

## ПРАКТИЧНА РОБОТА №2. ВИВЧЕННЯ БАЗОВИХ ФУНКЦІЙ. РОБОТА З РЕКУРСИВНИМИ ФУНКЦІЯМИ

**Мета роботи** – вивчити та засвоїти базові навички використання функцій у Python. Дізнатися про різновиди функцій та можливі аргументи і результати таких функцій.

### Основні теоретичні відомості

#### Базові поняття про функції

**Функція** в python - об'єкт, який приймає аргументи і повертає значення. Зазвичай функція визначається за допомогою інструкції **def**.

```
def double(x) :  
    #Тут, коли потрібно, розмішують  
    #багаторядковий документує коментар docstring,  
    #який пояснює, що саме функція обчислює.|  
    #Наприклад, дана функція множить вхідне значення на 2  
    return x * 2  
y=double(5)  
print(y)
```

10

Рисунок 2.1 – Визначення функції

Інструкція **return** каже, що потрібно повернути значення. У цьому випадку функція повертає  $2x$ . Функції в Python розглядаються як об'єкти першого класу. Це означає, що їх можна привласнювати змінним і передавати в інші функції так само, як будь-які інші аргументи. Функція може бути будь-якої складності і повертати будь-які об'єкти (списки, кортежі, і навіть функції!):

```
def newfunc(n):  
    def myfunc(x):  
        return x + n  
    return myfunc  
new = newfunc(100) # new - це функція  
print(new(200))
```

300

## Рисунок 2.2 – Демонстрація використання інструкції return

Функція може і не закінчуватися інструкцією return, при цьому функція поверне значення None:

```
def func():  
    pass  
print(func())
```

None

## Рисунок 2.3 – Демонстрація функції без інструкції return

### Аргументи функції

Функція може приймати будь-яку кількість аргументів чи не приймати їх зовсім. Також поширені функції з довільним числом аргументів, функції з позиційними і іменованими аргументами, обов'язковими і необов'язковими.

```
def func(a, b, c=2): # c - необов'язковий аргумент  
    return a + b + c  
x=func(1, 2) # a = 1, b = 2, c = 2 (за замовчуванням)  
print(x)  
x=func(1, 2, 3) # a = 1, b = 2, c = 3  
print(x)  
x=func(a=1, b=3) # a = 1, b = 3, c = 2  
print(x)  
x=func(a=3, c=6) # a = 3, c = 6, b не визначений  
print(x)
```

5

6

6

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-1-19e104a86b3c> in <module>  
      7 x=func(a=1, b=3) # a = 1, b = 3, c = 2  
      8 print(x)  
---->  9 x=func(a=3, c=6) # a = 3, c = 6, b не визначений  
     10 print(x)  
  
TypeError: func() missing 1 required positional argument: 'b'
```

## Рисунок 2.4 – Використання аргументів функції

Функція також може приймати змінну кількість позиційних аргументів, тоді перед ім'ям ставиться \*:

```
def func(*args):  
    return args  
print(func(1, 2, 3, 'abc'))  
print(func())  
print(func(1))  
  
(1, 2, 3, 'abc')  
()  
(1,)
```

Рисунок 2.5 – Змінна кількість позиційних аргументів

### Анонімні функції, інструкція lambda

Анонімні функції можуть містити лише один вислів, а й виконуються вони швидше. Анонімні функції створюються за допомогою інструкції lambda. Крім цього, їх не обов'язково привласнювати змінної, як робили ми інструкцією def func():

```
func = lambda x, y: x + y  
print(func(1, 2))  
print(func('a', 'b'))  
print((lambda x, y: x + y)(1, 2))  
print((lambda x, y: x + y)('a', 'b'))  
  
3  
ab  
3  
ab
```

Рисунок 2.6 – Створення анонімної функції

Анонімним функціям, на відміну від звичайних, не потрібна інструкція return, а взагалі вона поводить себе так само:

```
func = lambda *args: args  
print(func(1, 2, 3, 4))  
  
(1, 2, 3, 4)
```

Рисунок 2.7 – Робота анонімної функції

## Базові поняття про рекурсивні функції

Рекурсія в програмуванні – це можливість дати визначення функції, використовуючи в процесі саму визначену функцію. В математиці багато функцій визначені саме таким чином, тому і більшість мов програмування бере на озброєння цей підхід. Python тут не є винятком: зазвичай у визначенні функції ви можете використовувати тільки визначення, дані раніше, але є один виняток – функція в своєму тілі може викликати себе. Виглядає це так:

```
def shortStory():  
    print("У попа був собака, він його любив")  
    print("Вона з'їла шматок м'яса, він її вбив,")  
    print("У землю закопав і напис написав:")  
    shortStory()  
shortStory()
```

```
У попа був собака, він його любив  
Вона з'їла шматок м'яса, він її вбив,  
У землю закопав і напис написав:  
У попа був собака, він його любив  
Вона з'їла шматок м'яса, він її вбив,  
У землю закопав і напис написав:  
У попа був собака, він його любив  
Вона з'їла шматок м'яса, він її вбив,  
У землю закопав і напис написав:  
У попа був собака, він його любив  
Вона з'їла шматок м'яса, він її вбив,  
У землю закопав і напис написав:
```

Рисунок 4.9 – Приклад рекурсивної функції

Для більшого розуміння розберемо рішення класичної задачі «Ханойські вежі». Головоломка "Ханойські вежі" складається з трьох стрижнів, пронумерованих числами 1, 2, 3. На стрижень 1 надіта пірамідка з  $n$  дисків різного діаметру в порядку зростання діаметру. Диски можна перекладати з одного стрижня на інший строго по одному, при цьому диск можна класти на диск меншого діаметру. Необхідно перекласти всю пірамідку з стрижня 1 на стрижень 3 за мінімальне число перекладань.

Необхідно написати програму, яка для даного числа дисків  $n$  друкує послідовність перекладань, необхідну для вирішення головоломки. Спочатку потрібно подумати, як перекласти пірамідку з  $n$  дисків з одного стрижня на

інший. Для цього потрібно перш за все перенести найбільший диск. Але щоб перенести цей диск потрібно всю пірамідку без цього диска, тобто пірамідку з диска перенести на третій стрижень, потім перенести один найбільший диск, потім перенести пірамідку з диска на той стрижень, на який перемістили найбільший диск.

Напишемо рекурсивну функцію. Яка друкує послідовність перекладань, необхідних для переміщення пірамідки з дисків зі стрижня *start* на стрижень *finish*. крайній випадок - це випадок  $n==0$ , в цьому випадку для переміщення пірамідки з 0 дисків просто нічого не потрібно робити!

```
[7]: def move(n, start, finish):  
    if n > 0:  
        temp = 6-start-finish # Допоміжний стрижень  
        move(n-1, start, temp)  
        print("Перенести диск", n, "зі стрижня", start, "на стрижень", finish)  
        move(n-1, temp, finish)  
move(10,1,3)  
  
