Лабораторна робота 2

Тема: Рядки та опрацювання тексту.

Мета: Навчитися використовувати індекси, зрізи і методи рядків, розбирати структуровані записи та формувати текстові результати.

Зміст роботи

Завдання 1. Індекси та зрізи

Введіть слово зі щонайменше шести літер, без пробілів. За допомогою індексів, зрізів і len() виведіть:

1. Довжину слова, першу і останню літери.
2. Перші три і останні три літери.
3. Символи з індексами 0, 2, 4, … та слово у зворотному порядку.

У всіх завданнях цієї роботи дані коректні. Цикли, власні функції і обробка винятків не потрібні. Перевірте наведений приклад і власне слово. У контрольних таблицях наведено приклади виведення. Підписи результатів можна обрати самостійно. Наведені значення мають збігатися з результатами програми.

Таблиця 1 — Контрольна перевірка індексів і зрізів

Вхідне слово	**Приклад виведення**	**Що перевіряється**
програмування	Довжина слова: 13 Перша літера: п Остання літера: я Перші три літери: про Останні три літери: ння Символи з індексами 0, 2, 4, …: пормвня Слово у зворотному порядку: яннавумаргорп	Усі операції
python	Довжина слова: 6 Перша літера: p Остання літера: n Перші три літери: pyt Останні три літери: hon Символи з індексами 0, 2, 4, …: pto Слово у зворотному порядку: nohtyp	Мінімальна допустима довжина

Завдання 2. Нормалізація, пошук і заміна

Введіть окремо речення, шуканий фрагмент, непорожній після strip(), текст заміни та невід'ємне ціле m — максимальну кількість замін. Заміна може бути порожньою.

1. Застосуйте strip() і lower() до трьох текстових рядків.
2. Виведіть очищене речення, кількість входжень фрагмента через count() та індекс першого входження через find().
3. Через replace() замініть не більше ніж m входжень зліва направо. Виведіть результат, його довжину і різницю з довжиною очищеного речення.

Пошук виконуйте також усередині слів. Внутрішні пробіли збережіть. Результат заміни повторно не очищуйте. У контрольній таблиці лапки позначають межі тексту і показують крайні пробіли. Лапки не є частиною введення або результату. Заміна «» означає введення порожнього рядка. Перевірте власні дані, m = 0, відсутній фрагмент і порожню заміну.

Таблиця 2 — Контрольна перевірка пошуку і заміни

Вхідні дані	**Приклад виведення**	**Що перевіряється**
Речення: « КОД код » Фрагмент: «код» Заміна: «мова» Максимальна кількість замін (m): 1	Очищене речення: «код код» Кількість входжень: 2 Індекс першого входження: 0 Результат заміни: «мова код» Довжина результату: 8 Зміна довжини: +1	Одна заміна
Речення: « КОД код » Фрагмент: «код» Заміна: «мова» Максимальна кількість замін (m): 0	Очищене речення: «код код» Кількість входжень: 2 Індекс першого входження: 0 Результат заміни: «код код» Довжина результату: 7 Зміна довжини: 0	Без замін
Речення: « КОД код » Фрагмент: «код» Заміна: «мова» Максимальна кількість замін (m): 5	Очищене речення: «код код» Кількість входжень: 2 Індекс першого входження: 0 Результат заміни: «мова мова» Довжина результату: 9 Зміна довжини: +2	Ліміт більший за кількість входжень

Вхідні дані	Приклад виведення	Що перевіряється
Речення: « КОД код » Фрагмент: «дані» Заміна: «мова» Максимальна кількість замін (m): 1	Очищене речення: «код код» Кількість входжень: 0 Індекс першого входження: -1 Результат заміни: «код код» Довжина результату: 7 Зміна довжини: 0	Немає фрагмента
Речення: « КОД код » Фрагмент: «код» Заміна: «» Максимальна кількість замін (m): 1	Очищене речення: «код код» Кількість входжень: 2 Індекс першого входження: 0 Результат заміни: « код» Довжина результату: 4 Зміна довжини: -3	Видалення без повторного strip()

Завдання 3. Розбір структурованого запису

Введіть запис за форматом свого варіанта таблиці 3. Він містить рівно три непорожні поля; роздільник не входить до самих полів. Крайні пробіли дозволено.

