Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Алгоритми та методи обчислення

Лабораторна робота №1

**«Поняття алгоритму. Задавання алгоритмів у вигляді блок-схем»**

Виконала:

студентка групи ІО-64

Бровченко А. В.

Перевірив:

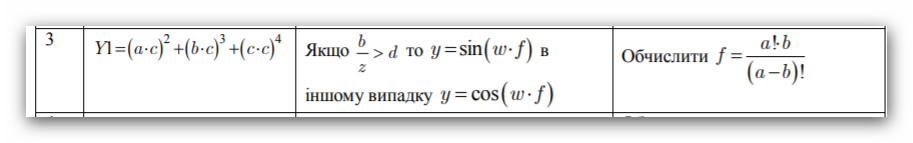
ст.вик. Порєв В. М.

Київ

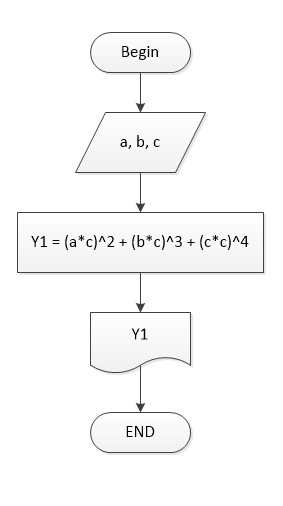
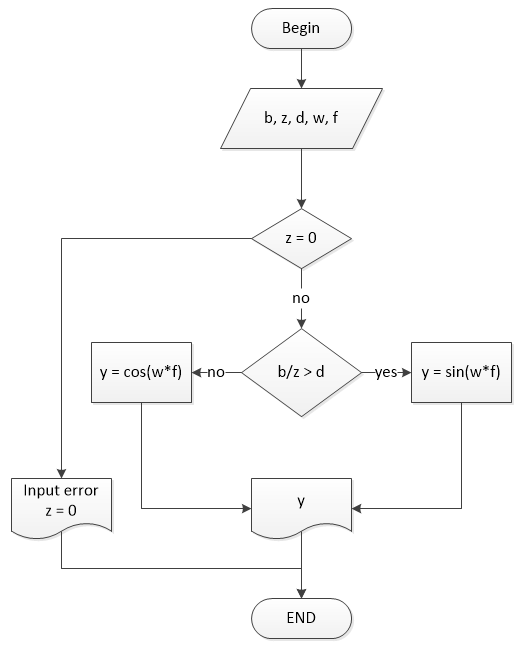
2017 р.

**Мета:** Навчитися створювати блок-схеми лінійного алгоритму; розгалуженого алгоритму та циклічного алгоритму за допомогою редактора блок-схем *afce*або іншого довільного редактора.

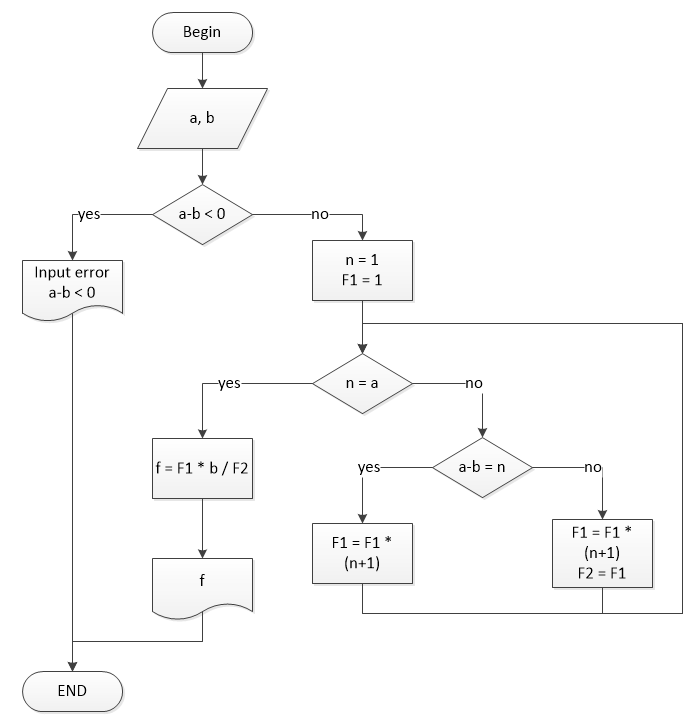
**Завдання:** Відповідно до варіанту завдання розробити блок-схеми обчислення виразів для лінійного алгоритму, алгоритму, що розгалужується та циклічного алгоритму. У відповідності до блок-схеми створити програму обчислення виразу алгоритмічною мовою, узгодженою з викладачем.



