

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 1

з дисципліни «Алгоритми та методи обчислень»

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

ВИКОНАЛА:
студентка 2 курсу
групи ІВ-92
Бабенко В.В.
Залікова - 9201

ПЕРЕВІРИВ:
Доцент кафедри ОТ
Порєв В.М.

Київ - 2021

Хід роботи

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

Завдання:

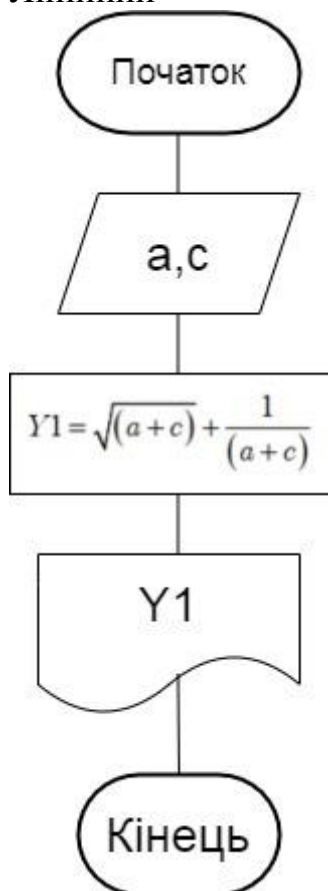
Відповідно до варіанту завдання розробити блок-схеми обчислення виразів для лінійного алгоритму, алгоритму, що розгалужується та циклічного алгоритму. У відповідності до блок-схеми створити програму обчислення виразу алгоритмічною мовою, узгодженою з викладачем.

Варіанти завдання

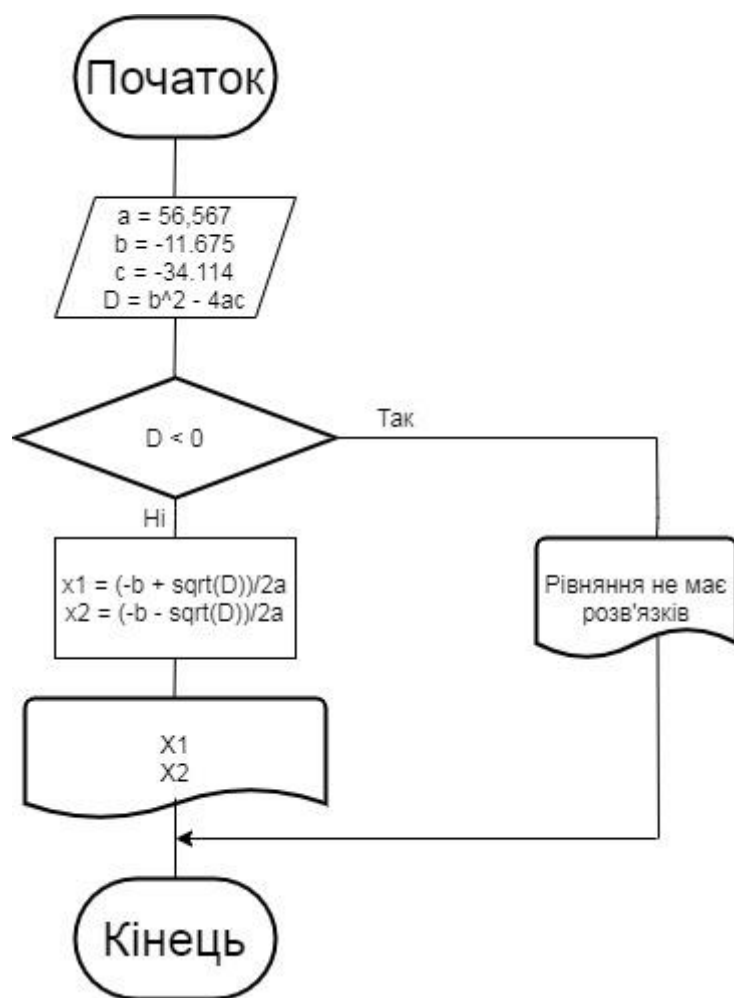
| | Лінійний | Що розгалужується | Циклічний |
|---|---------------------------------------|---|--|
| 1 | $Y1 = \sqrt{(a+c)} + \frac{1}{(a+c)}$ | Знайти корені рівняння: $57.567x^2 - 11.675x - 34.114 = 0$ | Обчислити: $f = \sum_{i=0}^{10} (a_i^2 + 56 \cdot c_i \cdot f \cdot g_i)$ |

