**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Дискретна математика**

**Лабораторна робота №1**

**«Множини: основні властивості та операції над ними, діаграми Венна».**

Виконав:

студент групи IО-63

Братун Андрій

Залікова книжка № 6305

Перевірив Новотарський М. А.

Київ 2016 р.

**Мета:** вивчити основні аксіоми, закони і теореми теорії множин, навчитися застосовувати їх на практиці. Обчислити логічний вираз шляхом послідовного застосування операцій над множинами.

**Загальне завдання:**

1. Повторити матеріал: «Бібліотека tkinter (віджети)» та виконати лабораторну роботу з застосуванням графічного інтерфейсу.

2. Спростити логічний вираз з застосуванням тотожностей алгебри множин. 3.

В окремому модулі написати функцію обчислення початкового логічного виразу (1), вибраного відповідно до індивідуального варіанта. 4.

В окремому модулі написати функцію обчислення спрощеного логічного виразу. 5.

В окремому модулі написати функцію виконання логічної операції (2), вибраної відповідно до індивідуального варіанта. 6.

В окремому модулі виконати порівняння результатів:

А) обчислення початкового та спрощеного виразу

Б) виконання логічної операції Вашою функцією та відповідною стандартною логічною операцією або функцією Python.

Варіант виразу відповідно до індивідуального завдання – 9

**Короткі теоретичні відомості:**

**Множина** – є сукупність визначених об'єктів, різних між собою, об’єднаних за певною ознакою чи властивістю.

**Об’єднання**. Об'єднанням множин A і B називають множину, що складається із всіх тих елементів, які належать хоча б одній з множин A або B.

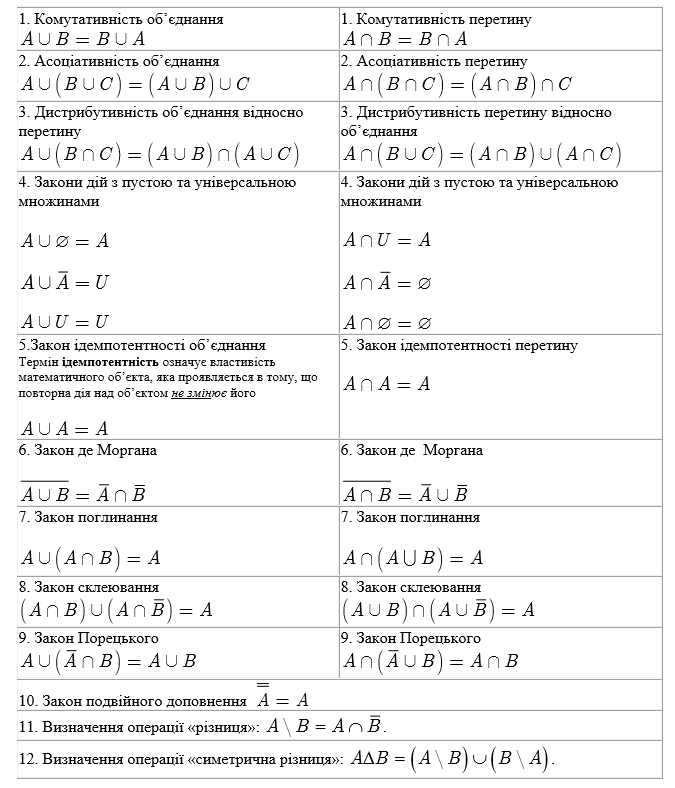
**Перетин**. Перетином множин A і B називають множину, що складається із всіх тих елементів, які належать як множині A, так і множині B.

**Доповнення**. Доповненням (або абсолютним доповненням) множини A називають множину, що складається із всіх елементів універсальної множини, які не належать A.

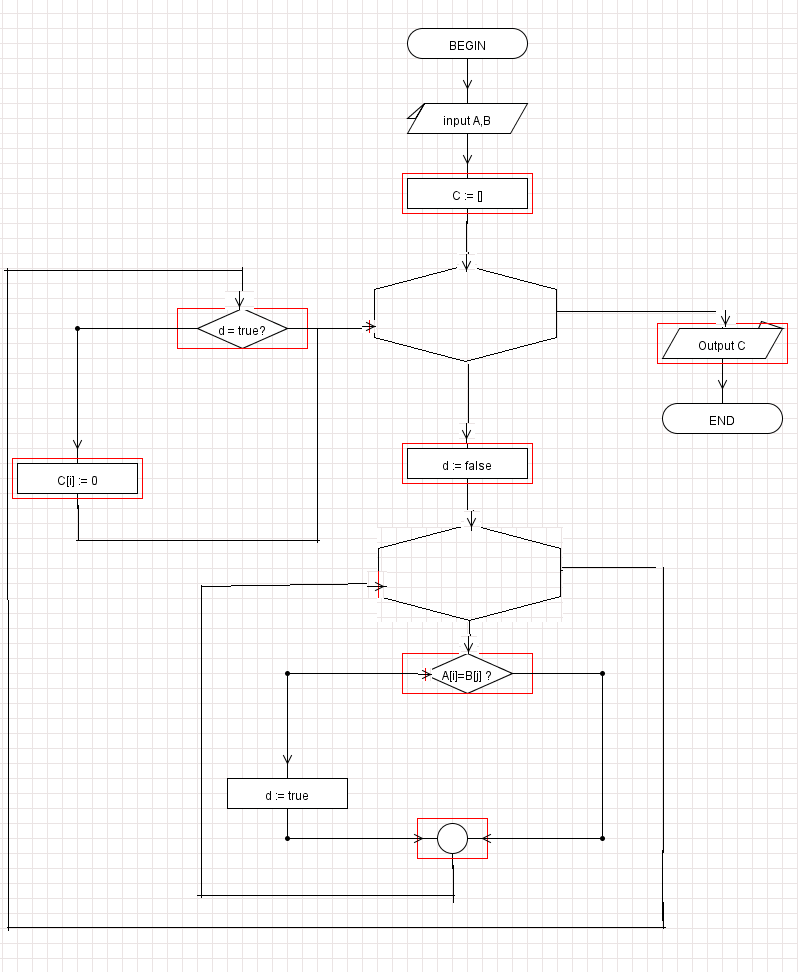
**Різниця**. Різницею множин A й B (або відносним доповненням) називають множину, що складається із всіх елементів множини A, які не належать B.