Лінійний алгоритм: Алгоритм з розгалудженням:



Циклічний алгоритм:



**Роздруківка тексту програми:**

algs.py

**def** factorial(x):  
 **if** x < 0:  
 **raise** ValueError  
 **else**:  
 res = 1  
 **for** i **in** list(range(x)):  
 res = res\*(i+1)  
 **return** res  
  
  
**def** alg1(a, b, c):  
 **return** (a\*c)\*\*2 + (b\*c)\*\*3 + c\*\*8  
  
  
**def** alg2(b, z, d, w, f):  
 **from** math **import** sin, cos  
  
 **if** z == 0: **return 'ZeroDivisonError. Check your input.'  
 elif** b/z > d: **return** sin(w\*f)  
 **else**: **return** cos(w\*f)  
  
  
**def** alg3(a, b):  
 **if** a-b < 0: **return 'a-b < 0. Enable to calculate (a-b)!.\n'** \  
 **'Check your input.'  
 else**: **return** factorial(a) \* b / factorial(a-b)

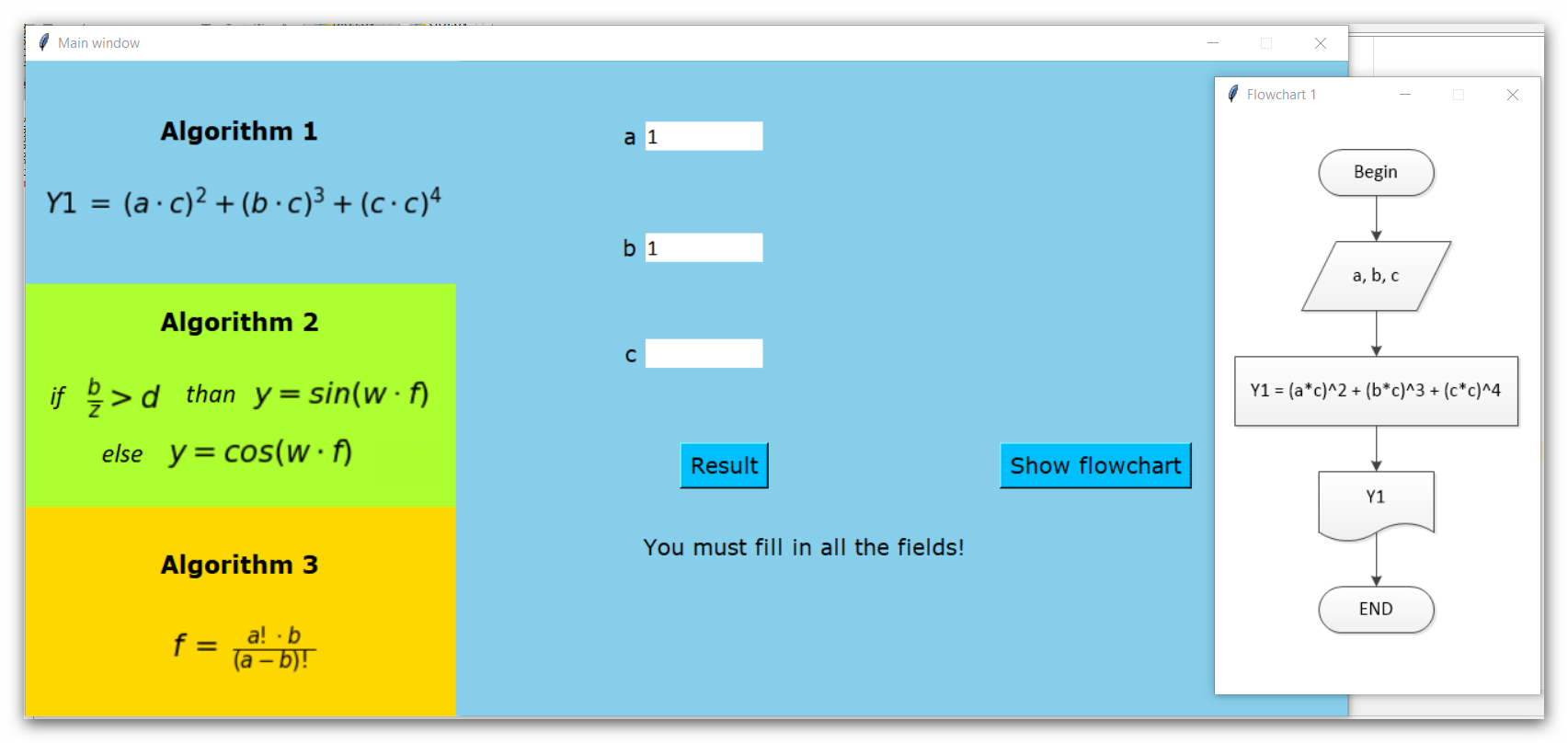
GUI.py

**from** tkinter **import** \*  
**from** PIL **import** Image, ImageTk  
**import** third\_sem.AMO.lab1.algs **as** algs  
  
  
**def** but\_result1():  
 *"""Shows results of the algorithm 1 or error message."""* **global** l1  
  
 *# check for input errors* **try**:  
 \_a = ent\_a1.get()  
 \_b = ent\_b1.get()  
 \_c = ent\_c1.get()  
 **if** \_a == **'' or** \_b == **'' or** \_c == **''**:  
 \_res = **'You must fill in all the fields!'  
 else**:  
 \_res = round(algs.alg1(int(\_a), int(\_b), int(\_c)), 3)  
 **except**: \_res = **'There are non-numeric symbols in your input.'** *# determination of the coordinates on which the result will appear* **if** len(str(\_res)) < 15: *# no errors* \_c = 4  
 \_r = 7  
 \_cs = 3  
 **else**:  
 \_c = 3  
 \_r = 8  
 \_cs = 7  
  