Блок-схеми алгоритмів

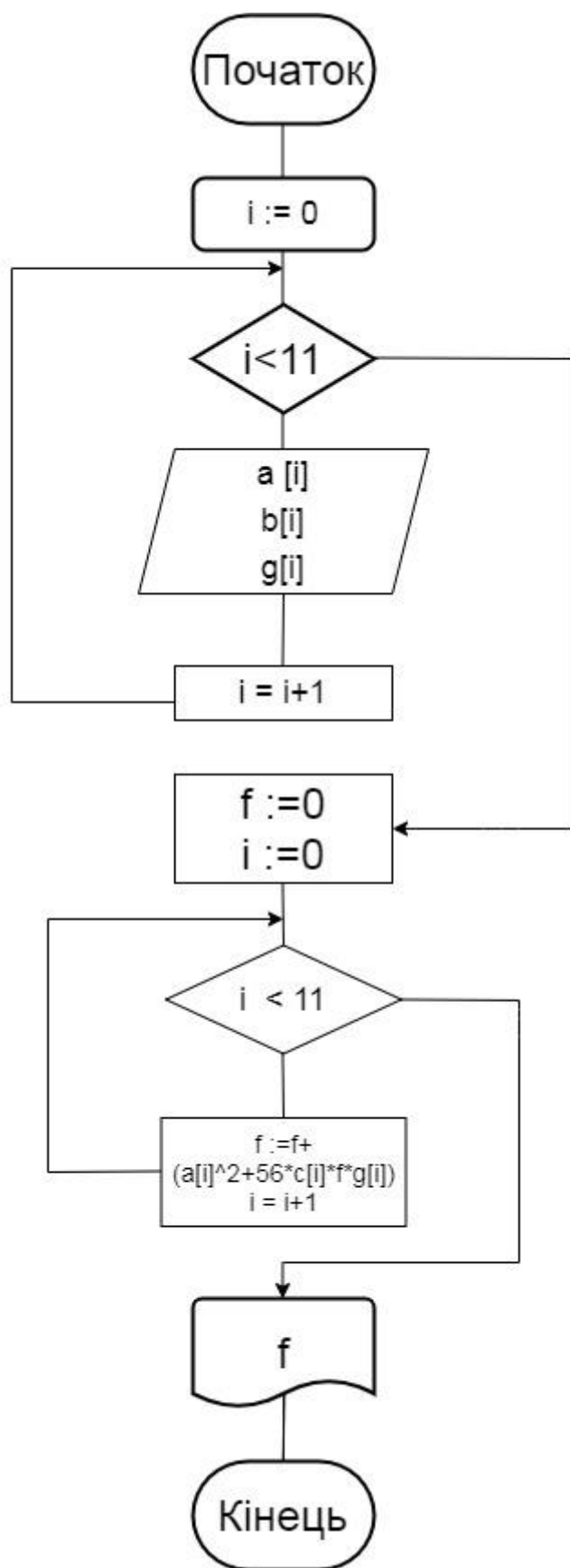
Лінійний



Розгалужений



Циклічний



Роздруківка тексту програми

Алгоритми

```
import math

def task1(a, c):
    y1 = math.sqrt(a + c) + 1 / (a + c)
    return y1

def task2():
    a = 57.567
    b = -11.675
    c = -34.114
    d = b * b - 4 * a * c
    x1 = (-b + math.sqrt(d)) / (2 * a)
    x2 = (-b - math.sqrt(d)) / (2 * a)
    return x1, x2

def task3(list_a, list_c, list_g, f):
    res = 0
    for i in range(10):
        solution = (list_a[i] ** 2 + 56 * list_c[i] * f * list_g[i])
        res += solution
    return res
```

