

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІКТА кафедра БІТ



**РОБОТА З БІБЛІОТЕКОЮ NUMPY та  
РЕГУЛЯРНІ ВИРАЗИ**

Звіт до лабораторної роботи № 2  
з навчальної дисципліни: “Програмування скриптовими мовами.”  
Варіант №6

виконав:  
студент групи КБ-212  
Зінько А.В.  
перевірив:  
Гасілін Д.Л.

Львів-2025

**Мета роботи.** навчитися використовувати функції, методи та типи даних бібліотеки NumPy. Навчитися використовувати регулярні вирази з модуля re для опрацювання текстової інформації

**Хід роботи. (Бібліотека NumPy)**

Написати і протестувати операції над матрицями NumPy згідно завдань з таблиці 1. Текст кожного завдання повинен виводитися на початку у вигляді рядка, після чого виводяться вхідні дані та результат виконання завдання.

Варіант	Номери завдань
6	2, 13, 15, 18, 21, 35, 46, 50, 55, 63

**Код програми:**

```
def matricesTask():
    import numpy as np
    # Створити матрицю 4x4 з заданим типом елементів і значеннями та вивести інформацію
    # про матрицю (розмірність, тип елементів, розмір елементу масиву в байтах, розмір масиву
    # в байтах) та саму матрицю:

    # 2. Тип: uint64, всі 3 - в першому рядку, всі 2 - в другому рядку і т. д.
    a = np.array([[3]*4,
                  [2]*4,
                  [1]*4,
                  [0]*4], dtype='uint64')

    print('Матриця 4x4 із елементами за умовою: ')
    print(a)
    print(f'Розмірність матриці: {a.shape} (рядки, стовпці)')
    print(f'Тип даних матриці: {a.dtype}')
    print(f'Розмір матриці: {a.size} байтів')
    print(f'Розмір елементу масиву в байтах: {a.itemsize}')

    # Вивести з допомогою операцій індексації, зміни розмірностей та зрізів:

    # 13. Непарні стовпці.
    print('Непарні стовпці:')
    print(a[:,0::2])

    # 15. Кожен третій рядок.
    print('Кожен третій рядок:')
    print(a[2::3,:])

    # 18. Елементи в непарних стовпцях у непарних рядках.
    print('Елементи в непарних стовпцях у непарних рядках:')
    print(a[0::2,0::2])

    # 21. Елементи в парних рядках менші за 2.
    print('Елементи в парних рядках менші за 2:')
    print(a[1::2]<2)

    # 35. Реверсувати порядок елементів у стовпцях.
    print('Реверсувати порядок елементів у стовпцях:')
    print(a[:,::-1,:])

    # 46. Підрахувати кількість непарних елементів у матриці.
    print('Кількість непарних елементів у матриці:')
```

```

odd= 0
for i in range(a.shape[0]):
    for j in range(a.shape[0]):
        if a[i,j]%2 == 1:
            odd +=1
print(odd)

# Знайти і вивести для матриці:

# 50. Максимальні значення по стовпцях.
print('Максимальні значення по стовпцях:')
for j in range(a.shape[0]):
    print(a[:,j].max(), end=', ')

# 55. Середнє значення.
print('\nСереднє значення матриці: ')
print(a.sum()/a.size)

# 63. Розв'язати задану систему лінійних рівнянь і показати, що розв'язок вірний:
# [2  3  7][x1]   [ -2 ]
# [11 4  9][x2] = [ 13 ]
# [17 1 -3][x3]   [ 6  ]

array_coefficients = np.array([
    [2,3,7],
    [11,4,9],
    [17,1,-3],
])
array_constants = np.array([-2,13,6])
solution = np.linalg.solve(array_coefficients,array_constants)
print(f"""Розв'язок системи лінійних рівнянь
    [2x1 + 3x2 + 7x3 = -2
    [11x1 + 4x2 + 9x3 = 13
    [17x1 + x2 + -3x3 = 6

    (x1, x2, x3 відповідно): {solution}""")
print('Доведення правильності розв'язку')
print(int(2*solution[0] + 3*solution[1] + 7*solution[2]))
print(int(11*solution[0] + 4*solution[1] + 9*solution[2]))
print(int(17*solution[0] + 1*solution[1] + -3*solution[2]))

matricesTask()

