

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Програмування

Лабораторна робота №3

«Робота з даними типу str, bytes та bytearray»

Виконав:
студент групи ІО-32
Крадожон М. Р.
Номер у списку групи: 16
Перевірив:
Пономаренко А. М.

Лабораторна робота №3

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:

1. Вивчити матеріал лекцій 7, 8, 9 та 10.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Короткі теоретичні основи:

Створення рядка указавши його між апострофами або подвійними лапками.

```
>>> print ('рядок1\nрядок2')  
      \n – перевід рядка.
```

Створення об'єкта типу bytearray

```
bytearray([<Рядок >,<Кодування> [,<Обробка помилок>]])
```

Методи bytearray

append(<Число>) – додає один елемент у кінець об'єкта

decode() – перетворює об'єкт типу bytearray в рядок

```
decode([encoding="utf-8"] [, errors="strict"])
```

Метод format()

```
<Рядок>=<Рядок спеціального формату>.format(*args, **kwargs)
```

У параметрі усередині символів фігурних дужок: { i } вказуються

специфікатори, що мають наступний синтаксис:

```
{[<Поле>] [!< Функція>] [:< Формат>]}
```

```
>>> print("Символи {{ i }} - {0}".format("спеціальні"))
```

Символи { i } – спеціальні

У параметрі <Формат> вказується значення, що має наступний синтаксис:

```
[[<Заповнювач>] <Вирівнювання>] [<Знак>] [#] [0] [<Ширина>]  
[,][.<Точність>] [<Перетворення>]
```

За замовчуванням значення усередині поля вирівнюється по правому краю.

Управляти вирівнюванням дозволяє параметр <Вирівнювання>.

Функція len(<Рядок>) – повертає кількість символів в рядку.

Завдання 1:

Відповідно до номера в списку групи вибрати індивідуальне завдання. Написати програму на мові Python . Забезпечити ввід даних з клавіатури комп'ютера та друк результатів обчислень. У звіті до лабораторної роботи описати алгоритм, за яким побудована програма. При виводі даних обов'язково використати форматування.

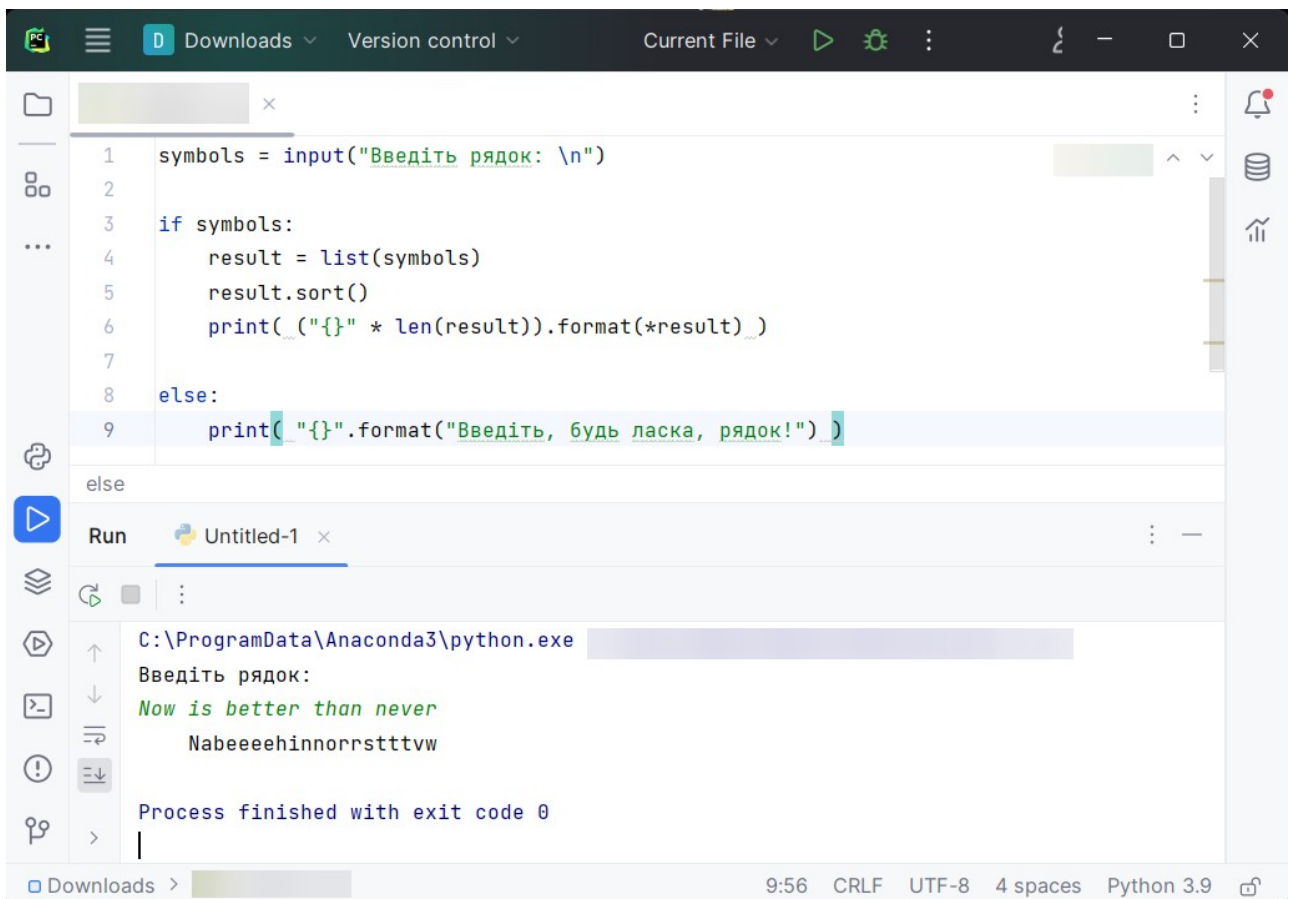
16	Ввести рядок. Розташувати всі його символи по зростанню їх кодів.
----	---