Графічний інтерфейс

```
import subprocess
from tkinter import *
from tkinter import messagebox
import math
import file1 as f1
import file3 as f3

root = Tk()
root.title("АМО_Лабораторна робота 1")
root.geometry('550x250')
root["bg"] = "yellow"

def linearAlgorithm():
    win1 = Toplevel(root)
    win1["bg"] = "yellow"
    global task1_path

    def result1():
        try:
            a = float(ent_a.get())
            c = float(ent_c.get())

        except:
            messagebox.showinfo("Error", "Перевірте коректність введених даних")

        try:
            y1_result = math.sqrt(a + c) + 1 / (a + c)
            y1["text"] = "\n\n Y1 = {}".format(y1_result)
        except:
            messagebox.showinfo("Error", "Обчислення неможливі!!!")

    def openFileForEditing():
        subprocess.call(["notepad.exe", "file1.py"])
```

```

def result1_2():
    try:
        a = f1.a
        c = f1.c
    except:
        messagebox.showinfo("Помилка", "Дані введені некоректно або не
введені!")
    try:
        y1_answer = math.sqrt(a + c) + 1 / (a + c)
        y1["text"] = "\n\nY1 = {}".format(y1_answer)
    except:
        messagebox.showinfo("Помилка", "Обрахунки неможливі!")

nameOfTask = Label(win1, text="Лінійний алгоритм")
nameOfTask.config(bg="aqua")

ent_a = Entry(win1)
ent_c = Entry(win1)
ent_a.config(bg="skyblue")
ent_c.config(bg="skyblue")

write_a = Label(win1, text="Значення а")
write_a.config(bg="skyblue")
write_c = Label(win1, text="Значення с")
write_c.config(bg="skyblue")
label_for_task_name = Label(win1, text="Формула (умова завдання)")
label_for_task_name.config(bg="blue")
image_for_task = Label(win1, image=task1_path)

res_button_1 = Button(win1, text="Розрахувати результат\n Зчитування даних",
activebackground="skyblue",
command=result1, activeforeground="blue")
edit_file_1 = Button(win1, text="Внести значення \n у файл",
activebackground="skyblue",
command=openFileForEditing, activeforeground="blue")
res_button_2 = Button(win1, text="Розрахувати результат\n Зчитування даних\n
з файлу", command=result1_2,
activebackground="skyblue", activeforeground="blue")

y1 = Label(win1, text="\nY1 = \n")
y1.config(bg="aqua")
nameOfTask.grid(column=0, row=0)
label_for_task_name.grid(column=0, row=1)
image_for_task.grid(column=0, row=2)
write_a.grid(column=0, row=4)
ent_a.grid(column=1, row=4)
write_c.grid(column=0, row=5)
ent_c.grid(column=1, row=5)
res_button_1.grid(column=1, row=6)
edit_file_1.grid(column=0, row=7)
res_button_2.grid(column=1, row=7)
y1.grid(column=0, row=8)

def branched_algoritm():
    win2 = Toplevel(root)
    win2.config(bg="blue")
    global task2_path
    def equation():
        a = 57.567
        b = -11.675
        c = -34.114
        d = b * b - 4 * a * c
        x1_res = (-b + math.sqrt(d)) / (2 * a)
        x2_res = (-b - math.sqrt(d)) / (2 * a)

