Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

**ДОМАШНЯ РОБОТА №2.1**

**з дисципліни «Математичне моделювання та оптимізація процесів і систем»**

**тема «Чисельні методи наближення функцій. Інтерполяційний многочлен Лагранжа»**

**Варіант № \_13\_**

**Виконав:**

**Студент 3 курсу, групи *\_ТІ-01\_\_***

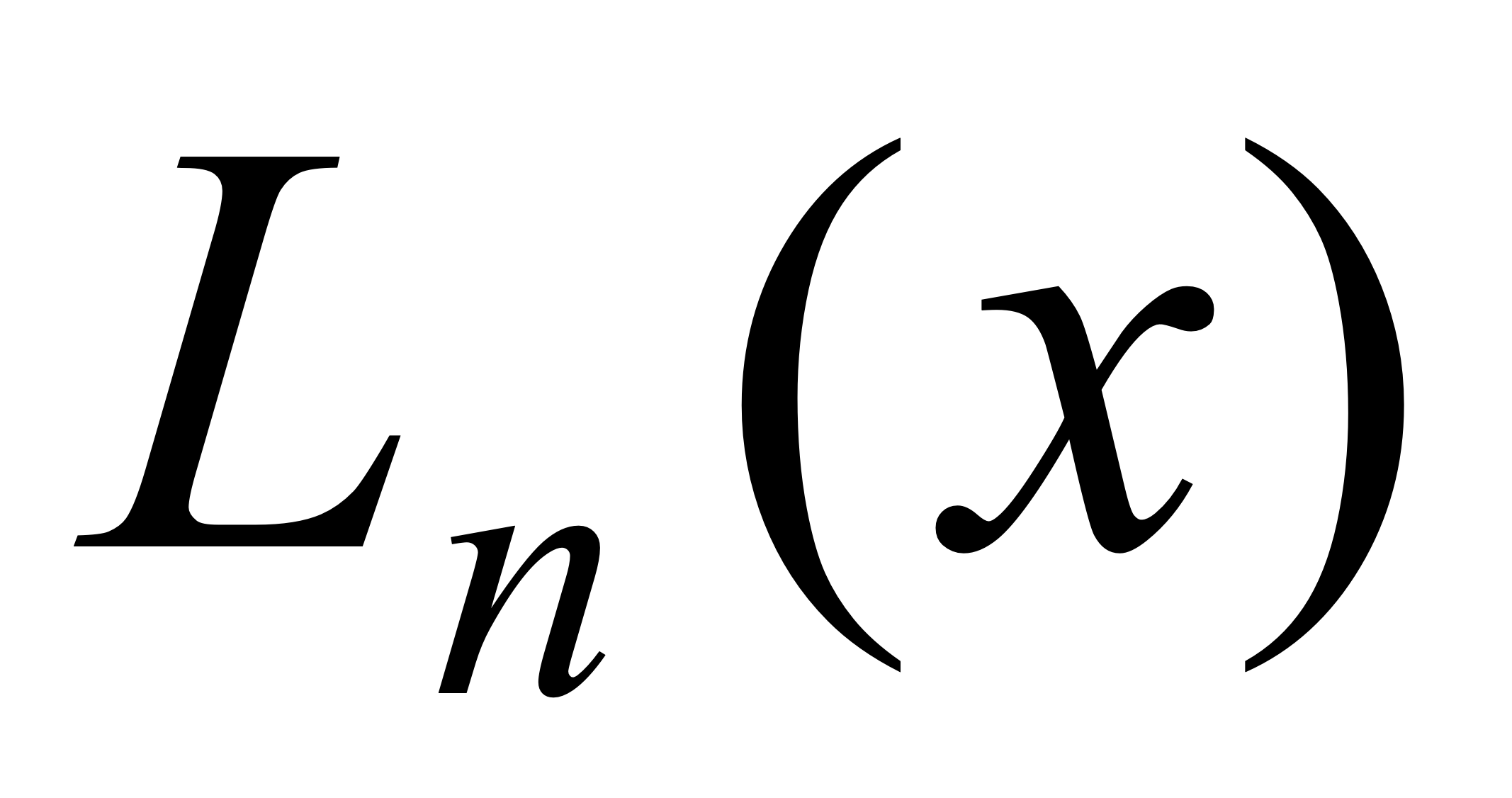
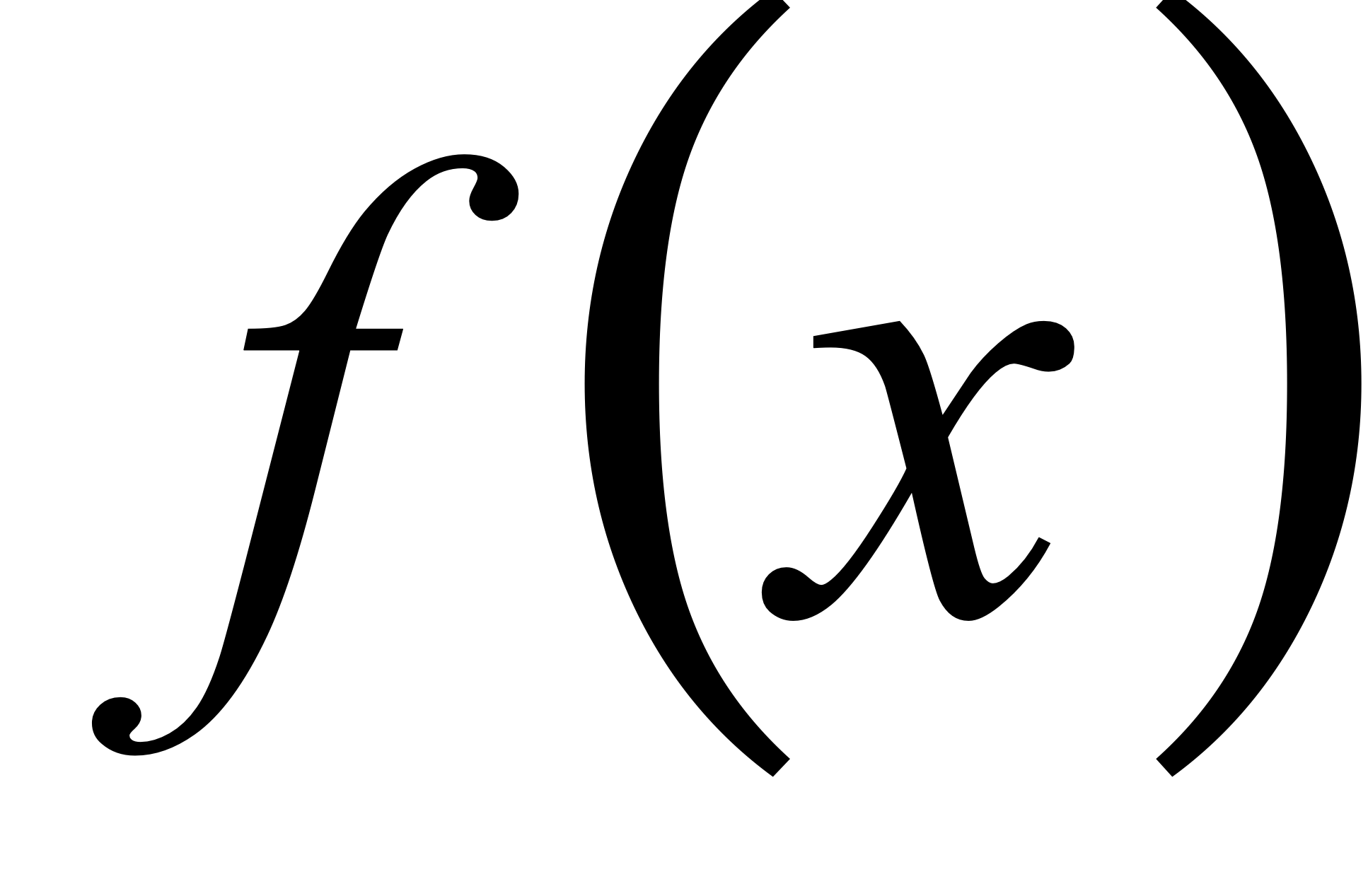
***\_\_\_Круть Катерини\_\_\_\_***