Роздруківка коду:

```
1 symbols = input("Введіть рядок: \n")
2
3 if symbols:
4     result = list(symbols)
5     result.sort()
6     print( ("{}" * len(result)).format(*result) )
7
8 else:
9     print( "{}".format("Введіть, будь ласка, рядок!") )
```

Код має відмінний від типового для цього документа шрифт для зручного читання.

Знімок екрана тексту програми:



The screenshot shows a code editor with the following Python code:

```
1 symbols = input("Введіть рядок: \n")
2
3 if symbols:
4     result = list(symbols)
5     result.sort()
6     print( ("{}" * len(result)).format(*result) )
7
8 else:
9     print( "{}".format("Введіть, будь ласка, рядок!") )
```

The output of the program is displayed in the console:

```
Введіть рядок:
Now is better than never
Nabeeeehinnorrstttvw

Process finished with exit code 0
```

Алгоритм: Користувач вводить будь-яку фразу чи будь-що в рядок. У тому випадку, коли рядок непустий, то програма перетворює його у тип `list` і сортує його; згодом друкується сортований рядок за допомогою форматування; якщо рядок порожній, то програма виводить повідомлення про те, що немає що розташовувати.

Завдання 2:

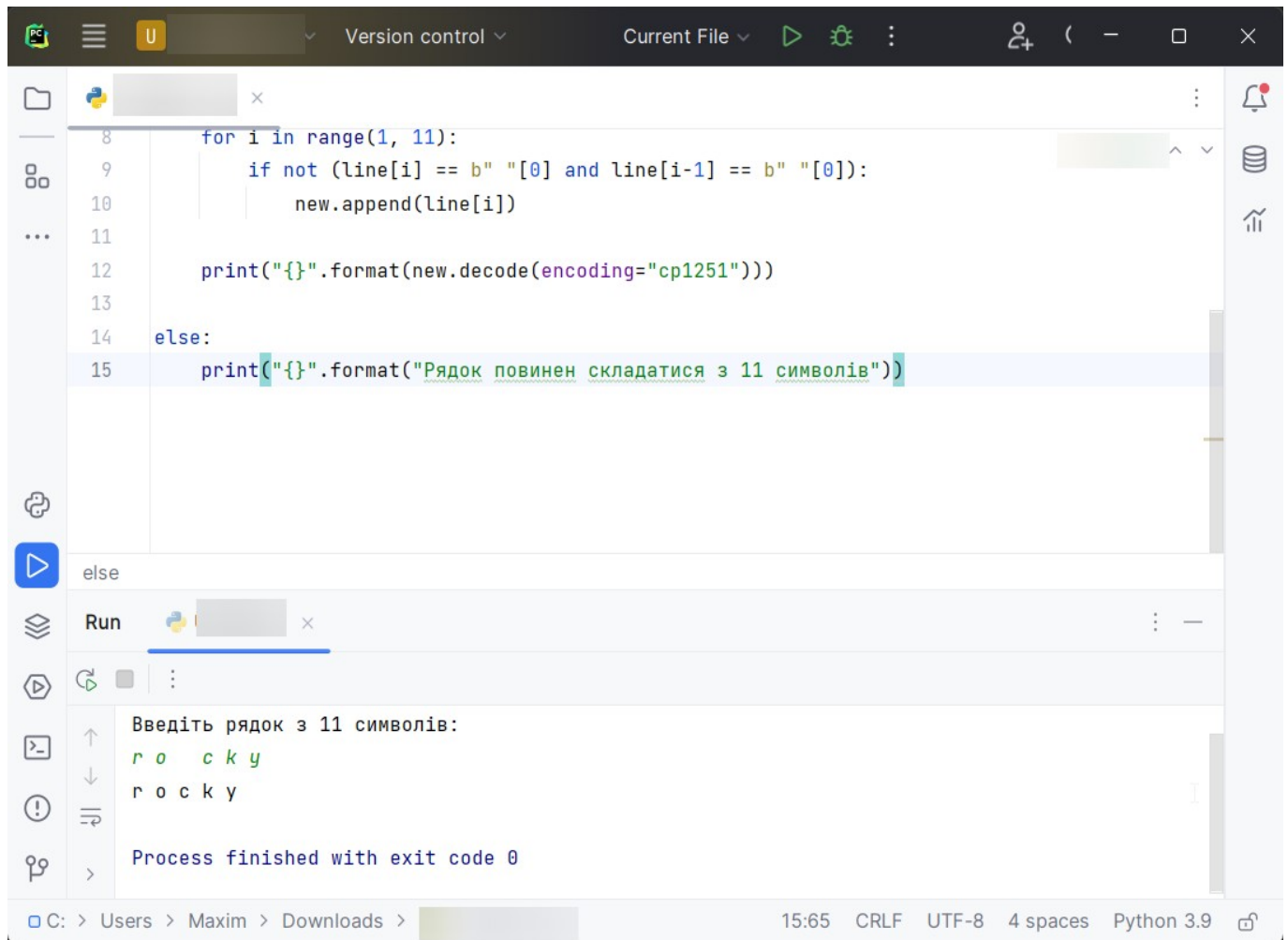
Відповідно до номера в списку групи вибрати індивідуальне завдання. Написати програму на мові Python з використанням типів даних `byte` та `bytearray`. Забезпечити ввід даних з клавіатури комп'ютера та друк результатів обчислень. У звіті до лабораторної роботи описати алгоритм, за яким побудована програма. При виводі даних обов'язково використати форматування.

16	1.Ввести послідовність символів з 11 елементів. 2.Прибрати зайві пробіли (Більше одного поспіль).
17	1.Ввести 5 послідовностей символів довжиною 8 елементів

Роздруківка коду:

```
1 line = input("Введіть рядок з 11 символів:\n")
2
3 if len(line) == 11:
4     line = bytearray(line, "cp1251")
5     new = bytearray()
6     new.append(line[0])
7
8     for i in range(1, 11):
