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

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-93 | ст.вик. Дрозденко О. М. |
| Довгаль Є. О. |  |

Київ — 2019

[1 ПОСТАНОВКА ЗАДАЧІ 3](#_Toc524472651)

[1.1 Мета роботи 3](#_Toc524472652)

[1.2 Завдання до лабораторної роботи 3](#_Toc524472653)

[2 ОПИС ПРОГРАМИ 4](#_Toc524472654)

[2.1 Змінні та функції 4](#_Toc524472655)

[Додаток А 6](#_Toc524472656)

[Текст програми 6](#_Toc524472657)

[Додаток Б 8](#_Toc524472658)

[Скріншоти тестувань програми 8](#_Toc524472659)

# 1 ПОСТАНОВКА ЗАДАЧІ

## Мета роботи

Придбання навичок по роботі з інтегрованим середовищем Python IDLE та PyCharm. Вивчення принципів розробки програм лінійної структури. Отримання навичок роботи з функціями уведення / виведення, різними типами даних, прийнятих в цій мові. Вивчення керуючих структур мови Python і отримання навичок складання програм з розгалуженнями.

## 1.2 Завдання до лабораторної роботи

1) Вивчити основні можливості мови програмування Python для підготовки тексту програми і запуску її на виконання.

2) Вивчити структуру програми на мові Python, способи оголошення змінних і перетворення типів, функції введення і виведення даних.

3) Розробити лінійну програму відповідно до варіанта завдання.

4) Вивчити принципи побудови програм із застосуванням умовних операторів.

5)Розробити 2 програми з операторами розгалуження відповідно до варіанта завдання.

6) Вхідні дані і результат роботи супроводжувати відповідною інформацією на екрані.

7) Показати розроблену програму викладачеві.

8) Письмово відповісти на питання для самоперевірки.

9) Оформити звіт відповідно до вимог.

# 2 ОПИС ПРОГРАМИ

## 2.1 Змінні та функції

0.1) Перевірка: чи є значення типу int (цілим числом)

Функція get\_int використовується як допоміжний засіб при перевірки деяких введених користувачем чисел. Працює через цикл while True та конструкцію try-except (ValueError).

Використовується по аналогії з input: get\_int(faq: str) – викликається input(f"{faq}: ") та зберігає дані за умовою, якщо input – ціле число.

0.2) Перевірка: чи є значення типу float (дійсним числом)

Функція get\_float використовується як допоміжний засіб при перевірки деяких введених користувачем чисел. Працює через цикл while True та конструкцію try-except (ValueError).

Використовується по аналогії з input: get\_float(faq: str) – викликається input(f"{faq}: ") та зберігає дані за умовою, якщо input – дійсне число.

1) Розв’язок певного прикладу з трьома змінним.

У лабораторній роботі використовуються 3 змінні типу int – (a, b, c), що вводяться користувачем за допомогою операції get\_int.

Програма обчислює вираз в залежності від введених користувачем значень.

Відповідь виводиться за допомогою операції print.

2) Розв’язок певного прикладу за певними умовами з двома змінними

У лабораторній роботі використовуються 2 змінні типу float – (a, b), що вводяться користувачем за допомогою операції get\_float, причому a і b спочатку приймають однакове значення для подальшої перевірки на a == b через while. Програма просить ввести нове значення b, поки не виконається a!=b.

Далі виконується розгалуження. Менше з цих двох чисел замінюється на половину їх суми, а більше – їх подвоєним добутком.

Відповідь виводиться за допомогою операції print.

3) Обчислення функції в залежності від значення введеної змінної.

У лабораторній роботі використовуються 1 змінна типу float - (x), що вводиться користувачем за допомогою операції get\_float.

Аналіз виконується через скорочений запис if-else. Якщо х ≤ 1, то у відповідь отримаємо 0. Якщо ж інше число, то виконається f(x) = , а результат округлюється до тисячних (для зручності). Так як х = -6 вже за умовою не буде працювати для цього завдання, перевірку на ZeroDivisionError робити не має сенсу.

Відповідь виводиться за допомогою операції print.

Програма може виконуватися безкінечно, поки користувач відповідає на питання «Продовжити? y/n» символом «y» (велика не підтримується). Все почнеться з самого початку.

# Додаток А

## Текст програми

Зовнішній вигляд збережено для більш зручного читання

def get\_int(faq: str):  
 # Фукнція перевіряє, чи є введене значення цілим числом  
 while True:  
 try:  
 num = int(input(f'{faq}: '))  
 break  
 except ValueError:  
 print('ПОМИЛКА! Вводити потрібно лише цілі числа.')  
 return num  
  
  
def get\_float(faq: str):  
 # Фукнція перевіряє, чи є введене значення дійсним числом  
 while True:  
 try:  
 num = float(input(f'{faq}: '))  
 break  
 except ValueError:  
 print('ПОМИЛКА! Вводити потрібно лише дійсні числа.')  
 return num  
  
  
def task1():  
 print('\n<<< Виконання першого завдання >>>')  
 """  
 Користувач уводить три числа. Збільшити перше число в два рази,  
 друге числа зменшити на 3, третє число звести в квадрат  
 і потім знайти суму нових трьох чисел.  
 """  
  
 # Отримуємо бажані 3 числа за допомогою ф-ції get\_int().  
 a = get\_int('Перше число')  
 b = get\_int('Друге число')  
 c = get\_int('Третє число')  
 return a \* 2 + b - 3 + c \*\* 2  
  
  
def task2():  
 print('<<< Виконання другого завдання >>>')  
 """  
 Увести з клавіатури дійсні числа х і у, не рівні одне одному.  
 Менше з цих двох чисел замінити половиною їх суми,  
 а більше - їх подвоєним добутком.  
 """  
  
 # Отримуємо бажані 2 числа за допомогою ф-ції get\_float().  
 a = b = get\_float('Перше число')  
 # Перевірка на виконання першої умови завдання  
 while a == b:  
 b = get\_float('Друге число')  
 if a == b: print('ПОМИЛКА! Змінні x та y не повинні бути рівні одне одному.')  
  
