

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

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

## **Програмування**

### **Лабораторна робота №4**

«Списки, кортежі, множини і діапазони в Python»

Виконав:

студент групи ІО-41

*Давидчук А. М.*

Залікова книжка № 4108

Перевірив

*Пономаренко А. М.*

**Тема:** «Списки, кортежі, множини і діапазони в Python».

**Мета:** вивчити способи створення списків, кортежів, множин та задавання діапазонів. Операції над списками, кортежами та діапазонами. Функції для перетворень списків, кортежів та множин.

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

### **Короткі теоретичні відомості:**

from import – конструкція операторів імпортування, яка дозволяє імпортувати лише потрібний функціонал модуля та їх подальше використання без імені модуля;  
random – бібліотека Python, за допомогою якої можна генерувати різні структури даних значення яких будуть випадковими;  
def – оператор створення функції чи процедури;  
return – повернення значення;  
 $x \% 2$  – дістати остачу після ділення  $x$  на 2;  
sample() – функція з бібліотеки random, яка дозволяє створити список, значення якого будуть раптові з переданого діапазону;  
range() – функція створення діапазону;  
max(), min() – функція виводу максимального та відповідно мінімального значення послідовності;  
lambda – безіменна функція, блок якої записується в один рядок після двокрапки та немає оператора return;  
abs() – отримати модуль числа (абсолютне значення);  
remove() – метод видалення елемента зі списку;  
[x for x in first\_list] – спискове включення чи генератор списків – один зі способів генерування списків;  
uniform() – функція з бібліотеки random, яка виводить випадкове десяткове значення на інтервалі двох переданих чисел;  
round() – функція округлення певного числа, з певною точністю;  
for – оператор циклу, кількість проходження якого залежить від кількості елементів певного ітерованого об'єкта;  
 $x*=2$  – операція множення-присвоєння. Еквівалентна запису:  $x = x*2$ .

## Задання №1

- |   |   |
|---|---|
| 8 | Випадковим чином створити цілочисельний список. Визначте індекси найбільшого з непарних за значенням додатних елементів та найбільшого з парних за значенням додатних елементів. Знайдіть індекс елемента, найближчого за значенням до різниці двох попередніх елементів. Видаліть цей елемент, а також сформууйте новий список з тих елементів, що у сумі з виведеним перевищують число 10 . |
|---|---|

### Код:

```
from random import sample

def parity(x): return x if x % 2 == 0 else False
def oddity(x): return x if x % 2 != 0 else False

first_list = sample(range(-100, 101), 10)
print(f"Список: {first_list}")

max_parity = max(first_list, key=parity)
max_oddity = max(first_list, key=oddity)
difference = max_oddity - max_parity
closest = min(first_list, key = lambda x: abs(x - difference))

print(f"Індекс максимального парного значення: {first_list.index(max_parity)}")
print(f"Індекс максимального непарного значення: {first_list.index(max_oddity)}")
print(f"Індекс найближчого значення до різниці максимумів: {first_list.index(closest)}")

first_list.remove(closest)
new_list = [x for x in first_list if x + closest > 10]

print(f"Новий список: {new_list}")
```

### Знімок екрана контрольного прикладу:

```
Список: [-17, 32, 89, 2, -58, -89, -85, 64, -48, -36]
Індекс максимального парного значення: 7
Індекс максимального непарного значення: 2
Індекс найближчого значення до різниці максимумів: 1
Новий список: [-17, 89, 2, 64]
```

## Завдання №2

8	Згенерувати матрицю $A(m,n)$ розмірністю $m \times n$ з випадкових елементів, що є дійсними числами. Знайти спочатку мінімальні елементи в кожному рядку матриці, а потім обчислити їх добуток.
---	---

### Код:

```
from random import uniform

minimum = 1

m = int(input('Введіть число рядків: '))
n = int(input('Введіть число стовпців: '))

matrix = [[round(uniform(-50, 50), 4) for j in range(n)] for i in range(m)]
print(f"Матриця {m} на {n}: {matrix}")

for string in matrix: minimum *= min(string)

print(f"Добуток мінімальних значень кожного рядка: {minimum}")
```

### Знімок екрана контрольного прикладу:

```
Введіть число рядків: 3
Введіть число стовпців: 3
Матриця 3 на 3: [[9.7908, 41.8474, 43.0163], [-44.0217, 10.4532, -42.5228], [-8.6226, -11.1737, 35.748]]
Добуток мінімальних значень кожного рядка: 4815.950294564533

Введіть число рядків: 1
Введіть число стовпців: 1
Матриця 1 на 1: [[26.0032]]
Добуток мінімальних значень кожного рядка: 26.0032

Введіть число рядків: 1
Введіть число стовпців: 10
Матриця 1 на 10: [[21.1794, 19.315, -31.3374, -12.853, 44.7723, 27.0562, -34.7711, 33.9788, -40.3026, 16.187]]
Добуток мінімальних значень кожного рядка: -40.3026

Введіть число рядків: 10
Введіть число стовпців: 1
Матриця 10 на 1: [[-2.7674], [-10.6032], [39.319], [-17.5766], [10.3855], [-34.1901], [-23.1494], [-18.7381], [42.2473], [45.9822]]
Добуток мінімальних значень кожного рядка: 6067754492581.2
```

### Висновок:

Я написав дві програми з використанням списків, діапазонів та модуля random для генерації їх значень. Доволі успішно продемонстрував вірність виконання програм та розуміння концептів послідовностей, діапазонів та роботи з ними. Також додатково написав три унарні функції, двоє з яких повертають булеві значення парності чи непарності певного числа, а інша – безіменна, яка повертала число. Написані функції застосував як ключі вбудованих функцій max() та min() задля більш модифікованого їх використання. При виводі використав форматування.