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»**

Виконав:

студент групи ІВ-82

Бережнюк А. І.

Перевірив:

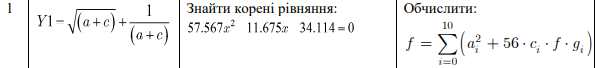
ст.вик. Порєв В. М.

Київ

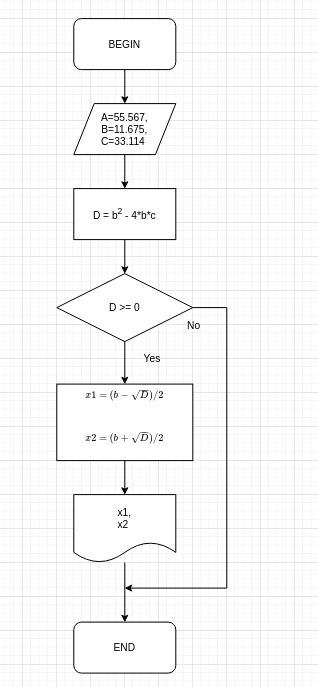
2020 р.

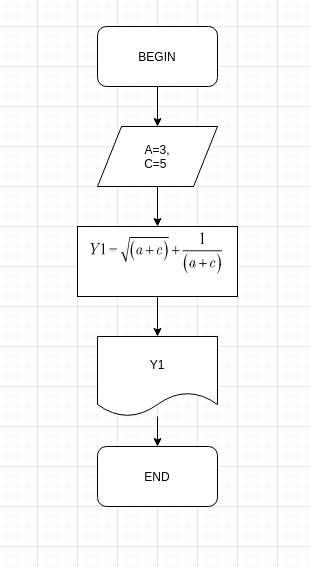
**Мета:** Навчитися створювати блок-схеми лінійного алгоритму; розгалуженого алгоритму та циклічного алгоритму за допомогою редактора блок-схем *afce*або іншого довільного редактора.

**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.

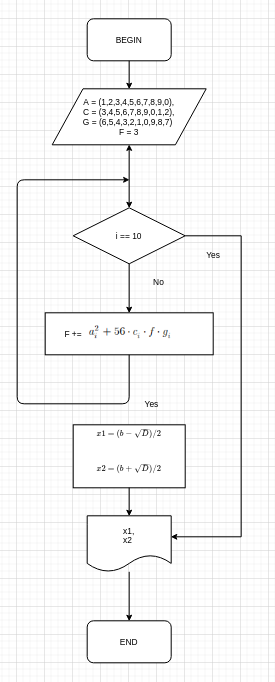


Лінійний алгоритм: Алгоритм з розгалудженням:





Циклічний алгоритм:



**Роздруківка тексту програми:**

import tkinter as tk

from PIL import Image, ImageTk

from math import sqrt

class Main\_Window:

def \_\_init\_\_(self, root):

self.root = root

self.root.title('Menu')

self.frame = tk.Frame(root)

self.butnew('Algorithm 1', Algorithm\_1)

self.butnew('Algorithm 2', Algorithm\_2)

self.butnew('Algorithm 3', Algorithm\_3)

self.frame.pack()

def butnew(self, text, \_class,):

tk.Button(self.frame, text=text, height=6, width=30,

command=lambda: self.new\_window(\_class)).pack()

def new\_window(self, \_class):

self.new = tk.Toplevel(self.root)

self.app = \_class(self.new)

class Algorithm\_1:

def \_\_init\_\_(self, master):

self.master = master

self.master.title('Algorithm\_1')

self.master.geometry('600x600')

self.frame\_left = tk.Frame(self.master)

self.frame\_left.pack(side=tk.LEFT)

# Label

self.label = tk.Label(self.frame\_left,

text='Expression')

self.label.grid(row=0, columnspan=2)

# Variant

self.img = ImageTk.PhotoImage(Image.open('pics/task1.png'))

self.panel = tk.Label(self.frame\_left)

self.panel.image = self.img

self.panel.configure(image=self.img)

self.panel.grid(row=1, columnspan=2)