 **try**:  
 l1.destroy()  
 **finally**:  
 l1 = Label(root, text=\_res, font=main\_font, bg=bg\_color1)  
 l1.grid(column=\_c, row=\_r, columnspan=\_cs, sticky=W)  
  
  
**def** but\_result2():  
 *"""Shows results of the algorithm 2 or error message."""* **global** l2  
  
 *# check for input errors* **try**:  
 \_b = int(ent\_b2.get())  
 \_z = int(ent\_z2.get())  
 \_d = int(ent\_d2.get())  
 \_w = int(ent\_w2.get())  
 \_f = int(ent\_f2.get())  
 **if** \_b == **'' or** \_z == **'' or** \_d == **'' or** \_w == **'' or** \_f == **''**:  
 \_res = **'You must fill in all the fields!'  
 else**:  
 \_res = round(algs.alg2(int(\_b), int(\_z), int(\_d), int(\_w), int(\_f)), 3)  
 **except**: \_res = **'There are non-numeric symbols in your input.'** *# determination of the coordinates on which the result will appear* **if** len(str(\_res)) < 15: *# no errors* \_c = 4  
 \_r = 7  
 \_cs = 3  
 **else**:  
 \_c = 3  
 \_r = 8  
 \_cs = 7  
  
 **try**:  
 l2.destroy()  
 **finally**:  
 l2 = Label(root, text=\_res, font=main\_font, bg=bg\_color2)  
 l2.grid(column=\_c, row=\_r, columnspan=\_cs, sticky=W)  
  
  
**def** but\_result3():  
 *"""Shows results of the algorithm 3 or error message."""* **global** l3  
  
 *# check for input errors* **try**:  
 \_a = int(ent\_a3.get())  
 \_b = int(ent\_b3.get())  
 **if** \_a == **'' or** \_b == **''**:  
 \_res = **'You must fill in all the fields!'  
 else**:  
 \_res = round(algs.alg3(int(\_a), int(\_b)), 3)  
 **except**: \_res = **'There are non-numeric symbols in your input.'** *# determination of the coordinates on which the result will appear* **if** len(str(\_res)) < 15: *# no errors* \_c = 4  
 \_r = 7  
 \_cs = 3  
 **else**:  
 \_c = 3  
 \_r = 8  
 \_cs = 7  
  
 **try**:  
 l3.destroy()  
 **finally**:  
 l3 = Label(root, text=\_res, font=main\_font, bg=bg\_color3)  
 l3.grid(column=\_c, row=\_r, columnspan=\_cs, sticky=W)  
  
  
**def** show\_flowchart1():  
 *"""Shows block diagram of a linear algorithm in new Toplevel window."""* \_flowchart1 = PhotoImage(file=**r"D:\Python\Brovchenko\_projects\third\_sem\AMO\lab1\al1.png"**)  
  
 slave1 = Toplevel(root)  
 slave1.focus\_set()  
 slave1.title(**'Flowchart 1'**)  
  
 \_l1 = Label(slave1, image=\_flowchart1)  
 \_l1.image = \_flowchart1  
 \_l1.place(x=0, y=0, relwidth=1, relheight=1)  
  
 \_w = \_flowchart1.width()  
 \_h = \_flowchart1.height()  
 slave1.geometry(**'%dx%d+0+0'** % (\_w, \_h))  
 slave1.resizable(width=**False**, height=**False**)  
  
  
**def** show\_flowchart2():  
 *"""Shows block diagram of branching algorithm in new Toplevel window."""* \_flowchart2 = PhotoImage(file=**r"D:\Python\Brovchenko\_projects\third\_sem\AMO\lab1\al2.png"**)  
  
 slave2 = Toplevel(root)  
 slave2.focus\_set()  
 slave2.title(**'Flowchart 2'**)  
  
 \_l1 = Label(slave2, image=\_flowchart2)  
 \_l1.image = \_flowchart2  
 \_l1.place(x=0, y=0, relwidth=1, relheight=1)  
  
 \_w = \_flowchart2.width()  
 \_h = \_flowchart2.height()  
 slave2.geometry(**'%dx%d+0+0'** % (\_w, \_h))  
 slave2.resizable(width=**False**, height=**False**)  
  
  
**def** show\_flowchart3():  
 *"""Shows block diagram of a cyclic algorithm in new Toplevel window."""* \_flowchart3 = PhotoImage(file=**r"D:\Python\Brovchenko\_projects\third\_sem\AMO\lab1\al3.png"**)  
  
 slave3 = Toplevel(root)  
 slave3.focus\_set()  
 slave3.title(**'Flowchart 3'**)  
  
