**ЗАКЛАД ВИЩОЇ ОСВІТИ**

**«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»**

**Факультет суспільних і прикладних наук**

**Кафедра інформаційних технологій**

ПРОГРАМУВАННЯ МОВОЮ PYTHON

**ПРАКТИЧНА РОБОТА № 3**

Виконав:

студент групи ІПЗс-21-1,

Бойчук Богдан Романочич

Перевірив:

викладач

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата \_\_\_\_\_\_\_\_\_\_

**Івано-Франківськ**

**2024**

**Тема:** Функції користувача

**Мета:** ознайомитися з принципами побудови функцій

користувача на мові Python, з використанням локальних і глобальних

змінних.

**Хід роботи**

**Завдання 1.**

Увести з клавіатури три дійсних числа. Піднести до квадрата ті з них,

значення яких невід'ємні, і в четверту ступінь - від`ємні .

**Код:**

def square\_or\_fourth\_power():

    num1 = float(input("Введіть перше число: "))

    num2 = float(input("Введіть друге число: "))

    num3 = float(input("Введіть третє число: "))

    result = []

    for num in [num1, num2, num3]:

        if num >= 0:

            result.append(num \*\* 2)

        else:

            result.append(num \*\* 4)

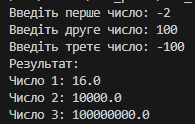
    print("Результат:")

    for i, res in enumerate(result, 1):

        print(f"Число {i}: {res}")

square\_or\_fourth\_power()

**Результат:**



**Пояснення:**

* Спочатку функція square\_or\_fourth\_power() запитує в користувача три дійсних числа за допомогою функції input(). Кожне введене значення конвертується у тип float, щоб обробити як дійсні числа.
* Потім використовується цикл for, щоб пройтися по кожному введеному числу. Для кожного числа перевіряється його знак. Якщо число додатне або рівне нулю, воно підноситься до квадрата (num \*\* 2). Якщо ж число від'ємне, воно підноситься до четвертої ступеня (num \*\* 4). Отримані результати додаються до списку result.
* Після обробки всіх введених чисел програма виводить результати. Вона виводить кожне число разом з його порядковим номером (1 для першого числа, 2 для другого і т.д.) і відповідним піднесенням до степеня.

**Завдання 2.**

Увести з клавіатури координати двох точок А (х1, у1) і В (х2, у2). Скласти алгоритм, який визначає, яка з точок знаходиться ближче до початку координат.

**Код:**

def distance\_from\_origin(x, y):

    return (x \*\* 2 + y \*\* 2) \*\* 0.5

def closer\_point():

    x1 = float(input("Введіть координату x для точки А: "))

    y1 = float(input("Введіть координату y для точки А: "))

    x2 = float(input("Введіть координату x для точки В: "))

    y2 = float(input("Введіть координату y для точки В: "))

    distance\_a = distance\_from\_origin(x1, y1)

    distance\_b = distance\_from\_origin(x2, y2)

    if distance\_a < distance\_b:

        print("Точка А знаходиться ближче до початку координат.")

    elif distance\_b < distance\_a:

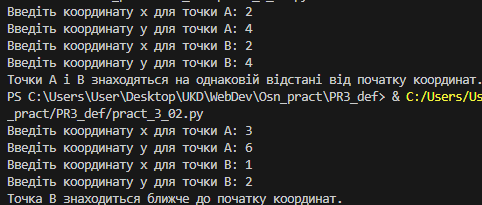
        print("Точка В знаходиться ближче до початку координат.")

    else:

        print("Точки А і В знаходяться на однаковій відстані від початку координат.")

closer\_point()

**Результат:**



**Пояснення:**

* Функція distance\_from\_origin(x, y) обчислює відстань від початку координат до точки (𝑥,𝑦)(*x*,*y*) за допомогою формули відстані від точки до початку координат у двовимірному просторі (𝑥2+𝑦2*x*2+*y*2​).
* У функції closer\_point() користувач вводить координати точок 𝐴 і 𝐵.
* Після введення координат обчислюється відстань кожної точки від початку координат за допомогою функції distance\_from\_origin(x, y).
* За допомогою умовного оператора if-elif-else програма визначає, яка з точок знаходиться ближче до початку координат і виводить відповідне повідомлення.

**Завдання 3.**

Увести з клавіатури величини двох кутів трикутника (в градусах).

Визначити, чи існує такий трикутник, і якщо так, то чи буде він прямокутним.