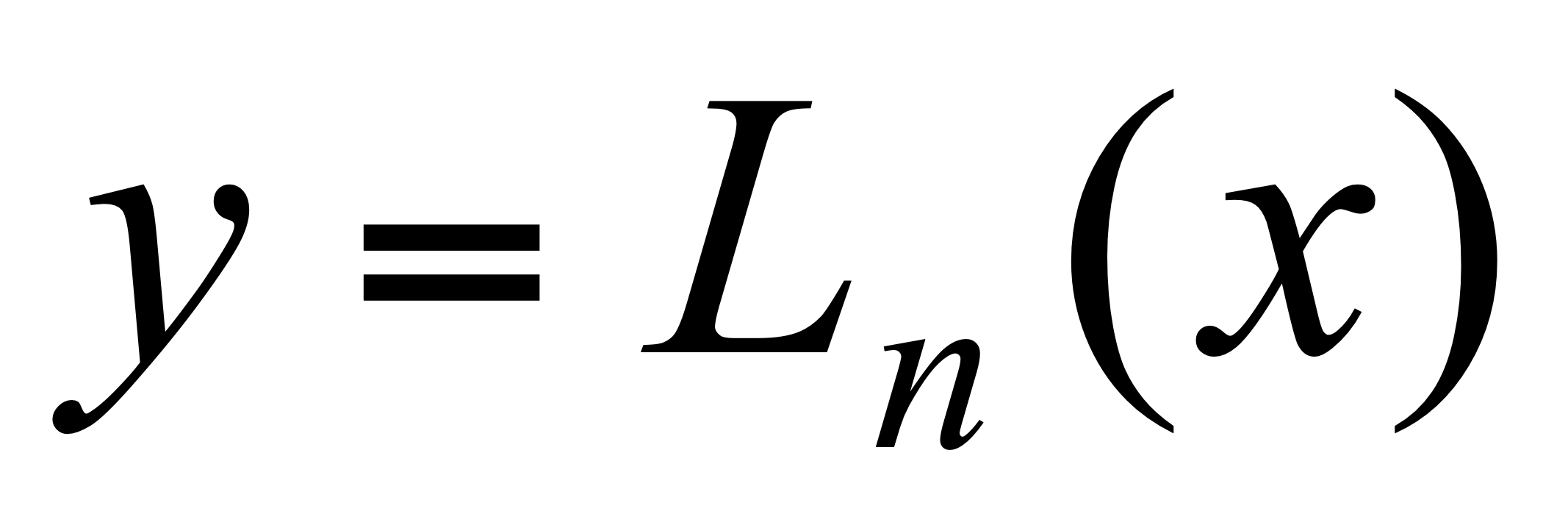
**(прізвище ім’я)**

**Дата здачі \_\_\_\_2023-03-09\_\_\_\_\_\_\_\_**

Київ – 2023

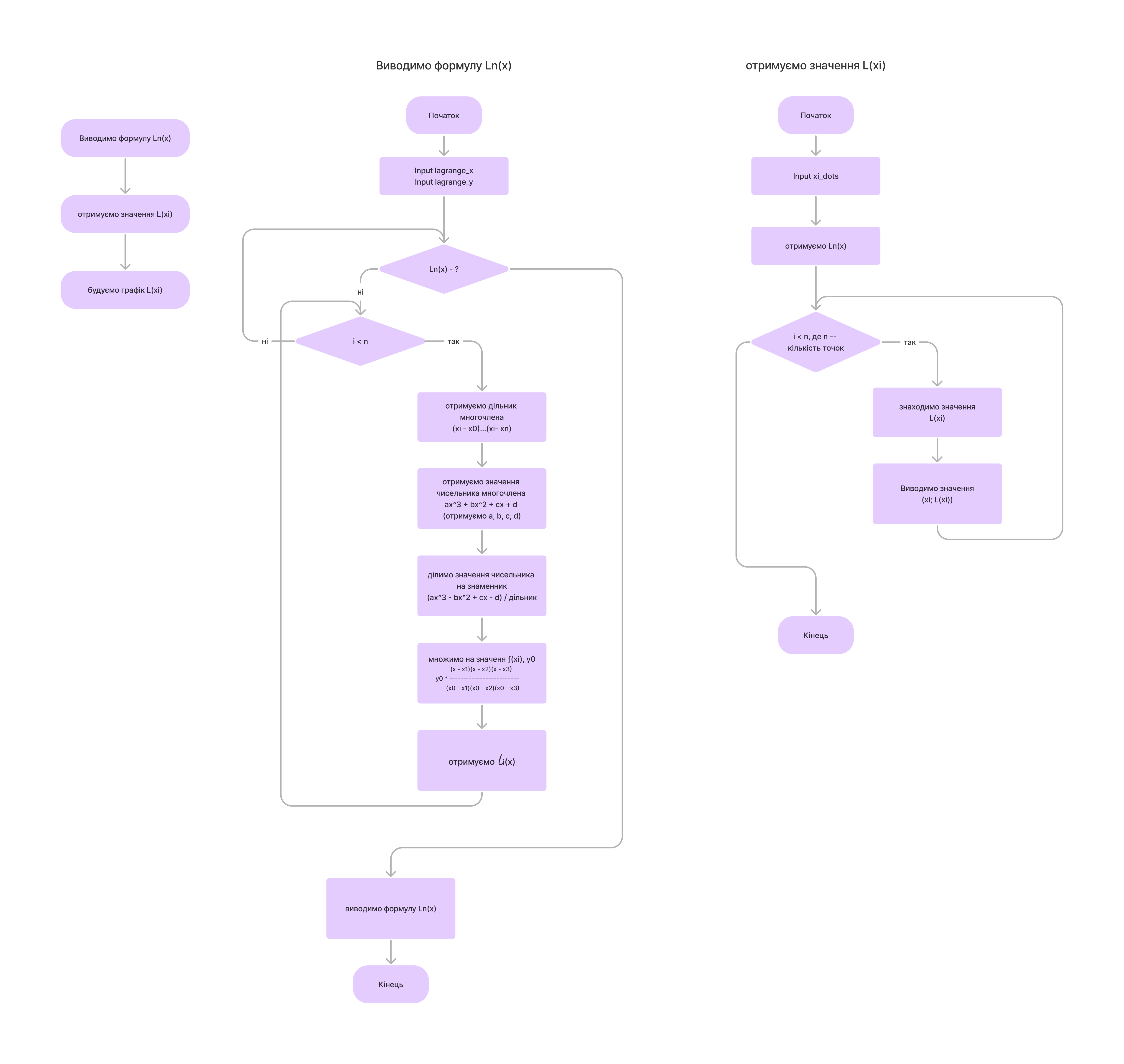
**Завдання:**

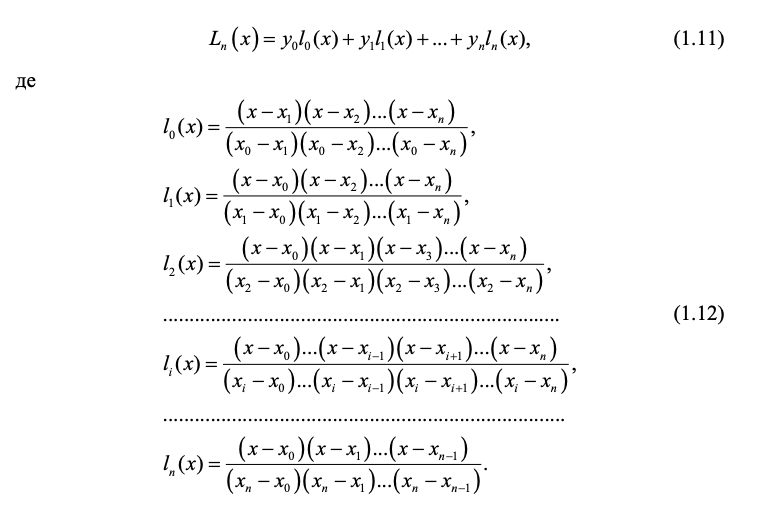
Побудувати інтерполяційний многочлен Лагранжа  для функції , що задана таблицею, та обчислити значення функції у заданих точках. Побуду

вати графік інтерполяційної функції  за наявним набором точок. Для виконання завдання розробити програму на одній з мов програмування.

**Дані:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Варіант** | **Задана таблиця значень функції** | | | | | **Задані точки** | | | | |
| **13** | Изображение | -4 | -3 | 0 | 1 | Изображение | -2 | -0,5 | 0,5 | 2 |
| Изображение | -7 | 10 | -11 | -22 | Изображение | ? | ? | ? | ? |

**Блок-схема:**

**Використані формули:**

**Код:**

import copy

