|  |  |  |  |
| --- | --- | --- | --- |
| **AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU**  Wydział Nauk Inżynieryjnych, Katedra informatyki | | | |
| Przedmiot: | Inteligentne systemy budynkowe, mgr inż. Mariusz Mikulski | | |
| Grupa: | P1 | Nr sprawozdania: | 2 |
| Imię i Nazwisko: | Bartłomiej Cetera | Data: | 22.03.2023 |

**Zadanie 6**

# Importowanie biblioteki matplotlib.pyplot jako plt

**import matplotlib.pyplot as plt**

# Importowanie biblioteki numpy jako np

**import numpy as np**

# Ustalenie wartości a i b

**a = 2 #** Wartość a równa 2

**b = 5 #** Wartość b równa 5

# Ustalenie przedziału

# Przedział Df, czyli zbiór liczb od -10 do 10 (włącznie)

# jest definiowany za pomocą funkcji arange() z biblioteki numpy.

**Df = np.arange(-10, 11)**

# Obliczenie wartości funkcji liniowej

# Wyliczenie wartości funkcji liniowej y=ax+b dla każdej wartości x z przedziału Df.

**y = a \* Df + b**

# Stworzenie wykresu

# Narysowanie wykresu funkcji liniowej za pomocą funkcji plot()

# z biblioteki pyplot. Na osi x znajdują się wartości

# z przedziału Df, a na osi y wartości funkcji liniowej y.

**plt.plot(Df, y)**

# Dodanie tytułu i etykiet osi

# Dodanie tytułu do wykresu za pomocą funkcji title() z biblioteki pyplot.

**plt.title("Wykres funkcji liniowej y = ax + b")**

# Dodanie etykiety osi x za pomocą funkcji xlabel() z biblioteki pyplot.

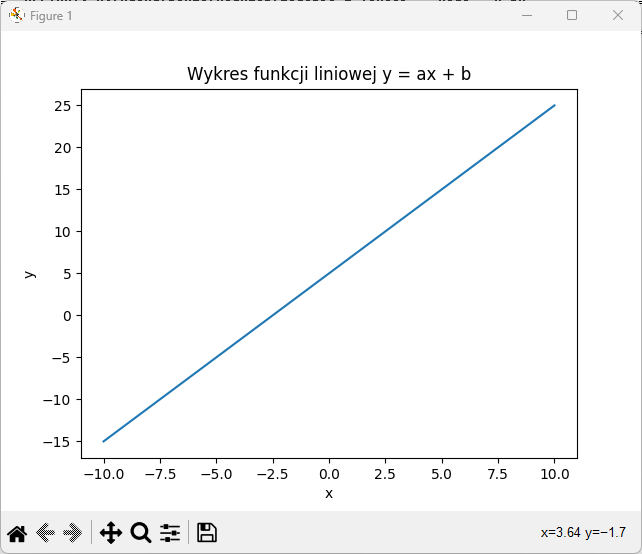
**plt.xlabel("x")**

# Dodanie etykiety osi y za pomocą funkcji ylabel() z biblioteki pyplot.

**plt.ylabel("y")**

# Wyświetlenie gotowego wykresu za pomocą funkcji show() z biblioteki pyplot.

**plt.show()**

****

Wynik powyższego kodu:

2)