1. Розділіть запис і приберіть крайні пробіли кожного поля.
2. Нормалізуйте поля та сформуйте два результати за правилом варіанта.
3. Виведіть обидва результати і довжину кожного.

Числові позначення залишайте рядками, зберігаючи початкові нулі. Назви та імена складаються з літер і пробілів. Артикули, серії та введені коди — з латинських літер або цифр. Позначення відділу, групи і залу мають вигляд «літери-номер» з латинськими або кириличними літерами. Домен — латинські літери, цифри, крапки і дефіси, без пробілів. Перевірте контрольний запис і власні дані. У всій роботі використовуйте один номер варіанта. У контрольному стовпці розмежовано вхідний запис і приклад виведення. Для кожного результату його довжину зазначено окремим рядком.

Вхідний запис вводьте одним рядком за форматом, наведеним у колонці «Умова та потрібні результати».

Таблиця 3 — Розбір записів і формування результатів

Варіант	Умова та потрібні результати	Методи і операції	Контрольний приклад
1	**Запис:** прізвище;ім’я;група Підпис «Прізвище І. (ГРУПА)». Код — перші 3 літери прізвища і номер групи через «-».	Імена — title() Група, код — upper() Номер групи — split("-")	**Вхід:** коваль ; олена ; кн-02 **Виведення:** Підпис: Коваль О. (КН-02) Довжина підпису: 17 Код: КОВ-02 Довжина коду: 6
2	**Запис:** назва;артикул;колір Підпис «назва / колір». Код — артикул і перші 3 літери кольору через «_».	Назва, колір — lower() Артикул, код — upper() Колір — [:3]	**Вхід:** ЧАШКА ; ab007 ; СИНІЙ **Виведення:** Підпис: чашка / синій Довжина підпису: 13 Код: AB007_СИН Довжина коду: 9
3	**Запис:** місто|вулиця|номер Адреса «Місто, Вулиця, номер». Код — перші 2 літери міста і номер через «-».	Назви — title() Код — upper() Місто — [:2]	**Вхід:** львів | зелена | 007 **Виведення:** Адреса: Львів, Зелена, 007 Довжина адреси: 18 Код: ЛЬ-007 Довжина коду: 6
4	**Запис:** прізвище|ім’я|відділ Підпис «Ім’я Прізвище — ВІДДІЛ». Код — ініціали імені і прізвища та номер відділу через «_».	Імена — title() Відділ, ініціали — upper() Ініціали — [0] Номер — split("-")	**Вхід:** мельник | іван | it-03 **Виведення:** Підпис: Іван Мельник — IT-03 Довжина підпису: 20 Код: ІМ_03 Довжина коду: 5
5	**Запис:** країна;місто;індекс Адреса «індекс Місто, Країна». Код — перші 3 літери міста і індекс через «-».	Назви — title() Код — lower() Місто — [:3]	**Вхід:** україна ; київ ; 01001 **Виведення:** Адреса: 01001 Київ, Україна Довжина адреси: 19 Код: киї-01001 Довжина коду: 9
6	**Запис:** назва|серія|номер Підпис «Назва [СЕРІЯ-номер]». Код — номер та останні 2 літери серії через «/».	Назва — title() Серія — upper(), [-2:]	**Вхід:** практикум | pyab | 004 **Виведення:** Підпис: Практикум [PYAB-004] Довжина підпису: 20 Код: 004/AB Довжина коду: 6

