

Лабораторна робота №2

*Програмування алгоритмів лінійної структури та розгалужень.
Обчислення арифметичних виразів.*

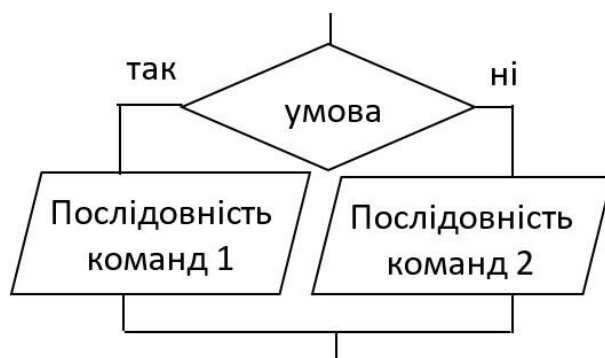
Мета роботи: ознайомлення зі стандартними математичними функціями модуля `math` та конструкціями розгалужень мови `Python`.

Короткі теоретичні відомості

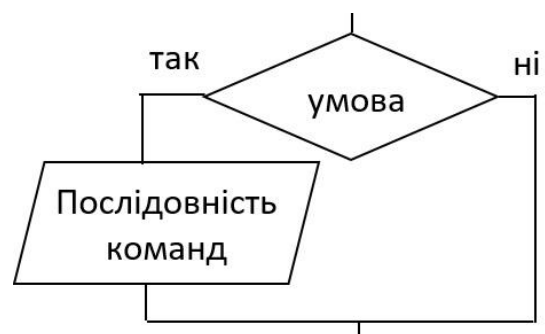
В різних обчислювальних задачах, на етапі обчислення розв'язку, програмі доводиться обирати послідовність команд, які вона виконуватиме для отримання результату, залежно від того чи дані, задані користувачем відповідають тим чи іншим критеріям. Така ситуація в програмуванні називається **розгалуженням**. Тут частина команд алгоритму може виконуватися чи не виконуватися залежно від результату перевірки деякої умови (значення **логічного виразу**).

Розгалуження зустрічаються у двох формах: **повна форма розгалуження** передбачає дві послідовності команд, одна з яких виконуватиметься, якщо значення логічного виразу істинне, інша, – якщо ні; **коротка форма розгалуження** дозволяє виконати деяку послідовність команд, якщо значення логічного виразу істинне, і не виконувати нічого у протилежному випадку. Графічне зображення обидвох форм подано нижче.

Повна форма розгалуження



Коротка форма розгалуження



Враховуючи, що в синтаксичних конструкціях **Python** беруть участь не лише вертикальні, а й горизонтальні розділювачі тексту, можна припустити, що синтаксис умовного оператора тут складний і заплутаний. Насправді, це не зовсім так, оскільки різних конструкцій оператора **`if`** в **Python** доволі багато. З іншого боку, розмаїття синтаксичних форм умовного оператора дозволяє писати на **Python** добре структурований і лаконічний код. В цьому розділі звернемо увагу лише на ті форми оператора **`if`**, що використовуються в програмах розв'язування задач до лабораторних робіт.

Повна форма умовного оператора в **Python** має структуру:

```
if <умова> :  
    <блок інструкцій 1>  
else :  
    <блок інструкцій 2>
```

Поділ коду на стрічки в цій конструкції важливий, насамперед, до уваги береться вертикальне розташування початку стрічок з кодом. Якщо не дотримуватися наступних правил запису умовного оператора, **Python** може не впоратися з інтерпретацією коду і виводити повідомлення про помилки.

- ✓ *пов'язані стрічки коду з **if** та **else** повинні починатися з однакової кількості відступів зліва (пробілів, чи знаків табуляції), це вказує інтерпретатору на один рівень вкладеності цих команд;*
- ✓ *блоки команд після **if** чи **else** повинні складатися зі стрічок з коду з однаковою кількістю відступів зліва, але більшою ніж у оператора, до якого вони належать (вкладаються), дужок для об'єднання блоку чи додаткових розділових знаків при цьому не потрібно;*
- ✓ *якщо блок команд після **if** чи **else** містить лише одну команду, його можна записати в тій же стрічці, після знаку ":";*
- ✓ *послідовність інструкцій одного рівня вкладеності можна розмістити в одній стрічці, розділяючи їх знаком ";"*;
- ✓ *блок інструкцій після **if** чи **else**, записаний в одну стрічку з розділювачем ";" можна помістити в тій же стрічці, після знаку ":".*

Коротка форма розгалуження, очевидно, утворюється з описаної вище конструкції відкиданням **else** та відповідного блоку інструкцій.