**Код:**

import math

def is\_triangle\_possible(angle1, angle2):

    if angle1 + angle2 < 180:

        return True

    else:

        return False

def is\_right\_triangle(angle1, angle2):

    if angle1 == 90 or angle2 == 90:

        return True

    else:

        return False

def main():

    angle1 = float(input("Введіть величину першого кута трикутника (в градусах): "))

    angle2 = float(input("Введіть величину другого кута трикутника (в градусах): "))

    if is\_triangle\_possible(angle1, angle2):

        print("Так, цей трикутник існує.")

        if is\_right\_triangle(angle1, angle2):

            print("Цей трикутник є прямокутним.")

        else:

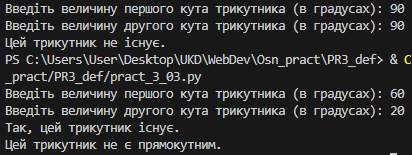
            print("Цей трикутник не є прямокутним.")

    else:

        print("Цей трикутник не існує.")

main()

**Результат:**



**Пояснення:**

* Функція is\_triangle\_possible(angle1, angle2) перевіряє, чи сума двох кутів трикутника менша за 180 градусів. Якщо так, повертається значення True, інакше - False.
* Функція is\_right\_triangle(angle1, angle2) перевіряє, чи один з кутів дорівнює 90 градусів, щоб визначити, чи є трикутник прямокутним. Якщо так, повертається значення True, інакше - False.
* У функції main() користувач вводить значення для першого та другого кута трикутника у градусах.
* Після цього програма викликає функцію is\_triangle\_possible() для перевірки, чи можливий такий трикутник з введеними кутами. Якщо так, програма виводить, що трикутник існує, і перевіряє, чи є він прямокутним за допомогою функції is\_right\_triangle().
* Якщо функція is\_triangle\_possible() повертає False, програма повідомляє, що трикутник не існує.

**Завдання 4.**

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.

**Код:**

def replace\_numbers(x, y):

    if x != y:

        if x < y:

            smaller = x

            greater = y

        else:

            smaller = y

            greater = x

        x = (smaller + greater) / 2

        y = smaller \* greater \* 2

        return x, y

    else:

        print("Помилка: Введені числа рівні одне одному.")

        return None, None

def main():

    x = float(input("Введіть число x: "))

    y = float(input("Введіть число y: "))

    new\_x, new\_y = replace\_numbers(x, y)

    if new\_x is not None and new\_y is not None:

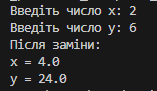
        print("Після заміни:")

        print("x =", new\_x)

        print("y =", new\_y)

main()

**Результат:**



**Пояснення:**

* Функція replace\_numbers(x, y) перевіряє, чи *x* не дорівнює *y*. Якщо це так, вона знаходить менше з двох чисел (*smaller*) і більше з двох чисел (*greater*). Потім обчислює нове значення *x* як середнє арифметичне *smaller* і *greater*, а також нове значення *y* як добуток *smaller* і *greater* удвічі.
* Якщо *x* та *y* рівні, програма виводить повідомлення про помилку.
* У функції main() користувач вводить значення *x* та *y*.
* Потім програма викликає функцію replace\_numbers(x, y), яка замінює числа *x* та *y*, якщо вони не рівні, і повертає їхні нові значення.
* Якщо значення *x* та *y* були успішно замінені, програма виводить їхні нові значення.

**Завдання 5.**

На площині ХОY задана своїми координатами точка А (координати ввести з клавіатури). Вказати, де вона розташована (на якій осі або в якому координатном куті).

**Код:**

def locate\_point(x, y):

    if x == 0 and y == 0:

        print("Точка розташована в початку координат.")

    elif x == 0:

        print("Точка розташована на вісі Y.")

    elif y == 0:

        print("Точка розташована на вісі X.")

    elif x > 0 and y > 0:

        print("Точка розташована в першому квадранті.")

    elif x < 0 and y > 0:

        print("Точка розташована в другому квадранті.")

    elif x < 0 and y < 0:

        print("Точка розташована в третьому квадранті.")

    elif x > 0 and y < 0:

        print("Точка розташована в четвертому квадранті.")

def main():

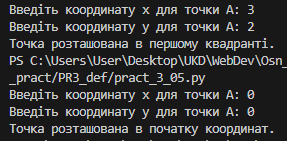
    x = float(input("Введіть координату x для точки А: "))

    y = float(input("Введіть координату y для точки А: "))

    locate\_point(x, y)

main()