# Importowanie biblioteki matplotlib.pyplot jako plt

import matplotlib.pyplot as plt

# Definicja wartości x i dwóch funkcji y

# Lista wartości argumentów x dla funkcji

**x = [1, 2, 3, 4, 5]**

# Lista wartości funkcji liniowej y=-x+1

**y = [0, -1, -2, -3, -4]**

# Lista wartości funkcji kwadratowej y=x^2

**y2 = [1, 4, 9, 16, 25]**

# Rysowanie dwóch funkcji na jednym wykresie

# Rysowanie funkcji liniowej y=-x+1 na wykresie

**plt.plot(x, y, label='-x+1')**

# Rysowanie funkcji kwadratowej y=x^2 na wykresie

**plt.plot(x, y2, label='x^2')**

# Dodanie etykiet osi x i y oraz legendy

# Dodanie etykiety osi x za pomocą funkcji xlabel() z biblioteki pyplot.

**plt.xlabel('x')**

# Dodanie etykiety osi y za pomocą funkcji ylabel() z biblioteki pyplot.

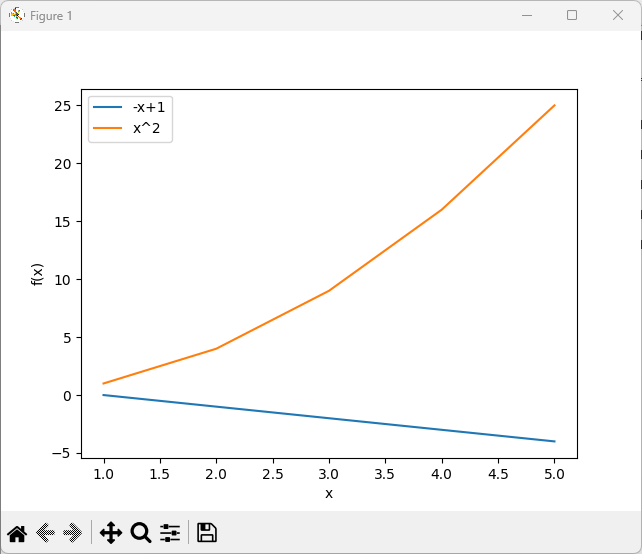