При кодуванні розгалужень з багатьма вітками доводиться використовувати вкладені умовні конструкції, враховуючи використання горизонтальних відступів, таке вкладення може бути заплутаним та важким для читання. Уникнути цього можна за рахунок команди **elif**, що поєднує в собі дію команди **else** та подальшої перевірки умови для наступної вітки розгалуження.

Окрім згаданих форм умовного оператора в **Python** є цікава конструкція, – так званий *короткий if...else*. Його записують в одну стрічку за такою схемою:

```
<інструкція 1> if <умова> else <інструкція 2>
```

Його дія аналогічна до звичайної повної форми розгалуження: виконується перевірка умови, якщо результат – істина, то виконується інструкція зліва від **if**, якщо ні, – інструкція справа від **else**. Якщо такою конструкцією виконувати обчислення значення деякої змінної у вітках розгалуження, то присвоєння записують лише в інструкції зліва від **if**:

```
<змінна> = <значення 1> if <умова> else <значення 2>
```

Короткий *if...else* допускає вкладення, щоправда, тут матимемо доволі громіздкі арифметичні вирази і запис їх обчислення у вигляді двох вкладених один в один коротких ***if...else*** в одній стрічці буде надзвичайно довгим. Саме для такого випадку у **Python** передбачено спеціальний розділювач “\”, щоб повідомити інтерпретатору, що наступна стрічка коду є продовженням попередньої.

Завдання 1. Скласти алгоритм розв’язування задачі відповідно до варіанта та зобразити його графічно у вигляді блок-схеми. Записати алгоритм у вигляді коду програми на **Python**. Виконати складену програму, використовуючи командну стрічку інтерпретатора для декількох різних наборів вхідних даних. У звіт подати:

- умову задачі варіанта;
- блок-схему алгоритму розв’язування задачі;
- складений Python-код;
- результати обчислень хоча б для 2-х різних наборів вхідних даних.

Варіанти завдань:

Запрограмувати обчислення виразу за заданими формулами для введених користувачем вхідних даних:

$$1. y = |v^2 - u^2|^{2z+3}, \text{ де } u = \sin t + \cos t, v = |\sin t|^{\cos t};$$

$$2. y = \frac{v^{\sin p} - v^{\cos p}}{e^t}, \text{ де } p = x^3 + 3x^2 + 18, v = \sin t + x, x > 1;$$

$$3. y = \frac{e^{\sin p} - e^{\cos p}}{\ln|z|}, \text{ де } p = |x^3 \operatorname{tg} t|, z = \frac{x+t}{xt};$$

$$4. z = \ln(v^2 + u^2), \text{ де } u = \sin^2 t + xt, v = \frac{5x^3}{t^2+2t};$$

$$5. z = \operatorname{tg}(p^3 + 12q), \text{ де } p = \log_5 x + \log_7 y, q = 5x^y, x, y > 0;$$

$$6. z = \cos(p^{q+2} + 2pq^p), \text{ де } q = \ln x + |\operatorname{ctg} y|, p = y^x + x^y, x, y > 1;$$

$$7. p = \lg x - y^q \ln|e^{xq^2} v + e^{-qv}|, \text{ де } v = \lg x + y, q = y^3 - x^5, x, y > 0;$$

$$8. v = \operatorname{tg}(p^3 + 15q), \text{ де } p = \lg x - y^x \ln|x + e^{-4y}|, q = 5x^y, x > 1;$$

$$9. z = |v^{2t} - u^{2t}|^{2x+3}, \text{ де } u = \sin t + \cos t, v = \sin t + x;$$

$$10. y = \frac{e^{\sin p} - e^{\cos p}}{\log_3 z}, \text{ де } p = x^2 \sin t, z = \left| \frac{x-t}{xt} \right|;$$

$$11. z = \frac{p^{\sin t} - v^{\cos t}}{e^{p+v}}, \text{ де } p = x^3 + 3t^2, v = |\sin x|^{\cos t}, x > 0;$$

$$12. y = (v^2 + u^2)^{2u+3v}, \text{ де } u = \sin t - \cos t, v = \sin(z^2 + t^2);$$

$$13. z = \cos(p^{q+2} + 2p - q^p), \text{ де } p = |x \operatorname{ctg} y|, q = |y^x - x^y|, x, y > 0;$$

$$14. z = \ln(v^2 + u^2) + \cos u, \text{ де } u = \sin^2 t + xt, v = \frac{5x^3}{x^2+2x};$$