Варіант	Умова та потрібні результати	Методи і операції	Контрольний приклад
7	**Запис:** ім’я;прізвище;домен Підпис «Прізвище, Ім’я». Адреса — перша літера імені, крапка, прізвище та @домен. Домен без пробілів.	Імена — title() Адреса — lower() Ініціал — [0]	**Вхід:** ОЛЕНА ; КОВАЛЬ ; EXAMPLE.ORG **Виведення:** Підпис: Коваль, Олена Довжина підпису: 13 Адреса: о.коваль@example.org Довжина адреси: 20
8	**Запис:** курс|група|номер Підпис «Курс — ГРУПА (номер)». Код — літери групи, номер групи і номер запису через «_».	Курс — title() Група — upper(), split("-")	**Вхід:** python | кн-02 | 005 **Виведення:** Підпис: Python — КН-02 (005) Довжина підпису: 20 Код: КН_02_005 Довжина коду: 9
9	**Запис:** книга;автор;рік Підпис «Автор: Книга (рік)». Код — перші 3 літери автора і останні 2 цифри року через «-». Рік — 4 цифри.	Назви — title() Код — upper() Автор — [:3] Рік — [-2:]	**Вхід:** алгоритми ; коваль ; 2026 **Виведення:** Підпис: Коваль: Алгоритми (2026) Довжина підпису: 24 Код: КОВ-26 Довжина коду: 6
10	**Запис:** подія|зал|місце Підпис «Подія / ЗАЛ / місце». Код — зал без дефіса та місце через «_».	Подія — title() Зал — upper() Код залу — replace("-", "")	**Вхід:** семінар | b-02 | 007 **Виведення:** Підпис: Семінар / B-02 / 007 Довжина підпису: 20 Код: B02_007 Довжина коду: 7
11	**Запис:** пристрій;код;кімната Підпис «пристрій (кімната)». Новий код — перші 2 і останні 2 символи коду, «-», кімната. Введений код має ≥ 4 символів.	Назва — lower() Код — upper(), [:2], [-2:]	**Вхід:** Ноутбук ; ab1234 ; 009 **Виведення:** Підпис: ноутбук (009) Довжина підпису: 13 Код: AB34-009 Довжина коду: 8
12	**Запис:** ім’я|позивний|команда Підпис «Ім’я [позивний]». Код — команда і позивний без внутрішніх пробілів через «-».	Ім’я — title() Позивний — lower(), replace(" ", "") Команда — upper()	**Вхід:** анна | blue fox | team **Виведення:** Підпис: Анна [blue fox] Довжина підпису: 15 Код: TEAM-bluefox Довжина коду: 12

Варіант	Умова та потрібні результати	Методи і операції	Контрольний приклад
13	**Запис:** назва;розмір;колір Підпис «Назва, РОЗМІР, колір». Код — перші 3 літери назви і розмір через «_».	Назва — title(), [:3] Розмір, код — upper() Колір — lower()	**Вхід:** сорочка ; xl ; білий **Виведення:** Підпис: Сорочка, XL, білий Довжина підпису: 18 Код: СОР_XL Довжина коду: 6
14	**Запис:** прізвище|група|рік Підпис «Прізвище — ГРУПА / рік». Код — літери прізвища з парними індексами і номер групи через «-». Рік — 4 цифри.	Прізвище — title(), [::2] Група, код — upper() Номер групи — split("-")	**Вхід:** бондар | кн-04 | 2026 **Виведення:** Підпис: Бондар — КН-04 / 2026 Довжина підпису: 21 Код: БНА-04 Довжина коду: 6
15	**Запис:** місто;станція;код Підпис «Станція (Місто)». Новий код — перші 2 літери міста, перші 2 літери станції та введений код через «-».	Назви — title(), [:2] Код — upper()	**Вхід:** київ ; дарниця ; r007 **Виведення:** Підпис: Дарниця (Київ) Довжина підпису: 14 Код: КИ-ДА-R007 Довжина коду: 10

Завдання 4. Перетворення короткого тексту

Введіть рівно три слова зі щонайменше трьох літер кожне, без розділових знаків. Переведіть текст у нижній регістр і розділіть через split(). Позначення слово_1, слово_2, слово_3 відповідають першому, другому і третьому словам.

1. Утворіть новий рядок за правилом таблиці 4, обов’язково застосувавши join().
2. Виведіть цей рядок, його довжину та кількість входжень літери «а».

Перевірте власні слова та повторні крайні і внутрішні пробіли. Спільні контрольні слова — «мова python дані». У таблиці наведено сформований рядок, його довжину та кількість входжень літери «а».

Рахуйте саме малу українську літеру «а» у сформованому рядку, без повторної зміни його регістру. Велика літера «А» не враховується. Методи і зрізи варіанта наведено окремо.