**Симетрична різниця**. Симетричною різницею множин A і B називають множину, що складається з об'єднання всіх елементів, що належать множині A і не містяться в B, і елементів, що належать множині B і не містяться в A.

**Тотожності алгебри множин**

****

**Блок-схеми, які відповідають алгоритмам, що використані в лабораторній роботі.**

****

No

No

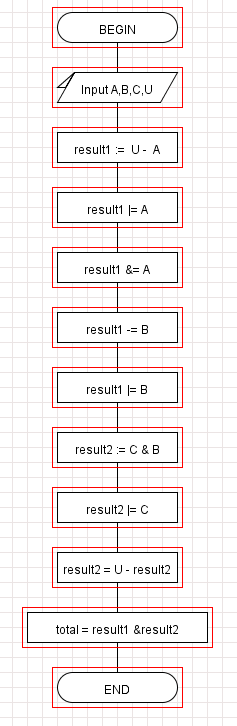
Yes

Yes

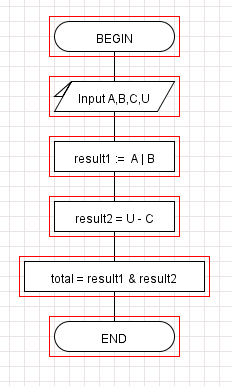
j := 1…|B|

i := 1…|A|

**Блок-схема послідовності обчислення початкового логічного виразу**

****

**Блок-схема послідовності обчислення спрощеного логічного виразу**

****

**Докладний опис спрощення логічного виразу з посиланням на тотожності алгебри множин, що застосовувалися при спрощенні.**

Умова



Спрощення

 Закон дії з пустою та універсальною множиною

Закон поглинання

Отримали



Спрощуємо далі

 Закон дії з пустою та універсальною множиною

Отримали



Спрощуємо далі

 За визначенням операції «різниця»

 Закон Порецького

Результат

.

Роздруківка того фрагменту тексту програми, який написаний індивідуально

Модуль main

**from** tkinter **import** \*  
**import** func,window2, window3, window4, window5  
**from** tkinter **import** messagebox  
  
  
root = Tk()  
root.title("Main Window")  
root.maxsize(width=895,height=615)  
root.minsize(width=895,height=615)  
  
  
#My uniq task  
Num = 5  
G = 63  
number = (Num+G%60)%30+1  
task = "Personal task number - " + str(number)  
  
  
######################MAINLOOOOOOOP################  
  
  
#Info about student  
lab1 = Label(root, text='Create by Andrew Bratun', font='arial 20')  
lab2 = Label(root, text='Group number - IO-63', font='arial 20')  
lab3 = Label(root, text=task , font='arial 20')  
  
#Info about doings1  
lab4 = Label(root, text='Give a power to set:', font='arial 20')  
lab5 = Label(root, text='Create random set\n(elements from 0 to 255):', font='arial 20')  
lab6 = Label(root, text='Create a set themselves:', font='arial 20')  
lab7 = Label(root, text='Create a universal set:', font='arial 20')  
  
#Give a power  
but1 = Button(root, text='Give a power to A', font='arial 20')  
but1.bind("<Button-1>", func.give\_power\_A)  
but2 = Button(root, text='Give a power to B', font='arial 20')  
but2.bind("<Button-1>", func.give\_power\_B)  
but3 = Button(root, text='Give a power to C', font='arial 20')  
but3.bind("<Button-1>", func.give\_power\_C)  
  
#Create random set  
but4 = Button(root, text='Create A\nBase power = 5', font='arial 20')  
but4.bind("<ButtonRelease-1>", func.create\_A)  
but5 = Button(root, text='Create B\nBase power = 6', font='arial 20')  
but5.bind("<ButtonRelease-1>", func.create\_B)  
but6 = Button(root, text='Create C\nBase power = 7', font='arial 20')  
but6.bind("<ButtonRelease-1>", func.create\_C)  
  
#Create a set themselves:  
but7 = Button(root, text='Write themselves A', font='arial 20')  
but7.bind("<Button-1>",func.create\_themselves\_A)  
but8 = Button(root, text='Write themselves B', font='arial 20')  
but8.bind("<Button-1>",func.create\_themselves\_B)  
but9 = Button(root, text='Write themselves C', font='arial 20')  
but9.bind("<Button-1>",func.create\_themselves\_C)  
  
#Universal set  
but10 = Button(root, text='Universal set', font='arial 20')  
but10.bind("<Button-1>",func.create\_universal\_set)  
  
#Info about student  
lab1.grid(row=1,column=1, sticky=W+E+N+S, pady=5, padx=5)  
lab2.grid(row=2,column=1, sticky=W+E+N+S, pady=5, padx=5)  
lab3.grid(row=3,column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
#Info about doings1  
lab4.grid(row=4,column=0, sticky=W+E+N+S, pady=5, padx=5)  
lab5.grid(row=4,column=1, sticky=W+E+N+S, pady=5, padx=5)  
lab6.grid(row=4,column=2, sticky=W+E+N+S, pady=5, padx=5)  
  
#Give a power  
but1.grid(row=5,column=0, sticky=W+E+N+S, pady=5, padx=5)  
but2.grid(row=6,column=0, sticky=W+E+N+S, pady=5, padx=5)  
but3.grid(row=7,column=0, sticky=W+E+N+S, pady=5, padx=5)  
  
#Create random set  
but4.grid(row=5,column=1, sticky=W+E+N+S, pady=5, padx=5)  
but5.grid(row=6,column=1, sticky=W+E+N+S, pady=5, padx=5)  
but6.grid(row=7,column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
#Create a set themselves:  
but7.grid(row=5,column=2, sticky=W+E+N+S, pady=5, padx=5)  
but8.grid(row=6,column=2, sticky=W+E+N+S, pady=5, padx=5)  
but9.grid(row=7,column=2, sticky=W+E+N+S, pady=5, padx=5)  
  
#Universal set  
lab7.grid(row=8,column=1, sticky=W+E+N+S)  
but10.grid(row=9,column=1, sticky=W+E+N+S)  
  
