

Вступна частина

Вітаємо в IT-книзі!

Символьний калькулятор — це важливий інструмент криптоаналізу, тобто розшифрування текстів, і сьогодні ми завершимо його розробку. Поки що він у нас рахує кількість входжень і частоту символу «а», а на цьому уроці ми розширимо програму так, щоб вона рахувала кількість входжень і частоту всіх символів, які є в тексті.

Сьогодні вам буде запропоновано 12 завдань, кожне з яких оцінюється 1 балом.

Якщо підсумковий результат вас не задовольнить, то наприкінці уроку ви зможете виконати додаткове завдання, отримавши додатково 1 бал.

Вправи, які потрібно виконати в середовищі Python, позначено олівцем.
Крім того, наприкінці уроку буде практична робота —
змагання на швидкість розшифрування текстів!

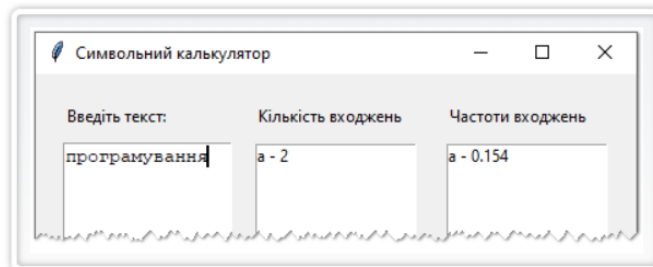


Вправа

Вправа №1



Вправа 1. Додайте до команди виведення частоти входжень літери **"а"** функцію округлення дробової частини до трьох знаків після коми.



Слайд №1

Підрахуємо кількість та частоти входжень до тексту всіх маленьких літер українського алфавіту.

Введіть текст:	Кількість входжень	Частоти входжень
продовжуємо	а - 3	а - 0.064
знайомство з	б - 1	б - 0.021
рядками більш	в - 2	в - 0.043
детально	г - 0	г - 0.0
	д - 3	д - 0.064
	е - 1	е - 0.021
	є - 1	є - 0.021
	ж - 1	ж - 0.021
	з - 2	з - 0.043

Обчислити

Слайд №2

Для визначення кількості входжень кожної літери алфавіту програмі необхідно вказати, які літери вона буде підраховувати. Для цього можна створити рядок, що міститиме послідовність літер українського алфавіту та символ пробілу:

```
alphabet="абвггдеежзіїйклмнопрстуфхццшщьюя "
```

Слайд №3

Отже, загальний вигляд алгоритму визначення кількості входжень кожної літери та пробілу буде таким:

```
Пошук кількості входжень літери "а"
Пошук кількості входжень літери "б"
...
Пошук кількості входжень літери "я"
Пошук кількості входжень пробілу " "
```

Літери "а", "б", ..., "я" та символ " " є послідовними елементами рядка **alphabet**. Кожний із цих символів можна перебрати за допомогою циклу

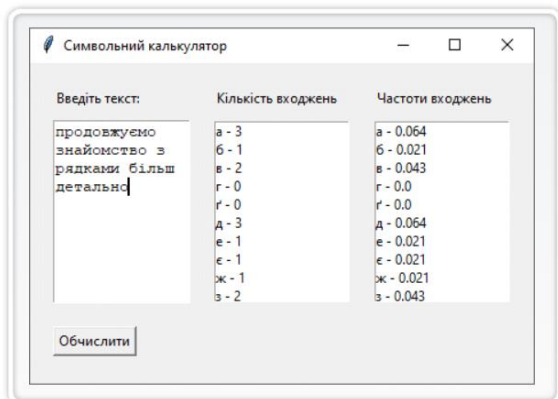
```
for j in alphabet:
```

Вправа

Вправа №2



Вправа 2. Змініть програму так, щоб виконувався пошук кількостей та частот входжень до тексту усіх маленьких літер українського алфавіту та пробілу.



Підказка до вправи №2

Підказка до вправи 2.

У код необхідно внести такі зміни:

- 1) додати список **alphabet="абвґґеєжзіїйклмнопрстуфхцщьюя"**
- 2) додати зовнішній цикл **for j in alphabet:**
- 3) цикл знаходження кількості літер **"а"** та частоти її входження до тексту розташувати у тілі зовнішнього циклу
- 4) у внутрішньому циклі замінити умову на **i==j**
- 5) у командах виведення значень до списків літеру **"а"** замінити змінною **j**

Вивчення нового матеріалу

Слайд №4

Щоб списки було зручніше переглядати, додамо до них смуги прокручування. Конструктор смуг прокручування називається **Scrollbar**.

