НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №*1*

із дисципліни «Основи програмування»

на тему

ПРОГРАМУВАННЯ ЛІНІЙНИХ АЛГОРИТМІВ ТА РОЗГАЛУЖЕНИХ ПРОЦЕСІВ

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-93 | *асистент Дрозденко О. М.* |
| *Пиндиківський Т. Р.* |  |

Київ — 2019

Зміст

[МЕТА РОБОТИ 3](#_Toc19567015)

[ПОСТАНОВКА ЗАДАЧІ 4](#_Toc19567016)

[ОСНОВНА ЧАСТИНА 5](#_Toc19567017)

[ВИСНОВКИ 7](#_Toc19567018)

[ДОДАТОК А 8](#_Toc19567019)

[ДОДАТОК Б 11](#_Toc19567020)

# МЕТА РОБОТИ

Отримати навички по роботі з інтегрованим середовищем Python IDLE. Вивчити принципи розробки програм лінійної структури. Здобути навички роботи з функціями уведення / виведення, різними типами даних, прийнятих в цій мові. Вивчити керуючі структури мови Python і отримати навички складання програм з розгалуженнями.

# ПОСТАНОВКА ЗАДАЧІ

1. ***Завдання 1:***

Створити програму для обчислення тривалості року на двох планетах по введених їх радіусах орбіт і швидкості руху по орбітах. З'ясувати, чи правда, що рік на першій планеті довше, ніж на другий.

1. ***Завдання 2:***

Скласти програму, що визначає, чи пройде графік функції у = ах2 + b х + с через задану з клавіатури точку з координатами (x, y).

1. ***Завдання 3:***

Обчислити значення конкретної функції, в залежності від введеного аргументу :

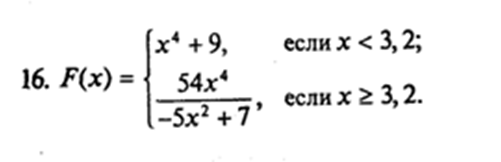


Рис. 1 – завдання 3 лабораторної роботи №1

# ОСНОВНА ЧАСТИНА

1. ***Завдання 1 :***

Для обчислення тривалості року на планетах використовуються змінні radius\_first\_planet, speed\_first\_planet, radius\_second\_planet, speed\_second\_planet, значення яких вводиться з командного рядка за допомогою функції input(). Попередньо зазначимо, що змінні будуть набувати значень дійсного типу, тому перед функцією присвоєння значення змінній вказуємо її тип у вигляді:



Потрібно також згадати, що на початку коду програми імпортуємо модуль math для забезпечення наявності математичних констант, які будуть застосовані для обчислення тривалості року на кожній планеті. Для знаходження значень змінних year\_duration\_first, year\_duration\_second (змінні мають дійсне значення, оскільки вони присвоюють значення з діленням на число) згідно формули використовується константа π (math.pi у вигляді коду). Згодом, викорисовуючи умовний оператор if() знаходимо більше значення тривалості року на планетах та виводимо його у консоль за допомогою інструкції print().

1. ***Завдання 2 :***

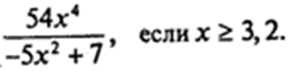
Для визначення графіка квадратичної функції потрібно вказати коефіцієнти квадратного рівняння a, b, c. Вони вводяться з командного рядка за допомогою функції input() з типом дійсних чисел, оскільки коефіцієнтами квадратного рівняння можуть бути будь-які числа. Також з командного рядка вводяться координати x і y точки А. Потім, за допомогою умовної конструкції if() визначаємо, чи належить точка А графіку, підклавши значення відповідних координат у рівняння квадратичної функції.



При утворенні тотожних частин рівняння можна судити про приналежність точки графіку та вивести у командний рядок за допомогою функції print() результат перевірки.