 \_l1 = Label(slave3, image=\_flowchart3)  
 \_l1.image = \_flowchart3  
 \_l1.place(x=0, y=0, relwidth=1, relheight=1)  
  
 \_w = \_flowchart3.width()  
 \_h = \_flowchart3.height()  
 slave3.geometry(**'%dx%d+0+0'** % (\_w, \_h))  
 slave3.resizable(width=**False**, height=**False**)  
  
  
**def** but1\_bind():  
 *"""Appear fields to determine the variables of the algorithm 1."""* **global** ent\_a1, ent\_b1, ent\_c1  
 root.geometry(**'1150x570'**)  
  
 Label(root, text=**''**, font=**'Calibri 14'**, width=80, height=25, bg=bg\_color1)\  
 .grid(row=0, column=1, rowspan=12, columnspan=10)  
  
 Label(root, text=**'a'**, justify=RIGHT, font=main\_font, bg=bg\_color1).grid(column=2, row=1, sticky=E)  
 Label(root, text=**'b'**, justify=RIGHT, font=main\_font, bg=bg\_color1).grid(column=2, row=3, sticky=E)  
 Label(root, text=**'c'**, justify=RIGHT, font=main\_font, bg=bg\_color1).grid(column=2, row=5, sticky=E)  
  
 ent\_a1 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
 ent\_b1 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
 ent\_c1 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
  
 ent\_a1.grid(column=3, row=1, columnspan=2, sticky=W, padx=5)  
 ent\_b1.grid(column=3, row=3, columnspan=2, sticky=W, padx=5)  
 ent\_c1.grid(column=3, row=5, columnspan=2, sticky=W, padx=5)  
  
 Button(root, text=**'Result'**, font=main\_font, bg=**'deep sky blue'**, command=but\_result1).grid(column=3, row=7)  
 Button(root, text=**'Show flowchart'**, font=main\_font, bg=**'deep sky blue'**, command=show\_flowchart1)\  
 .grid(column=6, row=7, columnspan=3)  
  
  
**def** but2\_bind():  
 *"""Appear fields to determine the variables of the algorithm 2."""* **global** ent\_b2, ent\_z2, ent\_d2, ent\_w2, ent\_f2  
 root.geometry(**'1150x570'**)  
  
 Label(root, text=**''**, font=**'Calibri 14'**, width=80, height=25, bg=bg\_color2)\  
 .grid(row=0, column=1, rowspan=12, columnspan=10)  
  
 Label(root, text=**'b'**, justify=RIGHT, font=main\_font, bg=bg\_color2).grid(column=2, row=1, sticky=E)  
 Label(root, text=**'z'**, justify=RIGHT, font=main\_font, bg=bg\_color2).grid(column=2, row=3, sticky=E)  
 Label(root, text=**'d'**, justify=RIGHT, font=main\_font, bg=bg\_color2).grid(column=2, row=5, sticky=E)  
 Label(root, text=**'w'**, justify=RIGHT, font=main\_font, bg=bg\_color2).grid(column=6, row=1, sticky=E)  
 Label(root, text=**'f'**, justify=RIGHT, font=main\_font, bg=bg\_color2).grid(column=6, row=3, sticky=E)  
  
 ent\_b2 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
 ent\_z2 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
 ent\_d2 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
 ent\_w2 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
 ent\_f2 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
  
 ent\_b2.grid(column=3, row=1, columnspan=2, sticky=W, padx=5)  
 ent\_z2.grid(column=3, row=3, columnspan=2, sticky=W, padx=5)  
 ent\_d2.grid(column=3, row=5, columnspan=2, sticky=W, padx=5)  
 ent\_w2.grid(column=7, row=1, columnspan=2, sticky=W, padx=5)  
 ent\_f2.grid(column=7, row=3, columnspan=2, sticky=W, padx=5)  
  
 Button(root, text=**'Result'**, font=main\_font, bg=**'chartreuse2'**, command=but\_result2).grid(column=3, row=7)  
 Button(root, text=**'Show flowchart'**, font=main\_font, bg=**'chartreuse2'**, command=show\_flowchart2)\  
 .grid(column=6, row=7, columnspan=3)  
  
  
**def** but3\_bind():  
 *"""Appear fields to determine the variables of the algorithm 3."""* **global** ent\_a3, ent\_b3  
 root.geometry(**'1150x570'**)  
  
 Label(root, text=**''**, font=**'Calibri 14'**, width=80, height=25, bg=bg\_color3)\  
 .grid(row=0, column=1, rowspan=12, columnspan=10)  
  
 Label(root, text=**'a'**, justify=RIGHT, font=main\_font, bg=bg\_color3).grid(column=2, row=1, sticky=E)  
 Label(root, text=**'b'**, justify=RIGHT, font=main\_font, bg=bg\_color3).grid(column=2, row=3, sticky=E)  
  
 ent\_a3 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
 ent\_b3 = Entry(root, font=**'Calibri 14'**, width=10, bd=0)  
  
 ent\_a3.grid(column=3, row=1, columnspan=2, sticky=W, padx=5)  
 ent\_b3.grid(column=3, row=3, columnspan=2, sticky=W, padx=5)  
  