Ось код для додавання вертикальної смуги прокручування у списку **Lbox1**:

#Створення смуги прокручування

```
Scroll1 = Scrollbar(Lbox1, command=Lbox1.yview)
```

Команда прив'язки до об'єкта смуги прокручування по осі **y**

Об'єкт, для якого встановлюється смуга прокручування

#Розташування смуги прокручування

```
Scroll1.place(x=100,height=160)
```

Координати вказано відносно верхнього лівого кута об'єкта **Lbox1**

#Прив'язка створеної смуги прокручування до списку Lbox1

```
Lbox1.config(yscrollcommand=Scroll1.set)
```

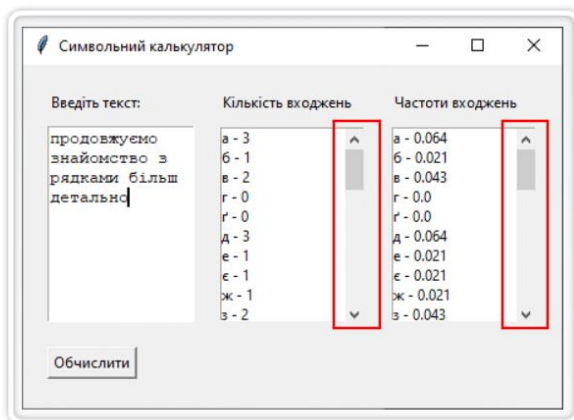
Опція встановлення смуги прокручування

Ім'я смуги прокручування

Вправа №3


Вправа 3. Додайте команди створення смуг прокручування для списків Lbox1 та Lbox2.

Запустіть програму та перевірте роботу смуг прокручування.



Підказка до вправи №3

Підказка до вправи 3.

#Створення смуги прокручування

```
Scroll1 = Scrollbar(Lbox1, command=Lbox1.yview)
```

Команда прив'язки до об'єкта смуги прокручування по осі **y**

Об'єкт, для якого встановлюється смуга прокручування

#Розташування смуги прокручування

```
Scroll1.place(x=100,height=160)
```

Координати вказано відносно верхнього лівого кута об'єкта **Lbox1**

#Прив'язка створеної смуги прокручування до списку Lbox1

```
Lbox1.config(yscrollcommand=Scroll1.set)
```

Опція встановлення смуги прокручування

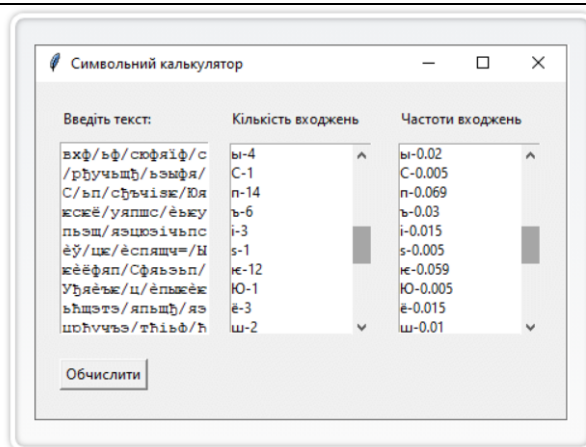
Ім'я смуги прокручування

Вивчення нового матеріалу

Слайд №5

Наш символічний калькулятор рахує частоти літер в українському тексті. Однак зашифрований текст не є українським! Подивіться на приклад: багатьох символів із зашифрованого повідомлення немає в українському алфавіті.

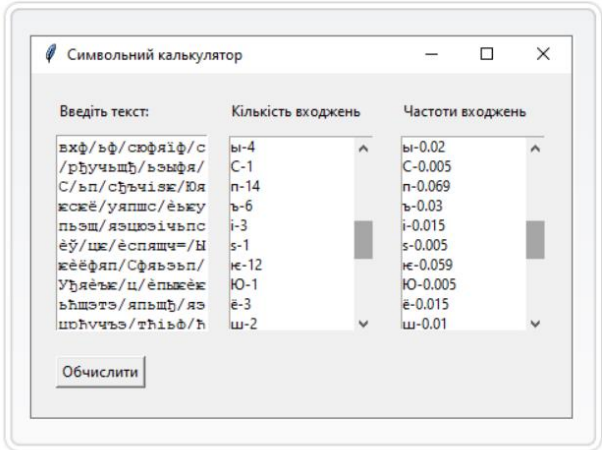
Щоб символічний калькулятор допоміг у криптоаналізі, він має рахувати частоти саме тих символів, які є у вхідному тексті.





Вправа 4. Забезпечте обчислення кількості входжень та частот саме тих символів, з яких складається текст.

Для цього замість присвоєння `alphabet="абвггдеєжзиіїйклмнопрстуфхцчшщьюя "` потрібно сформувати рядок `alphabet` за розглянутим вище алгоритмом.



Підказка до вправи №4

Алгоритм формування рядка `alphabet`:

- 1. Рядковій змінній `x` присвоїти текст, а `alphabet` — порожній рядок.
- 2. Для кожного символу в рядку `x` виконати крок 3.
- 3. Якщо символ не належить рядку `alphabet`, додати його до рядка `alphabet`.



Практична робота. Змагання з криптоаналізу

Варіант змагання 1

Знайдіть в Інтернеті будь-який текст українською мовою (не менше 2 речень), виберіть ключ (число від 1 до 15), зашифруйте цей текст за допомогою програми **Шифр Цезаря**, створеної на уроці 27, і надішліть комусь з однокласників для криптоаналізу. Домовтеся, щоб одночасно однокласники надішлють зашифрований текст вам.

Варіант змагання 2

Зашифрований текст публікує вчитель.

У змаганні перемагає той, хто першим розшифрує текст.

Для розшифрування

- 1) За допомогою **Символьного калькулятора** і таблиці символів знайдіть ключ.
- 2) Перевірте, чи розшифровується текст цим ключем, у програмі **Шифр Цезаря**.

Таблиця символів

символ	пробіл	!	“	#	\$	%	&	'
код	32	33	34	35	36	37	38	39
символ	(	)	*	+	,	-	.	/
код	40	41	42	43	44	45	46	47