15. $y = |v^{2t} - u^{2t}|^{2u+3v}$, де $u = \sin t - \cos t$, $v = \sin t + x$;

16. $p = \frac{|e^{u^2} + e^{-3u}|}{\log_3 v}$, де $u = \cos x + \sin y$, $v = y^{3x} + x^{5y}$, $x, y > 0$.

Приклад виконання завдання 1

Варіант № 16. Умова задачі: запрограмувати обчислення виразу за заданими формулами для введених користувачем вхідних даних

$$p = \frac{|e^{u^2} + e^{-3u}|}{\log_3 v}, \text{ де } u = \cos x + \sin y, v = y^{3x} + x^{5y}, x, y > 0.$$

Єдине, що може викликати роздуми, – які саме змінні слугують вхідними даними, а які є результатом обчислень, а також звідки брати значення числа e .

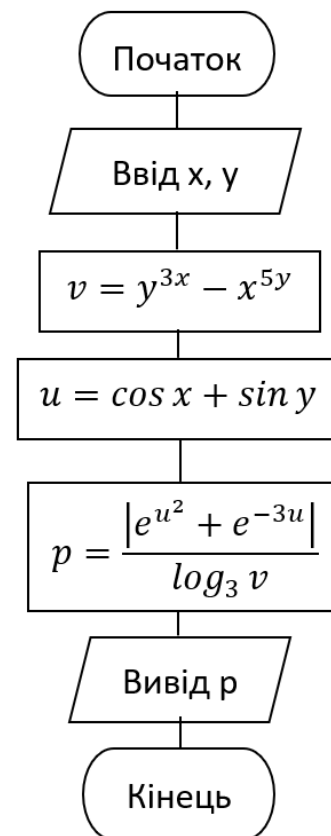
Щодо числа e , то знати його значення не потрібно, математичні бібліотеки усіх мов програмування містять функцію $\exp(x)$ («експонента») для обчислення виразу e^x . За потреби ж дізнатися значення самого числа e достатньо обчислити значення цієї функції при $x = 1$.

Щоб встановити вхідні дані для обчислень слід уважно розглянути формули для обчислення p . Окрім стандартних елементарних функцій тут фігурують дві невідомі величини u та v . Для знаходження u та v задано два наступні вирази, які, очевидно є функціями від невідомих x та y . Отже, задавши деякі значення для x та y , ми зможемо обчислити відповідні значення u та v , а за ними знайти результат p .

Тобто, схематично задачу можна сформулювати так: за заданими значеннями x та y обчислити значення p , використовуючи наступні формули. Алгоритм розв'язання задачі такий:

- користувач вводить два числа – значення невідомих x та y ;
- програма знаходить значення «проміжних» змінних u та v за відповідними формулами;
- за отриманими значеннями u та v програма обчислює результат, тобто значення величини p ;
- обчислений результат програма повідомляє користувачеві і завершує свою роботу.

Графічне зображення сформульованого вище словесно алгоритму подано у вигляді блок-схеми на малюнку.