1. ***Завдання 3 :***

Значення функції залежить від аргументу x, який вводиться з консолі з використанням функції введення input(). Потім, за допомогою if() інструкції визначаємо хід обчислень:

* 
* 

Відповідно виконуємо обчислення для одного з випадків відповідно до значення x. Для розрахунків використовуємо бібліотеку math, яку ми імпортували на початку програмного коду. Застосовуємо функцію pow(x, y), яка підносить число х до степеня у. Використання цієї функції реалізовано у рядках 7 та 9 відповідно :





Слід також зазначити, що функція набуватиме дійсні значення через наявність дії ділення у виразі, який присвоюється змінною function. Після цього значення функції виводиться у консоль.

# ВИСНОВКИ

На цій лабораторній роботі я отримав навички по роботі з інтегрованим середовищем Python IDLE. Вивчив принципи розробки програм лінійної структури. Здобув навички роботи з функціями уведення / виведення, різними типами даних, прийнятих в цій мові. Вивчив керуючі структури мови Python і отримав навички складання програм з розгалуженнями. Були розроблені три програми, кожна з яких реалізовує поставлену задачу відповідно до вимог. Кожну з них можна застосовувати для вирішення однотипних завдань. Є багато можливостей для розширення функціоналу з метою забезпечення виконання складніших завдань.