**Результат:**



**Пояснення:**

* Функція locate\_point(x, y) приймає координати *x* та *y* точки і визначає її розташування відносно координатних вісей та квадрантів.
* Умовні конструкції перевіряють різні випадки розташування точки:
  + Якщо *x* та *y* рівні 0, то точка знаходиться в початку координат.
  + Якщо *x* або *y* дорівнює 0, то точка знаходиться на відповідній вісі.
  + Якщо *x* і *y* додатні, то точка знаходиться в першому квадранті.
  + Якщо *x* від'ємний, а *y* додатний, то точка знаходиться в другому квадранті.
  + Якщо *x* і *y* від'ємні, то точка знаходиться в третьому квадранті.
  + Якщо *x* додатний, а *y* від'ємний, то точка знаходиться в четвертому квадранті.
* Функція main() вводить координати *x* та *y* для точки *A* з клавіатури і викликає функцію locate\_point(x, y).

**Завдання 6.**

Увести з клавіатури цілі числа a, b. Якщо числа не рівні, то замінити кожне з них одним і тим же числом, рівним більшому із вихідних, а якщо рівні, тозамінити числа нулями.

**Код:**

def modify\_numbers(a, b):

    if a != b:

        result = max(a, b)

    else:

        result = 0

    return result

def main():

    a = int(input("Введіть число a: "))

    b = int(input("Введіть число b: "))

    modified\_number = modify\_numbers(a, b)

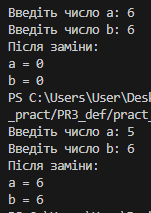
    print("Після заміни:")

    print("a =", modified\_number)

    print("b =", modified\_number)

main()

**Результат:**



**Пояснення:**

* Функція modify\_numbers(a, b) приймає два аргументи *a* та *b* і повертає модифіковане значення.
* Умовна конструкція перевіряє, чи *a* не рівне *b*. Якщо це так, змінній result присвоюється більше з двох чисел за допомогою функції max(a, b). Якщо числа рівні, змінній result присвоюється 0.
* Функція main() вводить числа *a* та *b* з клавіатури.
* Потім програма викликає функцію modify\_numbers(a, b) для модифікації чисел.
* Після модифікації програма виводить нові значення *a* та *b* на екран. Обидва числа *a* та *b* замінюються на одне і те ж саме значення, яке відповідає більшому з двох введених чисел, або на 0, якщо числа рівні.

**Завдання 7.**

Підрахувати кількість негативних серед чисел а, b, с (ввести з клавіатури).

**Код:**

def count\_negative\_numbers(a, b, c):

    count = 0

    if a < 0:

        count += 1

    if b < 0:

        count += 1

    if c < 0:

        count += 1

    return count

def main():

    a = float(input("Введіть число a: "))

    b = float(input("Введіть число b: "))

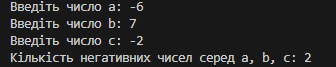
    c = float(input("Введіть число c: "))

    negative\_count = count\_negative\_numbers(a, b, c)

    print("Кількість негативних чисел серед a, b, c:", negative\_count)

main()

**Результат:**



**Пояснення:**

* Функція count\_negative\_numbers(a, b, c): Ця функція приймає три аргументи *a*, *b* та *c* і повертає кількість негативних чисел серед них.
* Змінна count: Ця змінна ініціалізується значенням 0. У кожній з трьох умовних конструкцій перевіряється, чи відповідне число *a*, *b* або *c* менше за 0. Якщо так, значення count збільшується на 1.
* Функція main(): Ця функція відповідає за введення чисел *a*, *b* та *c* з клавіатури.
* Виклик функції та вивід результату: Після того, як користувач ввів числа, програма викликає функцію count\_negative\_numbers(a, b, c) для обчислення кількості негативних чисел. Потім вона виводить кількість негативних чисел на екран.

**Завдання 8.**

Підрахувати кількість додатних серед чисел а, b, с (ввести з клавіатури).