Перенести диск 1 зі стрижня 1 на стрижень 2  
Перенести диск 2 зі стрижня 1 на стрижень 3  
Перенести диск 1 зі стрижня 2 на стрижень 3  
Перенести диск 3 зі стрижня 1 на стрижень 2  
Перенести диск 1 зі стрижня 3 на стрижень 1  
Перенести диск 2 зі стрижня 3 на стрижень 2  
Перенести диск 1 зі стрижня 1 на стрижень 2  
Перенести диск 4 зі стрижня 1 на стрижень 3  
Перенести диск 1 зі стрижня 2 на стрижень 3  
Перенести диск 2 зі стрижня 2 на стрижень 1  
Перенести диск 1 зі стрижня 3 на стрижень 1  
Перенести диск 3 зі стрижня 2 на стрижень 3  
Перенести диск 1 зі стрижня 1 на стрижень 2  
Перенести диск 2 зі стрижня 1 на стрижень 3  
Перенести диск 1 зі стрижня 2 на стрижень 3  
Перенести диск 5 зі стрижня 1 на стрижень 2  
Перенести диск 1 зі стрижня 3 на стрижень 1
```

Рисунок 4.10 – Рішення класичної задачі «Ханойські вежі»

### Завдання до виконання

Створити рядок довжиною  $(10 + N \bmod 5)$ , де  $N$  – номер студента у журналі.

Створити та продемонструвати роботу трьох функцій згідно власного варіанту (таб. 2.1). **Примітка: Функцію сортування масиву написати самостійно, тобто не використовувати вбудовані функції сортування.**

Таблиця 2.1 – Варіанти завдань

№	Завдання
1	Створити функцію для вирішення наступного завдання: Масив $X=(x_1, x_2, \dots, x_n)$ містить велику кількість нульових елементів. Визначити положення і розмір найбільш довгої серії таких елементів.
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = 3x * y^2 - 9z$
2	Створити функцію для вирішення наступного завдання: Заданий масив $X=(x_1, x_2, \dots, x_n)$ , в якому можуть бути однакові числа. Знайти максимальний і мінімальний елементи серед неповторюваних чисел.
	Створити функцію сортування в порядку убутання.
	Створити анонімну функцію для обчислення функції: $f = 5x * y^3 + 7z$
3	Створити функцію для вирішення наступного завдання: З масиву чисел $X=(x_1, x_2, \dots, x_n)$ вилучити всі парні за значенням елементи.
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = 8x * y^2 - 9z$
4	Створити функцію для вирішення наступного завдання: У масиві $X=(x_1, x_2, \dots, x_n)$ поміняти місцями перший і другий негативні елементи, третій і четвертий негативні елементи тощо.
	Створити функцію сортування в порядку убутання.
	Створити анонімну функцію для обчислення функції: $f = \frac{3x}{y^4} - 2z$
5	Створити функцію для вирішення наступного завдання: Елементи масиву $X = (x_1, x_2, \dots, x_n)$ – це послідовність цифр цілого числа. Переставити цифри числа у зворотному порядку

№	Завдання
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = x - y^2 * 4z$
6	Створити функцію для вирішення наступного завдання: Відомо, що в цілочисельному масиві $X=(x_1, x_2, \dots, x_n)$ три і тільки три числа, що є рівними між собою. Знайти ці числа
	Створити функцію сортування в порядку убутання.
	Створити анонімну функцію для обчислення функції: $f = 2x + y^2 - 5z$
7	Створити функцію для вирішення наступного завдання: За однократний перегляд масиву знайти його максимальний позитивний елемент $X_{\max}$
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = 7x * y^3 * 2z$
8	Створити функцію для вирішення наступного завдання: Перетворити масив $X$ , розташувавши спочатку його негативні, а потім позитивні елементи, зберігши при цьому в групі негативних та позитивні елементів їх вихідний відносний порядок.
	Створити функцію сортування в порядку убутання.
	Створити анонімну функцію для обчислення функції: $f = 11x - y^{-2} + 6z$
9	Створити функцію для вирішення наступного завдання: У масиві $X=(x_1, x_2, \dots, x_n)$ поміняти місцями перший і другий позитивні елементи, третій і четвертий позитивні елементи тощо.
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = 6x + y^3 * 8z$
10	Створити функцію для вирішення наступного завдання: Заданий масив цілих чисел $X=(x_1, x_2, \dots, x_n)$ . Сформувати масив $Y=(y_1, y_2, \dots, y_m)$ , помістивши в нього в порядку убутання всі позитивні числа, що входять у масив $X$ .
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = \frac{2x}{y^1} + 5z$
11	Створити функцію для вирішення наступного завдання:

№	Завдання
	Заданий цілочисельний масив $X=(x_1, x_2, \dots, x_n)$ , у якому можуть бути однакові числа. Підрахувати кількість повторюваних чисел у масиві.
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = 5x + \frac{y^2}{7z}$
12	Створити функцію для вирішення наступного завдання: Виконати циклічне зрушення масиву $X=(x_1, x_2, \dots, x_n)$ на 5 елементів вліво.
	Створити функцію сортування в порядку убутання.
	Створити анонімну функцію для обчислення функції: $f = 4x * y^2 - \frac{9}{z}$
13	Створити функцію для вирішення наступного завдання: Виконати циклічне зрушення масиву $X=(x_1, x_2, \dots, x_n)$ на 3 елементів вправо.
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = 11x * \frac{y^3}{3} + 8z$
14	Створити функцію для вирішення наступного завдання: Масив $X=(x_1, x_2, \dots, x_n)$ містить велику кількість нульових елементів. Визначити положення і розмір найбільш довгої серії ненульових елементів.
	Створити функцію сортування в порядку убутання.
	Створити анонімну функцію для обчислення функції: $f = 2xz * \frac{y^2}{2}$
15	Створити функцію для вирішення наступного завдання: Заданий масив цілих чисел $X=(x_1, x_2, \dots, x_n)$ . Сформувати масив $Y=(y_1, y_2, \dots, y_m)$ , помістивши в нього в порядку убутання всі негативні числа, що входять у масив $X$ .
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції: $f = 2(x - y) * \frac{z^2}{4}$
16	Створити функцію для вирішення наступного завдання: Заданий масив цілих чисел $X=(x_1, x_2, \dots, x_n)$ . Сформувати масив $Y=(y_1, y_2, \dots, y_m)$ , помістивши в нього в порядку убутання всі різні (неповторювані) числа, що входять у масив $X$ .
	Створити функцію сортування масиву за зростанням.
	Створити анонімну функцію для обчислення функції:



№	Завдання
	$f = 2(x - 4z) * \frac{y^{-2}}{5}$

У даній лабораторній роботі необхідно виконати рішення задачі згідно свого варіанту за допомогою побудови рекурсивної функції, при цьому в своєму рішенні не можна використовувати цикли та зрізи з кроком відмінним від 1.

Таблиця 2.2 – Варіанти завдань

№	Завдання
1	Написати рекурсивну функцію для обчислення факторіала натурального числа n.
2	Написати рекурсивну функцію: обчислення суми цифр натурального числа.
3	Написати рекурсивну функцію: обчислення кількості цифр натурального числа.
4	Дано перший член і різницю арифметичної прогресії. Написати рекурсивну функцію для знаходження: знаходження n-го члена прогресії;
5	Написати рекурсивну функцію для обчислення k-го члена послідовності Фібоначчі.
6	Дано перший член і різницю арифметичної прогресії. Написати рекурсивну функцію для знаходження: знаходження суми n перших членів прогресії
7	Написати рекурсивну функцію знаходження цифрового кореня натурального числа. Цифровий корінь даного числа отримують у такий спосіб. Якщо скласти всі цифри цього числа, потім все цифри знайденої суми і повторювати цей процес, то в результаті буде отримано однозначне число (цифра), яка і називається цифровим коренем даного числа.
8	Дано перший член і знаменник геометричної прогресії. Написати рекурсивну функцію: знаходження n-го члена прогресії;
9	Дано перший член і знаменник геометричної прогресії. Написати рекурсивну функцію: знаходження суми n перших членів прогресії;
10	Написати рекурсивну процедуру для виведення на екран цифр натурального числа у зворотному порядку.
11	Дано натуральне число n. Виведіть всі числа від 1 до n.
12	Дано два цілих числа A і B (кожне в окремому рядку). Виведіть всі числа від A до B включно, в порядку зростання, якщо A < B, або в порядку убутання в іншому випадку.
13	Дано натуральне число N. Виведіть всі його цифри по одній, в зворотному порядку, розділяючи їх пробілами.
14	Дано натуральне число n > 1. Виведіть всі прості множники цього числа в порядку неспадання з урахуванням кратності.

№	Завдання
15	Дано слово, що складається лише з малих латинських букв. Перевірте, чи є це слово паліндромом. Виведіть YES або NO.
16	Дано натуральне число $n > 1$ . Перевірте, чи є воно простим. Програма повинна вивести слово YES, якщо число просте і NO

## Приклад виконання

Варіант - 16.

