

Урок 42. Символьний калькулятор

Вступна частина	Вітаємо в IT-книзі! На минулих уроках ми навчилися шифрувати та розшифровувати шифр Цезаря із заданим ключем. А якщо ключ невідомий? Чи можна тоді зашифрований текст розшифрувати? У часи Давнього Риму це вважалося неможливим, а сьогодні є доволі простою задачею, яку ми розв'яжемо на наступних трьох уроках. Але для цього потрібно рахувати, скільки разів трапляється в тексті той чи інший символ. Цим ми займемося сьогодні. Сьогодні ви зможете виконати 10 завдань на 12 балів у сумі. Якщо підсумковий результат вас не задовольнить, то наприкінці уроку ви зможете виконати додаткове завдання, отримавши додатково 1 бал. Практичні вправи потрібно виконати в середовищі Python. Їх позначено олівцем. 
------------------------	--

Вивчення нового матеріалу

Слайд №1	Коли для зашифрованого повідомлення є ключ, то розшифрувати його не становить труднощів. 
Слайд №2	У цьому випадку звертаються до так званого частотного криптоаналізу. Його принцип такий: 1) визначаємо частоти появи кожного символу в шифрованому тексті; 2) символам шифру ставимо у відповідність літери мови, що мають схожі частоти. Найкращий спосіб розшифрувати шифр Цезаря – це визначити, який символ у ньому зустрічається найчастіше. Найвірогідніше йому відповідатиме пробіл у початковому тексті, оскільки саме пробіл між словами в українському тексті трапляється частіше за всі інші символи. Різниця між кодом символу, що трапляється у шифрі найчастіше, і кодом пробілу й буде ключем. Криптоаналіз – наука про розшифрування повідомлень. Криптографія – наука про шифрування.

Слайд №3

Проте в короткому тексті найуживанішим символом може бути не пробіл, а деяка літера.

У таблиці оранжевим виділено літери, які найчастіше зустрічаються у текстах українською мовою.

а	0,072	ї	0,006	у	0,04
б	0,017	й	0,008	ф	0,001
в	0,052	к	0,035	х	0,012
г	0,016	л	0,036	ц	0,006
д	0,035	м	0,031	ч	0,018
е	0,017	н	0,065	ш	0,012
є	0,008	о	0,094	щ	0,001
ж	0,009	п	0,029	ь	0,029
з	0,023	р	0,047	ю	0,004
и	0,061	с	0,041	я	0,029
і	0,057	т	0,055	пробіл	0,17

Справа біля кожної літери вказано її частоту. Ця величина показує, яка частина тексту припадає на цю літеру.

Слайд №4

Сьогодні ми зробимо **перший крок у частотному криптоаналізі** — створимо програму символьного калькулятора. У ньому користувач вводитиме текст, а програма рахуватиме, скільки разів у цьому тексті трапляється кожна літера.

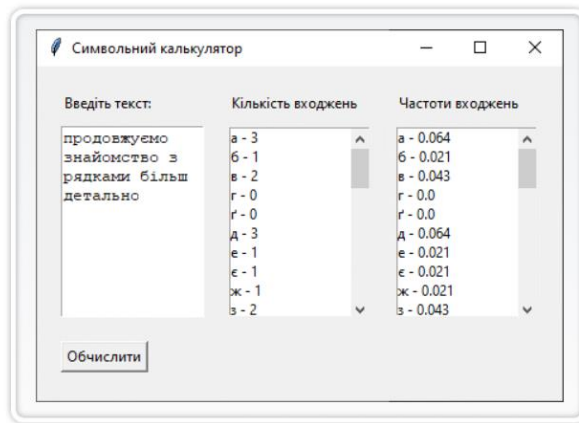
Обчислюватимемо також **відносну частоту** — кількість входжень символу, поділену на загальну кількість символів.

Наприклад, кількість символів "а" = 3, усього символів у тексті – 47, отже відносна частота входжень даного символу = $3/47 = 0,064$.

Слайд №5

Розглянемо, як працюватиме програма.

Створимо спочатку інтерфейс програми.



Для вхідного тексту скористаємося багаторядковим текстовим полем, а для підрахунку кількості літер та їхніх частот будемо використовувати списки.