9         if not (line[i] == b" "[0] and line[i-1] == b"
"[0]):
10             new.append(line[i])
11
12     print("{}".format(new.decode(encoding="cp1251")))
13
14 else:
15     print("{}".format("Рядок повинен складатися з 11
символів"))
```

Знімок екрана тексту програми:



The screenshot shows a code editor with a dark theme. The main editor area displays Python code with line numbers 8 to 15. Line 15 is highlighted. Below the editor is a 'Run' panel showing the execution output. The output indicates that the program finished with exit code 0. The status bar at the bottom shows the file path, time, and encoding.

```
8     for i in range(1, 11):
9         if not (line[i] == b" "[0] and line[i-1] == b" "[0]):
10             new.append(line[i])
11
12     print("{}".format(new.decode(encoding="cp1251")))
13
14 else:
15     print("{}".format("Рядок повинен складатися з 11 символів"))
```

Run

Введіть рядок з 11 символів:
r o c k y
r o c k y

Process finished with exit code 0

C: > Users > Maxim > Downloads > 15:65 CRLF UTF-8 4 spaces Python 3.9

Алгоритм побудови програми: Користувач вводить слово 11 символів, використовуючи два та більше пробілів. Рядок повинен бути строго 11 символів, інакше програма не запрацює. Потім програма перетворює рядок у тип `bytearray`, після цього створюється нова змінна «new», також типу `bytearray`, де додається рядок, який було введено користувачем. Далі у new додається елементи з рядку, якщо ті та попередні до них не були пробілами, проте якщо один з елементів усе ж таки є пробілом, то також додає поточний елемент; і так робиться для елементів від одного до десяти. Наприкінці new перетворюється у `str` та друкується рядок за допомогою форматування.

Висновок: Виконавши цю лабораторну роботу, я зміг здобути відповідні навички в роботі з даними типу `str`, `bytes` та `bytearray`. Під час виконання лабораторної роботи проблем не виникало, а складність була в структуруванні коду та приведенні його до більш гарного вигляду.