**Звіт до Теми №1**

**Функції та змінні**

**Завдання №1**

“Перевернення” слова

C:\Users\Мой компьютер\AppData\Local\Microsoft\Windows\INetCache\Content.Word\76g.png

Перший рядок створює змінну і присваює їй значення(яке потрібно перевернути). В другому рядку я створила нову змінну reversed\_а, яка використовує [::-1]. В результаті значення змінної "а" перевертається позаду вперед і зберігається в змінній reversed\_а.

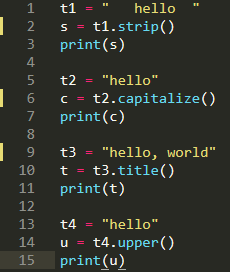
Останній рядок виводить “перевернуте” слово на екран за допомогою функцій print.

Результат:

C:\Users\Мой компьютер\AppData\Local\Microsoft\Windows\INetCache\Content.Word\fuy.png

**Завдання №2**

Тестування функцій strip, capitalize, titale, upper



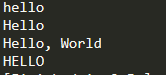
strip() видаляє початкові та кінцеві пробіли зі строки.

capitalize() робить першу букву строки заголовної.

title() робить першу літеру кожного слова в рядку великою, а інші залишаються малими.

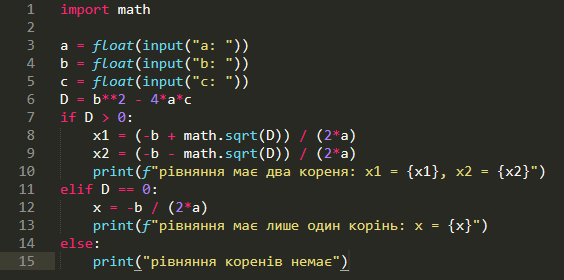
upper() перетворює всі літери строки в великі.

Результат:



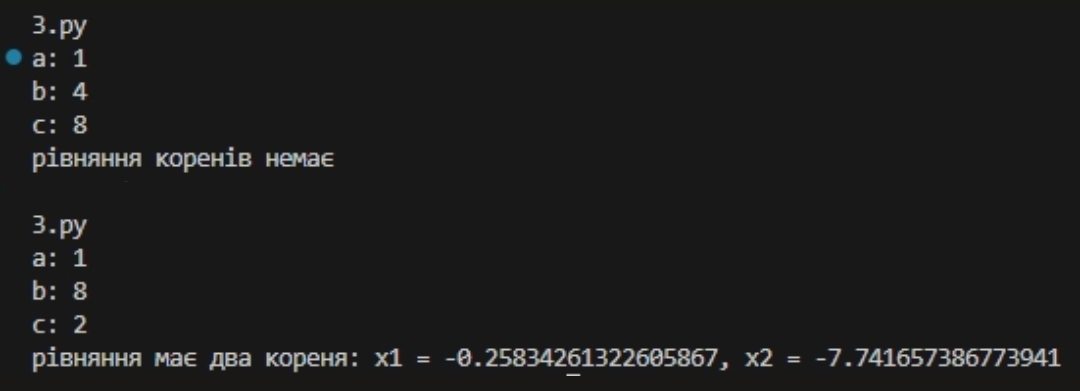
**Завдання №3**

Квадратне рівняння



Цей код спочатку запитує значення коефіцієнтів a, b і c квадратного рівня. Потім вираховує дискримінант і, в залежності від значення, виводить корні рівняння. Якщо D > 0, то у рівняння два дійсних корня. Якщо D = 0, то у рівняння один дійсний корінь. Якщо D < 0, то у рівняння немає дійсних корнів.