#Menu  
m = Menu(root)  
root.config(menu=m)  
fm = Menu(m)  
m.add\_cascade(label="Windows",menu=fm)  
fm.add\_command(label="Window№2", command=window2.create\_window\_2)  
fm.add\_command(label="Window№3", command=window3.create\_window\_3)  
fm.add\_command(label="Window№4", command=window4.create\_window\_4)  
fm.add\_command(label="Window№5", command=window5.create\_window\_5)  
  
  
**def exitwindow**():  
 **if** messagebox.askokcancel("Quit", "Do you really wish to quit?"):  
 root.destroy()  
root.protocol("WM\_DELETE\_WINDOW", exitwindow)  
  
  
#Xz  
root.mainloop()

Модуль window2

**rom** tkinter **import** \*  
**import** func\_win\_2  
  
  
#Create window 2  
**def create\_window\_2**():  
  
 #Create Toplevel for window 2  
 window2 = Toplevel()  
 window2.title("Window 2")  
 window2.maxsize(width=475,height=290)  
 window2.minsize(width=475,height=290)  
  
  
 #Top label  
 lab1 = Label(window2, text='Window №2', font='arial 20')  
 lab2 = Label(window2, text='See set now', font='arial 20')  
 lab3 = Label(window2, text='Do something on the set', font='arial 20')  
  
  
 #Left Button  
 but1 = Button(window2, text="See set A", font='arial 20')  
 but1.bind("<Button-1>", func\_win\_2.see\_set\_A)  
 but2 = Button(window2, text="See set B", font='arial 20')  
 but2.bind("<Button-1>", func\_win\_2.see\_set\_B)  
 but3 = Button(window2, text="See set C", font='arial 20')  
 but3.bind("<Button-1>", func\_win\_2.see\_set\_C)  
  
  
 #Right Button  
 but4 = Button(window2, text="Do next step", font='arial 20')  
 but4.bind("<Button-1>", func\_win\_2.open\_toplevel\_next\_step)  
 but5 = Button(window2, text="See set D", font='arial 20')  
 but5.bind("<Button-1>", func\_win\_2.see\_set\_D)  
 but6 = Button(window2, text="Save set D", font='arial 20')  
 but6.bind("<ButtonRelease-1>", func\_win\_2.save\_set\_D)  
  
  
 #Top label  
 lab1.grid(row=1,column=1,columnspan=2, sticky=W+E+N+S, pady=5, padx=5)  
 lab2.grid(row=2,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 lab3.grid(row=2,column=2, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Lefr Button  
 but1.grid(row=4,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 but2.grid(row=5,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 but3.grid(row=6,column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Right Button  
 but4.grid(row=4,column=2, sticky=W+E+N+S, pady=5, padx=5)  
 but5.grid(row=5,column=2, sticky=W+E+N+S, pady=5, padx=5)  
 but6.grid(row=6,column=2, sticky=W+E+N+S, pady=5, padx=5)

Модуль window3

**from** tkinter **import** \*  
**import** func\_win\_2,func\_win\_3  
  
  
#Create window 3  
**def create\_window\_3**():  
  
 #Create Toplevel for window 3  
 window3 = Toplevel()  
 window3.title("Window 3")  
 window3.maxsize(width=475,height=290)  
 window3.minsize(width=475,height=290)  
  
  
 #Top label  
 lab1 = Label(window3, text='Window №3', font='arial 20')  
 lab2 = Label(window3, text='See set now', font='arial 20')  
 lab3 = Label(window3, text='Do something on the set', font='arial 20')  
  
  
 #Left Button  
 but1 = Button(window3, text="See set A", font='arial 20')  
 but1.bind("<Button-1>", func\_win\_2.see\_set\_A)  
 but2 = Button(window3, text="See set B", font='arial 20')  
 but2.bind("<Button-1>", func\_win\_2.see\_set\_B)  
 but3 = Button(window3, text="See set C", font='arial 20')  
 but3.bind("<Button-1>", func\_win\_2.see\_set\_C)  
  
  
 #Right Button  
 but4 = Button(window3, text="Do next step", font='arial 20')  
 but4.bind("<Button-1>", func\_win\_3.open\_toplevel\_next\_step)  
 but5 = Button(window3, text="See set D", font='arial 20')  
 but5.bind("<Button-1>", func\_win\_3.see\_set\_D)  
 but6 = Button(window3, text="Save set D", font='arial 20')  
 but6.bind("<ButtonRelease-1>", func\_win\_3.save\_set\_D)  
  
  
 #Top label  
 lab1.grid(row=1,column=1,columnspan=2, sticky=W+E+N+S, pady=5, padx=5)  
 lab2.grid(row=2,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 lab3.grid(row=2,column=2, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Lefr Button  
 but1.grid(row=4,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 but2.grid(row=5,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 but3.grid(row=6,column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Right Button  
 but4.grid(row=4,column=2, sticky=W+E+N+S, pady=5, padx=5)  
 but5.grid(row=5,column=2, sticky=W+E+N+S, pady=5, padx=5)  
 but6.grid(row=6,column=2, sticky=W+E+N+S, pady=5, padx=5)

Модуль window4

**from** tkinter **import** \*  
**import** func\_win\_4  
  
  
#Create window 4  
**def create\_window\_4**():  
  
 #Create Toplevel for window 4  
 window4 = Toplevel()  
 window4.title("Window 4")  
 window4.maxsize(width=475,height=290)  
 window4.minsize(width=475,height=290)  
  
  
 #Top label  
 lab1 = Label(window4, text='Window №4', font='arial 20')  
 lab2 = Label(window4, text='See set now', font='arial 20')  
 lab3 = Label(window4, text='Do something on the set', font='arial 20')  
  
  
 #Left Button  
 but1 = Button(window4, text="See set X", font='arial 20')  
 but1.bind("<Button-1>", func\_win\_4.see\_set\_X)  
 but2 = Button(window4, text="See set Y", font='arial 20')  
 but2.bind("<Button-1>", func\_win\_4.see\_set\_Y)  
  
  
 #Right Button  
 but3 = Button(window4, text="X \ Y", font='arial 20')  
 but3.bind("<Button-1>", func\_win\_4.step)  
 but4 = Button(window4, text="See set Z", font='arial 20')  
 but4.bind("<Button-1>", func\_win\_4.see\_set\_Z)  
  
