МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ І ІНФОРМАТИКИ

Практична робота №2

з дисципліни «Методи та системи штучного інтелекту»

на тему «ВИВЧЕННЯ БАЗОВИХ ФУНКЦІЙ. РОБОТА

З РЕКУРСИВНИМИ ФУНКЦІЯМИ»

Варіант 4

Виконав:

студент гр. КІБ-21

Андрій Севастьянов

Перевірив:

асистент кафедри ПМІ

Андрій НІКІТЕНКО

Луцьк – 2024

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

**Завдання:**

Перша частина

1. Створити функцію для вирішення наступного завдання: У масиві X=(x1,x2,...,xn) поміняти місцями перший і другий негативні елементи, третій і четвертий негативні елементи тощо.
2. Створити функцію сортування в порядку убування.
3. Створити анонімну функцію для обчислення функції:

f = (3x)/(y^4) - 2z

Друга частина

Дано перший член і різницю арифметичної прогресії. Написати рекурсивну функцію для знаходження: знаходження n-го члена прогресії;

Програмний код першої частини:

def swap\_negatives(arr):  
 fl = False  
 first\_min\_ind = 0  
 for index in range(len(arr)):  
 if arr[index] < 0 and fl == False:  
 *# finding the first minimum* first\_min\_ind = index  
 fl = True  
 elif arr[index] < 0 and fl == True:  
 *# swap* arr[index], arr[first\_min\_ind] = arr[first\_min\_ind], arr[index]  
 *# normal value* fl = False  
 return arr  
  
numb = [1, -2, 3, -4, 5, -6, 7, -8, 9, -10, 11, -12, 13, -14]  
print("Масив до обробки:\t\t" + str(numb))  
result = swap\_negatives(numb)  
print("Масив після обробки:\t" + str(result))  
  
  
*# Створити функцію сортування в порядку убування.  
# Сортування включенням*def descending\_insertion\_sort(arr):  
 *# починаємо з другого* for i in range(1, len(arr)):  
 key = arr[i] *# key = наш елемент для якого ми шукаємо місце  
 # находимо правильне місце для вставки елемента* j = i - 1  
 while j >= 0 and arr[j] < key:  
 arr[j + 1] = arr[j]  
 j -= 1  
 arr[j + 1] = key  
  
 return arr  
  
  
arr = [64, 34, 25, 12, 22, 11, 90]  
print("Масив до сортування:", arr)  
descending\_insertion\_sort(arr)  
print("Масив після сортування:", arr)  
  
*# Створити анонімну функцію для обчислення функції f = ((3x)\(y^4))-2z*f = lambda x, y, z: ((3\*x) / (y\*\*4)) - 2\*z  
result = f(2, 3, 4)  
print("Результат = " + str(result))

Програмний код другої частини:

def func\_rec(a, d, n):  
 if n == 1:  
 return a  
 else:  
 return d + func\_rec(a, d, n-1)  
def func(a, d, n):  
 return a + (n - 1) \* d  
  
a = 1 *# перший член прогресії*d = 2 *# різниця прогресії*n = 5 *# член прогресії, який потрібно знайти*print("Результат з рекурсією = " + str(func\_rec(a, d, n)))  *# перевірка*print("Результат без рекурсії = " + str(func(a, d, n))) *# Виводимо n-й член прогресії*

Скріншот роботи програмного коду першої частини:

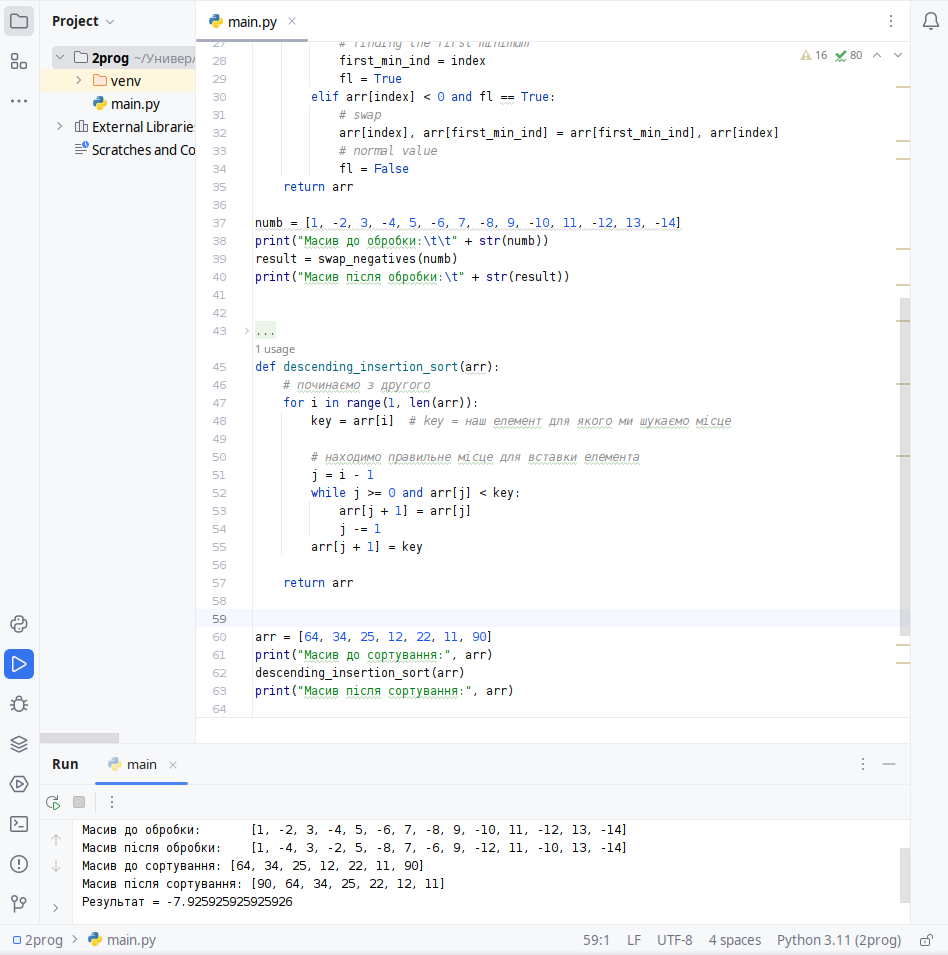


Рисунок 1 – робота першої частини

Скріншот роботи програмного коду другої частини:

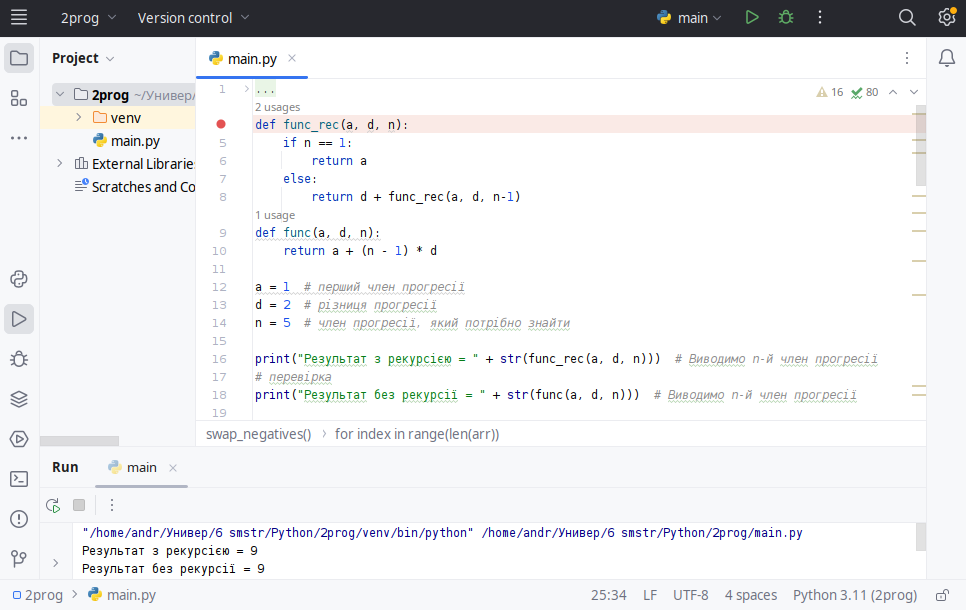


Рисунок 2 – робота другої частини

Висновок:  
 У результаті виконання даної роботи було ознайомлено з базовими навичками використання функцій у мові програмування Python. Це включає вивчення різновидів функцій, таких як звичайні функції та анонімні lambda-функції, а також розгляд можливостей аргументів і результатів функцій. Оволодівши цими навичками, програмісти зможуть створювати більш організований та ефективний код у Python.