Результат:

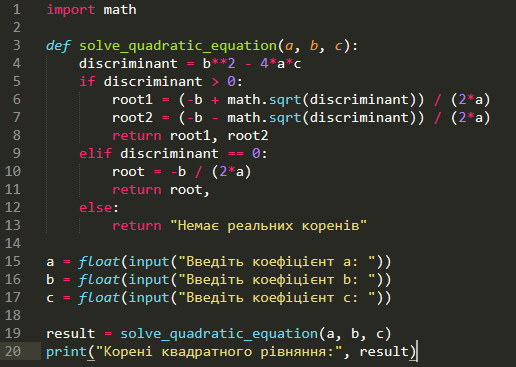


**Звіт до теми №2**

**Умовний перехід**

**Завдання №1**

Функція пошуку коренів квадратного рівняння



Функція "find\_quadratic\_roots" отримує три коефіцієнти a, b і c, які вводяться користувачем. Функція розраховує дискримінант за допомогою формули b \*\* 2 - 4 \* a \* c.

Після обчислення дискримінанта функція використовує умовні переходи для визначення коренів квадратного рівняння залежно від значення дискримінанта:

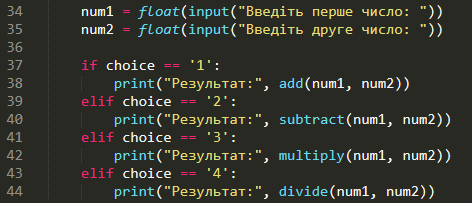
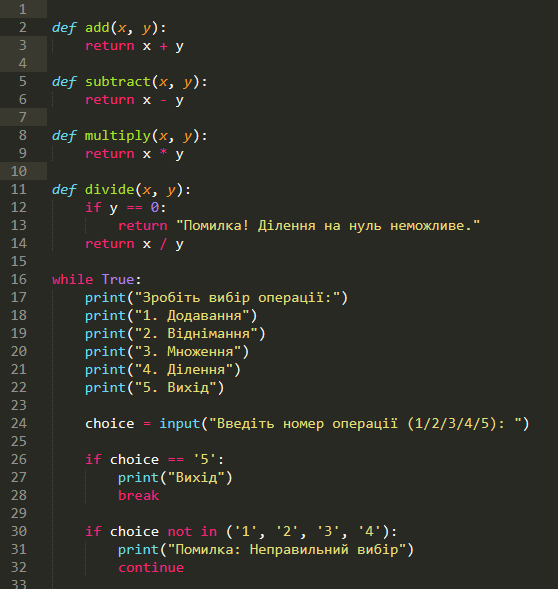
Якщо дискримінант більше 0, це означає, що рівняння має два різних корені. Функція використовує формулу квадратного кореня для знаходження цих двох коренів.

Якщо дискримінант рівний 0, це означає, що рівняння має один корінь (корені рівні між собою).

Якщо дискримінант менше 0, це означає, що рівняння не має реальних коренів.

**Завдання №2**

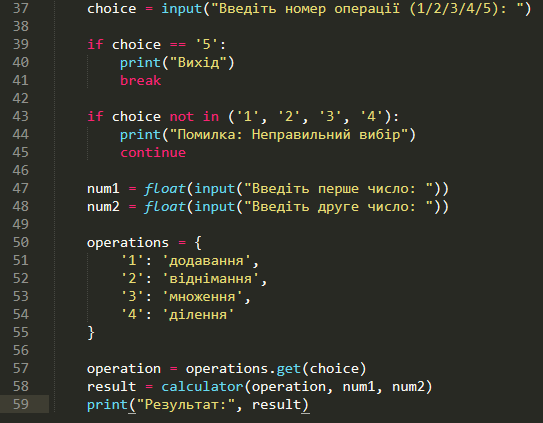
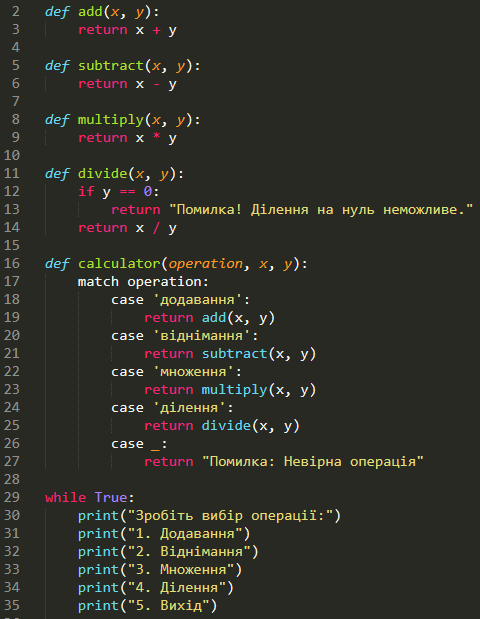
Програма калькулятор, з використанням if else конструкції.