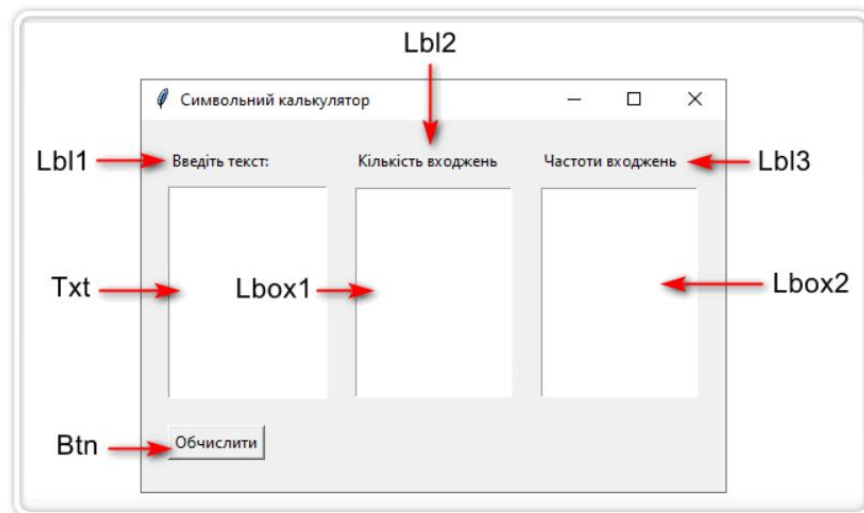
Вправа

Вправа №1



Вправа 1. Створіть інтерфейс вікна програми відповідно до зразка.

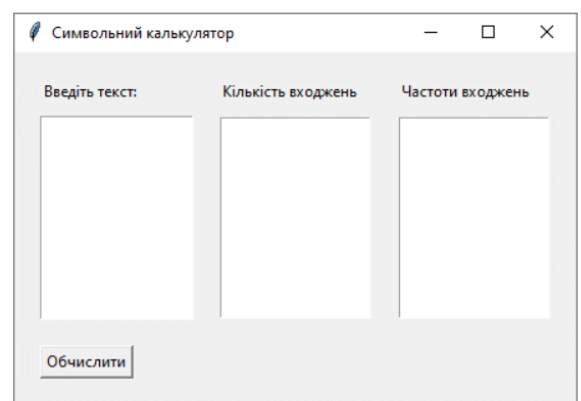
Надайте елементам керування імена, що вказані на малюнку.



Підказка до вправи №1

Підказка до вправи 1.

```
#створення графічного вікна
from tkinter import*
tk = Tk()
tk.title("Символьний калькулятор")
tk.geometry("440x280")
```



Підказка до вправи №1

Підказка до вправи 1.

```
Lbl1=Label(text="Введіть текст:")
Lbl1.place(x=20, y=20)

Txt=Text()
Txt.place(x=20, y=50, width=120, height=160)

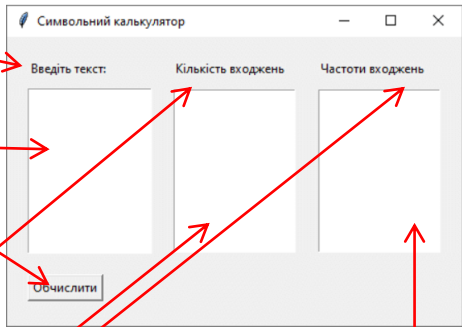
Btn=Button(text="Обчислити")
Btn.place(x=20, y=230)

Lbl2=Label(text="Кількість входжень")
Lbl2.place(x=160, y=20)

Lbox1=Listbox()
Lbox1.place(x=160, y=50, height=160, width=120)

Lbl3=Label(text="Частоти входжень")
Lbl3.place(x=300, y=20)

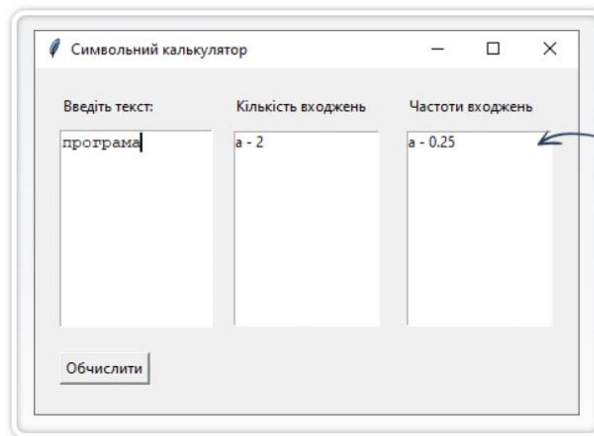
Lbox2=Listbox()
Lbox2.place(x=300, y=50, height=160, width=120)
```



Вивчення нового матеріалу

Слайд №7

Спочатку підрахуємо кількість входжень до введеного поля літери "а".