```

```

x1["text"] = "\n\nX1 = {}".format(x1_res)
x2["text"] = "\n\nX2 = {}".format(x2_res)

task_2 = Label(win2, text="Завдання 2\n Розгалужений алгоритм")
task_2.grid(column=0, row=0)
task_2.config(bg="yellow")
name_task_2 = Label(win2, text="Квадратне рівняння")
name_task_2.grid(column=0, row=2)
name_task_2.config(bg="orange")
task_2_image = Label(win2, image=task2_path)
task_2_image.grid(column=0, row=3)
button_for_equation = Button(win2, text="Розв'язати рівняння",
command=equation)
button_for_equation.config(activeforeground="skyblue")
button_for_equation.config(bg="yellow")
button_for_equation.grid(column=0, row=4)
x1 = Label(win2, text="\n\nX1 = \n")
x2 = Label(win2, text="\n\nX2 = \n")
x1.config(bg="orange")
x2.config(bg="yellow")
x1.grid(column=0, row=5)
x2.grid(column=0, row=6)

def cyclic_aloritm():
    global task3_path
    win3 = Toplevel(root)
    win3.config(bg="blue")
    def edit_file3():
        subprocess.call(["notepad.exe", "file3.py"])

    def result_of_3_1():
        try:
            list_a = (entry_a.get()).split()
            list_c = (entry_c.get()).split()
            list_g = (entry_g.get()).split()
            value_f = float(entry_f.get())
            for i in range(len(list_a)):
                list_a[i] = float(list_a[i])
                list_c[i] = float(list_c[i])
                list_g[i] = float(list_g[i])
        except:
            messagebox.showinfo("Error", "Введено некоректні значення!")

        try:
            if not list_g or not list_c or not list_a or value_f == '':
                messagebox.showinfo("Error", "Введіть, будь ласка, числові
значення")
            if len(list_a) != 10 or len(list_c) != 10 or len(list_g) != 10:
                messagebox.showinfo("Error", "Введіть, будь ласка, числові
значення")
            res = 0
            for i in range(10):
                solution = (list_a[i] ** 2 + 56 * list_c[i] * value_f *
list_g[i])
                res += solution
            res_3["text"] = "\n\nf = {}".format(res)
        except:
            messagebox.showinfo("Error", "Обчислення не можливі!!!")

    def result_of_3_2():
        try:

```

```

list_a = f3.a
list_g = f3.g
list_c = f3.c
value_f = f3.f
# value_f = float(value_f)
for i in range(len(list_a)):
    list_a[i] = float(list_a[i])
    list_c[i] = float(list_c[i])
    list_g[i] = float(list_g[i])

except:
    messagebox.showinfo("Error", "Введено некоректні значення!")

try:
    if not list_g or not list_c or not list_a or value_f == '':
        messagebox.showinfo("Error", "Введіть, будь ласка, числові
значення")
    if len(list_a) != 10 or len(list_c) != 10 or len(list_g) != 10:
        messagebox.showinfo("Error", "Введіть, будь ласка, числові
значення")
    res = 0
    for i in range(10):
        solution = (list_a[i] ** 2 + 56 * list_c[i] * value_f *
list_g[i])
        res += solution
    res_3["text"] = "\n\nf = {}".format(res)
except:
    messagebox.showinfo("Error", "Обчислення не можливі!!!")

task3 = Label(win3, text="Завдання 3\n Циклічний алгоритм")
task3.config(bg="yellow")
task3.grid(column=0, row=1)
task3_image = Label(win3, image=task3_path)
task3_image.grid(column=0, row=2)

label_a = Label(win3, text="Введіть 10 значень a(a1, a2...a10)")
label_a.config(bg="orange")
label_c = Label(win3, text="Введіть 10 значень c(c1, c2...c10)")
label_c.config(bg="orange")
label_g = Label(win3, text="Введіть 10 значень g(g1, g2...g10)")
label_g.config(bg="orange")
label_f = Label(win3, text="Введіть значення f")
label_f.config(bg="orange")

entry_a = Entry(win3)
entry_a.config(bg="yellow")
entry_c = Entry(win3)
entry_c.config(bg="yellow")
entry_g = Entry(win3)
entry_g.config(bg="yellow")
entry_f = Entry(win3)
entry_f.config(bg="yellow")

label_a.grid(column=0, row=3)
label_c.grid(column=0, row=5)
label_g.grid(column=0, row=7)
label_f.grid(column=0, row=9)

entry_a.grid(column=0, row=4)
entry_c.grid(column=0, row=6)
entry_g.grid(column=0, row=8)
entry_f.grid(column=0, row=10)

button_1 = Button(win3, text="Зчитати данні\n Результат",