Таблиця 4 — Варіанти перетворення трьох слів

Варіант	Що сформувати	Методи і операції	Приклад виведення
1	З’єднайте слова у зворотному порядку через « / ».	Зворотний порядок слів.	Рядок: «дані / python / мова» Довжина: 20 Кількість літер «а»: 2
2	Виведіть слово_2, слово_1, слово_3 через «-»; слово_2 — у верхньому регістрі.	upper() для слово_2.	Рядок: «PYTHON-мова-дані» Довжина: 16 Кількість літер «а»: 2
3	Перші 2 літери кожного слова через «.».	Зріз [:2].	Рядок: «мо.ру.да» Довжина: 8 Кількість літер «а»: 1
4	Останні 2 літери слів слово_3, слово_2, слово_1 через «_».	Зріз [-2:].	Рядок: «ні_on_ва» Довжина: 8 Кількість літер «а»: 1
5	Кожне слово у зворотному порядку, з’єднати через пробіл.	Зріз [::-1] для кожного слова.	Рядок: «авом nohtyp інад» Довжина: 16 Кількість літер «а»: 2
6	Слова слово_1 і слово_3 — title(), між ними слово_2 у нижньому регістрі; роздільник « | ».	title() для слово_1 і слово_3.	Рядок: «Мова | python | Дані» Довжина: 20 Кількість літер «а»: 2
7	Початкові літери слів слово_3, слово_1, слово_2 у верхньому регістрі через «-».	Індекс [0], upper().	Рядок: «Д-М-Р» Довжина: 5 Кількість літер «а»: 0
8	З’єднайте слово_1, слово_3, слово_2 через « / »; у слові слово_2 замініть «о» на «0».	replace("о", "0") для слово_2.	Рядок: «мова / дані / pyth0n» Довжина: 20 Кількість літер «а»: 2
9	Символи кожного слова з індексами 0, 2, 4, … через «+».	Зріз [::2].	Рядок: «мв+рто+дн» Довжина: 9 Кількість літер «а»: 0
10	Виведіть слово_3 у верхньому регістрі, слово_2 у зворотному порядку і слово_1 через «:».	upper(), зріз [::-1].	Рядок: «ДАНІ:nohtyp:мова» Довжина: 16 Кількість літер «а»: 1
11	Першу і останню літери кожного слова разом; три пари через «-».	Індекси [0] і [-1].	Рядок: «ма-рn-ді» Довжина: 8 Кількість літер «а»: 1
12	Слова без першої літери в порядку слово_2, слово_3, слово_1 через « / ».	Зріз [1:].	Рядок: «ython / ані / ова» Довжина: 17 Кількість літер «а»: 2
13	Слова слово_1, слово_2, слово_3 без останньої літери через «_», увесь результат — upper().	Зріз [:-1], upper().	Рядок: «МОВ_РУТНО_ДАН» Довжина: 13 Кількість літер «а»: 0

Варіант	Що сформувати	Методи і операції	Приклад виведення
14	Подайте слово_1 у зворотному порядку, слово_3 у нижньому, слово_2 — title(); через «; ».	Зріз [::-1], title().	Рядок: «авом; дані; Python» Довжина: 18 Кількість літер «а»: 2
15	Перші 3 літери слів слово_2, слово_1, слово_3 у верхньому регістрі через « / ».	Зріз [:3], upper().	Рядок: «PYT / МОВ / ДАН» Довжина: 15 Кількість літер «а»: 0

Оформлення звіту

Звіт оформлюють за чинними вимогами курсу (документ «Вимоги до оформлення»).

Для кожного завдання у звіті подають формулювання, текст програми та скріншот результату в консолі.

Методичні рекомендації

Індекси, зрізи та незмінність рядків

Рядок str — послідовність символів. Індекси починаються з 0: "мова"[0] дає "м", а [-1] — "а". len("мова") дорівнює 4. Рядок незмінний: методи і зрізи утворюють новий результат, який потрібно присвоїти змінній.

Зріз text[start:stop:step] не включає stop. Для "python" зріз [:3] дає "pyt", [-3:] — "hon", [::2] — "pto", [::-1] — "nohtyp". Пропущені межі задають початок або кінець; зріз поза межами безпечний, окремий неправильний індекс — ні.

Індекс і межі зрізу

Код Python

```
word = "дані"
print(word[0], word[-1])
print(word[:2], word[-2:])
print(word[1::2], word[::-1])
```

Результат виконання

```
д і
да ні
аі інад
```

Індекс 0 вибирає першу літеру, −1 — останню. Зріз [1::2] починається з індексу 1. Нові рядки не змінюють word. У завданні оберіть свої межі та кроки.