import itertools

import matplotlib.pyplot as plt

import numpy as np

def get\_coeff\_denominator(copy\_list, removed\_element):

*"""* Отримуємодільникдляінтерполяційногомногочлена*"""*

denominator = 1

for j in range(len(copy\_list)):

denominator \*= removed\_element - copy\_list[j]

return denominator

def get\_numerator\_elements(combinations):

*"""*

***:param*** *combinations:* всікомбінаціїдляперемноження

***:return****:* елементичисельника

*"""*

summ = 0

for combination in combinations:

mult = 1

for elem in combination:

mult \*= -1 \* elem

summ += mult

return summ

def coefficient(list\_x, list\_y):

*"""*

***:param*** *list\_x:* значееняряду *xi*

***:param*** *list\_y:* значенняряду *ƒ(xi)*

***:return****:* коефіцієнтимногочленаЛагранжа

*L3(x) = a\*x^3 - b\*x^2 + c\*x + d => [a, b, c, d]*

*L3(x) = 3/10\*x^3 - 13/6\*x^2 + 62/15\*x + 1 => [0.3, -2.16667, 4.13333, 1]*

*"""*

if not (isinstance(list\_x, list) or isinstance(list\_y, list)):

raise ValueError

if len(list\_x) != len(list\_y) or len(list\_x) < 2:

raise ValueError

list\_coefficient = [0] \* (len(list\_x))

# for i in range(1):

for i in range(len(list\_x)):

removed\_element = list\_x[i]

copy\_list = copy.deepcopy(list\_x)

copy\_list.pop(i)

# дільник (x0 - x1)(x0 - x2)(x0 - x3)

# (0 -2)(0 - 3)(0 - 5) = -30

denominator = get\_coeff\_denominator(copy\_list, removed\_element)

for j in range(0, len(copy\_list) + 1):

# комбінації для перемноження (x - x1)(x - x2)(x - x3)

# (x - 2)(x - 3)(x - 5)

combinations = list(itertools.combinations(copy\_list, len(copy\_list) - j))

# ax^3 + bx^2 + cx + d (тут отримуємо a, b, c, d)

# x^3 - 10x^2 + 31x - 30 (1, -10, 31, -30)

summ = get\_numerator\_elements(combinations)