  
 #Bottom Button  
 but5 = Button(window4, text="Save set Z", font='arial 20')  
 but5.bind("<ButtonRelease-1>", func\_win\_4.save\_set\_Z)  
  
  
 #Top label  
 lab1.grid(row=1,column=1,columnspan=2, sticky=W+E+N+S, pady=5, padx=5)  
 lab2.grid(row=2,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 lab3.grid(row=2,column=2, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Lefr Button  
 but1.grid(row=4,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 but2.grid(row=5,column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Right Button  
 but3.grid(row=4,column=2, sticky=W+E+N+S, pady=5, padx=5)  
 but4.grid(row=5,column=2, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Bottom Button  
 but5.grid(row=6,column=1, columnspan=2, sticky=W+E+N+S, pady=5, padx=5)

Модуль window5

**from** tkinter **import** \*  
**import** func\_win\_5  
  
  
#Create window 5  
**def create\_window\_5**():  
  
 #Create Toplevel for window 5  
 window5 = Toplevel()  
 window5.title("Window 5")  
 window5.maxsize(width=715,height=485)  
 window5.minsize(width=715,height=485)  
  
  
 #Top Tabel  
 lab1 = Label(window5, text='Window №5', font='arial 20')  
 lab2 = Label(window5, text='See set D into file', font='arial 20')  
 lab3 = Label(window5, text='See set Z into file', font='arial 20')  
 lab4 = Label(window5, text='Comparison sets', font='arial 20')  
  
  
 #Left button  
 but1 = Button(window5, text="See set D(don`t simplified) into file", font='arial 14')  
 but1.bind("<ButtonRelease-1>", func\_win\_5.see\_dont\_simplified\_D)  
 but2 = Button(window5, text="See set D(simplified) into file", font='arial 14')  
 but2.bind("<ButtonRelease-1>", func\_win\_5.see\_simplified\_D)  
  
  
 #Center button  
 but3 = Button(window5, text="See set Z(calculated with my functions) into file", font='arial 14')  
 but3.bind("<ButtonRelease-1>", func\_win\_5.see\_calc\_about\_my\_func\_C)  
 but4 = Button(window5, text="See and save set Z(calculated with Python functions) into file", font='arial 14')  
 but4.bind("<ButtonRelease-1>", func\_win\_5.step)  
  
  
 #Right button  
 but5 = Button(window5, text="comparison D(simplified) and D(don`t simplified)", font='arial 14')  
 but5.bind("<ButtonRelease-1>", func\_win\_5.comparison\_D\_sets)  
 but6 = Button(window5, text="comparison Z(calculated with my functions) and Z(calculated with Python functions)", font='arial 14')  
 but6.bind("<ButtonRelease-1>", func\_win\_5.comparison\_Z\_sets)  
  
  
 #Top Tabel  
 lab1.grid(row=1,column=1,columnspan=1, sticky=W+E+N+S, pady=5, padx=5)  
 lab2.grid(row=2,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 lab3.grid(row=5,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 lab4.grid(row=8,column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Left button  
 but1.grid(row=3,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 but2.grid(row=4,column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Center button  
 but3.grid(row=6,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 but4.grid(row=7,column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
  
 #Right button  
 but5.grid(row=9,column=1, sticky=W+E+N+S, pady=5, padx=5)  
 but6.grid(row=10,column=1, sticky=W+E+N+S, pady=5, padx=5)

Модуль My\_difference

#My function  
**def difference**(a, b):  
 res = []  
 **for** i **in** a:  
 d = **True  
 for** j **in** b:  
 **if** i == j:  
 d = **False  
 break  
 if** d:  
 res.append(i)  
 **return** set(res)

Модуль func

**from** tkinter **import** \*  
**from** tkinter **import** messagebox  
**import** random  
  
  
#Firstly var  
power\_A = 5  
power\_B = 6  
power\_C = 7  
A = set()  
B = set()  
C = set()  
U = set()  
  
#Power for A  
**def give\_power\_A**(event):  
 top = Toplevel(height=500, width=500, relief = GROOVE)  
 top.maxsize(width=300,height=75)  
 top.minsize(width=300,height=75)  
 top.title("Power A")  
 ent = Entry(top,width=50, bd=3)  
 ent.pack()  
  
 **def callback**():  
 **global** power\_A  
 **try**:  
 power\_A = int(ent.get())  
 msg = messagebox.showinfo('Power A', 'Power A = ' + str(power\_A))  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
 b = Button(top, text="Save", width=10, command=callback, font = ("Arial",20))  
 b.pack()  
  
  
#Power for B  
**def give\_power\_B**(event):  
 top = Toplevel(height=500, width=500, relief = GROOVE)  
 top.title("Power B")  
 top.maxsize(width=300,height=75)  
 top.minsize(width=300,height=75)  
 ent = Entry(top,width=50, bd=3)  
 ent.pack()  
  
 **def callback**():  
 **global** power\_B  
 **try**:  
 power\_B = int(ent.get())  
 msg = messagebox.showinfo('Power B', 'Power B = ' + str(power\_B))  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
 b = Button(top, text="Save", width=10, command=callback, font = ("Arial",20))  
 b.pack()  
  
  
#Power for C  
**def give\_power\_C**(event):  
 top = Toplevel(height=500, width=500, relief = GROOVE)  
 top.title("Power C")  
 top.maxsize(width=300,height=75)  
 top.minsize(width=300,height=75)  
 ent = Entry(top,width=50, bd=3)  
 ent.pack()  
  
 **def callback**():  
 **global** power\_C  
 **try**:  
 power\_C = int(ent.get())  
 msg = messagebox.showinfo('Power C', 'Power C = ' + str(power\_C))  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
 b = Button(top, text="Save", width=10, command=callback, font = ("Arial",20))  
 b.pack()  
  
  
#Random create A  
**def create\_A**(event):  
 **global** A  
 A = set()  
  
 **while** len(A) < power\_A:  
 elem = random.randint(0, 255)  
 A.add(elem)  
 msg = messagebox.showinfo('Random create A', 'Random set A was created')  
  
