**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №4**

з дисципліни  
«Інтелектуальні вбудовані системи»

на тему  
«Дослідження алгоритму швидкого перетворення фур'є з проріджуванням відліків сигналів у часі»

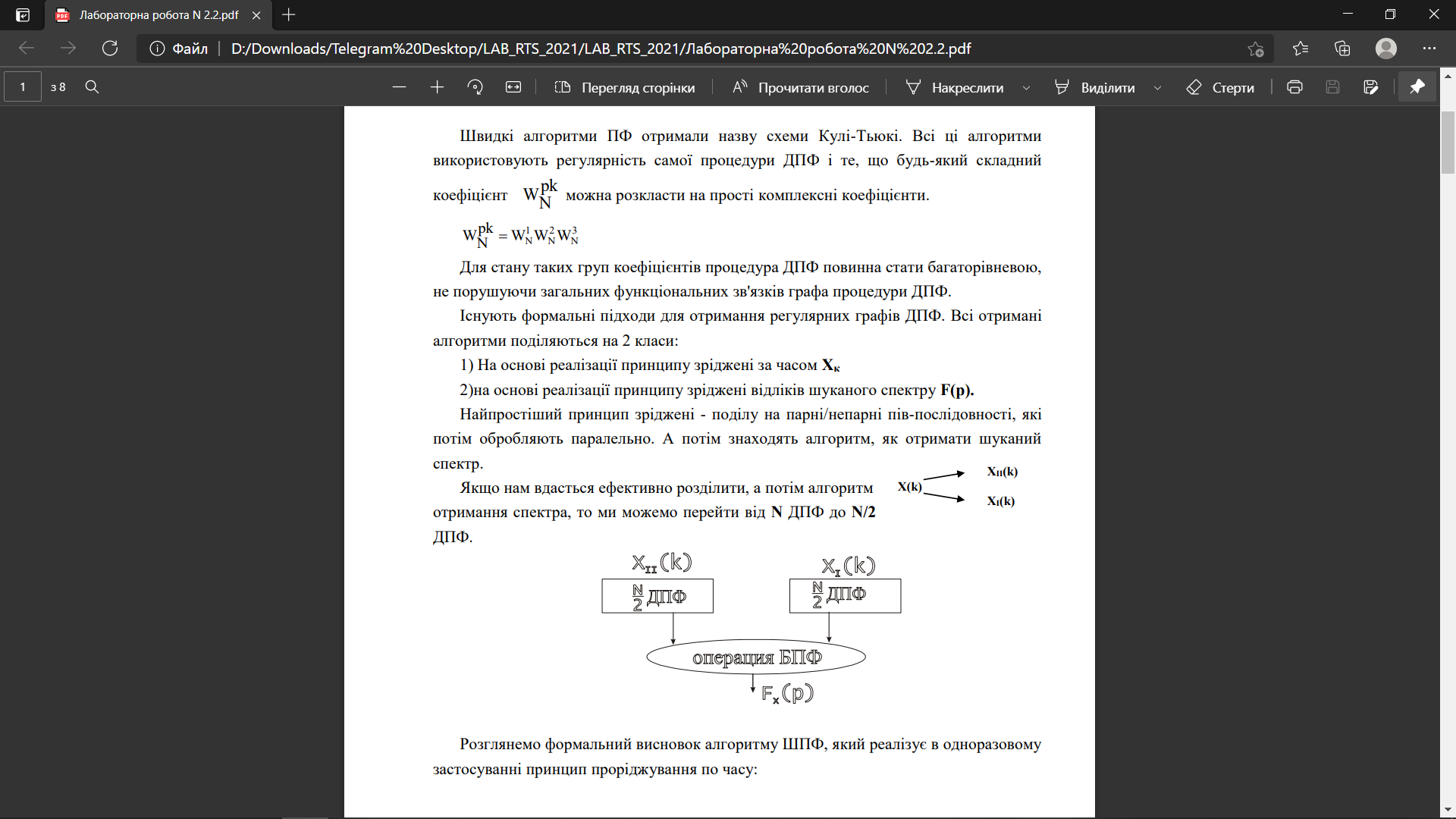
Виконала: Перевірив:

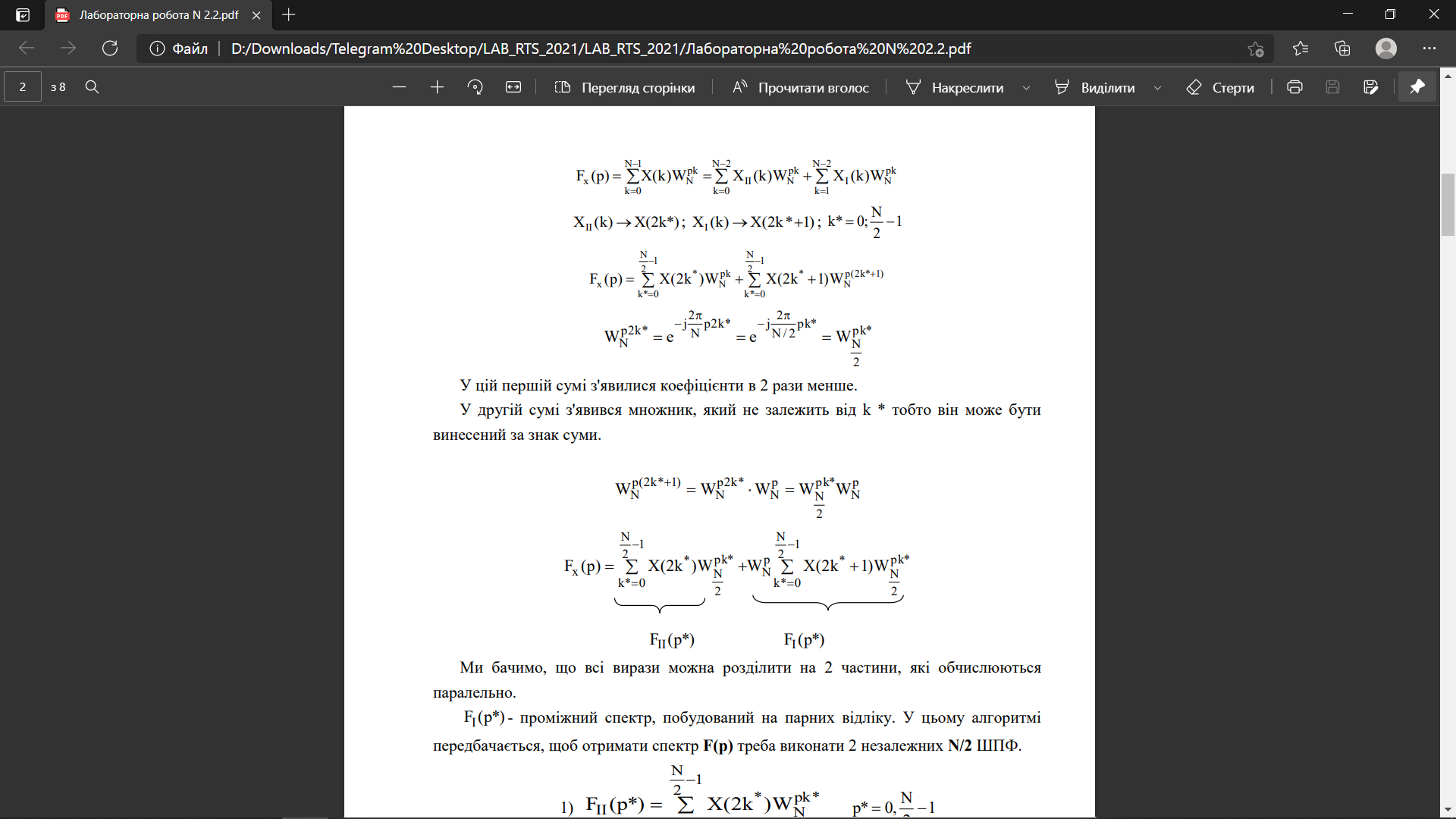
студентка групи ІП-84 ас. Регіда П. Г.