Нормалізація, пошук та заміна

strip() прибирає пробільні символи лише з країв; lower() та upper() змінюють регістр, title() починає кожне слово з великої літери. " код дані ".strip() дає "код дані" — два внутрішні пробіли збережено.

count(fragment) рахує входження без перекриття. find(fragment) повертає індекс першого або −1, коли входжень немає. "код код".count("код") дає 2, find("дані") — −1. replace(old, new, m) замінює не більше ніж m входжень. Для m = 0 рядок не змінюється; порожня заміна видаляє фрагмент. Зміну довжини обчислюйте як len(result) − len(original).

Метод повертає новий рядок

Код Python

```
raw = "  МОВА  Дані  "
text = raw.strip().lower()
print(repr(text))
print(text.upper())
print(text.title())
print(repr(raw))
```

Результат виконання

```
'мова  дані'
МОВА  ДАНІ
Мова  Дані
'  МОВА  Дані  '
```

repr() показує лапки і крайні пробіли. strip() прибрав лише краї, а два внутрішні пробіли збережено. upper() і title() повернули нові значення; raw залишився початковим.

Входження, індекс і обмежена заміна

Код Python

```
text = "мова мова"
print(text.count("мова"), text.find("мова"))
print(text.find("дані"))
result = text.replace("мова", "код", 1)
print(result)
print(len(result) - len(text))
```

Результат виконання

```
2 0
-1
код мова
-1
```

Індекс 0 означає знайдений фрагмент на початку, −1 — його відсутність. Одна заміна скоротила рядок на один символ. Ліміт 0 залишив би текст незмінним.

Розділення полів і складання результату

split(";") розділяє за конкретним символом: "А;Б;007".split(";") дає ["А", "Б", "007"]. Присвоєння first, second, third = parts розпаковує рівно три елементи. Крайні пробіли кожного поля приберіть окремо. "КН-02".split("-") дає ["КН", "02"]. Перетворення "02" на int знищить початковий нуль, тому позначення зберігайте рядками.

split() без аргументу об’єднує повторні пробільні роздільники: " код дані ".split() дає ["код", "дані"]. Метод separator.join(parts) з’єднує рядки: " / ".join(("А", "Б")) дає "А / Б". Роздільник записано перед join(), усі частини мають бути рядками. Для короткої фіксованої кількості слів достатньо розпакування; цикли не потрібні.

Для пояснення результату можна використати f-рядок: f"{name} ({code})". Враховуйте всі пробіли і розділові знаки у len(). Ініціал отримують як name[0], а для додавання крапки використовують name[0] + ".".

Розпакування та збереження початкових нулів

Код Python

```
left, right, number = " МОВА ; Код ; 007 ".split(";")
left = left.strip().title()
right = right.strip().lower()
number = number.strip()
print(left, right, number)
print(len(number))
```

Результат виконання

```
Мова код 007
3
```

Кількість змінних має дорівнювати кількості полів. Кожне поле очищується окремо. Рядок 007 зберігає три символи; перетворення на int дало б число 7.

split() і join() виконують різні дії

Код Python

```
words = "  код   мова  дані ".split()
print(words)
result = " / ".join(words[:2])
print(result)
print("КН-02".split("-"))
```

Результат виконання

```
['код', 'мова', 'дані']
код / мова
['КН', '02']
```

split() без аргументу розділяє за пробільними символами і не створює порожніх слів. join() з'єднує вибрані частини своїм роздільником; split("-") виділяє частини позначення групи.

Часті помилки

Не плутайте find() з перевіркою кількості: індекс 0 означає успішний пошук на початку. Не застосовуйте strip() повторно після видалення фрагмента, якщо крайній пробіл є частиною результату. Не перетворюйте коди на числа і не підміняйте join() циклом, що ще не вивчено.

Приклади виконання завдань

Приклад до завдання 1

Для слова утворимо позначення з крайніх літер та покажемо середину і символи з непарними індексами.