```

### Результат виконання програми:

```
PS A:\Microsoft VS Code> & C:\Users\Dream\AppData\Local\Programs\Python\Python313\python.exe
Матриця 4x4 із елементами за умовою:
[[3 3 3 3]
 [2 2 2 2]
 [1 1 1 1]
 [0 0 0 0]]
Розмірність матриці: (4, 4) (рядки, стовпці)
Тип даних матриці: uint64
Розмір матриці: 16 байтів
Розмір елементу масиву в байтах: 8
Непарні стовпці:
[[3 3]
 [2 2]
 [1 1]
 [0 0]]
Кожен третій рядок:
[[1 1 1 1]]
Елементи в непарних стовпцях у непарних рядках:
[[3 3]
 [1 1]]
Елементи в парних рядках менші за 2:
[[False False False False]
 [ True True True True]]
Реверсувати порядок елементів у стовпцях:
[[0 0 0 0]
 [1 1 1 1]
 [2 2 2 2]
 [3 3 3 3]]
Кількість непарних елементів у матриці:
8
Максимальні значення по стовпцях:
3, 3, 3, 3,
Середнє значення матриці:
1.5
Розв'язок системи лінійних рівнянь
      2x1 + 3x2 + 7x3 = -2
     11x1 + 4x2 + 9x3 = 13
     17x1 + x2 + -3x3 = 6

      (x1, x2, x3 відповідно): [ 2.08547009 -14.04273504  5.13675214]
Доведення правильності розв'язку
-2
13
6
PS A:\Microsoft VS Code> █
```

### Хід роботи. (Бібліотека re)

Написати і протестувати на декількох рядках регулярні вирази (РВ) для завдань з таблиці 1. Тестові рядки для перевірки регулярних виразів повинні показати, що РВ коректно працюють та відсіюють невірні варіанти.

Варіант	Номери завдань
6	3, 40

№3. Написати РВ для знаходження всіх слів у тексті, які закінчуються на літеру 'e'.

**Код програми:**

```
import re
import random as r
# №3. Написати РВ для знаходження всіх слів у тексті, які закінчуються на літеру 'e'.

string3 = '''Monosodium glutamate (MSG), also known as sodium glutamate, is a sodium
salt of glutamic acid.
MSG is found naturally in some foods including tomatoes and cheese in this glutamic
acid form.
MSG is used in cooking as a flavor enhancer with a savory taste that intensifies the
umami flavor of food,
as naturally occurring glutamate does in foods such as stews and meat soups.
MSG was first prepared in 1908 by Japanese biochemist Kikunae Ikeda, who tried to
isolate and duplicate
the savory taste of kombu, an edible seaweed used as a broth (dashi) ingredient in
Japanese cuisine.
MSG balances, blends, and rounds the perception of other tastes. MSG, along with disodium
ribonucleotides,
is commonly used and found in stock (bouillon) cubes, soups, ramen, gravy, stews,
condiments, savory snacks, etc.
The U.S. Food and Drug Administration has given MSG its generally recognized as safe
(GRAS) designation.
It is a popular misconception that MSG can cause headaches and other feelings of
discomfort, known as "Chinese restaurant syndrome".
Several blinded studies show no such effects when MSG is combined with food in normal
concentrations,
and are inconclusive when MSG is added to broth in large concentrations.
The European Union classifies it as a food additive permitted in certain foods and
subject to quantitative limits.
MSG has the HS code 2922.42 and the E number E621.'''

pattern3 = r"[A-Za-z]+e | [A-Za-z]+e\.[A-Za-z]+e, " #враховує те, що після слова що
закінчується на "e" може стояти кома або крапка
regularExpression3 = re.findall(pattern3, string3)
print('-'*70)
print("List of words, that end with the letter \"e\":")
print('-'*70)
print(regularExpression3)
print()
```