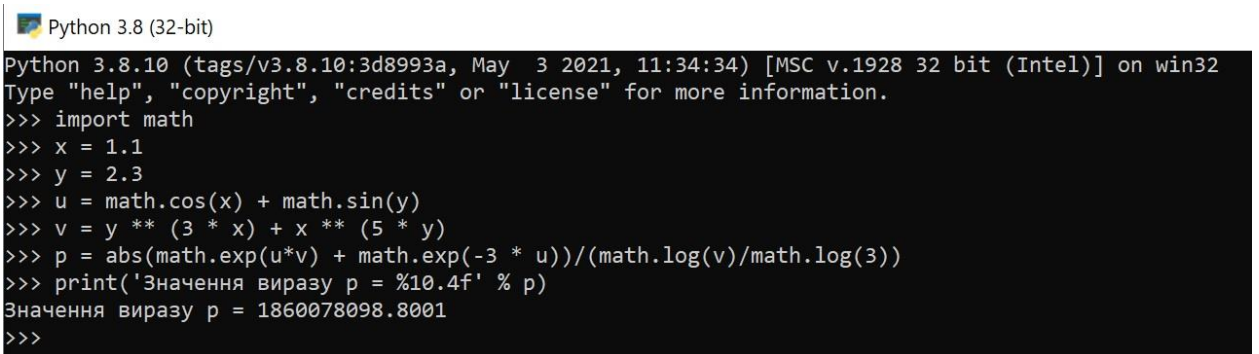


Код програми для розв’язування задачі на **Python**:

```
# Підключення модуля елементарних
import math # математичних функцій
x = float(input()) # ввід дійсних значень та створення
y = float(input()) # змінних x, y типу float
u = math.cos(x) + math.sin(y)
v = y ** (3*x) + x ** (5*y)
p = abs(math.exp(u*u) +
math.exp(-3*u)) / (math.log(v)/math.log(3))
# Вивід значення p з чотирма знаками
# після коми на 10 позицій
print("Значення виразу %10.4f\n" % p)
```

Оскільки ми виконуватимемо обчислення за допомогою командної стрічки, – коментарі в коді, що починаються з «#» вводити не потрібно. Так само для вводу даних обмежимося простим присвоєнням, без використання функції вводу даних `input()`.

Запускаємо інтерпретатор Python в режимі командної стрічки. Після запрошення «>>>» інтерпретатора вводимо послідовно команди складеної нами програми та записуємо результат у звіт.



```
Python 3.8 (32-bit)
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:34:34) [MSC v.1928 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> x = 1.1
>>> y = 2.3
>>> u = math.cos(x) + math.sin(y)
>>> v = y ** (3 * x) + x ** (5 * y)
>>> p = abs(math.exp(u*v) + math.exp(-3 * u))/(math.log(v)/math.log(3))
>>> print('Значення виразу p = %10.4f' % p)
Значення виразу p = 1860078098.8001
>>>
```

Виконуємо програму ще раз при інших значеннях x та y та записуємо результат у звіт.

Завдання 2. Скласти алгоритм розв’язування задачі відповідно до варіанта та зобразити його графічно у вигляді блок-схеми. Створити **Python**-програму для , використовуючи **IDE PyCharm**. У звіт подати:

- умову задачі варіанта;
- блок-схему алгоритму розв’язування задачі;
- код програми на Python;
- результат виконання програми для деяких вхідних даних.

Варіанти завдань:

Скласти програму для обчислення значення заданої функції $y(x)$ за введеним з клавіатури значенням змінної x :

$$1. y = \begin{cases} \sqrt{\cos x + \ln x}, & \text{якщо } x = 3, \text{ або } x = 7, \\ \sqrt[5]{ax} + \operatorname{tg} x, & \text{при } 3 < x < 7, \\ \frac{x}{x+a} & \text{у решті випадків.} \end{cases}$$

$$2. y = \begin{cases} a \lg x + \sqrt[3]{|x|}, & \text{якщо } x = 8, \text{ або } x = 10, \\ 2a \cos x + 3x^2, & \text{якщо } 8 < x < 12, \\ \frac{x+a}{a} & \text{у решті випадків.} \end{cases}$$

$$3. y = \begin{cases} x \sqrt[5]{3x - a}, & \text{якщо } x < a, \\ x \cos ax, & \text{якщо } x = a, \\ e^{-x} \cos ax & \text{якщо } x > a. \end{cases}$$

$$4. y = \begin{cases} \sqrt{ax^2 + \sin x}, & \text{якщо } x = 3 \text{ або } x = 7, \\ \arccos x, & \text{при } |x| < 1, \\ \log_2 |x| & \text{у решті випадків.} \end{cases}$$

$$5. y = \begin{cases} \sqrt{a + \lg x}, & \text{якщо } x > 1, \\ \arcsin x, & \text{якщо } |x| < 1, \\ x^a & \text{у решті випадків.} \end{cases}$$

$$6. y = \begin{cases} ax^8 \ln x, & \text{якщо } 1 \leq x \leq 2, \\ 1, & \text{якщо } x < 1, \\ e^{ax} \cos ax & \text{якщо } x > 2. \end{cases}$$