**plt.ylabel('f(x)')**

# Dodanie legendy do wykresu za pomocą funkcji legend() z biblioteki pyplot.

**plt.legend()**

# Wyświetlenie wykresu

**plt.show()**



Wynik powyższego kodu:

**Zadanie 7**

# Pobranie nazwy pliku

**plik = input("Podaj nazwę pliku: ")**

**x = input("Podaj słowo do wyszukania: ")**

# Otwarcie pliku

**try:**

**with open(plik, 'r') as file:**

# Odczytanie zawartości pliku

**content = file.read()**

# Sprawdzenie czy słowo znajduje się w pliku

**if content.find(x) != -1:**

**print("Słowo znajduje się w pliku.")**

**else:**

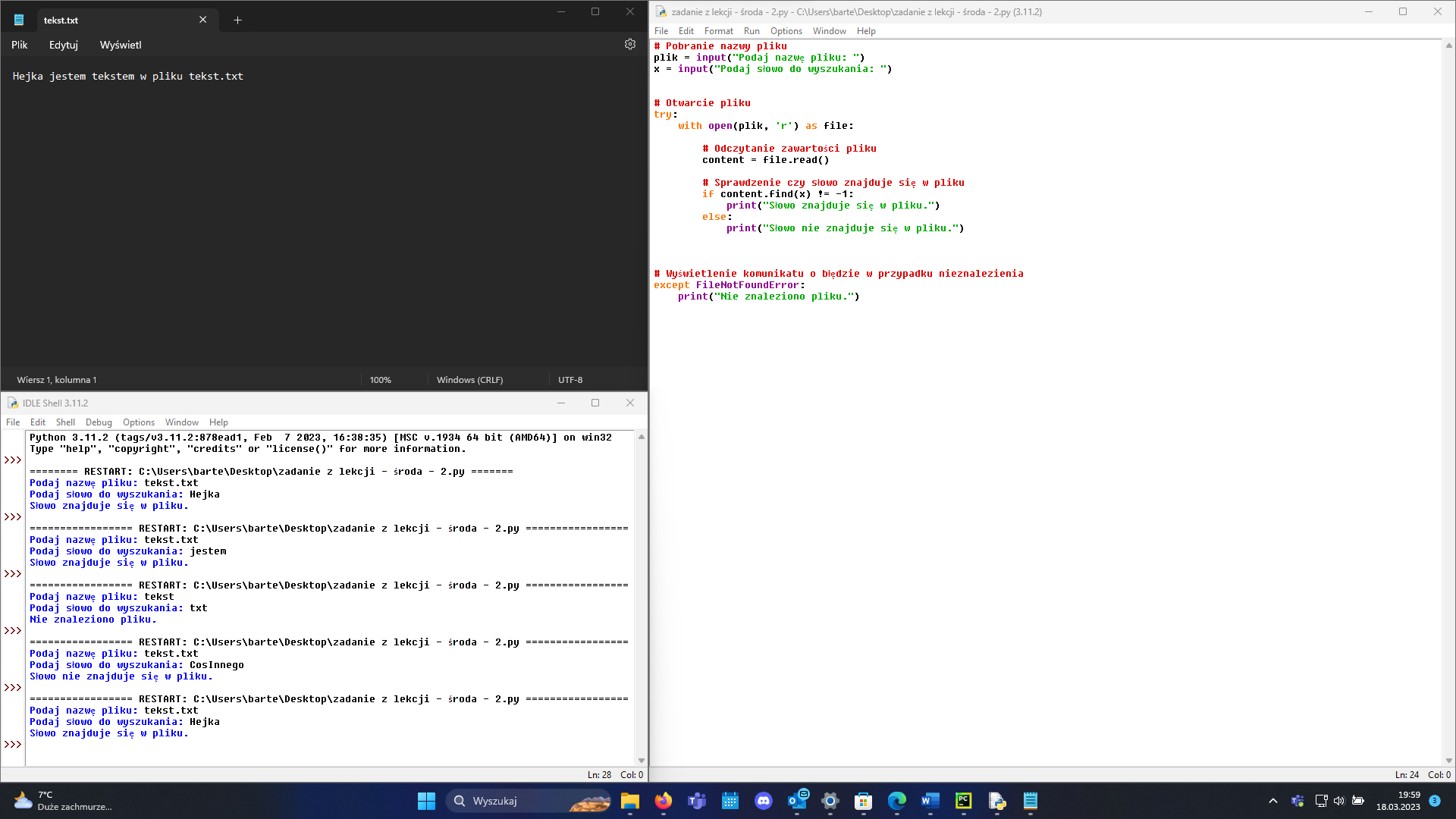
**print("Słowo nie znajduje się w pliku.")**