```
array=[1,2,2,4,5,8,4,3,18,11,11] # вхідний масив
def first_func(x):
    k=0 # лічильник
    imax=0 # номер максимального елементу
    max=x[0] # значення максимального елементу
    # сортування масиву
    while k<len(x)-1:
        j=k
        imax=k
        max=x[k]
        while j<len(x):
            if max<x[j]:
                max=x[j]
                imax=j
            j+=1
        x[imax]=x[k]
        x[k]=max
        k+=1
    y=[] # створення нового масиву
    # перенесення неповторюваних елементів до нового масиву
    for i in range(len(x)):
        buf=x[i]
        num=0
        for j in range(len(x)):
            if x[i]==x[j]:
                num+=1
        if num==1:
            y=y+[x[i]]
    return y
print(array) # демонстрація вхідного масиву
print(first_func(array)) # демонстрація результуючого масиву
```

[1, 2, 2, 4, 5, 8, 4, 3, 18, 11, 11]  
[18, 8, 5, 3, 1]

Рисунок 2.8 – Результат виконання першого завдання

```
import math
second_function = lambda x,y,z: 2*(x-4*z)*(pow(y,-2)/5)
print(second_function(2,5,8))
```

-0.48

Рисунок 2.9 – Результат виконання третього завдання

```
[1]: def rec(n,div):  
      if div==1:  
          print("YeS")  
      elif n%div==0:  
          print("No")  
      else:  
          rec(n,div-1)  
n=int(input("Введіть число"))  
rec(n,n-1)
```

Введіть число 17  
YeS

Рисунок 4.13 – Виконання четвертого завдання

## **Контрольні запитання**

1. Функції у Python, їх види та призначення.
2. Які аргументи може приймати функція, їх типи та кількість.
3. Анонімні функції, призначення та особливості використання.
4. Значення функції, що повертаються, їх типи та кількість.
5. Функції зі змінною кількістю аргументів, призначення та особливості використання.

## **Оформлення звіту**

- 1) Титульний лист;
- 2) Мета роботи;
- 3) Номер варіанту та умови завдання;
- 4) Програмний код з коментарями;
- 5) Скріни роботи програми;
- 6) Висновки