У цій програмі кожна операція виконується у відповідній функції (add(), subtract(), multiply(), divide()), і операція вибирається за допомогою конструкції if-else. Користувач може вибрати операцію, ввести два числа, і програма виведе результат відповідної операції. Цикл while True використовується для надання користувачеві можливості вибору інших операцій або виходу з програми.

**Завдання №3**

Програма калькулятор, з використанням match конструкції



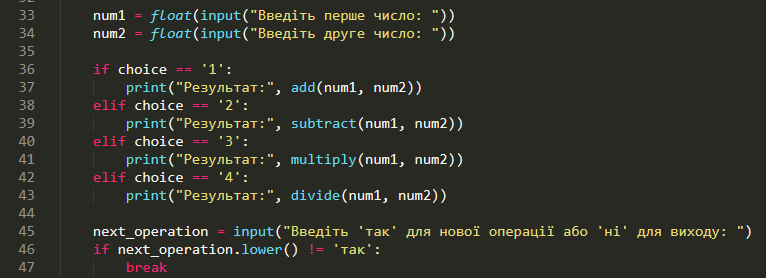
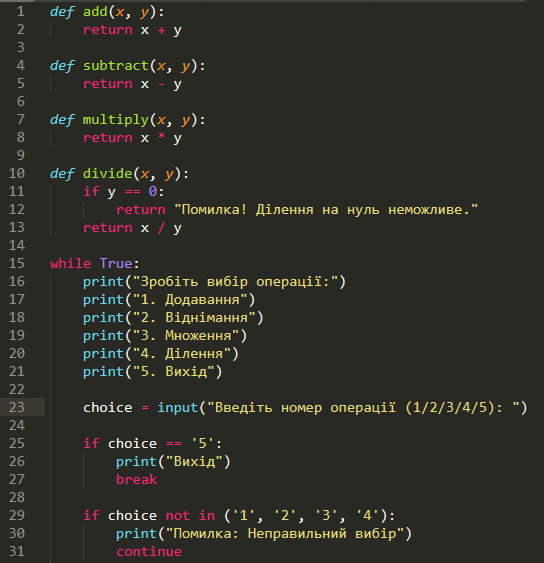
У цій програмі ми використовуємо конструкцію match для вибору операції відповідно до введеного користувачем номера. Кожна операція виконується в окремій функції, яка визначена вище. Після виконання операції програма виводить результат. Цикл while True дозволяє користувачеві вибирати інші операції або виходити з програми.

**Звіт до теми №3**

**Цикли**

**Завдання №1**

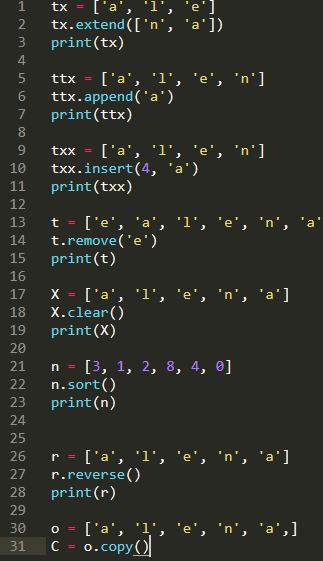
Програма калькулятор



У цій версії програми після виконання операції користувачу пропонується вибрати, чи виконувати ще операції. Він може ввести 'да' для нової операції або 'ні' для виходу. Це дозволить користувачеві виконувати декілька операцій підряд, не перезапускаючи програму кожного разу.