# ділимо на знаменник всі значення чисельника многочлена

# (x^3 - 10x^2 + 31x - 30) / -30

summ /= denominator

# множимо на значеня ƒ(xi), y0

# (x - x1)(x - x2)(x - x3) / (x0 - x1)(x0 - x2)(x0 - x3) \* y0

summ \*= list\_y[i]

list\_coefficient[j] += summ

return list\_coefficient

def lagrange(x):

*"""*

РозраховуємозначенямногочленаЛагранжадляякогосьзначення *x*

***:param*** *x:* аргументх*,* длякогопотрібнознайтизначення

*"""*

coefficients = coefficient(lagrange\_x, lagrange\_y)

return sum([coefficients[i] \* pow(x, i) for i in range(len(coefficients))])

def get\_formula():

*"""* Отримуємоформулуінтерполяційноїфункції *y = Ln(x)"""*

coeffs = coefficient(lagrange\_x, lagrange\_y)

formula = ""

for i in range(len(coeffs)):

formula = str(round(coeffs[i], 5)) + "x^" + str(i) + " " + formula

if coeffs[i] > 0 and i != len(coeffs) - 1:

formula = "+ " + formula

return formula

def get\_all\_dots(\*args):

*"""*

Знаходиомзначення *L(xi)* длярядузаданихточок

***:param*** *args: xi*

***:return****: L(xi)*

*"""*

# знаходиом значення L(xi)

for x in args:

print("\nX= ", x, "\nY= ", round(lagrange(x), 5))

def draw():

*"""Function for drawing graphic"""*

x = np.linspace(-10, 10, 100)

y = lagrange(x)

plt.plot(x, y, 'black')

plt.xlabel('x')

plt.ylabel('y')

plt.title('Lagrange:')

plt.grid(color='green', linestyle='--', linewidth=0.5)

plt.show()

"""Коменарі для l0(x) з прикладу"""

lagrange\_x = [-4, -3, 0, 1]

lagrange\_y = [-7, 10, -11, -22]

print(get\_formula())

get\_all\_dots(-2, -0.5, 0.5, 2)

draw()

"""Дані з прикладу в ДЗ"""

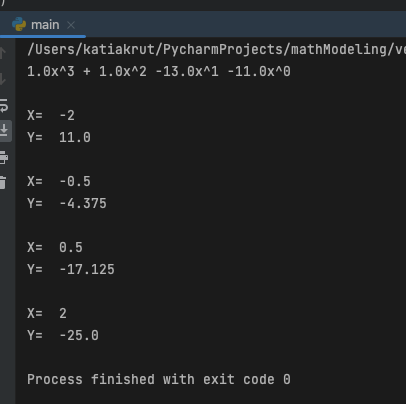
# lagrange\_x = [0, 2, 3, 5]

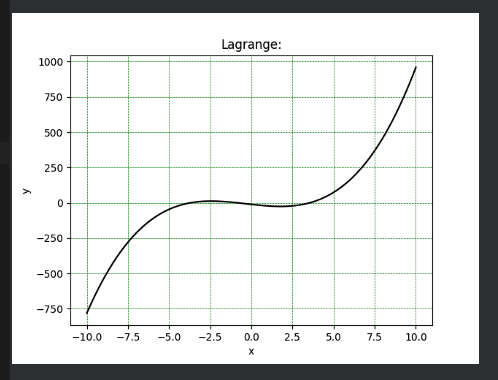
# lagrange\_y = [1, 3, 2, 5]

# print(get\_formula())

# get\_all\_dots(4)

# draw()

**Приклад роботи:**

****

**Висновки:**

Під часи виконання домашнього завдання було розроблено алгоритм для програмної реалізації побудови многочлена Лагранжа Ln(x) для функції ƒ(x), що задана таблицею. Алгоритм редставлений у вигляді блок схеми, а для пограмної реалізації було вибрано мову програмування Python. Крім того, розроблено алгорит для пошуку значення yi = L(xi) за наявним набором точок xi. Також доступний функціонал для побудови графіку інтерполяційної функції y = Ln(x) за знадений набором данних {xi; L(xi)}.