# ДОДАТОК А

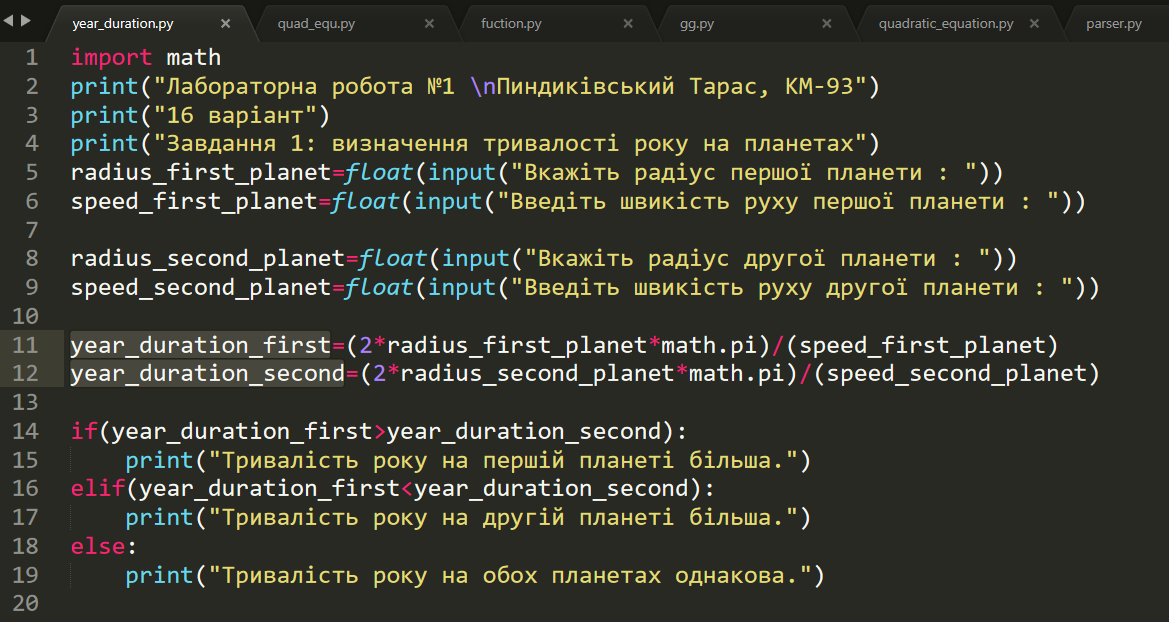


Рис. 2 - програмна реалізація задачі №1

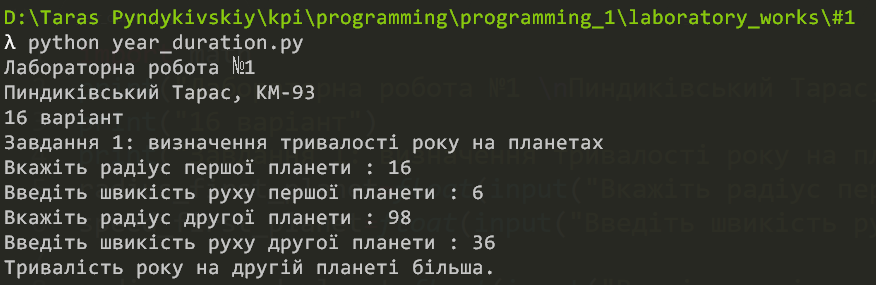


Рис. 3 - тестування програми задачі №1

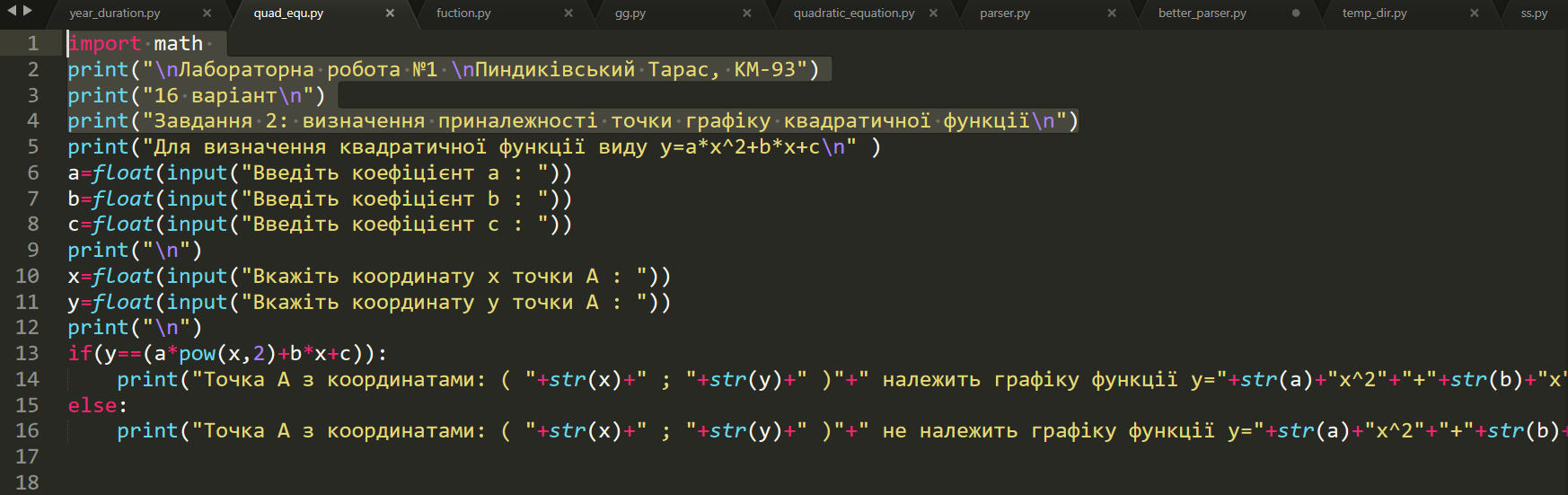


Рис. 4 – код програми для вирішення задачі №2

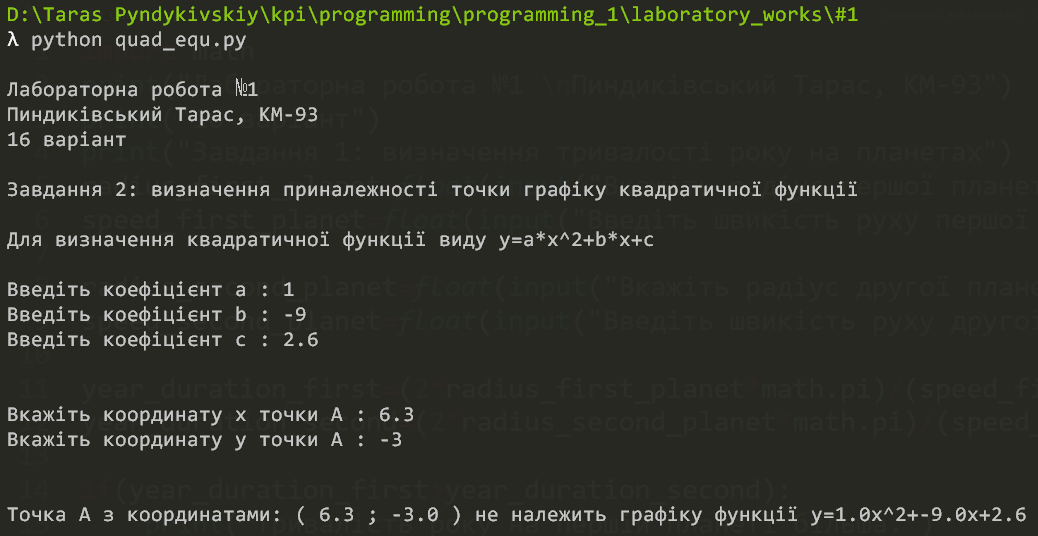


Рис. 5 – тестування програми для задачі №2

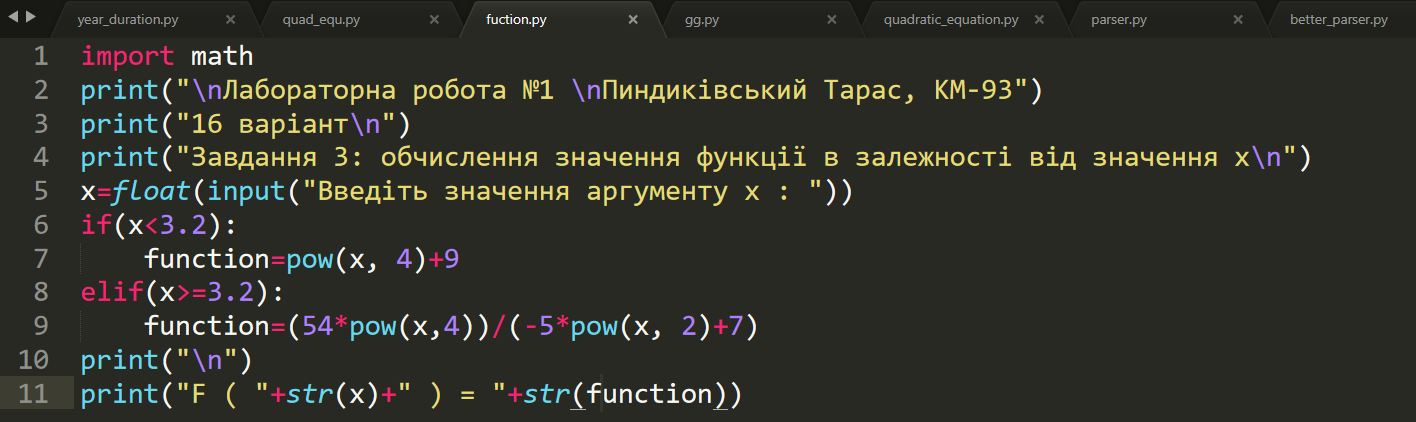


Рис. 6 – програмна реалізація задачі №3

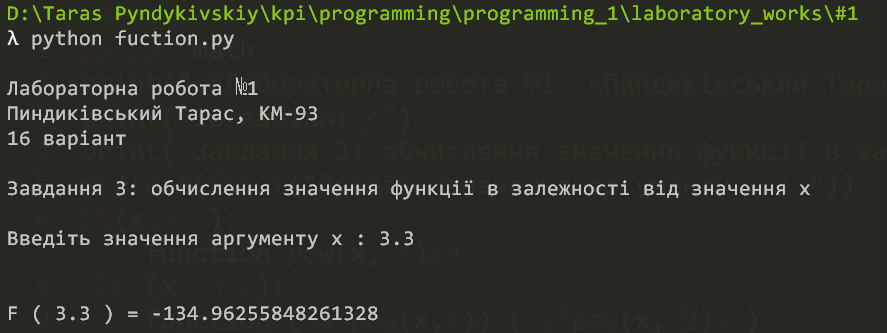


Рис. 7 – тестування програми для задачі №3

# ДОДАТОК Б

1. **Наведіть приклади операцій. Для чого призначена операція присвоєння?**

| **Оператор** | **Назва** | **Пояснення** | **Приклади** |
| --- | --- | --- | --- |
| + | Плюс | Додає два об’єкта. А також конкатенація для типу даних string | 3+5 дає 8. 'a'+'b' дає 'ab' |
| - | Мінус | Або дає число зі знаком - або віднімає одне число від іншого | -5.2 дає негативне число. 50-24 дає 26 |
| \* | Множення | Це або операція множення або операція "розмноження" для рядкового типу.Поверне рядок повторений задану к-ть разів | 2\*3 дорівнює 6. 'la'\*3 дає 'lalala' |
| \*\* | Піднесення до степені | Повертає x в степені y | 3 \*\* 4 дає 81 (іншими словами 3 \*3 \* 3 \* 3) |
| / | Ділення | Розділити x на y | 4/3 дає 1.3333333333333333 |
| // | Цілочислове ділення | Повертає цілочисельну частку від ділення | 4//3 дає 1 |
| % | Ділення по модулю | Повертає залишок від ділення | 8 % 3 дасть 2. -25.5 % 2.25 дасть 1.5. |