**Завдання №2**

Тестування та ознайомлення з extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy()



extend() дозволяє додати до списку всі елементи з іншого ітерабельного об'єкта, такого як інший список або кортеж.

append() додає вказаний елемент до кінця списку.

insert() вставляє вказаний елемент на певну позицію у списку за допомогою індексу.

remove() видаляє перший елемент із списку, який має задане значення.

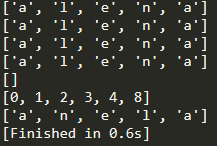
clear() видаляє всі елементи зі списку, роблячи його пустим.

sort() сортує елементи списку за зростанням (по замовчуванню) або відповідно до іншого ключа, якщо вказаний.

reverse() змінює порядок елементів у списку на протилежний.

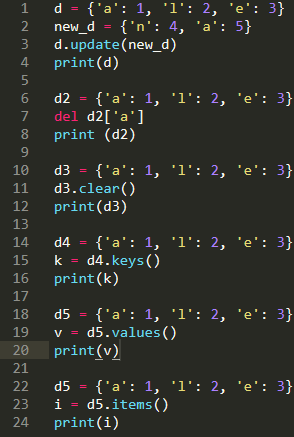
copy() створює копію списку. Ця операція не змінює оригінальний список.

Результат:



**Завдання №3**

Тестування та ознайомлення з update(), del, clear, keys(), values(), items()



update() дозволяє додавати або оновлювати елементи в словнику, використовуючи інші словники або послідовності пар "ключ-значення".

del використовується для видалення елементів зі словника за ключем.

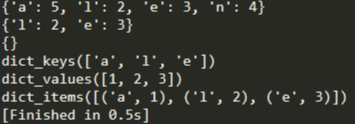
clear() видаляє всі елементи із словника, роблячи його пустим.

keys() повертає список ключів у словнику.

values() повертає список значень у словнику.

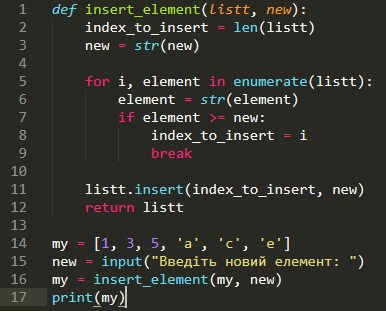
items() повертає список кортежів, кожен з яких містить пару "ключ-значення" зі словника.

Результат:



**Завдання №4**

Написати функцію пошуку позиції для вставки нового елементу в список



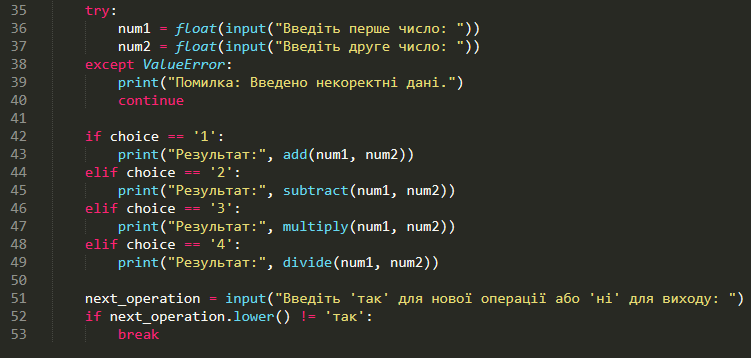
Функція приймає список listt та новий елемент new. Визначається індекс, на який потрібно вставити новий елемент. Цей індекс спочатку встановлюється як довжина списку та означає, що новий елемент додається в кінець списку за замовчуванням. Новий елемент перетворюється в строковий тип. Функція проходиться циклом for по елементам списку listt, кожен елемент також перетворюється в строковий тип. В кожній ітерації циклу функція порівнює елементи. Якщо елемент списку element більший або рівний за значенням нового елемента new, то зберігається індекс цього елемента, і цикл припиняється за допомогою break. Після того, як був знайдений відповідний індекс для вставки, функція використовує метод insert() для додавання нового елемента на визначену позицію в списку listt. Нарешті, функція повертає оновлений список listt з вставленим новим елементом.