$$7. y = \begin{cases} \pi x^2 - 7/x^2, & \text{якщо } x < 1,3, \\ ax^6 + 7/\sqrt{x}, & \text{якщо } x = 1,3, \\ \lg(x + 7\sqrt{x}), & \text{якщо } x > 1,3. \end{cases}$$

$$8. y = \begin{cases} ax^5 + bx + c, & \text{якщо } x < 1,2, \\ a/x + \sqrt{x^2 + 1}, & \text{якщо } x = 1,2, \\ (a + bx)/\sqrt{x^2 + 1}, & \text{якщо } x > 1,2. \end{cases}$$

$$9. y = \begin{cases} x \sqrt[3]{x - a}, & \text{якщо } x > a, \\ x \sin ax, & \text{якщо } x = a, \\ e^{-x} \cos ax, & \text{якщо } x < a. \end{cases}$$

$$10. y = \begin{cases} \frac{a+b}{e^{x+\cos x}}, & \text{якщо } x < 2,8, \\ (a + b)/(x + 1), & \text{якщо } 2,8 \leq x \leq 6, \\ e^{-x} \cos ax & \text{якщо } x > 6. \end{cases}$$

$$\begin{aligned}
11. y &= \begin{cases} bx - \lg b x, & \text{якщо } x < 1, \\ 1 + x^{10}, & \text{якщо } x = 1, \\ bx + \ln bx & \text{якщо } x > 1. \end{cases} \\
12. y &= \begin{cases} \frac{\ln^3 x + x^2}{\sqrt{x+t}}, & \text{якщо } x < 0,5, \\ \frac{\sqrt{x+t}+1}{x}, & \text{якщо } x = 0,5, \\ \cos x + t \sin^2 x, & \text{якщо } x > 0,5. \end{cases} \\
13. y &= \begin{cases} \sqrt{ax^2 + \sin x}, & \text{якщо } x = -5, \text{ або } x = 2, \\ \arcsin x, & \text{якщо } |x| < 1, \\ \log_3 |x|, & \text{у решті випадків.} \end{cases} \\
14. y &= \begin{cases} \pi x^2 - 7/x^2, & \text{якщо } x = 5, \\ ax^6 + 7/x, & \text{якщо } x < 5, \\ \lg(x + 7\sqrt{x}), & \text{якщо } x > 5. \end{cases} \\
15. y &= \begin{cases} x^3 \sqrt{3x - a}, & \text{якщо } x < a, \\ x \sin ax, & \text{якщо } x = a, \\ e^{-x} \cos ax & \text{якщо } x > a. \end{cases} \\
16. y &= \begin{cases} x^3 \sqrt{ax^3 + \sin x}, & \text{якщо } 0 \leq x \leq 5; \\ a \cdot \sin^3 x, & \text{якщо } x < 0; \\ \log_x |x^3 - e^x|, & \text{якщо } x > 5. \end{cases}
\end{aligned}$$

Приклад виконання завдання 2

Варіант № 16. Умова задачі: Скласти програму для обчислення значення заданої функції $y(x)$ за введенням з клавіатури значенням змінної x :

$$y = \begin{cases} x^3 \sqrt{ax^3 + \sin x}, & \text{якщо } 0 \leq x \leq 5; \\ a \cdot \sin^3 x, & \text{якщо } x < 0; \\ \log_x |x^3 - e^x|, & \text{якщо } x > 5. \end{cases}$$

Це типова задача, що полягає в обчисленні значення функції, заданої окремими формулами на різних проміжках числової осі. З використанням повної форми розгалуження схема її розв'язання могла б бути такою:

- якщо $x < 0$ обчислюємо значення функції y за формулою $y = a \cdot \sin^3 x$;
- інакше, але якщо $x > 5$ обчислюємо значення функції за формулою $y = \log_x |x^3 - e^x|$;
- у решті випадків значення функції обчислюємо за формулою $y = x^3 \sqrt{ax^3 + \sin x}$.

Попри те, що використання повної форми розгалуження дозволяє виконувати менше перевірок (обчислень логічних виразів), розробники надають перевагу короткій формі. Використання короткої форми розгалуження робить код програми зручнішим для читання, а логіку її роботи більш прозорою.

