

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»

Виконала:
Студентка групи ІО-15
Кушнерик Є.О.
Залікова книжка №1508

Перевірив:
Пономаренко А.М

Лабораторна робота №4

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.

Завдання:

1. Вивчити матеріал лекцій 11, 12, 13 та 14.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Теоретичні основи:

1. **Списки** є змінюваними типами даних.
2. **Кортежі** є незмінюваними типами даних.
3. **Множини** можуть бути як змінюваними, так і незмінюваними.
4. **Діапазони** є наборами чисел, сформованими на основі заданих початкового, кінцевого значень і величини кроку між числами.

Створити список можна такими способами:

1. За допомогою функції `list` (`[<Послідовність>]`).
2. Перелічивши **всі елементи** списку всередині квадратних дужок.
3. Застосувавши метод `append()` для заповнення списку поелементно.
4. За допомогою генератора списків.

Створити копію списку можна такими способами:

1. **Перший спосіб** – застосуванням операції добування зрізу.
2. **Другий спосіб** – використанням функції `list()`,
3. **Третій спосіб** – застосуванням методу `copy()`.
4. Щоб одержати глибоку копію списку, слід скористатися функцією `deepcopy()` з модуля `copy`.

Операції над списками

1. Присвоювання значень
елементам списку по індексу: `x[0]`

(Ця операція не застосовна до рядка!)

2. Присвоювання з кінця за від'ємним індексом: `y[-1]`

3. Присвоювання за допомогою зрізу:

Операція добування зрізу: `[<Початок>:<Кінець>:<Крок>]`

```
>>> arr = [2, 6, 10, 14, 18, 22]
```

```
>>> arr[1:3] = [8, 7]
```

```
>>> arr
```

```
[2, 8, 7, 14, 18, 22]
```

4. Оператори `+` та `+=`

5. Оператори `in` та `not in`

Багатовимірні списки

Будь-який елемент списку може містити список.

Створити вкладений список можна, наприклад, так:

Двовимірний список:

```
>>> n = [[1,2,3], [4,5,6], [7,8,9]]
>>> n
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>> n[0][0]
1
>>> n[2][1]
8
```

Тривимірний список:

```
>>> t = [[[1,2],[3,4]], [[5,6],[7,8]]]
>>> t
[[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
>>> t[0][0][0]
1
>>> t[1][1][0]
7
```

Чотиривимірний список:

```
>>> f = [[[[1,2],[3,4]], [[5,6],[7,8]]],
          [[9,10],[11,12]], [[13,14],[15,16]]]
>>> f[0][0][0][0]
1
>>> f[0][1][1][0]
7
>>> f[1][1][1][1]
16
```

Перебір елементів списку

1. За допомогою циклу `for`: `for i in arr: print(i, end=" ")`
2. За допомогою циклу `while`.
3. За допомогою генераторів списків

```
>>> arr = [i for i in range(1,10)]
>>> arr
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Функція `map()`

Дозволяє застосувати функцію до кожного елемента послідовності

Повертає об'єкт, що підтримує ітерації. Щоб одержати список, необхідно результат передати у функцію `list()`.

```
map(<Функція>, <Послідовність1>[, ..., <Послідовність N>])
```

Індивідуальні завдання

Завдання 1

Відповідно до номера в списку вибрати індивідуальне завдання. Написати програму мовою 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.**

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 програми

```
from random import randint
import math
arr = [randint(-100, 100) for i in range(10)]
print("Початковий список {0}".format(arr));
max1 = 0;max2 = 0
for i in arr:
    if i%2 == 1:
        if(i > max1):
            max1 = i
    else:
        if(i > max2):
            max2 = i
print("Індекс найбільшого з:\nнепарних за значенням додатних елементів: "
      "{0}\nпарних за значенням додатних елементів: {1}".format(arr.index(max1), arr.index(max2)))

difference_arr = []
for i in range(len(arr)):
    difference_arr.append(math.fabs((arr[i-2]-arr[i-1])-arr[i]))#знаходимо модуль різниці елемента і двох його
попередніх елементів
min_difference_index = difference_arr.index(min(difference_arr))#знаходимо найменшу різницю і її індекс. Тобто, чим
менша різниця ((i-2)-(i-1)) та i, тим більше i наближене до ((i-2)-(i-1))
min_difference_element = arr[min_difference_index]
arr.pop(min_difference_index)
sum_arr = [i for i in arr if i + min_difference_element > 10]
print("Список з тих елементів, що у сумі з виведеним перевищують число 10: {0}".format(sum_arr))
```

Результат програми

D:\ProgramData\Anaconda3\envs\pythonProject\python.exe C:/Users/HP840/PycharmProjec

Початковий список [-32, 15, -26, 0, 85, 9, 66, -99, -36, -26]

Індекс найбільшого з:

непарних за значенням додатних елементів: 4

парних за значенням додатних елементів: 6

Список з тих елементів, що у сумі з виведеним перевищують число 10: [0, 85, 9, 66]

Process finished with exit code 0

Завдання 2

Відповідно до номера в списку вибрати індивідуальне завдання. Написати програму мовою 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.**

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 програми

```
from random import uniform

col = input("Введіть кількість стовпців матриці: ")
row = input("Введіть кількість рядків матриці: ")
while not (row + col).isdigit():
    print("Введіть цілі числа")
    col = input("Введіть кількість стовпців матриці: ")
    row = input("Введіть кількість рядків матриці: ")
row, col = int(row), int(col)
array = [[round(uniform(-100, 100), 2) for i in range(col)] for j in range(row)]
min_el_arr = [min(i) for i in array]
mult = 1
print("Матриця")

for row in array:
    for elem in row:
        print("{:>10}".format(elem), end=" ")
    print("")

for i in range(len(min_el_arr)):
    mult *= min_el_arr[i]
print("Добуток мінімальних елементів = {0}".format(round(mult, 2)))
```

Результат програми

```
D:\ProgramData\Anaconda3\envs\pythonProject\pythor
```

```
Введіть кількість стовпців матриці: 3
```

```
Введіть кількість рядків матриці: 3
```

```
Матриця
```

54.55	-90.74	68.59
-14.4	-82.2	36.47
-17.98	54.33	2.55

```
Добуток мінімальних елементів = -134109.73
```

```
Process finished with exit code 0
```

```
D:\ProgramData\Anaconda3\envs\pythonProject\python.exe C:/Users/
```

```
Введіть кількість стовпців матриці: e
```

```
Введіть кількість рядків матриці: c
```

```
Введіть цілі числа
```

```
Введіть кількість стовпців матриці: 3
```

```
Введіть кількість рядків матриці: 4
```

```
Матриця
```

82.64	-57.56	55.79
-82.77	48.59	67.38
-11.47	-38.16	-11.29
44.22	-64.62	-11.84

```
Добуток мінімальних елементів = 11748138.56
```

```
Process finished with exit code 0
```

Висновок:

Під час лабораторної роботи я вивчила способи створення списків, кортежів, множин та задання діапазонів, операції над списками, кортежами та діапазонами, функції для перетворень списків, кортежів та множин.