 # ...Менше з цих двох чисел замінити половиною їх суми, а більше - їх подвоєним добутком:  
 # Якщо перше число меньше, то міняємо значення місцями  
 if a < b:  
 return f"x={(a + b) / 2}, y={2 \* a \* b}"  
 else:  
 return f"x={2 \* a \* b}, y={(a + b) / 2}"  
  
  
def task3():  
 print('<<< Виконання третього завдання >>>')  
 """  
 Обчислення конкретної функції,  
 в залежності від введеного значення х  
 """  
  
 # Отримуємо бажане число за допомогою ф-ції get\_float().  
 x = get\_float('Введіть значення х')  
  
 # Скорочений запис if-else:  
 return 0 if x <= 1 else round(1.0 / (x + 6.0), 3)  
 # Якщо умова не виконується, то розв'язується рівняння, а відповідь округлюється до тисячних.

while True:  
 print(f'Результат виконання першого завдання: {task1()}\n')  
 print(f'Результат виконання другого завдання: {task2()}\n')  
 print(f'Результат виконання третього завдання: {task3()}\n'))  
 # Для постійних тестувань  
 go = input('\nПродовжити? y/n\n')  
 if go != 'y': break

# Автор програми, номер лаб. роботи, варіант завдань  
print('\n>>> Лабораторну роботу №1 виконала\n'  
 '>>> студентка КМ-93 Довгаль Єва\n'  
 '>>> Варіанти кожного завдання: 4')

# Додаток Б

## Скріншоти тестувань програми

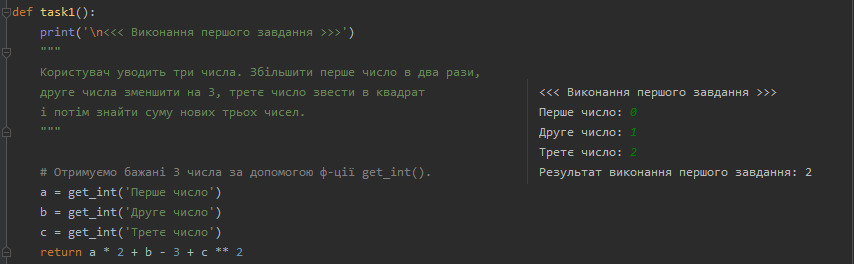


Рисунок 3.1 – Розв’язок певного прикладу з трьома змінним

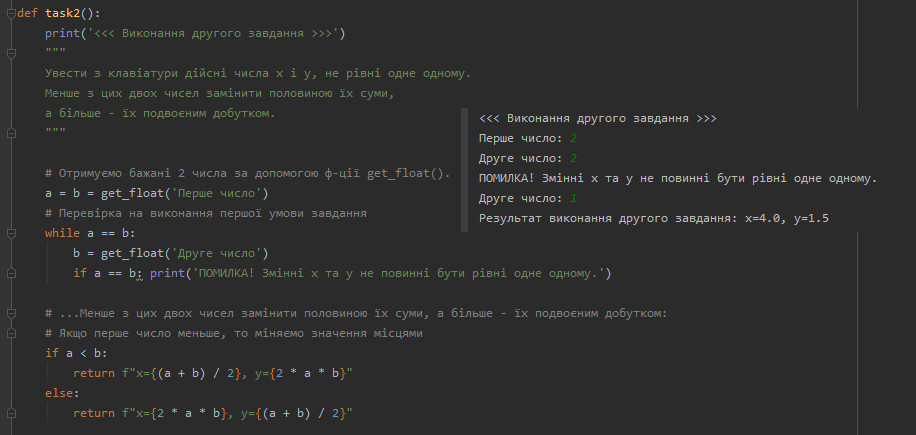


Рисунок 3.2 – Розв’язок прикладу за певними умовами з двома змінними

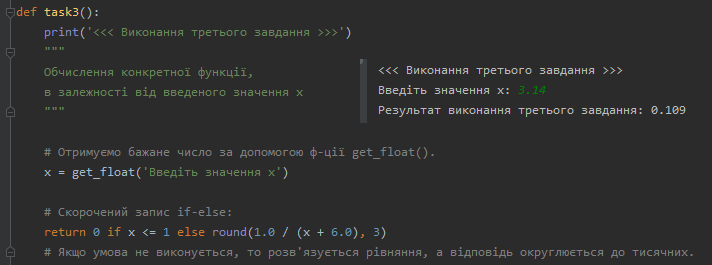


Рисунок 3.3 – Обчислення функції в залежності від значення змінної