# Wyświetlenie komunikatu o błędzie w przypadku nieznalezienia

**except FileNotFoundError:**

**print("Nie znaleziono pliku.")**

Wynik powyższego kodu:

****

**Zadanie 8**

**import os**

def search\_file(file\_name, search\_word):

for root, dirs, files in os.walk('.'):

for name in files:

if name == file\_name:

file\_path = os.path.join(root, name)

with open(file\_path, 'r') as f:

content = f.read()

if search\_word in content:

print(f"Słowo {search\_word} znajduje się w pliku {file\_path}")

else:

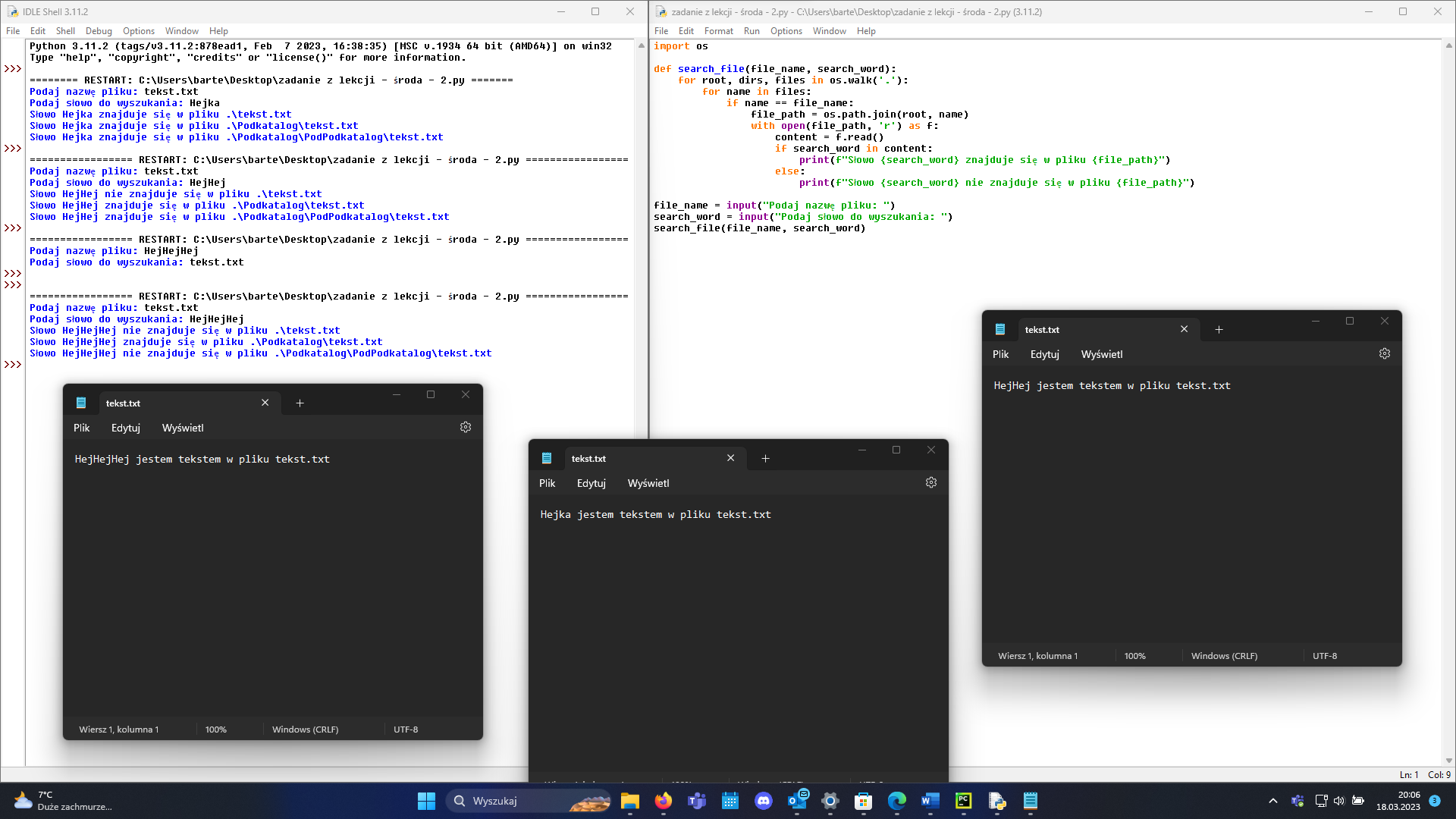
print(f"Słowo {search\_word} nie znajduje się w pliku {file\_path}")