Код Python

```
word = input("Введіть слово: ")
short = word[0] + word[-1]
middle = word[1:-1]
print(f"Коротке позначення: {short}")
print(f"Без крайніх літер: {middle}")
print(f"Довжина середньої частини: {len(middle)}")
print(f"Символи з індексами 1, 3, 5, ...: {word[1::2]}")
```

Приклад виконання в консолі

```
Введіть слово: навчання
Коротке позначення: ня
Без крайніх літер: авчанн
Довжина середньої частини: 6
Символи з індексами 1, 3, 5, ...: ачня
```

Пояснення до прикладу

Індекси 0 і −1 вибирають крайні літери. Зріз [1:-1] залишає середину, а [1::2] починає з індексу 1 і пересувається через один символ.

Приклад до завдання 2

Перетворимо позначення теми з підкресленнями на заголовок; водночас визначимо першу частину і кількість частин. Введіть щонайменше дві непорожні частини, розділені одиночними підкресленнями. Крайні пробіли дозволені.

Код Python

```
raw = input("Позначення теми: ")
topic = raw.strip()
separator = topic.find("_")
short = topic[:separator].upper()
title = topic.replace("_", " ").title()
parts = topic.count("_") + 1
print(f"Коротке позначення: {short}")
print(f"Назва теми: {title}")
print(f"Кількість частин: {parts}")
```

Приклад виконання в консолі

```
Позначення теми:   python_рядки_методи
Коротке позначення: PYTHON
Назва теми: Python Рядки Методи
Кількість частин: 3
```

Пояснення до прикладу

strip() прибирає крайні пробіли, тому topic містить python_рядки_методи. find("_") повертає індекс 6. Зріз topic[:separator] вибирає python без підкреслення, а upper() утворює PYTHON.

replace("_", " ") замінює підкреслення пробілами, а title() утворює назву Python Рядки Методи. У рядку два підкреслення, тому topic.count("_") + 1 дає 3 частини.

Приклад до завдання 3

Для складського запису товар:полиця:номер утворимо підпис і коротку позначку. Тут інший роздільник та інші правила, ніж у таблиці варіантів.

Код Python

```
record = input("Товар:полиця:номер: ")
item, shelf, number = record.split(":")
item = item.strip().lower()
shelf = shelf.strip().upper()
number = number.strip()
caption = f"{shelf} -> {item}"
mark = f"{number}/{item[-2:].upper()}"
print(f"Підпис: {caption}; довжина {len(caption)}")
print(f"Позначка: {mark}; довжина {len(mark)}")
```

Приклад виконання в консолі

```
Товар:полиця:номер:  ЛАМПА : b03 : 007
Підпис: B03 -> лампа; довжина 12
Позначка: 007/ПА; довжина 6
```

Пояснення до прикладу

split(":") розділяє введений запис на три поля, які розпаковуються в item, shelf і number. Після strip() та зміни регістру вони містять лампа, B03 і 007. Номер зберігається рядком, тому його початкові нулі не зникають.

f-рядок caption утворює B03 -> лампа, а item[-2:].upper() дає ПА для позначки 007/ПА. Довжина підпису становить 12: 3 символи B03, 2 символи ->, 2 пробіли й 5 літер слова лампа. Довжина позначки — 6: три цифри, символ / та дві літери.

Приклад до завдання 4

Із чотирьох слів утворимо рядок лише з двома середніми словами, а окремо — поєднаємо крайні.

Код Python

```
words = input("Чотири слова: ").lower().split()
middle = " / ".join(words[1:3])
edges = " + ".join((words[0].title(), words[-1].title()))
print(f"Середина: {middle}; довжина {len(middle)}")
print(f"Крайні слова: {edges}")
```

Приклад виконання в консолі

```
Чотири слова: Я код МОВА дані
Середина: код / мова; довжина 10
Крайні слова: Я + Дані
```

Пояснення до прикладу

split() нормалізує розділення, а зріз [1:3] містить друге і третє слова. join() не додає роздільника ні перед першою частиною, ні після останньої.

Контрольні питання

1. Як відрізняються індекси 0 і −1?
2. Чи входить stop до зрізу?
3. Що дають [::2] та [::-1]?
4. Чим split() відрізняється від split(";")?
5. Які пробіли прибирає strip()?
6. Що означають результати find() 0 та −1?
7. Як працює count() за перекриття входжень?
8. Як обмежити кількість замін і видалити фрагмент?
9. Як працює join() і які типи частин він приймає?
10. Чому код із початковими нулями потрібно залишати рядком?