**Звіт до теми №4**

**Виняткові ситуації**



Я додала до програми конструкцію try-except, яка відловлює помилки, що можуть виникнути при спробі перетворити введені дані користувача на числа. Якщо введені дані не можна перетворити в числа (наприклад, якщо користувач вводить букви замість чисел), програма виведе повідомлення про помилку і попросить користувача ввести дані знову. Також використовується конструкція try-except для обробки можливих винятків, які можуть виникнути при діленні на нуль.



Також ознайомилася з винятками ось деякі зних:

ZeroDivisionError: Це виняток, який виникає, коли ви намагаєтеся поділити на нуль.

TypeError: Виняток, який виникає, коли операція несумісна з типом даних.

ValueError: Виняток, який виникає, коли функція отримує аргумент правильного типу, але недопустимого значення.

FileNotFoundError: Цей виняток виникає, коли програма намагається відкрити файл, який не існує.

KeyError: Виняток, який виникає, коли ключ не знайдено в словнику.

IndexError: Виняток, який виникає, коли індекс виходить за межі діапазону.

NameError: Це виняток, який виникає, коли ви використовуєте змінну або ім'я, яке не було оголошено.

AssertionError: Виняток, який виникає, коли вираз assert повертає False.

EOFError: Виняток, який виникає, коли досягнуто кінця файлу і програма намагається зчитати більше даних.

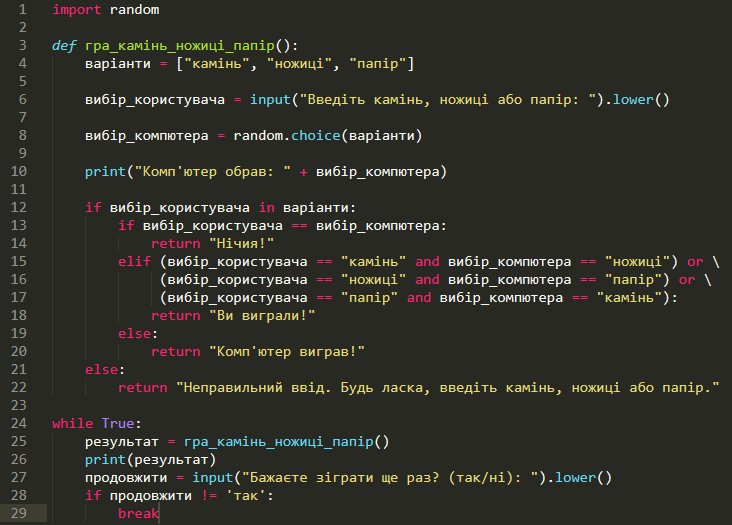
KeyboardInterrupt: Це виняток, який виникає, коли користувач вводить комбінацію клавіш Ctrl+C для переривання виконання програми.

**Звіт до теми №5**

**Бібліотеки**

**Завдання №1**

Гра "камінь, ножиці, папір"



Ця програма працює в нескінченному циклі, дозволяючи користувачеві грати доти, доки він не вирішить припинити гру. Користувач вводить свій вибір ("камінь", "ножиці" або "папір"), програма обирає випадковий варіант для комп'ютера, а потім порівнює їх, щоб визначити переможця чи нічию. Після кожної гри програма запитує користувача, чи він хоче зіграти ще раз. Якщо користувач вводить "ні", програма завершує роботу.

**Завдання №2**

Конвертор валют



Результат:



**Завдання №3**

Модулі для програми калькулятор



Результат:

