

Національний технічний університет України «Київський  
політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

## **Програмування**

### **Лабораторна робота №3**

«Робота з даними типу str, bytes та bytearray»

Виконав:

студент групи ІО-41

*Давидчук А. М.*

Залікова книжка № 4108

Перевірив

*Пономаренко А. М.*

**Тема:** «Робота з даними типу str, bytes та bytearray».

**Мета:** Вивчити способи створення рядків та даних типу bytes і bytearray, операції над ними. Форматування рядків. Функції та методи роботи з рядками. Налаштування локалі.

**Загальне завдання:** Написати 2 програми, які вирішують 2 індивідуальні задачі.

### **Короткі теоретичні відомості:**

str – тип даних “рядок” для відображення тексту;

bytes – незмінний тип даних “байти”, являє собою незмінний байтовий масив;

bytearray – тип даних “байтовий масив”, являє собою змінний байтовий масив;

True – тип даних “Boolean” для представлення істинності, правдивості;

a1 – змінна, здатна набувати/посилатись на різноманітні типи даних;

del – операція видалення;

ord() – вбудована функція виводу десяткового представлення літери;

len() – вбудована функція виводу кількості елементів певного масиву;

range() – вбудований генератор чисел в заданому проміжку;

print() – вбудована функція виводу;

input() – вбудована функція вводу;

if, elif, else – оператори умовності, що дозволяють створювати різноманітні блоки коду та логічні конструкції;

while – оператор циклу, що дозволяє виконувати блок циклу більше одного разів;

for – оператор циклу, кількість циклічних повторень якого дорівнює кількості елементів масиву;

+, +=, == – операції конкатенації (додавання рядків), конкатенації-присвоєння та порівняння відповідно;

### **Завдання №1**

8	Задати два однакові за довжиною рядки. Побудувати новий рядок, в якому на парних місцях розташовані елементи першого рядка, а на непарних – елементи другого рядка.
---	---

### **Опис алгоритму:**

1. Отримати два рядки та присвоїти значення кожній певній змінній;
2. Перевірити кількість символів кожного рядка: в разі однакової кількості створити новий пустий результуючий рядок та почати цикл (крок 3), інакше – сповістити про те, що рядки мають різну кількість символів та завершити програму;
3. Проводимо конкатенацію результуючого рядка та елементів при індексі “i” (0 при першому разі) з першого та переписуємо результуючий рядок на щойно сформований;
4. До значення “i” додаємо одиницю. Якщо “i” > (кількість елементів початкового рядка - 1), то завершуємо цикл та переходимо до кроку 5, інакше – повертаємось до кроку 3;
5. Виводимо результуючий рядок та завершуємо програму.

### Код:

```
a = input("Введіть перший рядок: ") #Одержання першого рядка
b = input("Введіть другий рядок: ") #Одержання другого рядка

if len(a) == len(b): #Перевірка їх розмірності

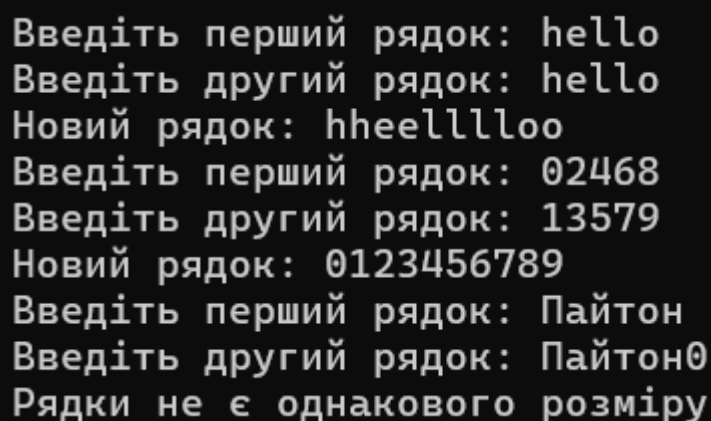
    new_string = "" #Результуючий рядок

    #Цикл обходу кожного елемента з 1 та 2 рядків, їх конкатенація-присвоєння з
    минулим рядком:
    for i in range(len(a)):
        new_string += a[i] + b[i]

    print(f"Новий рядок: {new_string}")

else:
    print("Рядки не є однакового розміру")
```