 Button(root, text=**'Result'**, font=main\_font, bg=**'orange'**, command=but\_result3).grid(column=3, row=7)  
 Button(root, text=**'Show flowchart'**, font=main\_font, bg=**'orange'**, command=show\_flowchart3)\  
 .grid(column=6, row=7, columnspan=3)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 root = Tk()  
 root.title(**'Main window'**)  
 root.resizable(width=**False**, height=**False**)  
 root.geometry(**'370x570'**)  
  
 *# set background* bg\_color1 = **'sky blue'** bg\_color2 = **'green yellow'** bg\_color3 = **'gold'** main\_font = **'Verdana 14'** *# main buttons* photo1 = PhotoImage(file=**r"D:\Python\Brovchenko\_projects\third\_sem\AMO\lab1\1.png"**)  
 photo2 = PhotoImage(file=**r"D:\Python\Brovchenko\_projects\third\_sem\AMO\lab1\2.png"**)  
 photo3 = PhotoImage(file=**r"D:\Python\Brovchenko\_projects\third\_sem\AMO\lab1\3.png"**)  
  
 Button(root, compound=BOTTOM, image=photo1, bd=0, text=**'Algorithm 1\n'**, relief=GROOVE, width=370, height=190,  
 font=**'Verdana 16 bold'**, bg=bg\_color1, activebackground=bg\_color1, command=but1\_bind)\  
 .grid(column=0, row=0, rowspan=4,)  
  
 Button(root, compound=BOTTOM, image=photo2, bd=0, text=**'Algorithm 2\n'**, relief=GROOVE, width=370, height=190,  
 font=**'Verdana 16 bold'**, bg=bg\_color2, activebackground=bg\_color2, command=but2\_bind)\  
 .grid(column=0, row=4, rowspan=4,)  
  
 Button(root, compound=BOTTOM, image=photo3, bd=0, text=**'Algorithm 3\n'**, relief=GROOVE, width=370, height=190,  
 font=**'Verdana 16 bold'**, bg=bg\_color3, activebackground=bg\_color3, command=but3\_bind)\  
 .grid(column=0, row=8, rowspan=4,)  
  
 root.mainloop()

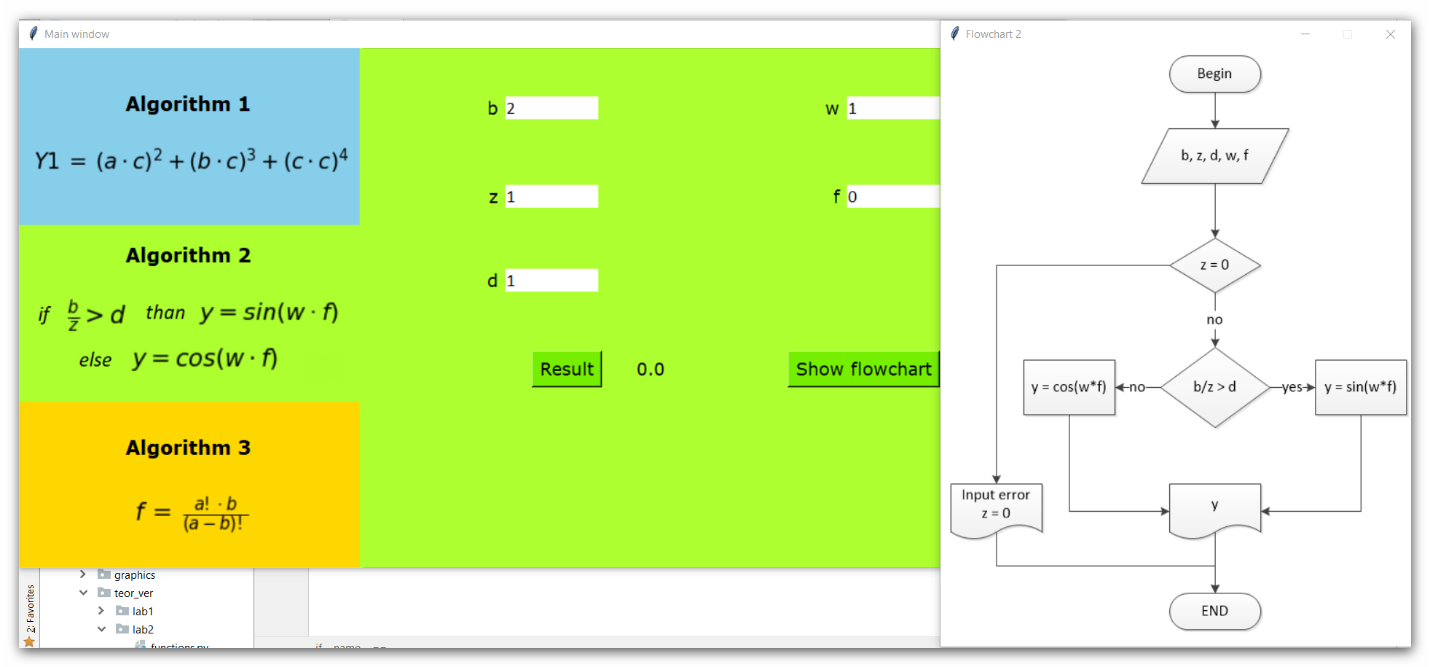
**Роздруківки результатів виконання програми**