  
  
#Random create B  
**def create\_B**(event):  
 **global** B  
 B = set()  
  
 **while** len(B) < power\_B:  
 elem = random.randint(0, 255)  
 B.add(elem)  
 msg = messagebox.showinfo('Random create B', 'Random set B was created')  
  
  
  
#Random create C  
**def create\_C**(event):  
 **global** C  
 C = set()  
  
 **while** len(C) < power\_C:  
 elem = random.randint(0, 255)  
 C.add(elem)  
 msg = messagebox.showinfo('Random create C', 'Random set C was created')  
  
  
#Create themselves A  
**def create\_themselves\_A**(event):  
 **global** A  
 A = set()  
  
 **def callback\_to\_range**(event):  
 **try**:  
 begin = int((ent1.get()))  
 over = int(ent2.get())  
 Add = set(range(begin,over))  
 A.update(Add)  
 msg = messagebox.showinfo('Create themselves A', 'Elements was add in set A')  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
  
  
 #Create toplevel  
 top = Toplevel(height=500, width=500, relief = GROOVE)  
 top.title("Created set A themselves")  
 top.maxsize(width=485,height=250)  
 top.minsize(width=485,height=250)  
  
 #Top label  
 lab1 = Label(top, text='Add diapasone to set', font='arial 20')  
  
  
 #Label For => To  
 lab2 = Label(top, text='From', font='arial 20')  
 lab3 = Label(top, text='To(write +1 more)', font='arial 20')  
  
 #Entry elements  
 ent1 = Entry(top,width=15, bd=3)  
 ent2 = Entry(top,width=15, bd=3)  
 but1 = Button(top, text='Add elements', font='arial 20')  
 but1.bind("<ButtonRelease-1>", callback\_to\_range)  
  
  
  
 **def callback\_to\_one\_element**(event):  
 **try**:  
 value = int(ent3.get())  
 A.add(value)  
 msg = messagebox.showinfo('Create themselves A', 'Element was add in set A')  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
  
 #Label add one element  
 lab4 = Label(top, text='Add one element to set', font='arial 20')  
  
  
 #Entry one element  
 ent3 = Entry(top,width=25, bd=3)  
 but2 = Button(top, text='Add one element', font='arial 20')  
 but2.bind("<ButtonRelease-1>", callback\_to\_one\_element)  
  
  
 #Top label  
 lab1.grid(row=1,column=1,columnspan=2)  
  
  
 #Label For => To  
 lab2.grid(row=2,column=0)  
 lab3.grid(row=2,column=2)  
  
  
 #Entry elements  
 ent1.grid(row=2,column=1)  
 ent2.grid(row=2,column=3, sticky='w')  
 but1.grid(row=3,column=1,columnspan=2)  
  
  
 #Label add one element  
 lab4.grid(row=4,column=1,columnspan=2)  
  
  
 #Entry one element  
 ent3.grid(row=5,column=1, columnspan=2)  
 but2.grid(row=6,column=1,columnspan=2)  
  
  
#Create themselves B  
**def create\_themselves\_B**(event):  
 **global** B  
 B = set()  
  
 **def callback\_to\_range**(event):  
 **try**:  
 begin = int((ent1.get()))  
 over = int(ent2.get())  
 Add = set(range(begin,over))  
 B.update(Add)  
 msg = messagebox.showinfo('Create themselves B', 'Elements was add in set B')  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
  
 #Create toplevel  
 top = Toplevel(height=500, width=500, relief = GROOVE)  
 top.title("Created set B themselves")  
 top.maxsize(width=485,height=250)  
 top.minsize(width=485,height=250)  
  
  
 #Top label  
 lab1 = Label(top, text='Add diapasone to set', font='arial 20')  
  
  
 #Label For => To  
 lab2 = Label(top, text='From', font='arial 20')  
 lab3 = Label(top, text='To(write +1 more)', font='arial 20')  
  
  
 #Entry elements  
 ent1 = Entry(top,width=15, bd=3)  
 ent2 = Entry(top,width=15, bd=3)  
 but1 = Button(top, text='Add elements', font='arial 20')  
 but1.bind("<ButtonRelease-1>", callback\_to\_range)  
  
  
  
 **def callback\_to\_one\_element**(event):  
 **try**:  
 value = int(ent3.get())  
 B.add(value)  
 msg = messagebox.showinfo('Create themselves B', 'Element was add in set B')  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
  
 #Label add one element  
 lab4 = Label(top, text='Add one element to set', font='arial 20')  
  
  
 #Entry one element  
 ent3 = Entry(top,width=25, bd=3)  
 but2 = Button(top, text='Add one element', font='arial 20')  
 but2.bind("<ButtonRelease-1>", callback\_to\_one\_element)  
  
  
 #Top label  
 lab1.grid(row=1,column=1,columnspan=2)  
  
  
 #Label For => To  
 lab2.grid(row=2,column=0)  
 lab3.grid(row=2,column=2)  
  
  
 #Entry elements  
 ent1.grid(row=2,column=1)  
 ent2.grid(row=2,column=3, sticky='w')  
 but1.grid(row=3,column=1,columnspan=2)  
  
  
 #Label add one element  
 lab4.grid(row=4,column=1,columnspan=2)  
  
  
 #Entry one element  
 ent3.grid(row=5,column=1, columnspan=2)  
 but2.grid(row=6,column=1,columnspan=2)  
  
  
#Create themselves C  
**def create\_themselves\_C**(event):  
 **global** C  
 C = set()  
  
 **def callback\_to\_range**(event):  
 **try**:  
 begin = int((ent1.get()))  
 over = int(ent2.get())  
 Add = set(range(begin,over))  
 C.update(Add)  
 msg = messagebox.showinfo('Create themselves C', 'Elements was add in set C')  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
 #Create toplevel  
 top = Toplevel(height=500, width=500, relief = GROOVE)  
 top.title("Created set C themselves")  
 top.maxsize(width=485,height=250)  
 top.minsize(width=485,height=250)  
  
  
 #Top label  
 lab1 = Label(top, text='Add diapasone to set', font='arial 20')  
  
  
 #Label For => To  
 lab2 = Label(top, text='From', font='arial 20')  
 lab3 = Label(top, text='To(write +1 more)', font='arial 20')  
  