### Знімок екрана контрольного прикладу:



```
Введіть перший рядок: hello
Введіть другий рядок: hello
Новий рядок: hheelllloo
Введіть перший рядок: 02468
Введіть другий рядок: 13579
Новий рядок: 0123456789
Введіть перший рядок: Пайтон
Введіть другий рядок: Пайтон0
Рядки не є однакового розміру
```

### Завдання №2:

8	1.Ввести послідовність символів. 2.Перетворити її, видаливши кожен символ *, і повторивши кожен символ, відмінний від *.
---	---

### Опис алгоритму:

1. Отримуємо рядок, та перетворюємо його в тип “bytearray” кодуючи кожен символ за UTF-8. Присвоюємо значенню “i” (індекс масиву) нулю;
2. Якщо “i” менше за довжину байтового масиву, переходимо до кроку 3, інакше – до кроку 4;
3. Порівнюємо десяткове представлення елемента байтового масиву з індексом “i” з десятковим представленням знаку “\*”. Якщо числа збігаються: видаляємо даний елемент з масиву і повертаємось до кроку 2. Якщо числа не збігаються, то вставляємо копію цього елемента поруч (на місце “i”+1), та додаємо-присвоюємо значення “i” на 2 і переходимо до кроку 2;
4. Виводимо модифікований байтовий масив та завершуємо програму.

## Код:

```
#Одержую та перетворюю його на bytearray з кодуванням UTF-8:
byte_array = bytearray(input("Введіть рядок: "), encoding="utf-8")

i = 0 #Початкове значення індексу цього масиву

#Цикл, який реалізує повне проходження всіх елементів в байтовому масиві:
while i < len(byte_array):

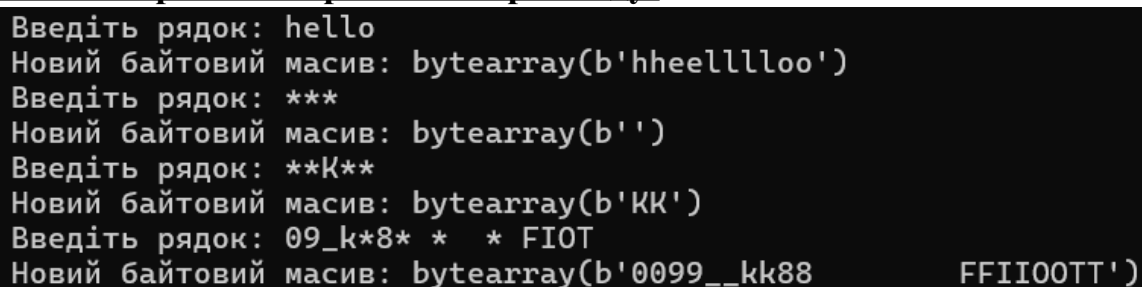
    if byte_array[i] == ord("*"): #Знаходжу символ "*" якщо такий існує
        del byte_array[i] #Видаляю його. Змінювати індекс немає сенсу, адже він вже
        вказує на наступний елемент

    else:

        #Вставляю символ на місце i+1, тим самим не видаляючи елемент, який
        знаходився на місці i+1:
        byte_array.insert(i+1, byte_array[i])
        i += 2

print(f"Новий байтовий масив: {byte_array}")
```

## Знімок екрана контрольного прикладу:



```
Введіть рядок: hello
Новий байтовий масив: bytearray(b'hheelllloo')
Введіть рядок: ***
Новий байтовий масив: bytearray(b'')
Введіть рядок: **K**
Новий байтовий масив: bytearray(b'KK')
Введіть рядок: 09_k*8* * * FIOT
Новий байтовий масив: bytearray(b'0099__kk88      FFII00TT')
```

## Висновок:

Я більш розглянув типи даних str, bytes та bytearray, а також операції між ними. Створив дві програми, які забезпечують ввід та вивід результатів обчислень відповідно до мого індивідуального завдання. Програми працюють коректно. Набув глибших навичок розуміння різноманіття типів даних та особливостей їх використання та реалізації.