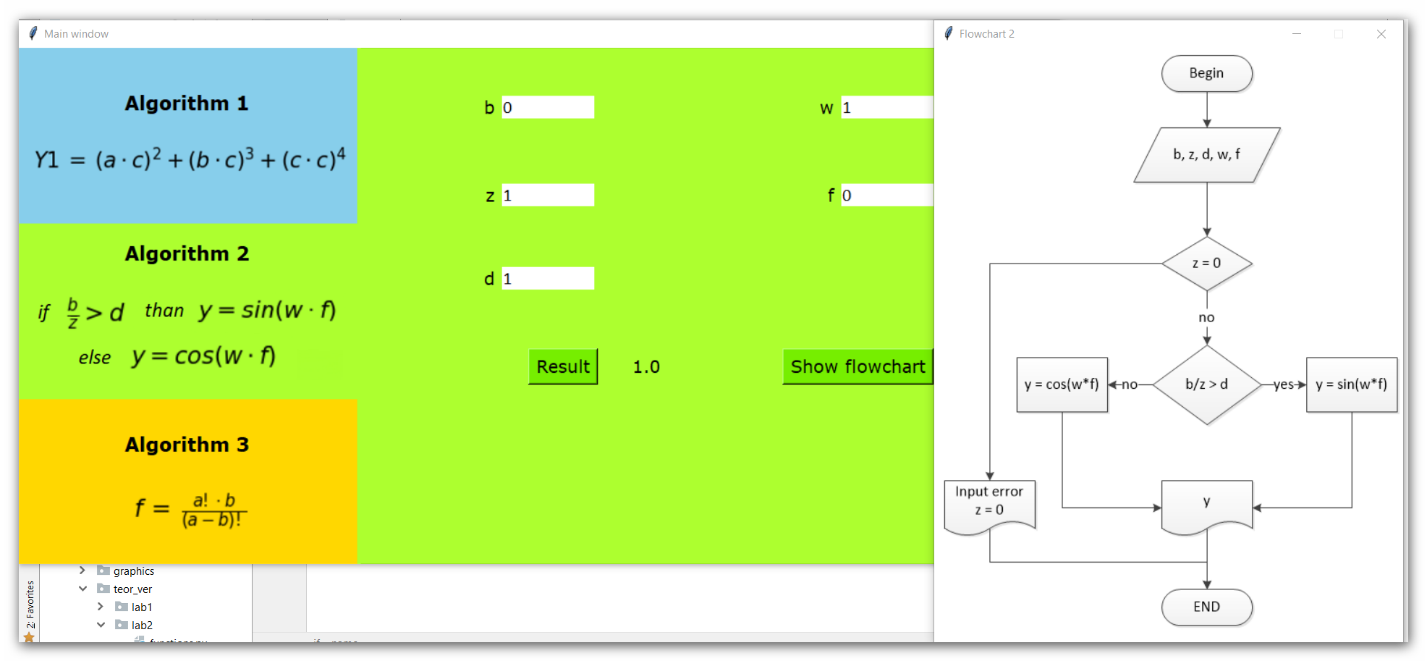
**Завдання 1**

****

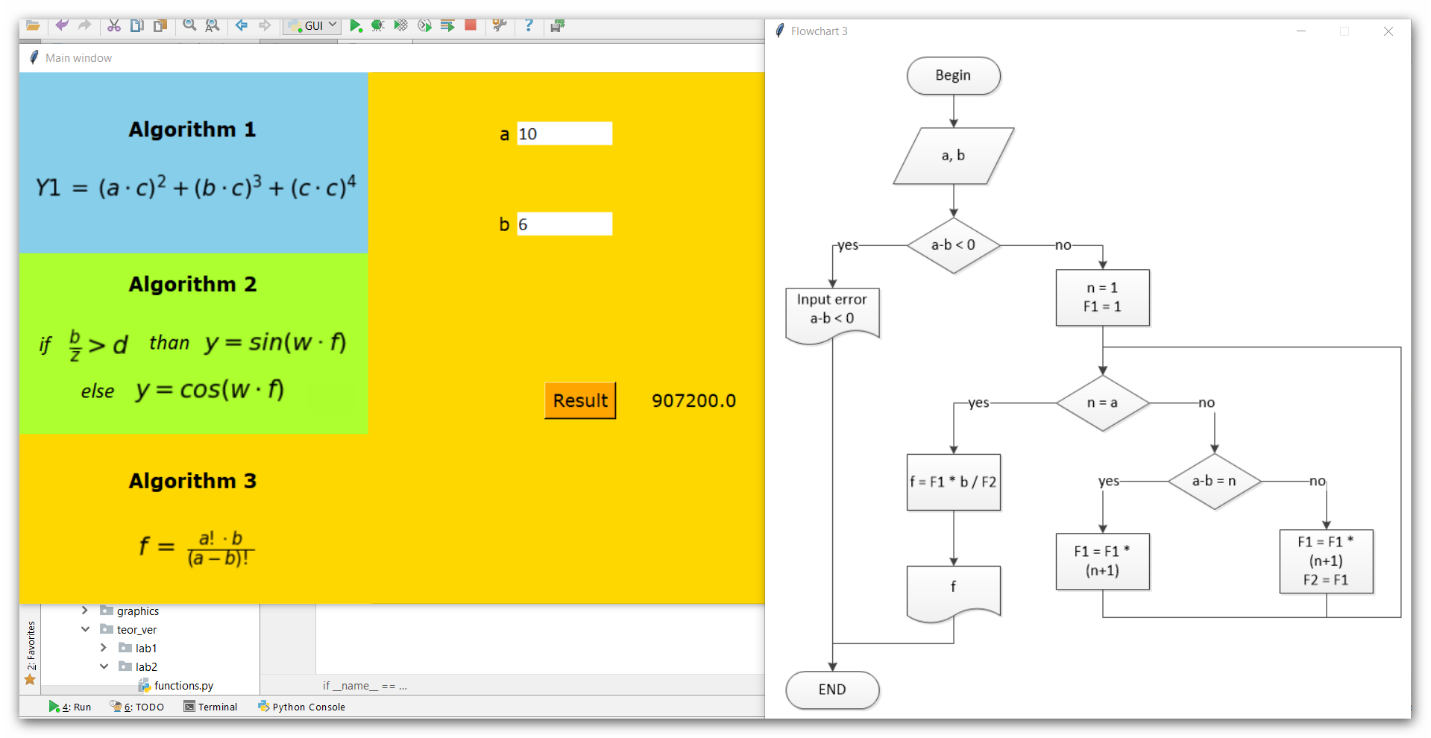
****

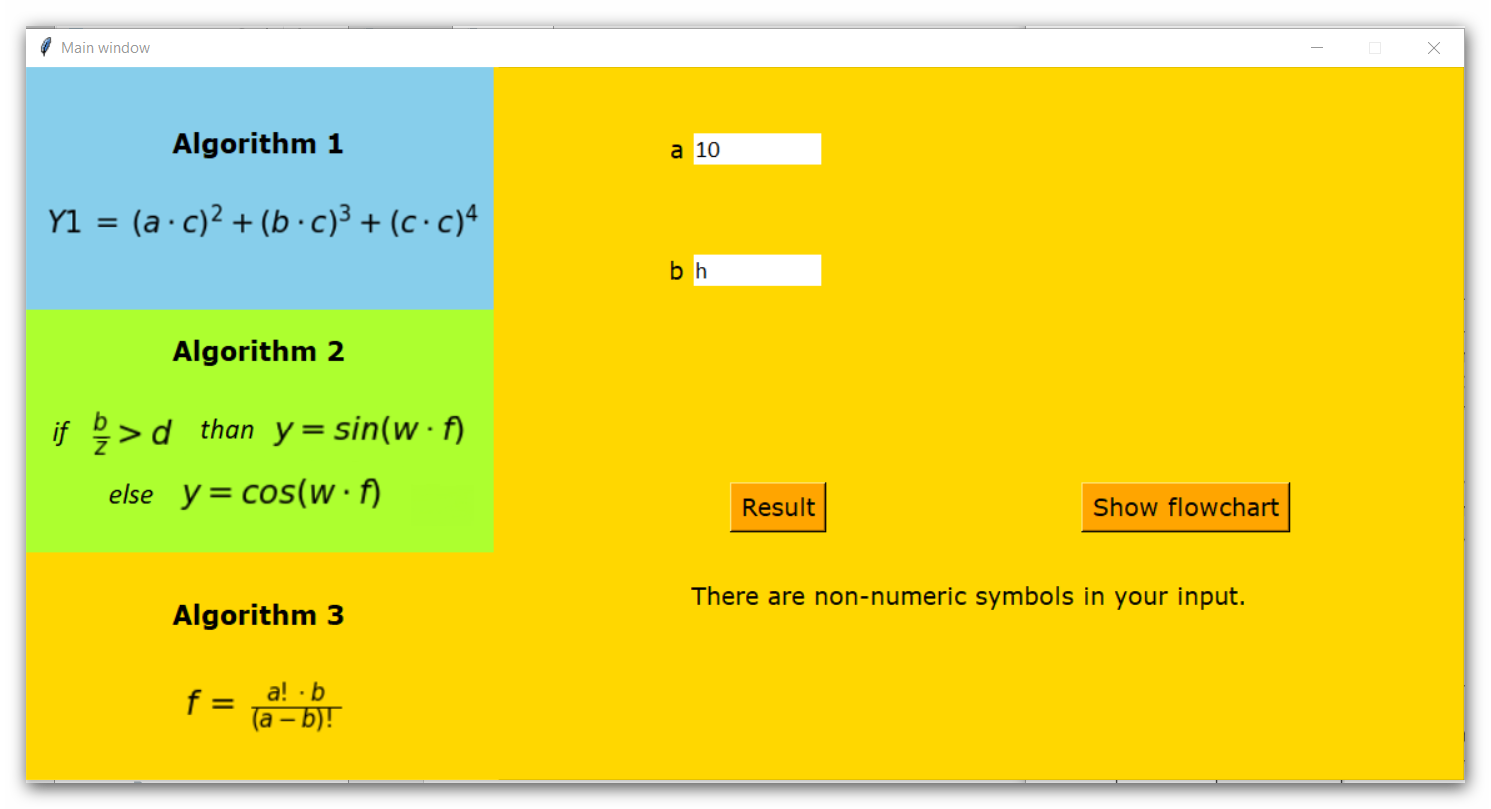