file\_name = input("Podaj nazwę pliku: ")

search\_word = input("Podaj słowo do wyszukania: ")

search\_file(file\_name, search\_word)

Wynik powyższego kodu:

****

**Zadanie 9**

**import matplotlib.pyplot as plt**

**import numpy as np**

# funkcja zwracająca wartość funkcji kwadratowej dla danego x i współczynników a, b, c

**def f\_kwadratowa(x, a, b, c):**

**return a \* x \*\* 2 + b \* x + c**

# pobieranie od użytkownika wartości współczynników a, b, c

**a = float(input("Podaj wartość współczynnika a: "))**

**b = float(input("Podaj wartość współczynnika b: "))**

**c = float(input("Podaj wartość współczynnika c: "))**

# tworzenie tablicy wartości x od -10 do 10 z krokiem 0.1

**x = np.arange(-10, 10, 0.1)**

# obliczenie wartości y dla każdego x z tablicy x przy użyciu funkcji quadratic\_func

**y = f\_kwadratowa(x, a, b, c)**

# rysowanie wykresu

**plt.plot(x, y)**

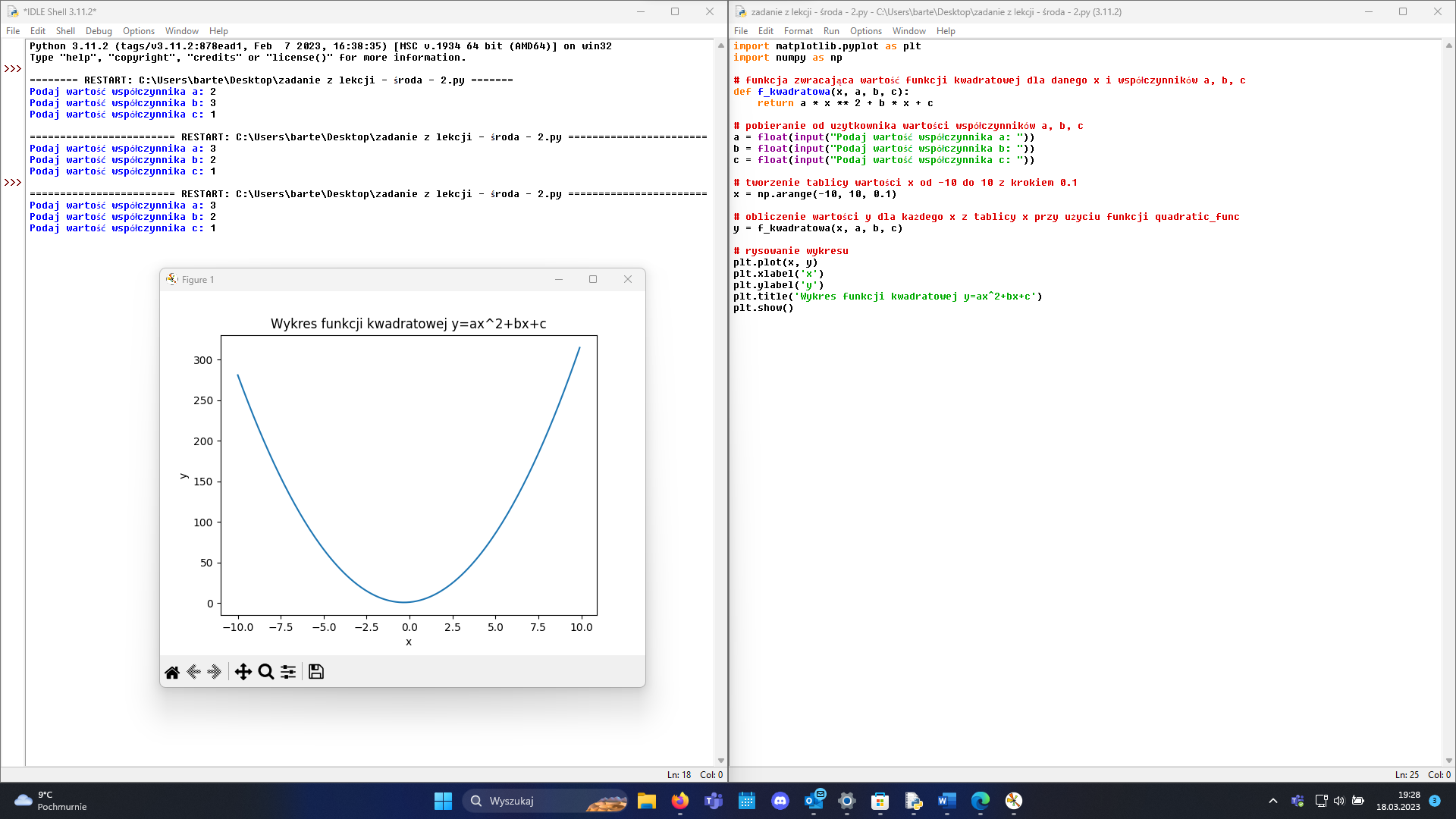
**plt.xlabel('x')**

**plt.ylabel('y')**

**plt.title('Wykres funkcji kwadratowej y=ax^2+bx+c')**

**plt.show()**

Wynik powyższego kodu:

****

**Zadanie 10**

# Definicja funkcji obliczającej silnię w sposób iteracyjny

**def iteracyjny(x):**

**wynik = 1**

**for i in range(1, x+1):**

**wynik \*= i**

**return wynik**

# Definicja funkcji obliczającej silnię w sposób rekurencyjny

**def rekurencyjny(x):**

**if x == 0:**

**return 1**

**else:**

**return x \* rekurencyjny(x-1)**

# Pobranie liczby od użytkownika

**x = int(input("Podaj liczbę, dla której chcesz obliczyć silnię: "))**

# Obliczenie silni dla liczby n przy użyciu

# funkcji iteracyjny i rekurencyjny

**wynik\_iteracyjny = iteracyjny(x)**

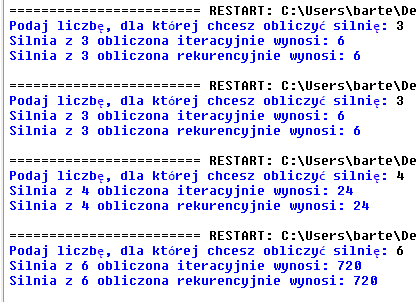
**wynik\_rekurencyjny = rekurencyjny(x)**

# Wypisanie wyników obliczeń na ekran

**print("Silnia z", x, "obliczona iteracyjnie wynosi:", wynik\_iteracyjny)**

**print("Silnia z", x, "obliczona rekurencyjnie wynosi:", wynik\_rekurencyjny)**

Wynik powyższego kodu:

****

**Zadanie 11)**

import os

# definiujemy funkcję, która zapisuje liczby do pliku na podstawie ich znaku

**def save\_number\_to\_file(number):**

**if number < 0:**

# jeśli liczba jest ujemna, zapisujemy ją do pliku minus.txt

**with open('minus.txt', 'a') as file:**

**file.write(str(number) + '\n')**

**elif number > 0:**

# jeśli liczba jest dodatnia, zapisujemy ją do pliku plus.txt

**with open('plus.txt', 'a') as file:**

**file.write(str(number) + '\n')**

# jeśli liczba jest równa 0, to nie zapisujemy jej do pliku

# pętla, w której prosimy użytkownika o podanie liczby i zapisujemy ją do pliku

**while True:**

**number = int(input('Podaj liczbę całkowitą (0 kończy program): '))**

**if number == 0:**

**break** # kończymy program, jeśli wpisano 0

**save\_number\_to\_file(number)**

# wyświetlamy zawartość plików plus.txt i minus.txt

**print('Zawartość pliku plus.txt:')**

**with open('plus.txt', 'r') as file:**

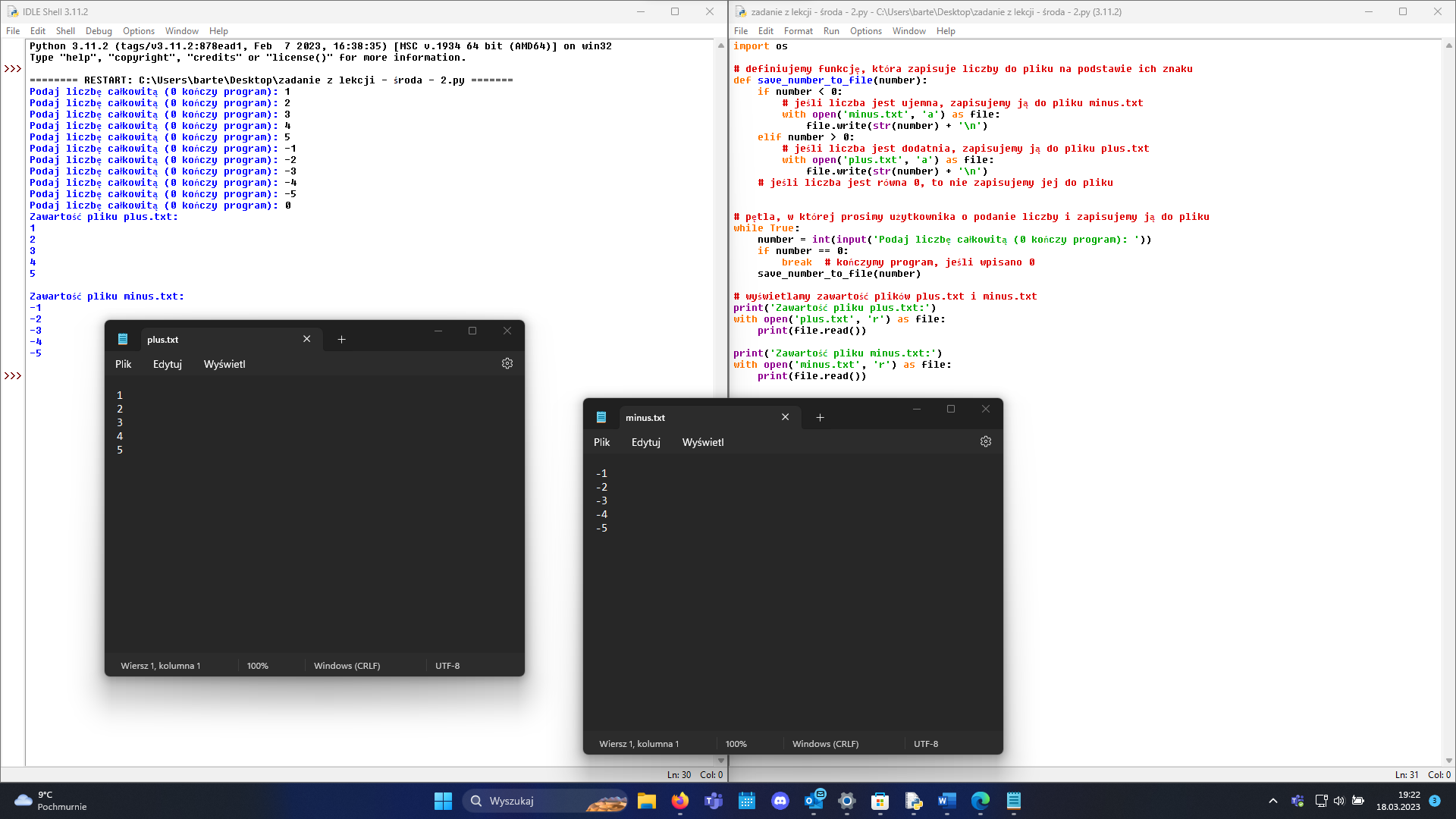
**print(file.read())**

**print('Zawartość pliku minus.txt:')**

**with open('minus.txt', 'r') as file:**

**print(file.read())**

Wynik powyższego kodu:

****