**Результат виконання програми:**

```
PS C:\Users\Dream\Desktop\налітєх_2курс\PCM\labs\2> & C:\Users\Dream\AppData\Local\Programs\Python\Python
313\python.exe c:/Users/Dream/Desktop/налітєх_2курс/PCM/labs/2/lab2PSM.py
-----
List of words, that end with the letter "e":
-----
['glutamate ', 'glutamate, ', 'some ', 'cheese ', 'taste ', 'the ', 'glutamate ', 'Japanese ', 'Kikunae ',
'isolate ', 'duplicate ', 'the ', 'taste ', 'edible ', 'Japanese ', 'cuisine.', 'the ', 'The ', 'safe ',
'cause ', 'Chinese ', 'are ', 'inconclusive ', 'large ', 'The ', 'additive ', 'quantitative ', 'the ',
'code ', 'the ']
```

№40. Написати РВ який би знаходив у тексті всі номери стільникових телефонів у форматі xxx-xxx-xxxx та вивести результат. Наприклад: "097-567-8901".

**Код програми:**

```
# №40. Написати РВ який би знаходив у тексті всі номери стільникових телефонів
# у форматі xxx-xxx-xxxx та вивести результат. Наприклад: "097-567-8901".
# for i in range(100):
#     print(f"{r.randint(10,999)}-{r.randint(10,9999)}-{r.randint(10,9999)}", end=' ')
string40 = '''539-5534-5220 501-2684-4713 357-5138-7698 444-1292-4207 705-3300-4977
777-4583-7020 276-8734-598 752-3705-9654 86-3968-1976 931-3193-6328 762-8240-6690 586-
6300-1117 992-1926-5363 952-6062-2353 671-7085-6169 236-9126-2840 408-6295-7495 594-
9342-8951 906-131-5257 921-9186-1658 90-3015-307 926-1995-4974 102-6665-5820 958-3180-
1532 818-4037-9242 150-8816-6884 738-8011-6522 160-7768-7873 738-6750-9737 730-1837-
8419 56-2103-3247 332-4018-8118 947-5233-6643 176-7310-5510 592-4441-7485 339-8915-6211
278-6959-8227 753-9684-4346 114-5896-7735 923-4842-1181 241-1456-464 840-137-1851 960-
2197-2083 498-3669-8537 455-1599-549 533-7367-6092 211-4787-8347 504-796-5156 633-5071-
5196 112-8513-4656 323-5144-2529 517-9398-5640 130-3264-9510 277-7920-7713
406-8304-5015 986-7911-1971 320-1634-9264 984-9388-9743 983-2705-906 461-134-9860 904-
8329-7845 512-6055-9431 591-3332-9160 827-1368-5231 788-3839-7952 28-2482-2222 348-465-
1640 327-6130-97 521-631-1076 699-1798-9947 121-825-4575 190-4624-4058 689-1229-5010
323-8622-8441 381-890-2884 111-3945-7426 406-2923-4152 845-7645-4099 128-5263-4480 594-
6909-7745 867-6414-6013 313-9366-8285 87-5936-7751 824-5010-9121 944-2012-5697 709-
6496-1113 18-1373-9394 575-3494-9251 19-965-6280 49-2842-2358 780-8767-1236 468-824-448
598-9135-6413 359-798-5443 279-141-5941 190-2060-1593 142-4089-9386 189-2527-5162 337-
5858-424 559-9839-5309'''

pattern40 = r"\d{3}-\d{3}-\d{4}"
regularExpression40 = re.findall(pattern40, string40)
print('-'*70)
print("List of phone numbers that match the \"xxx-xxx-xxxx\" pattern:")
print('-'*70)
print(regularExpression40)
```

**Результат виконання програми:**

```
PS C:\Users\Dream\Desktop\палітех_2курс\ПСМ\labs\2> & C:\Users\Dream\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Dream/Desktop/палітех_2курс/ПСМ/labs/2/lab2PSM.py
-----
List of phone numbers that match the "xxx-xxx-xxxx" pattern:
-----
['906-131-5257', '840-137-1851', '504-796-5156', '461-134-9860', '348-465-1640',
'521-631-1076', '121-825-4575', '381-890-2884', '359-798-5443', '279-141-5941']
PS C:\Users\Dream\Desktop\палітех_2курс\ПСМ\labs\2> █
```

**Висновок.** При виконанні цієї лабораторної роботи я навчився використовувати функції, методи та типи даних бібліотеки NumPy, а також навчився використовувати регулярні вирази з модуля re для опрацювання текстової інформації.