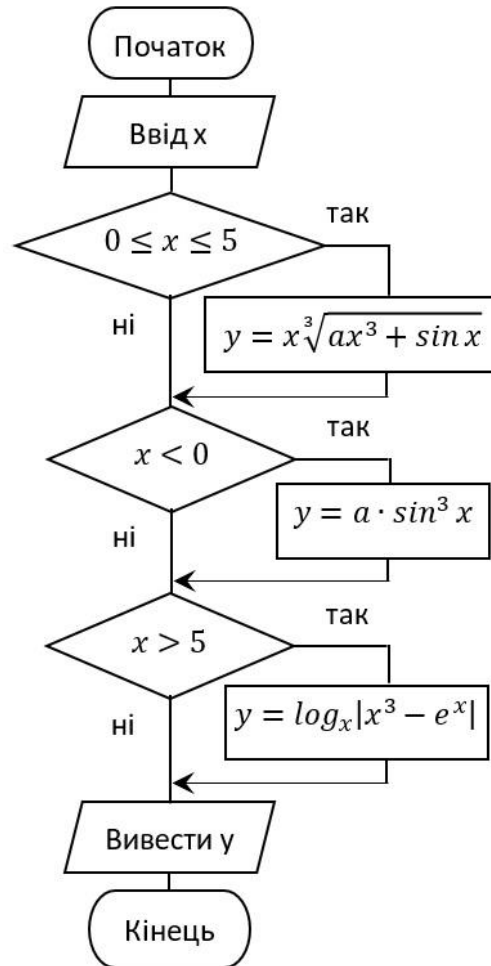
Тому в програмах для розв'язування цієї задачі використовуватимемо короткі розгалуження.

Алгоритм впливає безпосередньо з формули для обчислення y , його графічне зображення подано на малюнку.

Параметр a , що фігурує в формулах вважатимемо константою, яку задамо в коді програми за допомогою іменованої константи, присвоївши їй деяке довільне значення.

Код програми на **Python**:

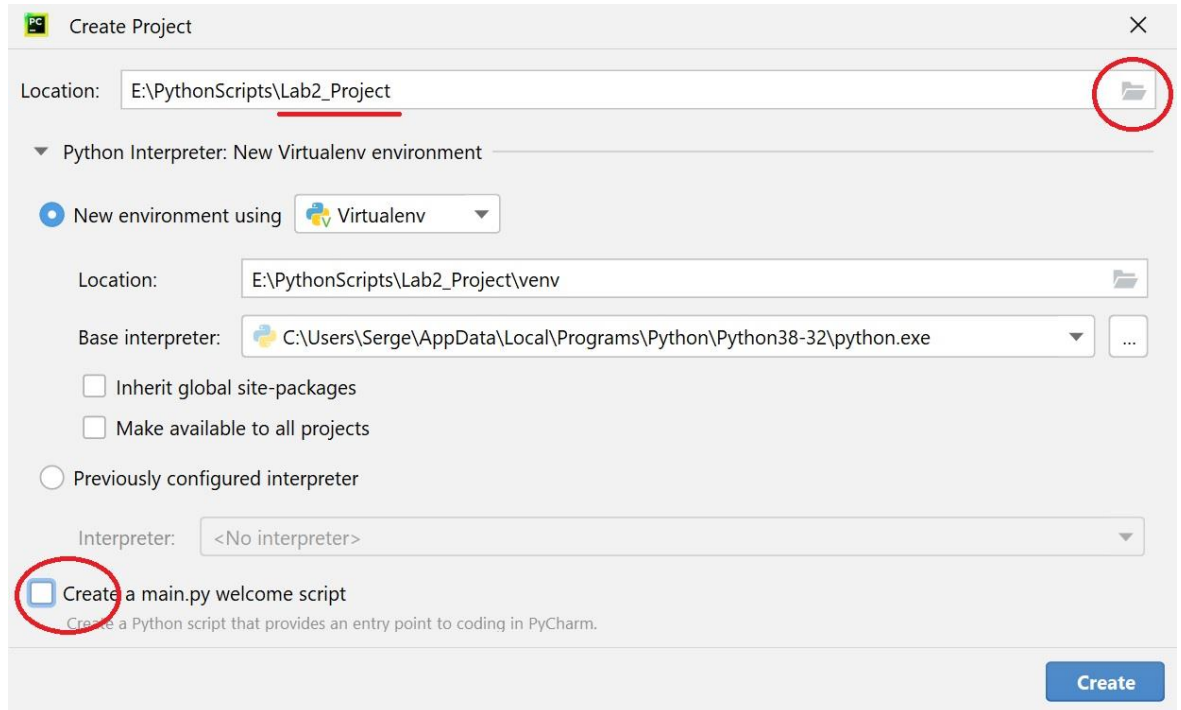
```
# використовуємо математичні
функції
import math
# задаємо значення параметра
a = 7.7
print('Введіть значення x:')
x = float(input())
y = a * math.sin(x) ** 3 if x < 0 else \
math.log(abs(x ** 3 - math.exp(x)))/math.log(x) if x > 5 \
else x * ((a * x ** 3 + math.sin(x)) ** (1.0/3))
print("Значення виразу %10.4f" % y)
```