| << | Зсув вліво | Зсуває біти числа вліво на задану к-ть позицій. Будь-яке число в оперативній пам’яті ПК представлено у вигляді бітів — або двійкових чисел, тобто 0 і 1 | 2 << 2 дасть 8. В двійковому вигляді 2 представляє собою 10. Зсув вліво на 2 біта дає 1000, що в десятковій системі числення означає 8. |
| >> | Зсув вправо | Зсуває біти числа вправо на задане число позицій | 11 >>1 дасть 5. В двійковій системі числення 11 представляє собою 1011, що будучи зміщеним на 1 біт вправо, дає 101, а це, в свою чергу, не що інше як десяткове 5 |
| & | Побітне І | Побітна операція І над числами | 5 & 3 дає 1 |
| | | Побітне АБО | Побітна операція АБО над числами | 5 | 3 дає 7 |
| ^ | Побітне ВИКЛЮЧНО АБО | Побітна операція ВИКЛЮЧНО АБО над числами | 5 ^ 3 дає 6 |
| ~ | Побітне НЕ | Побітне НЕ x це ~(x+1) | ~5 дає -6 |
| < | Менше ніж | Визначає чи вірно те, що x менше за y. Всі операції порівняння повертають True або False. Зауваж на заглавні букви в цих словах | 5 < 3 дає False 3 < 5 дає True. Можна складати довільні ланцюжки: 3 < 5 < 7 дає True |
| > | Більше ніж | Визначає чи вірно те, що x більше за y | 5 > 3 поверне True. Якщо обидва операнди числа, перед порівнянням їх буде приведено до однаковго типу даних. В протилежному випадку завжди повертає False |
| <= | Менше ніж або еквівалентно | Визначає чи вірно те, що x менше за y або x 0 еквівалентний y | x = 3; y = 6; x <= y поверне True. |