**Код:**

def count\_positive\_numbers(a, b, c):

    count = 0

    if a > 0:

        count += 1

    if b > 0:

        count += 1

    if c > 0:

        count += 1

    return count

def main():

    a = float(input("Введіть число a: "))

    b = float(input("Введіть число b: "))

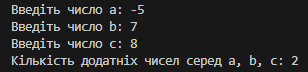
    c = float(input("Введіть число c: "))

    positive\_count = count\_positive\_numbers(a, b, c)

    print("Кількість додатніх чисел серед a, b, c:", positive\_count)

main()

**Результат:**



**Пояснення:**

* Функція count\_positive\_numbers(a, b, c): Ця функція приймає три аргументи *a*, *b* та *c* і повертає кількість додатних чисел серед них.
* Змінна count: Ця змінна ініціалізується значенням 0. У кожній з трьох умовних конструкцій перевіряється, чи відповідне число *a*, *b* або *c* більше за 0. Якщо так, значення count збільшується на 1.
* Функція main(): Ця функція відповідає за введення чисел *a*, *b* та *c* з клавіатури.
* Виклик функції та вивід результату: Після того, як користувач ввів числа, програма викликає функцію count\_positive\_numbers(a, b, c) для обчислення кількості додатних чисел. Потім вона виводить кількість додатних чисел на екран.

**Завдання 9.**

Підрахувати кількість цілих серед чисел а, b, с (ввести з клавіатури).

**Код:**

def count\_integer\_numbers(a, b, c):

    count = 0

    if a.is\_integer():

        count += 1

    if b.is\_integer():

        count += 1

    if c.is\_integer():

        count += 1

    return count

def main():

    a = float(input("Введіть число a: "))

    b = float(input("Введіть число b: "))

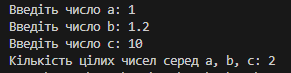
    c = float(input("Введіть число c: "))

    integer\_count = count\_integer\_numbers(a, b, c)

    print("Кількість цілих чисел серед a, b, c:", integer\_count)

main()

**Результат:**



**Пояснення:**

* Функція count\_integer\_numbers(a, b, c): Ця функція приймає три аргументи *a*, *b* та *c* і повертає кількість цілих чисел серед них.
* Змінна count: Ця змінна ініціалізується значенням 0. У кожній з трьох умовних конструкцій перевіряється, чи є відповідне число *a*, *b* або *c* цілим числом за допомогою методу .is\_integer(). Якщо так, значення count збільшується на 1.
* Функція main(): Ця функція відповідає за введення чисел *a*, *b* та *c* з клавіатури.
* Виклик функції та вивід результату: Після того, як користувач ввів числа, програма викликає функцію count\_integer\_numbers(a, b, c) для обчислення кількості цілих чисел. Потім вона виводить цю кількість на екран.

**Завдання 10.**

Визначити, дільником яких чисел а, b, с є число k (ввести з клавіатури).

**Код:**

def is\_divisor\_of\_numbers(a, b, c, k):

    if a % k == 0:

        print(f"{k} є дільником числа {a}.")

    else:

        print(f"{k} не є дільником числа {a}.")

    if b % k == 0:

        print(f"{k} є дільником числа {b}.")

    else:

        print(f"{k} не є дільником числа {b}.")

    if c % k == 0:

        print(f"{k} є дільником числа {c}.")

    else:

        print(f"{k} не є дільником числа {c}.")

def main():

    a = float(input("Введіть число a: "))

    b = float(input("Введіть число b: "))

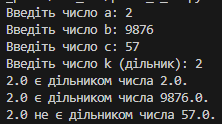
    c = float(input("Введіть число c: "))

    k = float(input("Введіть число k (дільник): "))

    is\_divisor\_of\_numbers(a, b, c, k)

main()

**Результат:**



**Пояснення:**

* Функція is\_divisor\_of\_numbers(a, b, c, k): Ця функція приймає чотири аргументи *a*, *b*, *c* та *k* і перевіряє, чи є *k* дільником для кожного з чисел *a*, *b* та *c*.
* Перевірка на дільник: Використовуючи операцію залишку від ділення %, функція перевіряє, чи ділиться кожне з чисел *a*, *b* та *c* на *k* без залишку:
  1. Якщо залишок від ділення *a* на *k* дорівнює 0, *k* є дільником числа *a*.
  2. Якщо залишок від ділення *b* на *k* дорівнює 0, *k* є дільником числа *b*.
  3. Якщо залишок від ділення *c* на *k* дорівнює 0, *k* є дільником числа *c*.
* Функція main(): Ця функція відповідає за введення чисел *a*, *b*, *c* та *k* з клавіатури.
* Виклик функції та вивід результату: Після того, як користувач ввів числа, програма викликає функцію is\_divisor\_of\_numbers(a, b, c, k) для перевірки, чи є *k* дільником кожного з чисел *a*, *b* та *c*. Програма виводить відповідні повідомлення для кожного числа.