```



```

command=result_of_3_1)
    button_1.config(bg="aqua")
    button_1.config(activeforeground="RoyalBlue")

    button_2 = Button(win3, text="Редагувати файл", command=edit_file3)
    button_2.config(activeforeground="RoyalBlue")
    button_2.config(bg="aqua")

    button_3 = Button(win3, text="Зчитати данні з файлу\n Результат",
command=result_of_3_2)
    button_3.config(activeforeground="royalblue")
    button_3.config(bg="aqua")
    button_1.grid(column=0, row=11)
    button_2.grid(column=0, row=12)
    button_3.grid(column=1, row=12)

    res_3 = Label(win3, text="\nf = \n")
    res_3.config(bg="yellow")
    res_3.grid(column=0, row=13)

task1_path = PhotoImage(file="first_task.png")
task2_path = PhotoImage(file="second_task.png")
task3_path = PhotoImage(file="third_task.png")

mainMenu = Menu(root)
root.config(menu=mainMenu)

l1 = Menu(mainMenu, tearoff=0)
l1.add_command(label="Лінійний алгоритм", command=linearAlgorithm)
l1.add_command(label="Розгалужений алгоритм", command=branched_algorithm)
l1.add_command(label="Циклічний алгоритм", command=cyclic_algorithm)

mainMenu.add_cascade(label="Меню виконаних завдань", menu=l1)
someAboutUsing = Label(root, text="Доброго дня!\n Для того, щоб переглянути
роботу\n використовуйте меню", font=('Arial', 25))
someAboutUsing.config(bg="blue")
dataAboutStudent = Label(root, text="Данні про студента:")
dataAboutStudent.config(bg="skyblue")
name = Label(root, text="Бабенко Вікторія Валентинівна")
name.config(bg="orange")
group = Label(root, text="ІВ-92, фІОТ")
group.config(bg="orange")
variant = Label(root, text="Варіант - 1")
variant.config(bg="orange")
someAboutUsing.pack()
dataAboutStudent.pack()
name.pack()
group.pack()
variant.pack()

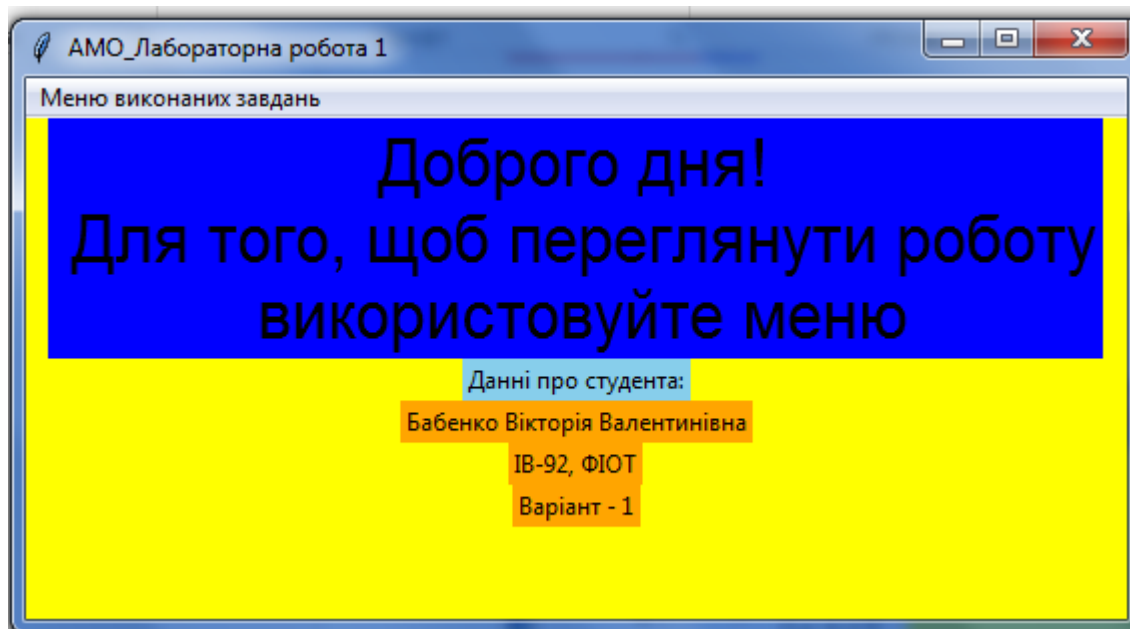
root.mainloop()