# Label A

self.label = tk.Label(self.frame\_left,

text='Enter A')

self.label.grid(row=2, column=0, padx=10, pady=20)

# Entry A

self.input\_a = tk.StringVar()

self.input\_field\_a = tk.Entry(self.frame\_left,

textvariable=self.input\_a)

self.input\_field\_a.grid(row=2, column=1, pady=20)

# Label C

self.label = tk.Label(self.frame\_left,

text='Enter B')

self.label.grid(row=3, column=0, padx=10, pady=20)

# Entry C

self.input\_c = tk.StringVar()

self.input\_field\_c = tk.Entry(self.frame\_left,

textvariable=self.input\_c)

self.input\_field\_c.grid(row=3, column=1, pady=20)

# Button Result

self.button = tk.Button(self.frame\_left, text='Result',

command=lambda: self.return\_result

(self.input\_a.get(), self.input\_c.get()))

self.button.grid(row=4, column=0, padx=10, pady=20)

# Result

self.result = tk.Label(self.frame\_left, text=" ")

self.result.grid(row=4, column=1, padx=10, pady=20)

# Button Default

self.button = tk.Button(self.frame\_left, text='Default',

command=lambda: self.set\_default())

self.button.grid(row=5, column=0, padx=10, pady=20)

# Right Frame

self.frame\_right = tk.Frame(self.master)

self.frame\_right.pack(side=tk.RIGHT)

# Label

self.label = tk.Label(self.frame\_right,

text='Chart')

self.label.pack()

# Variant

self.img = ImageTk.PhotoImage(Image.open('pics/chart1.png'))

self.panel = tk.Label(self.frame\_right, image=self.img)

self.panel.pack()

def return\_result(self, a, c):

try:

self.result['text'] = self.get\_answer(a, c)

except:

self.result['text'] = "Enter correct values"

def get\_answer(self, a, c):

a = float(a)

c = float(c)

return sqrt(a+c) + 1/(a+c)

def set\_default(self):

self.input\_field\_a.insert(tk.END, '3')

self.input\_field\_c.insert(tk.END, '5')

class Algorithm\_2:

def \_\_init\_\_(self, master):

self.master = master

self.master.title('Algorithm\_2')

self.master.geometry('700x700')

self.frame\_left = tk.Frame(self.master)

self.frame\_left.pack(side=tk.LEFT)

# Label Expression

self.label = tk.Label(self.frame\_left,

text='Expression')

self.label.grid(row=0, columnspan=2)

# Variant

self.img = ImageTk.PhotoImage(Image.open('pics/task2.png'))

self.panel = tk.Label(self.frame\_left)

self.panel.image = self.img

self.panel.configure(image=self.img)

self.panel.grid(row=1, columnspan=2)

# Label A

self.label = tk.Label(self.frame\_left,

text='Enter A')

self.label.grid(row=2, column=0, padx=10, pady=20)

# Entry A

self.input\_a = tk.StringVar()

self.input\_field\_a = tk.Entry(self.frame\_left,

textvariable=self.input\_a)

self.input\_field\_a.grid(row=2, column=1, pady=20)

# Label B

self.label = tk.Label(self.frame\_left,

text='Enter B')

self.label.grid(row=3, column=0, padx=10, pady=20)

# Entry B

self.input\_b = tk.StringVar()

self.input\_field\_b = tk.Entry(self.frame\_left,

textvariable=self.input\_b)

self.input\_field\_b.grid(row=3, column=1, pady=20)

# Label C

self.label = tk.Label(self.frame\_left,

text='Enter C')

self.label.grid(row=4, column=0, padx=10, pady=20)

# Entry C

self.input\_c = tk.StringVar()

self.input\_field\_c = tk.Entry(self.frame\_left,

textvariable=self.input\_c)

self.input\_field\_c.grid(row=4, column=1, pady=20)