  
 #Entry elements  
 ent1 = Entry(top,width=15, bd=3)  
 ent2 = Entry(top,width=15, bd=3)  
 but1 = Button(top, text='Add elements', font='arial 20')  
 but1.bind("<ButtonRelease-1>", callback\_to\_range)  
  
  
  
 **def callback\_to\_one\_element**(event):  
 **try**:  
 value = int(ent3.get())  
 C.add(value)  
 msg = messagebox.showinfo('Create themselves C', 'Element was add in set C')  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
  
 #Label add one element  
 lab4 = Label(top, text='Add one element to set', font='arial 20')  
  
  
 #Entry one element  
 ent3 = Entry(top,width=25, bd=3)  
 but2 = Button(top, text='Add one element', font='arial 20')  
 but2.bind("<ButtonRelease-1>", callback\_to\_one\_element)  
  
  
 #Top label  
 lab1.grid(row=1,column=1,columnspan=2)  
  
  
 #Label For => To  
 lab2.grid(row=2,column=0)  
 lab3.grid(row=2,column=2)  
  
  
 #Entry elements  
 ent1.grid(row=2,column=1)  
 ent2.grid(row=2,column=3, sticky='w')  
 but1.grid(row=3,column=1,columnspan=2)  
  
  
 #Label add one element  
 lab4.grid(row=4,column=1,columnspan=2)  
  
  
 #Entry one element  
 ent3.grid(row=5,column=1, columnspan=2)  
 but2.grid(row=6,column=1,columnspan=2)  
  
  
#Create universal set  
**def create\_universal\_set**(event):  
 **global** U  
 U = set()  
  
 **def callback\_to\_range**(event):  
 **try**:  
 begin = int((ent1.get()))  
 over = int(ent2.get())  
 Add = set(range(begin,over))  
 U.update(Add)  
 msg = messagebox.showinfo('Universal set', 'Universal set was created')  
 **except**:  
 msg = messagebox.showinfo('Error', 'Please input only numeric')  
  
  
 #Create toplevel  
 top = Toplevel(height=500, width=500, relief = GROOVE)  
 top.title("Created universal set")  
 top.maxsize(width=485,height=135)  
 top.minsize(width=485,height=135)  
  
  
 #Top label  
 lab1 = Label(top, text='Add diapasone to set', font='arial 20')  
  
  
 #Label For => To  
 lab2 = Label(top, text='From', font='arial 20')  
 lab3 = Label(top, text='To(write +1 more)', font='arial 20')  
  
  
 #Entry elements  
 ent1 = Entry(top,width=15, bd=3)  
 ent2 = Entry(top,width=15, bd=3)  
 but1 = Button(top, text='Add elements', font='arial 20')  
 but1.bind("<ButtonRelease-1>", callback\_to\_range)  
  
  
 #Top label  
 lab1.grid(row=1,column=1,columnspan=2)  
  
  
 #Label For => To  
 lab2.grid(row=2,column=0)  
 lab3.grid(row=2,column=2)  
  
  
 #Entry elements  
 ent1.grid(row=2,column=1)  
 ent2.grid(row=2,column=3, sticky='w')  
 but1.grid(row=3,column=1,columnspan=2)

Модуль func\_win\_2

**from** tkinter **import** \*  
**import** func  
**from** tkinter **import** messagebox  
  
  
D = set()  
result1 = set()  
result2 = set()  
#See set A  
**def see\_set\_A**(event):  
 #Create toplevel  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set A")  
 printer = "A = " + str(func.A)  
 #Create Text window  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printer)  
 tex.pack(fill=BOTH,expand=**True**)  
  
  
#See set A  
**def see\_set\_B**(event):  
 #Create toplevel  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set B")  
 printer = "B = " + str(func.B)  
 #Create Text window  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printer)  
 tex.pack(fill=BOTH,expand=**True**)  
  
  
 #See set A  
**def see\_set\_C**(event):  
 #Create toplevel  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set C")  
 printer = "C = " + str(func.C)  
 #Create Text window  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printer)  
 tex.pack(fill=BOTH,expand=**True**)  
  
  
#Step doing  
**def open\_toplevel\_next\_step**(event):  
  
  
 #Create toplevel  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Single-step execution expression")  
 top.maxsize(width=650,height=390)  
 top.minsize(width=650,height=390)  
 result1 = set()  
 result2 = set()  
 D = set()  
  
  
 #Step 1  
 **def step1**(event):  
 **global** result1  
 result1 = func.U - func.A  
 printed = '¬A = ' + str(result1)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 1\n ¬A")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but1 = Button(top, text="Step 1\n¬A", font='arial 14')  
 but1.bind("<Button-1>", step1)  
  
  
 #Step 2  
 **def step2**(event):  
 **global** result1  
 result1 = func.U - func.A | func.A  
 printed = '¬A v A= ' + str(result1)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 2\n ¬A v A")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but2 = Button(top, text="Step 2\n¬A v A", font='arial 14')  
 but2.bind("<Button-1>", step2)  
  
  
 #Step 3  
 **def step3**(event):  
 **global** result1  
 result1 = (func.U - func.A | func.A) & func.A  
 printed = '(¬A v A) ^ A = ' + str(result1)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 3\n (¬A v A) ^ A")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but3 = Button(top, text="Step 3\n(¬A v A) ^ A", font='arial 14')  
 but3.bind("<Button-1>", step3)  
  
  
 #Step 4  
 **def step4**(event):  
 **global** result1  
 result1 = ((func.U - func.A | func.A) & func.A) - func.B  
 printed = '((¬A v A) ^ A)\\B = ' + str(result1)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 4\n ((¬A v A) ^ A)\\B")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but4 = Button(top, text="Step 4\n((¬A v A) ^ A)\\B", font='arial 14')  
 but4.bind("<Button-1>", step4)  
  
  
 #Step 5  
 **def step5**(event):  
 **global** result1  
 result1 = (((func.U - func.A | func.A) & func.A) - func.B) | func.B  
 printed = '((¬A v A) ^ A)\\B v B = ' + str(result1)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 5\n ((¬A v A) ^ A)\\B v B")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but5 = Button(top, text="Step 5\n((¬A v A) ^ A)\\B v B", font='arial 14')  
 but5.bind("<Button-1>", step5)  
  