```

ПРИМІТКА

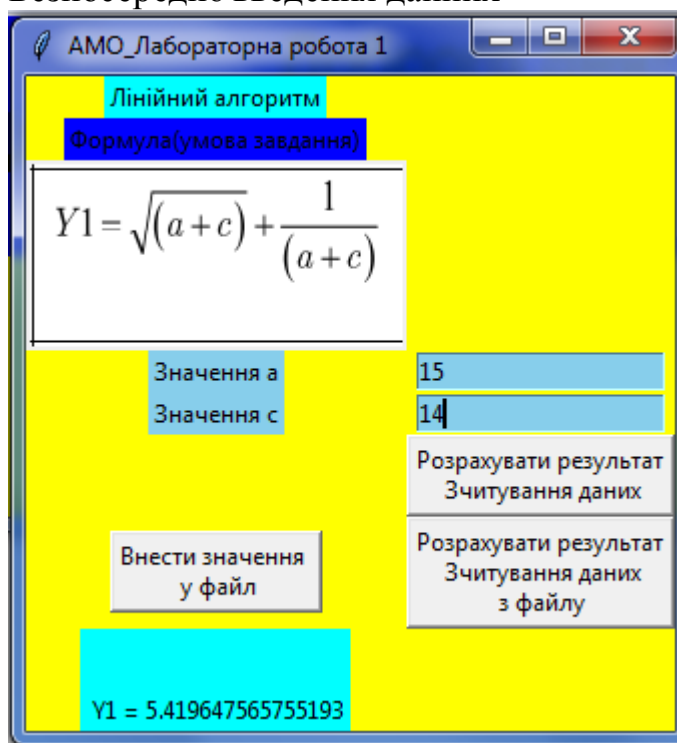
ПРИ ВВЕДЕННІ ДАНИХ В ФАЙЛ СПОЧАТКУ ЗБЕРЕЖІТЬ ЙОГО, А ПОТІМ ПЕРЕЗАПУСТІТЬ ПРОГРАМУ ТА НАТИСНІТЬ ВІДПОВІДНУ КНОПКУ

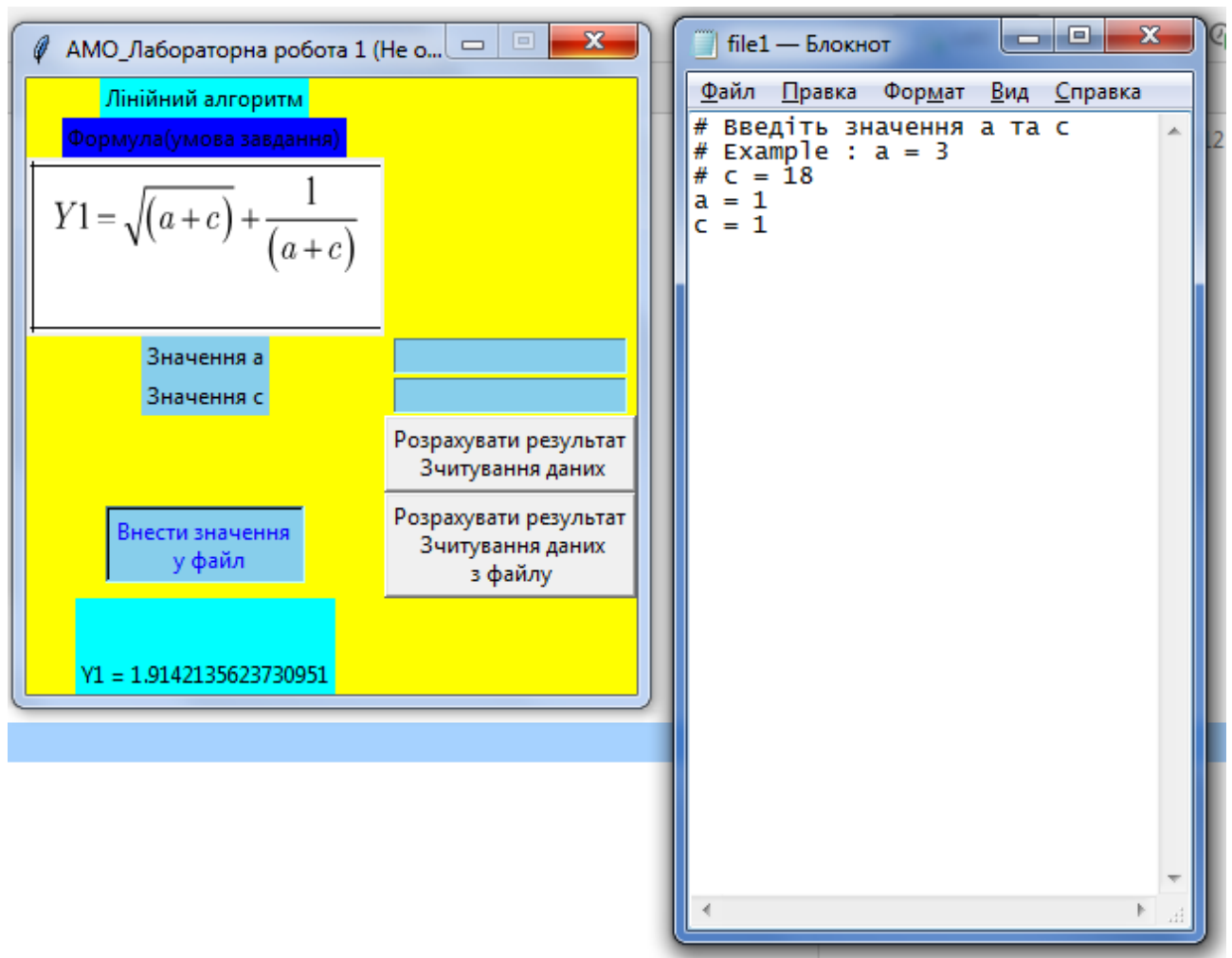
Роздруківка результатів виконання програми



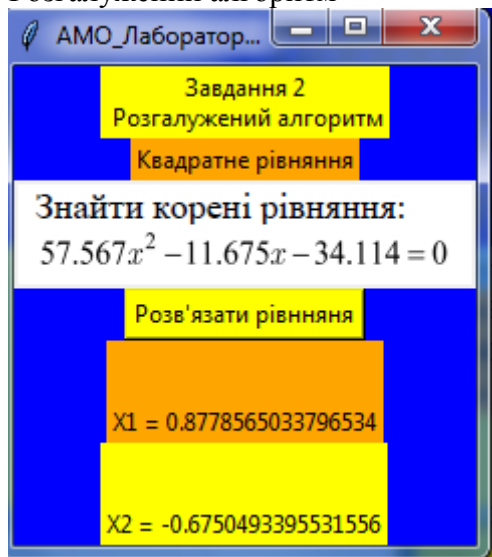
Лінійний алгоритм

Безпосереднє введення даних

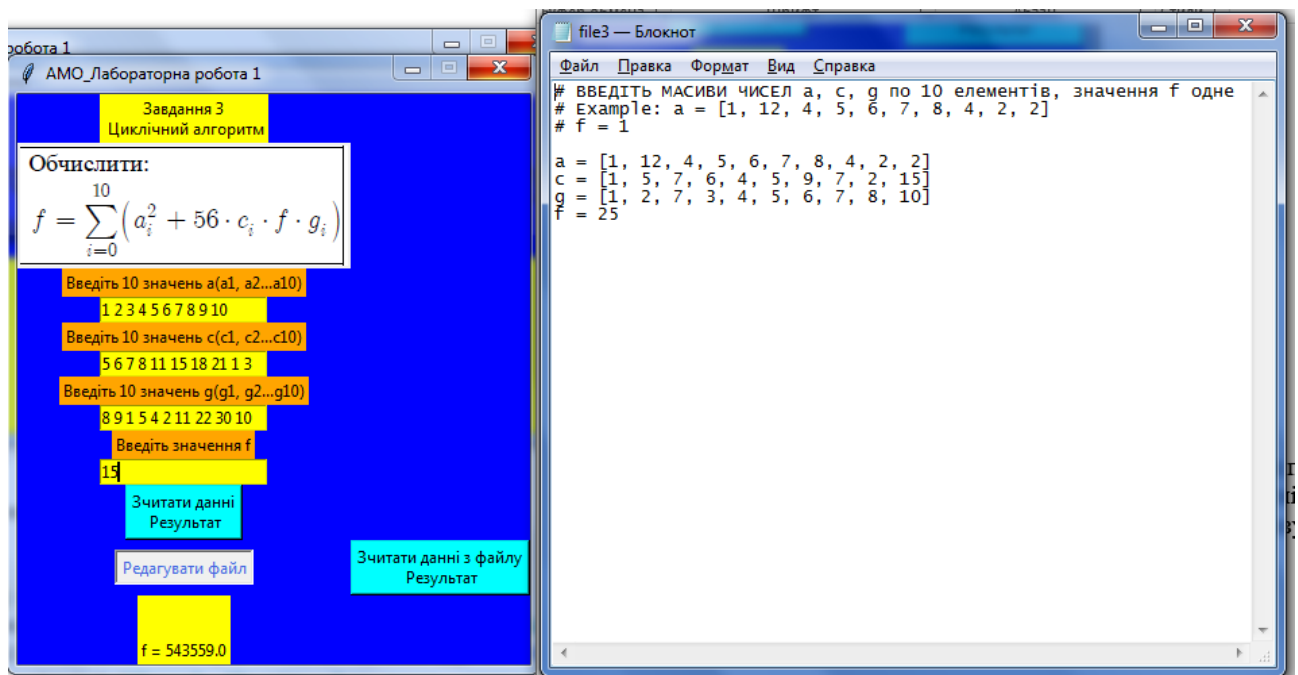
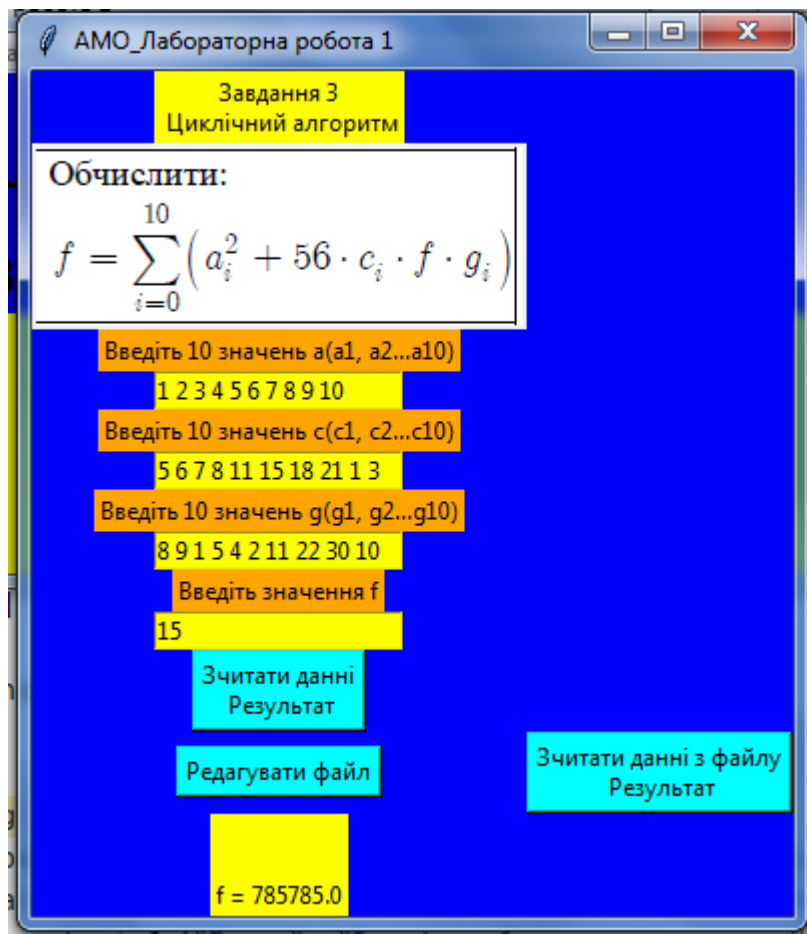




Розгалужений алгоритм



Циклічний алгоритм



Висновки

У ході виконання лабораторної роботи закріплено знання з базових понять алгоритмів, вивчено основні правила складання блок-схем алгоритмів та покращено навички роботи із онлайн сервісом draw.io. Отримані результати виконання програми є правильними.