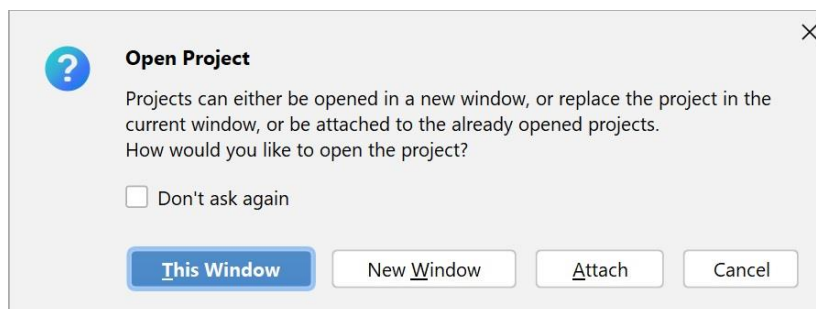
Для набору та виконання коду поданого вище використаємо **PyCharm**. Знаходимо цю програму в списку програм, використовуючи пошук чи будь-яким іншим способом та запускаємо її. Якщо основне вікно програми вже містить відкритий проєкт то створюємо новий проєкт команду **File=>New Project...**

Поле «**Location**» у вікні «**Create Project**» фактично задає повну назву папки (повний шлях до папки у файловій системі та саму назву папки), до якої будуть поміщатися файли проєкту. Назву папки (наприклад, **Lab2Project**) в цьому полі можна вписати з клавіатури, а обрати шлях до неї можна за

допомогою вікна вибору файлу, що відкривається кнопкою з зображенням папки в правому кінці поля. Заготовка шаблонного коду нам не потрібна, тому позначку з пункту «*Create a main.py welcome script*» знімемо. Після натискання кнопки «*Create*» у діалоговому вікні потрібно обрати спосіб відкриття нового проекту.