  
 #Step 6  
 **def step6**(event):  
 **global** result2  
 result2 = func.C & func.B  
 printed = 'C ^ B = ' + str(result2)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 6\n C ^ B")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but6 = Button(top, text="Step 6\nC ^ B", font='arial 14')  
 but6.bind("<Button-1>", step6)  
  
  
 #Step 7  
 **def step7**(event):  
 **global** result2  
 result2 = (func.C & func.B) | func.C  
 printed = '(C ^ B) v C = ' + str(result2)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 7\n (C ^ B) v C")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but7 = Button(top, text="Step 7\n(C ^ B) v C", font='arial 14')  
 but7.bind("<Button-1>", step7)  
  
  
 #Step 8  
 **def step8**(event):  
 **global** result2  
 result2 = func.U - ((func.C & func.B) | func.C)  
 printed = '¬(C ^ B) v C) = ' + str(result2)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 8\n ¬((C ^ B) v C)")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but8 = Button(top, text="Step 8\n¬(C ^ B) v C)", font='arial 14')  
 but8.bind("<Button-1>", step8)  
  
  
 #Step 9  
 **def step9**(event):  
 **global** D  
 **global** result1  
 **global** result2  
 D = ((((func.U - func.A | func.A) & func.A) - func.B) | func.B) & (func.U - ((func.C & func.B) | func.C))  
 printed = 'D = ' + str(D)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 9\n (((¬A v A) ^ A)\\B v B) ^ (¬(C ^ B) v C)")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but9 = Button(top, text="Step 9\n(((¬A v A) ^ A)\\B v B) ^ (¬(C ^ B) v C)", font='arial 14')  
 but9.bind("<Button-1>", step9)  
  
 lab1 = Label(top, text='Do with left example', font='arial 20')  
 lab2 = Label(top, text='Do with right example', font='arial 20')  
 lab3 = Label(top, text='Set D\n(Total with left and right example)', font='arial 20')  
  
 #Packed labels  
 lab1.grid(row=1,column=1)  
 lab2.grid(row=1,column=2)  
 lab3.grid(row=5,column=2)  
  
  
 #Packed buttons  
 but1.grid(row=2,column=1, sticky=W+E,padx=5,pady=5)  
 but2.grid(row=3,column=1, sticky=W+E,padx=5,pady=5)  
 but3.grid(row=4,column=1, sticky=W+E,padx=5,pady=5)  
 but4.grid(row=5,column=1, sticky=W+E,padx=5,pady=5)  
 but5.grid(row=6,column=1, sticky=W+E)  
 but6.grid(row=2,column=2, sticky=W+E,padx=5,pady=5)  
 but7.grid(row=3,column=2, sticky=W+E,padx=5,pady=5)  
 but8.grid(row=4,column=2, sticky=W+E,padx=5,pady=5)  
 but9.grid(row=6,column=2, sticky=W+E,padx=5,pady=5)  
  
  
#See set D  
**def see\_set\_D**(event):  
 printed = 'D = ' + str(D)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set D")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
  
#Save set D  
**def save\_set\_D**(event):  
 **with** open(r"Set D(don`t simplified).txt", "w") **as** f:  
 f.write('D = ' + str(D))  
 msg = messagebox.showinfo('Save set D', 'Set D was saved')

Модуль func\_win\_3

**from** tkinter **import** \*  
**import** func  
**from** tkinter **import** messagebox  
  
D = set()  
result = set()  
  
  
#Open toplevel to do next step  
**def open\_toplevel\_next\_step**(event):  
  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.maxsize(width=270,height=250)  
 top.minsize(width=270,height=250)  
 top.title("Single-step execution expression")  
 result = set()  
 D = set()  
  
 #Step 1  
 **def step1**(event):  
 **global** D  
 D = func.A | func.B  
 printed = 'A v B = ' + str(D)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 1\n A v B")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but1 = Button(top, text="Step 1\nA v B", font='arial 14')  
 but1.bind("<Button-1>", step1)  
  
  
 #Step 2  
 **def step2**(event):  
 **global** result  
 result = func.U - func.C  
 printed = '¬C = ' + str(result)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 2\n ¬C")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but2 = Button(top, text="Step 2\n¬C", font='arial 14')  
 but2.bind("<Button-1>", step2)  
  
  
 #Step 3  
 **def step3**(event):  
 **global** result  
 **global** D  
 D = (func.A | func.B) & (func.U - func.C)  
 printed = 'D = ' + str(D)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Step 3\n (A v B) ^ ¬C")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
 but3 = Button(top, text="Step 3\n(A v B) ^ ¬C ", font='arial 14')  
 but3.bind("<Button-1>", step3)  
  
 lab1 = Label(top, text='Do with easy example', font='arial 20')  
  
  
 #Packed labels  
 lab1.grid(row=1,column=1)  
  
  
 #Packed buttons  
 but1.grid(row=2,column=1, sticky=W+E,padx=5,pady=5)  
 but2.grid(row=3,column=1, sticky=W+E,padx=5,pady=5)  
 but3.grid(row=4,column=1, sticky=W+E,padx=5,pady=5)  
  
  
#See set D  
**def see\_set\_D**(event):  
 printed = 'D = ' + str(D)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set D")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
  
#Save set D  
**def save\_set\_D**(event):  
 **with** open(r"Set D(simplified).txt", "w") **as** f:  
 f.write('D = ' + str(D))  
 msg = messagebox.showinfo('Save set D', 'Set D was saved')

Модуль func\_win\_4

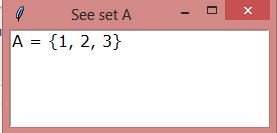
**from** tkinter **import** \*  
**from** tkinter **import** messagebox  
**import** func,My\_differense  
  
  
#See set X  
**def see\_set\_X**(event):  
 printed = 'X = ' + str(func.A)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set X")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
  
#See set Y  
**def see\_set\_Y**(event):  
 printed = 'Y = ' + str(func.C)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set Y")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
Z = set()  
  
  
#Step  
**def step**(event):  
 **global** Z  
 Z = My\_differense.difference(func.A ,func.C)  
 printed = 'Z = ' + str(Z)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("X \ Y")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
  
