(self.b)):  
 self.canv.create\_text(30+j\*50, 190, text=list(self.b)[j], font=('Courier New', 9))  
 self.canv.create\_oval([20+j\*50, 160], [40+j\*50, 180], fill="#B38D97")  
 dict\_b2.update({list(self.b)[j]: [30+j\*50, 160]})  
 for k in V:  
 if len(V) != 0:  
 self.canv.create\_line(dict\_b1[k[0]], dict\_b2[k[1]], arrow=LAST)  
  
 def btn5():  
 self.canv.delete("all")  
 self.canv.create\_text(150, 20, text='S⁻¹', font=('Courier New', 15))  
 dict\_b1 = {}  
 dict\_b2 = {}  
 V = copy.deepcopy(self.S)  
 for i in V:  
 i[0], i[1] = i[1], i[0]  
 for i in range(len(self.a)):  
 self.canv.create\_text(30+i\*50, 50, text=list(self.a)[i], font=('Courier New', 9))  
 self.canv.create\_oval([20+i\*50, 60], [40+i\*50, 80], fill="#424B54")  
 dict\_b1.update({list(self.a)[i]: [30+i\*50, 80]})  
 for j in range(len(self.b)):  
 self.canv.create\_text(30+j\*50, 190, text=list(self.b)[j], font=('Courier New', 9))  
 self.canv.create\_oval([20+j\*50, 160], [40+j\*50, 180], fill="#B38D97")  
 dict\_b2.update({list(self.b)[j]: [30+j\*50, 160]})  
 for k in V:  
 self.canv.create\_line(dict\_b2[k[0]], dict\_b1[k[1]], arrow=LAST)  
  
 self.union\_btn = Button(self.frm\_win4, text='R ∪ S', bg='#FCA311', width=5, command=btn1)  
 self.union\_btn.grid(row=1, column=0, sticky='w')  
 self.intersection\_btn = Button(self.frm\_win4, text='R ∩ S', bg='#FCA311', width=5, command=btn2)  
 self.intersection\_btn.grid(row=2, column=0, sticky='w')  
 self.difference\_btn = Button(self.frm\_win4, text='R \ S', bg='#FCA311', width=5, command=btn3)  
 self.difference\_btn.grid(row=3, column=0, sticky='w')  
 self.not\_btn = Button(self.frm\_win4, text='U \ R', bg='#FCA311', width=5, command=btn4)  
 self.not\_btn.grid(row=4, column=0, sticky='w')  
 self.reverse\_btn = Button(self.frm\_win4, text='S⁻¹', bg='#FCA311', width=5, command=btn5)  
 self.reverse\_btn.grid(row=5, column=0, sticky='w')  
 self.canv = Canvas(self.frm\_win4, width=600, height=250, bg='#C1FBA4')  
 self.canv.grid(row=1, rowspan=6, column=3)  
  
  
root = Tk()  
window = Main(root)  
root.mainloop()



7.**Висновок**

У ході виконання лабораторної роботи я засвоїла матеріал про операції над бінарними відношеннями та спробувала реалізувати їх в програмному коді.