| >= | Більше ніж або еквівалентно | Визначає чи вірно те, що x більше за y або екввівалентний y | x = 4; y = 3; x >= 3 поверне True. |
| == | Еквівалентно до | Визначає чи об’єкти еквівалентні (рівні) між собою | x = 2; y = 2; x == y поверне True. x = 'str'; y = 'stR'; x == y поверне False. x = 'str'; y = 'str'; x == y поверне True. |
| != | Не еквівалентно до | Визначає чи об’єкти не еквівалентні між собою | x = 2; y = 3; x != y поверне True. |
| not | Логічне НІ | Якщо x є True, то поверне False. Якщо x є False, то поверне True | x = True; not x поверне False. Операція not інвертує значення. |
| and | Логічне І | x and y поверне False якщо x є False, інакше значення False | x = False; y = True; x and y поверне False оскільки x є False. У цьому випадку Python не буде перевіряти значення y тому що він знає, що лівий операнд оператора and дорівнює False з чого випливає, що весь вираз дорівнює False незалежно від значення всіх інших величин. Це називається скорочено оцінкою логічних операторів |
| or | Логічне АБО | Якщо x є True, то поверне True. Інакше поверне значення y | x = True; y = False; x or y дає True. Тут також може бути виконана скорочена оцінка виразів |

2**. Які існують правила і рекомендації для іменування змінних?**

Імена змінних можуть бути будь-якими. Однак є рекомендації до їх написання: бажано давати змінним імена, що говорять про призначення даних, на які вони посилаються. Ім'я змінної не повинно збігатися з командами мови. Ім'я змінної має починатися з букви або символу підкреслення.