#See set Z  
**def see\_set\_Z**(event):  
 printed = 'Z = ' + str(Z)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set Z")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
  
  
#Save set Z  
**def save\_set\_Z**(event):  
 **with** open(r"Set Z(calc about my func).txt", "w") **as** f:  
 f.write('Z = ' + str(Z))  
 msg = messagebox.showinfo('Save set Z', 'Set Z was saved')

Модуль func\_win\_5

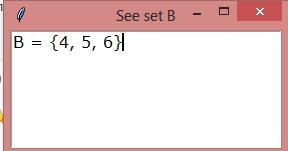
**import** func  
**from** tkinter **import** \*  
**from** tkinter **import** messagebox  
  
  
Z = set()  
#See set D(dont simplified) into file  
**def** see\_dont\_simplified\_D(event):  
 **try**:  
 **with** open(r"Set D(don`t simplified).txt", "r") **as** f:  
 p = f.read()  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set D(don`t simplified) into file")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,p)  
 tex.pack(fill=BOTH,expand=**True**)  
 **except**:  
 msg = messagebox.showinfo('Error', 'File not found')  
  
  
#See set D(simplified) into file  
**def see\_simplified\_D**(event):  
 **try**:  
 **with** open(r"Set D(simplified).txt", "r") **as** f:  
 p = f.read()  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set D(simplified) into file")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,p)  
 tex.pack(fill=BOTH,expand=**True**)  
 **except**:  
 msg = messagebox.showinfo('Error', 'File not found')  
  
  
#See set Z cacl about my func  
**def see\_calc\_about\_my\_func\_C**(event):  
 **try**:  
 **with** open(r"Set Z(calc about my func).txt", "r") **as** f:  
 p = f.read()  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("See set Z(calc about my func) into file")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,p)  
 tex.pack(fill=BOTH,expand=**True**)  
 **except**:  
 msg = messagebox.showinfo('Error', 'File not found')  
  
#Step(Difference about Python func)  
**def step**(event):  
 **global** Z  
 Z = func.A - func.C  
 printed = 'Z = ' + str(Z)  
 top = Toplevel(height=500, width=100, relief = GROOVE)  
 top.title("Calculated and see set Z about Python func")  
 tex = Text(top,width=180, height=20,font="Verdana 12",wrap=WORD)  
 tex.insert(END,printed)  
 tex.pack(fill=BOTH,expand=**True**)  
 **with** open(r"Set Z(calc about Python func).txt", "w") **as** f:  
 f.write('Z = ' + str(Z))  
 msg = messagebox.showinfo('Save set Z', 'Set Z was saved')  
  
  
#comparison sets D  
**def comparison\_D\_sets**(event):  
 **try**:  
 f1 = open(r"Set D(don`t simplified).txt", "r")  
 p1 = f1.read()  
 f2 = open(r"Set D(simplified).txt", "r")  
 p2 = f2.read()  
 **if** p1 == p2:  
 msg = messagebox.showinfo('Comparison D sets', 'Sets identical')  
 **else**:  
 msg = messagebox.showinfo('Comparison D sets', 'Sets unidentical')  
 **except**:  
 msg = messagebox.showinfo('Error', 'Files not found')  
  
  
#comparison sets Z  
**def comparison\_Z\_sets**(event):  
 **try**:  
 f1 = open(r"Set Z(calc about my func).txt", "r")  
 p1 = f1.read()  
 f2 = open(r"Set Z(calc about Python func).txt", "r")  
 p2 = f2.read()  
 **if** p1 == p2:  
 msg = messagebox.showinfo('Comparison Z sets', 'Sets identical')  
 **else**:  
 msg = messagebox.showinfo('Comparison Z sets', 'Sets unidentical')  
 **except**:  
 msg = messagebox.showinfo('Error', 'File not found')

Роздруківка результатів виконання програми з контрольним прикладом

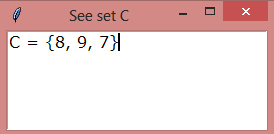
Множина А



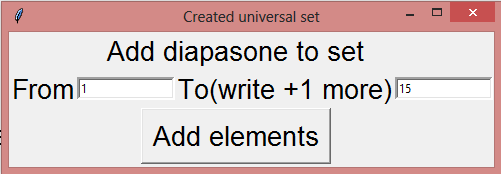
Множина В



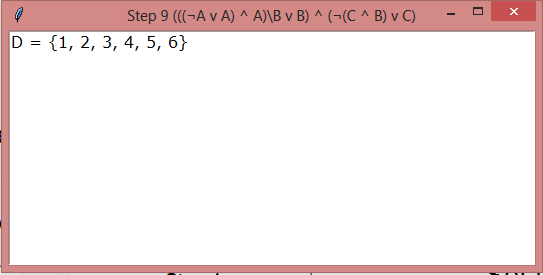
Множина С



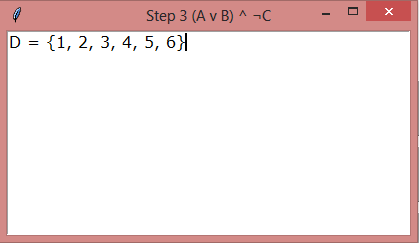
Універсальна множина



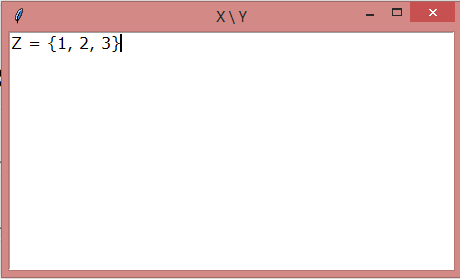
Порахований вираз



Порахований спрощений вираз



Різниця множин



**Аналіз результатів та висновки**

Протягом виконання роботи я ознайомився з алгеброю множин, вивчив тотожності алгебри множин, зрозумів принципи алгебри множин, та закріпив занання GUI в модулі tkinter в Python.

Завдання було цікавим, розробка своєї першої програми з інтерфейсом мені сподобалася. В ході розробки програми виникало багато помилок та незрозумілостей, але все вдалося налагодити за допомогою додаткової інформації в інтернеті.