**Завдання 2**

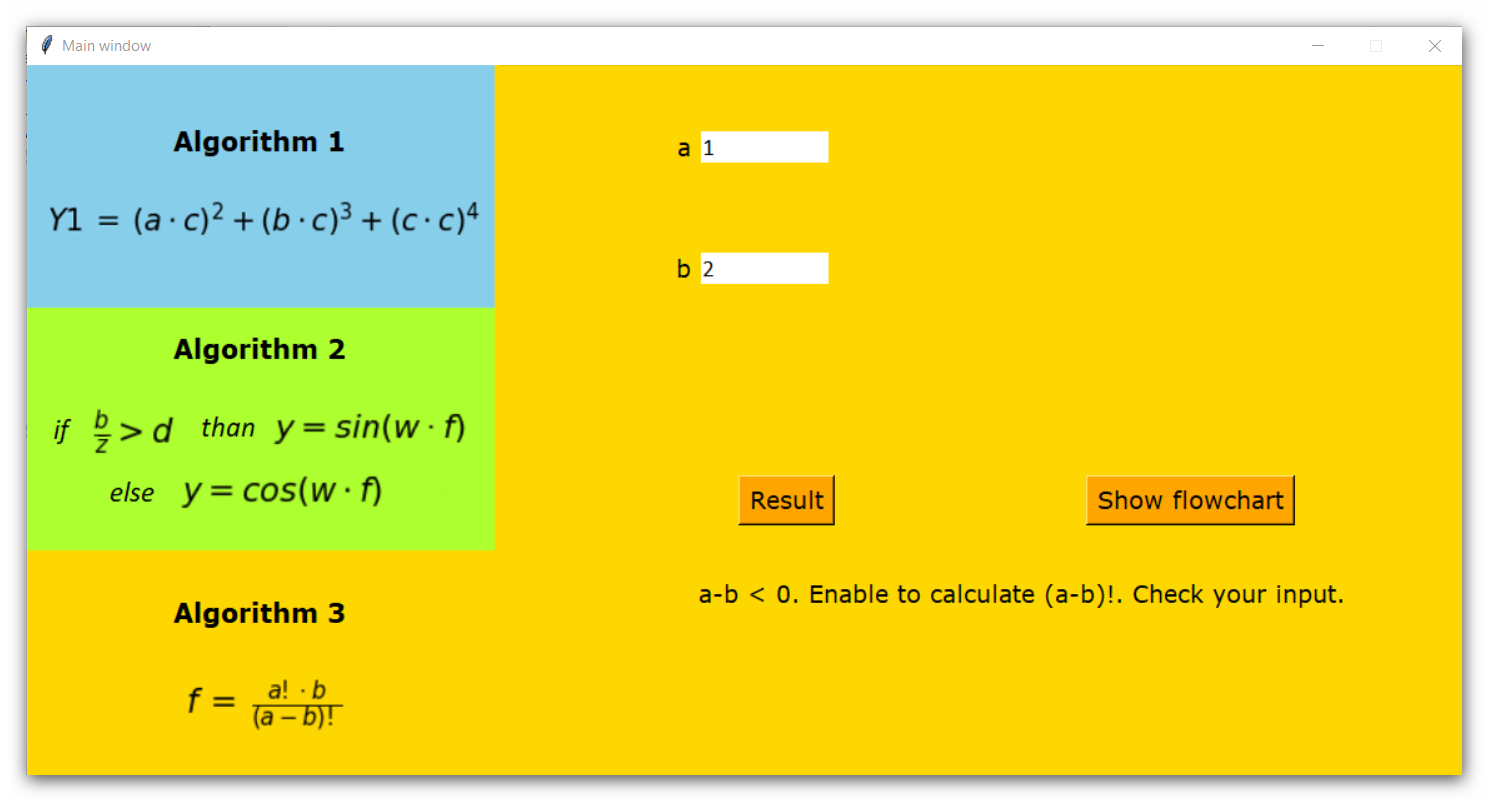
****

****

**Завдання 3**

****

****

****

**Висновки:** У ході виконання лабораторної роботи я закріпила знання з базових понять алгоритмів, вивчила основні правила складання блок-схем алгоритмів і покращила навички роботи з MS Visio. Отримані результати виконання програми є вірними.