3. **Які типи даних ви знаєте? Опишіть їх.**

Логічний, може приймати одне з двох значень - True (істина) або False (брехня).

Числа, можуть бути цілими (1 і 2), з плаваючою точкою (1.1 і 1.2), дробовими (1/2 і 2/3), і комплексними.

Рядки - послідовності символів Юнікоду, наприклад, HTML-документ.

Байти і масиви байтів, наприклад, файл зображення в форматі JPEG.

Списки - впорядковані послідовності значень.

Кортежі - впорядковані незмінні послідовності значень.

Множини- невпорядковані набори значень.

Словники - невпорядковані набори пар виду ключ-значення.

4**. Чому операція виду а <в = з недопустима, а операція виду а <в == с – допустима?**

Операція виду а <в = з недопустима, а операція виду а <в == с – допустима, тому що в першому випадку ми присвоюємо порівнянню змінну, а в другому прирівнюємо.

5. **Чим відрізняються результати операцій «/» і «//» для цілих чисел? А для дійсних чисел?**

Оператор / виконує ділення чисел з плаваючою крапкою. Він повертає float, навіть якщо чисельник та знаменник цілі.

Оператор // виконує хитрий вид цілочисельного ділення. Коли результат додатній, ви можете вважати його відкиданням (не округленням) дробової частини звичайного ділення.

При цілочисельному діленні від’ємних чисел оператор // округлює "вгору" до найближчого цілого. Хоча, говорячи формально, він округлює вниз, тому що -6 менше за -5.

Оператор // не завжди повертає цілі. Якщо чисельник чи знаменник дробові, // повертає float, яке, щоправда, все одно округлене до найближчого цілого.

6. **Чи можна перетворити дробове число в ціле? ціле в дробове? У яких випадках можна рядок перетворити в число?**

Так, дробове можна перетворити в ціле за допомогою функції int(), а ціле в дробове – за допомогою float(). Рядок можна перетворити в число, коли змінна частина у рядку складається з цифр.

7. **Який обчислювальний процес називається розгалуженим?**

Розгалуженим називають такий обчислювальний процес, в алгоритмі якого передбачене розгалуження деякої послідовності дій на два (іноді три) напрямки залежно від результату перевірки заданої умови. В алгоритмах розгалуженої структури завжди присутній блочний символ "Вибір", після якого дії виконуються по одній із двох (трьох) гілок.

8. **Як записується повна форма умовної інструкції?**

if (умова 1):

(блок1)

elif(умова 2):

(блок 2)

…

elif(умова n):

(блок n)

else:

(блок n+1)

9. **Яка інструкція називається неповним розгалуженням?**

У повній формі команди умовного переходу виконується серія команд 1, якщо умова істинна, і серія команд 2, якщо вона хибна. В скороченій формі - серія команд виконується тільки у випадку, якщо умова істинна, у протилежному випадку команда розгалуження закінчується і виконується наступний за нею оператор.

10. **Які оператори порівняння використовуються для перевірки умови?**

if-elif-else

11. **Який тип даних повертають оператори порівняння?**

Оператори порівняння повертають значення спеціального логічного типу bool.

12. **Перерахуйте логічні оператори.**

* not (ні)
* and (і)
* or (або)

13. **Наведіть синтаксис каскадної умовної інструкції.**

if (умова 1):

if(умова 2):

…

If(умова n):

Блок n

Elif(умова n+1):

Блок n+1

Else:

Блок n+2

…

Elif(умова 3):

Блок 3

Else:

Блок 4

Elif(умова 5):

Блок 5

Else:

Блок 6