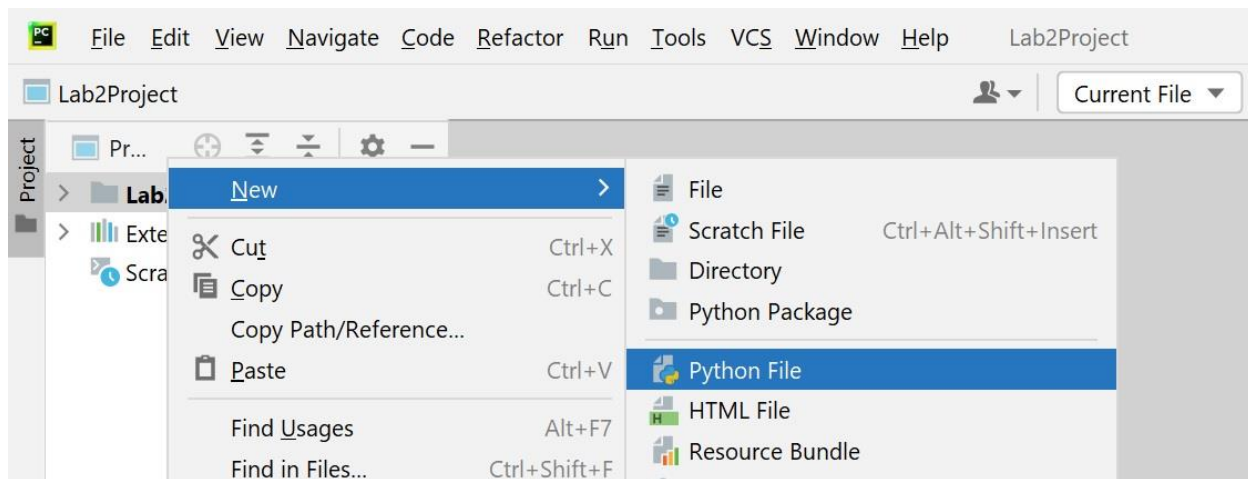


Обираємо «This Window», тоді новий проєкт буде відкрито у тому ж вікні, замість поточного проєкту. Якщо при цьому встановити позначку для опції «Don't ask again» програма запам'ятає наш вибір і наступного разу буде виконувати відкриття віна тим самим способом.

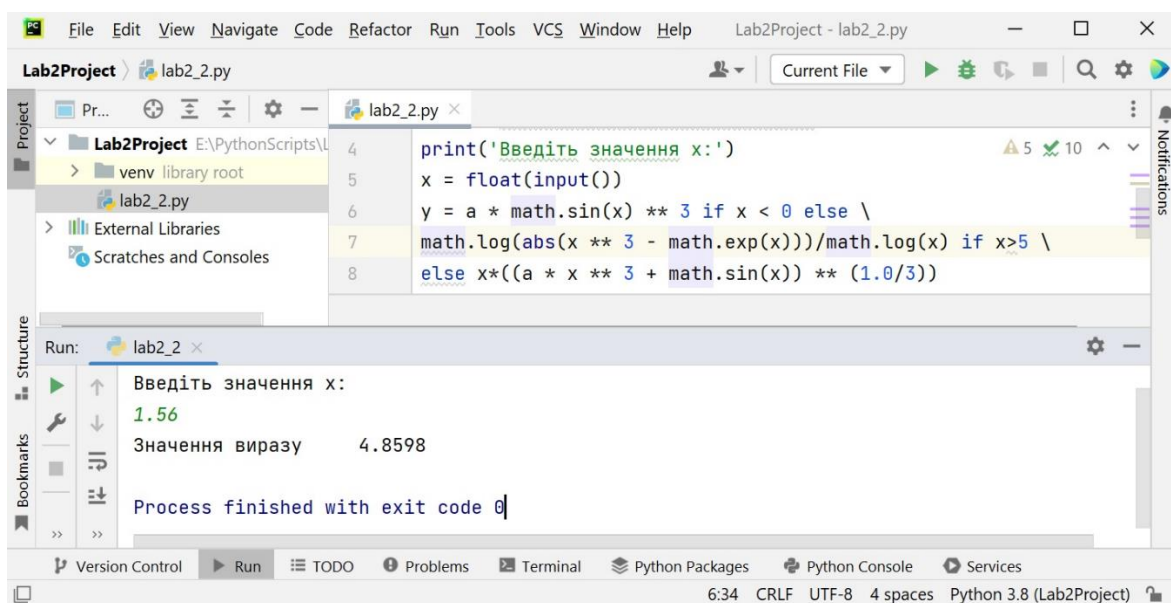


Після цього у вікні PyCharm відкриється порожній проєкт з назвою, що співпадає з заданою нами назвою папки. Новий проєкт не міститиме жодного файлу з кодом, – файл для написання нашого коду можна створити клацнувши правою клавішею миші на вузлі з назвою проєкту у вікні «*Project*» зліва.

Лабораторний практикум з програмування на Python



Далі слід обрати «**New=>Python File**» та задати назву файла. Тепер можна перейти до написання коду.



Роботу програми можна перевірити, запустивши її код на виконання кнопкою у вигляді зеленого трикутника на панелі інструментів, або командою «**Run => Run main**» з меню PyCharm.

Запитання для самоконтролю

1. Для чого в програмах застосовують команду розгалуження?
2. Опишіть процес виконання команди повного розгалуження **if-else**.
3. Як виконується команда короткого розгалуження **if**?
4. В чому проявляються відмінності між короткою та повною формами розгалуження?
5. Які розділові знаки та синтаксичні особливості використовуються при написанні команд розгалуження в **Python**?
6. Чи можна записати оператор **if-else** в одну стрічку? Якщо так, то яких умов при цьому слід дотримуватися?
7. Який оператор Python обчислює остачу від ділення цілих чисел?