Частота – це кількість входжень у текст певної літери (для літери **а** – 2), поділена на загальну кількість літер у тексті (у даному прикладі – 8).

$$2/8 = 0.25$$

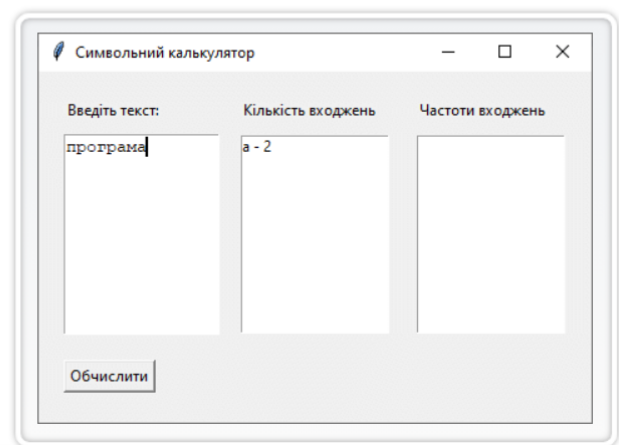
Вправа

Вправа №2



Вправа 2. Запрограмуйте обробник події натискання кнопки **Обчислити** так, щоб він підраховував кількість входжень до тексту літери "а".

Зауважте, що в команді перевірки умови літеру "а" необхідно вводити на українській розкладці клавіатури.



Підказка до вправи 2.

Команди, які необхідно додати до обробника події:

- 1) надання змінній **x** значення текстового поля **Txt**;
- 2) вилучення недрукованого символу "**\n**" зі змінної **x**;
- 3) очищення списку **Lbox1**;
- 4) визначення **k** кількості входжень літери "**a**";
- 5) виведення знайденого значення до списку **Lbox1**.

Вивчення нового матеріалу

Слайд №8

Під час введення великого тексту у багаторядковому текстовому полі текст автоматично переноситься на новий рядок, незалежно від того, чи закінчилося чергове слово. Для налаштування місця перенесення тексту використовують властивість **wrap**, що може набувати таких значень:

CHAR – перенесення на будь-якому символі (за умовчанням)

WORD – перенесення після слова

NONE – текст на новий рядок не переноситься

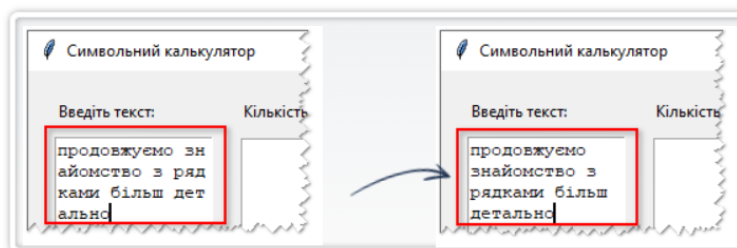
Вправа

Вправа №3



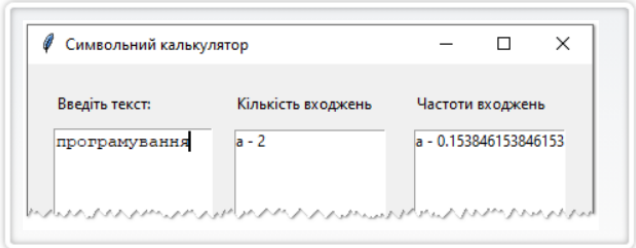
Вправа 3. Додайте до багаторядкового текстового поля властивість **wrap** з відповідним значенням, щоб перенесення на новий рядок відбувалося між словами.

У конструкторі поля допишіть **wrap=WORD**.



Слайд №9	Для обчислення довжини текстового рядка використовують функцію len(). Наприклад, після виконання коду <pre>length=len("моя програма")</pre> змінна length одержить значення 12.
----------	---

Вправа

Вправа №4	 Вправа 4. Додайте команди обчислення та виведення до списку Lbox2 частоти входжень літери "a" у текст. k – це кількість літер "a", x – текст. Самостійно сформулюйте та додайте команду очищення списку LBox2. 
Підказка до вправи №4	Підказка до вправи 4. Щоб очистити список Lbox2, скопіюйте команду очищення списку Lbox1 та замініть у ній Lbox1 на Lbox2. Команду відображення даних у списку Lbox2 також можна сформувати на основі аналогічної команди для списку Lbox1.