# Label Result

self.button = tk.Button(self.frame\_left, text='Result',

command=lambda: self.return\_result

(self.input\_a.get(), self.input\_b.get(),

self.input\_c.get()))

self.button.grid(row=5, column=0, padx=10, pady=20)

# Result

self.result = tk.Label(self.frame\_left, text=" ")

self.result.grid(row=5, column=1, padx=10, pady=20)

# Right Frame

self.frame\_right = tk.Frame(self.master)

self.frame\_right.pack(side=tk.RIGHT)

# Label Chart

self.label = tk.Label(self.frame\_right,

text='Chart')

self.label.pack()

# Variant

self.img = ImageTk.PhotoImage(Image.open('pics/chart2.png'))

self.panel = tk.Label(self.frame\_right, image=self.img)

self.panel.pack()

# Button Default

self.button = tk.Button(self.frame\_left, text='Default',

command=lambda: self.set\_default())

self.button.grid(row=6, column=0, padx=10, pady=20)

def return\_result(self, a, b, c):

try:

self.result['text'] = self.get\_answer(a, b, c)

except:

self.result['text'] = "Enter correct values"

def get\_answer(self, a, b, c):

a = float(a)

b = float(b)

c = float(c)

D = b\*\*2 - 4\*a\*c

if D < 0:

return 'None'

if abs(D) < 0.0001:

return ('x = {0:.2f}'.format(b / 2))

x1 = (b - sqrt(D)) / 2

x2 = (b + sqrt(D)) / 2

return ('x1 = {}\nx2 = {}'.format(round(x1, 3), round(x2, 3)))

def set\_default(self):

self.input\_field\_a.insert(tk.END, '57.567')

self.input\_field\_b.insert(tk.END, '11.675')

self.input\_field\_c.insert(tk.END, '34.114')

class Algorithm\_3:

def \_\_init\_\_(self, master):

self.master = master

self.master.title('Algorithm\_3')

self.master.geometry('900x750')

self.frame\_left = tk.Frame(self.master)

self.frame\_left.pack(side=tk.LEFT)

# Label Expression

self.label = tk.Label(self.frame\_left,

text='Expression')

self.label.grid(row=0, columnspan=2)

# Variant

self.img = ImageTk.PhotoImage(Image.open('pics/task3.png'))

self.panel = tk.Label(self.frame\_left)

self.panel.image = self.img

self.panel.configure(image=self.img)

self.panel.grid(row=1, columnspan=2)

# Label F

self.label = tk.Label(self.frame\_left,

text='Enter F')

self.label.grid(row=2, column=0, padx=10, pady=20)

# Entry F

self.input\_f = tk.StringVar()

self.input\_field\_f = tk.Entry(self.frame\_left,

textvariable=self.input\_f)

self.input\_field\_f.grid(row=2, column=1, pady=20)

# Label A

self.label = tk.Label(self.frame\_left,

text='Enter A (10 numbers saperated with \

spaces)')

self.label.grid(row=3, column=0, padx=10, pady=20)

# Entry A

self.input\_a = tk.StringVar()

self.input\_field\_a = tk.Entry(self.frame\_left,

textvariable=self.input\_a)

self.input\_field\_a.grid(row=3, column=1, pady=20)

# Label C

self.label = tk.Label(self.frame\_left,

text='Enter C (10 numbers saperated with \

spaces)')

self.label.grid(row=4, column=0, padx=10, pady=20)

# Entry C

self.input\_c = tk.StringVar()

self.input\_field\_c = tk.Entry(self.frame\_left,

textvariable=self.input\_c)

self.input\_field\_c.grid(row=4, column=1, pady=20)

# Label G

self.label = tk.Label(self.frame\_left,

text='Enter G (10 numbers saperated with \

spaces)')

self.label.grid(row=5, column=0, padx=10, pady=20)

# Entry G

self.input\_g = tk.StringVar()

self.input\_field\_g = tk.Entry(self.frame\_left,

textvariable=self.input\_g)

self.input\_field\_g.grid(row=5, column=1, pady=20)

# Label Result

self.button = tk.Button(self.frame\_left, text='Result',

command=lambda: self.return\_result

(self.input\_f.get(), self.input\_a.get(),

self.input\_c.get(), self.input\_g.get()))

self.button.grid(row=6, column=0, padx=10, pady=20)

# Result

self.result = tk.Label(self.frame\_left, text=" ")

self.result.grid(row=6, column=1, padx=10, pady=20)

# Right Frame

self.frame\_right = tk.Frame(self.master)

self.frame\_right.pack(side=tk.RIGHT)

# Label Chart

self.label = tk.Label(self.frame\_right,

text='Chart')

self.label.pack()

# Variant

self.img = ImageTk.PhotoImage(Image.open('pics/chart3.png'))

self.panel = tk.Label(self.frame\_right, image=self.img)

self.panel.pack()

# Button Default

self.button = tk.Button(self.frame\_left, text='Default',

command=lambda: self.set\_default())

self.button.grid(row=7, column=0, padx=10, pady=20)

def return\_result(self, f, a, c, g):

try:

self.result['text'] = self.get\_answer(f, a, c, g)

except:

self.result['text'] = "Enter correct values"

def get\_answer(self, f, a, c, g):

f = float(f)

a = list(map(float, a.split()))

c = list(map(float, c.split()))

g = list(map(float, g.split()))

for i in range(10):

f += a[i]\*\*2 + 56\*c[i]\*f\*g[i]

return 'F = {}'.format(round(f, 3))

def set\_default(self):

self.input\_field\_f.insert(tk.END, '3')

self.input\_field\_a.insert(tk.END, '1 2 3 4 5 6 7 8 9 0')

self.input\_field\_c.insert(tk.END, '3 4 5 6 7 8 9 0 1 2')

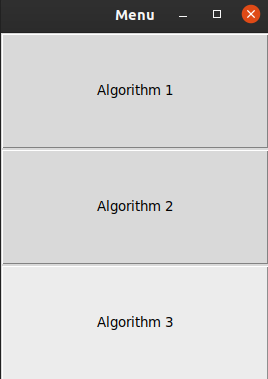
self.input\_field\_g.insert(tk.END, '6 5 4 3 2 1 0 9 8 7')

app = tk.Tk()

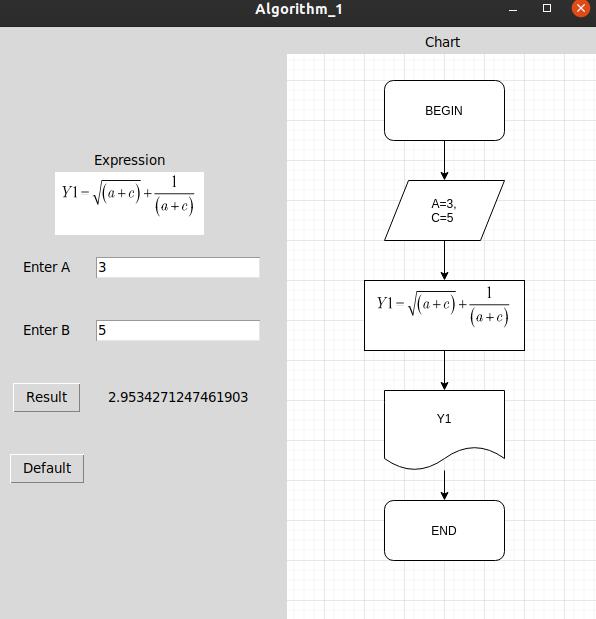
Main\_Window(app)

app.mainloop()

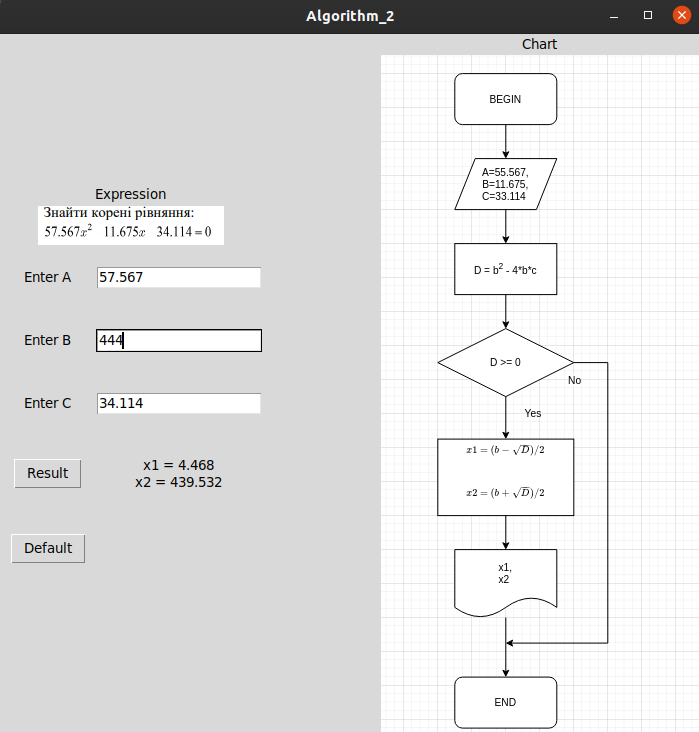
**Роздруківки результатів виконання програми**

****

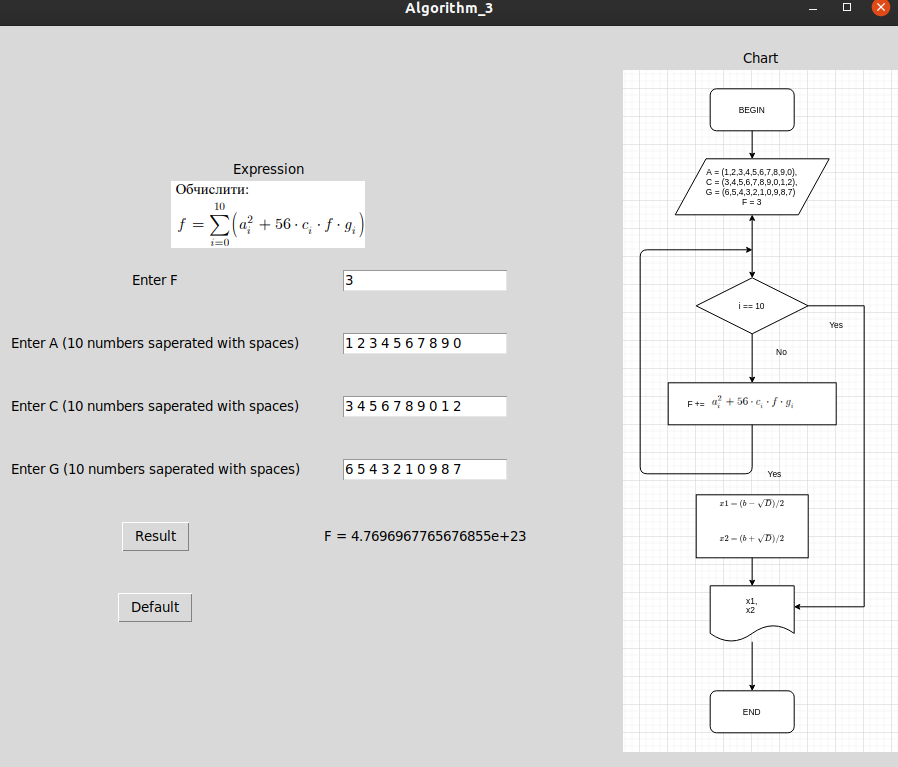
**Завдання 1**



**Завдання 2**



**Завдання 3**



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