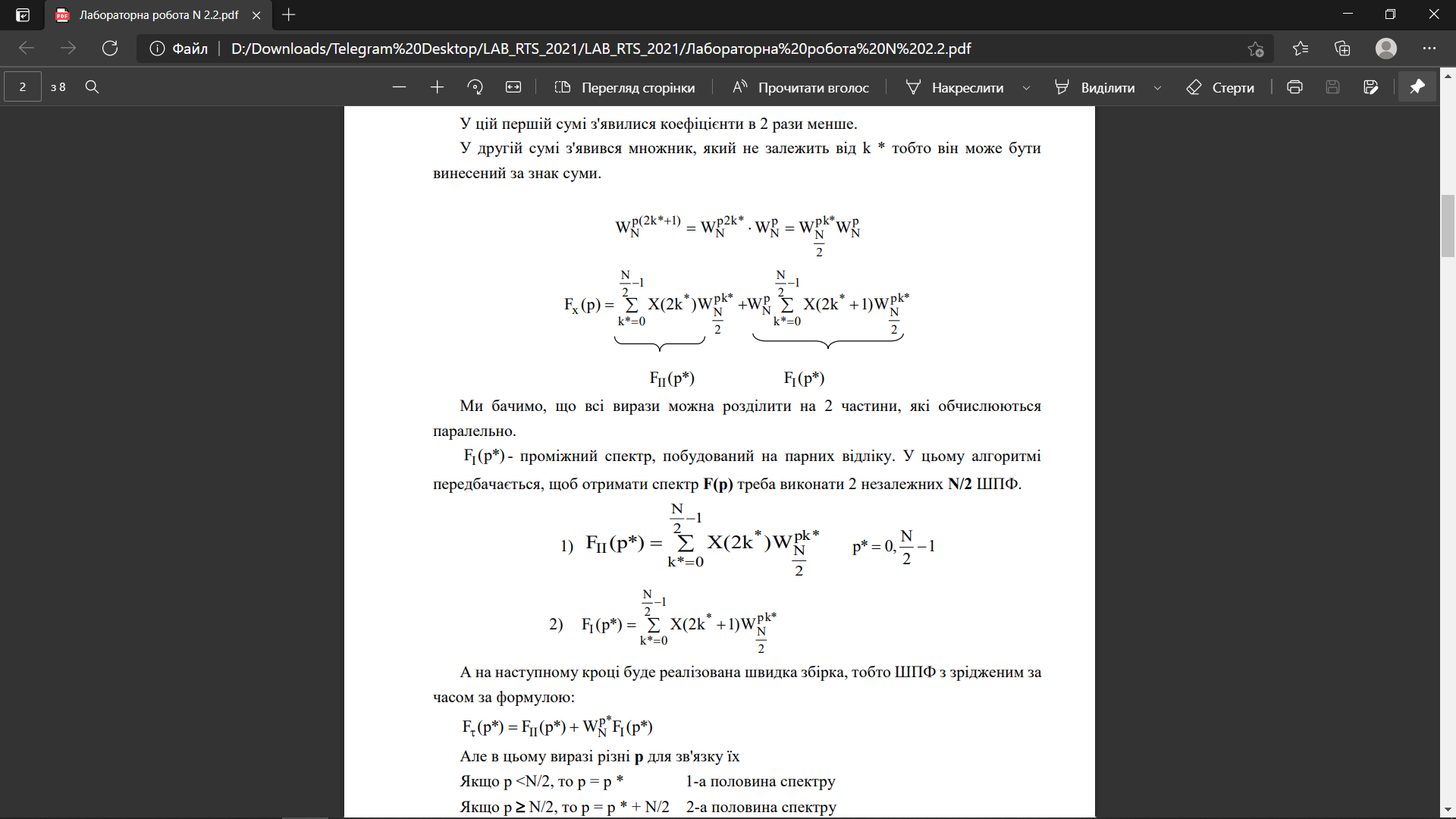
Коломієць Євгенія Валеріївна   
номер залікової книжки: 8412

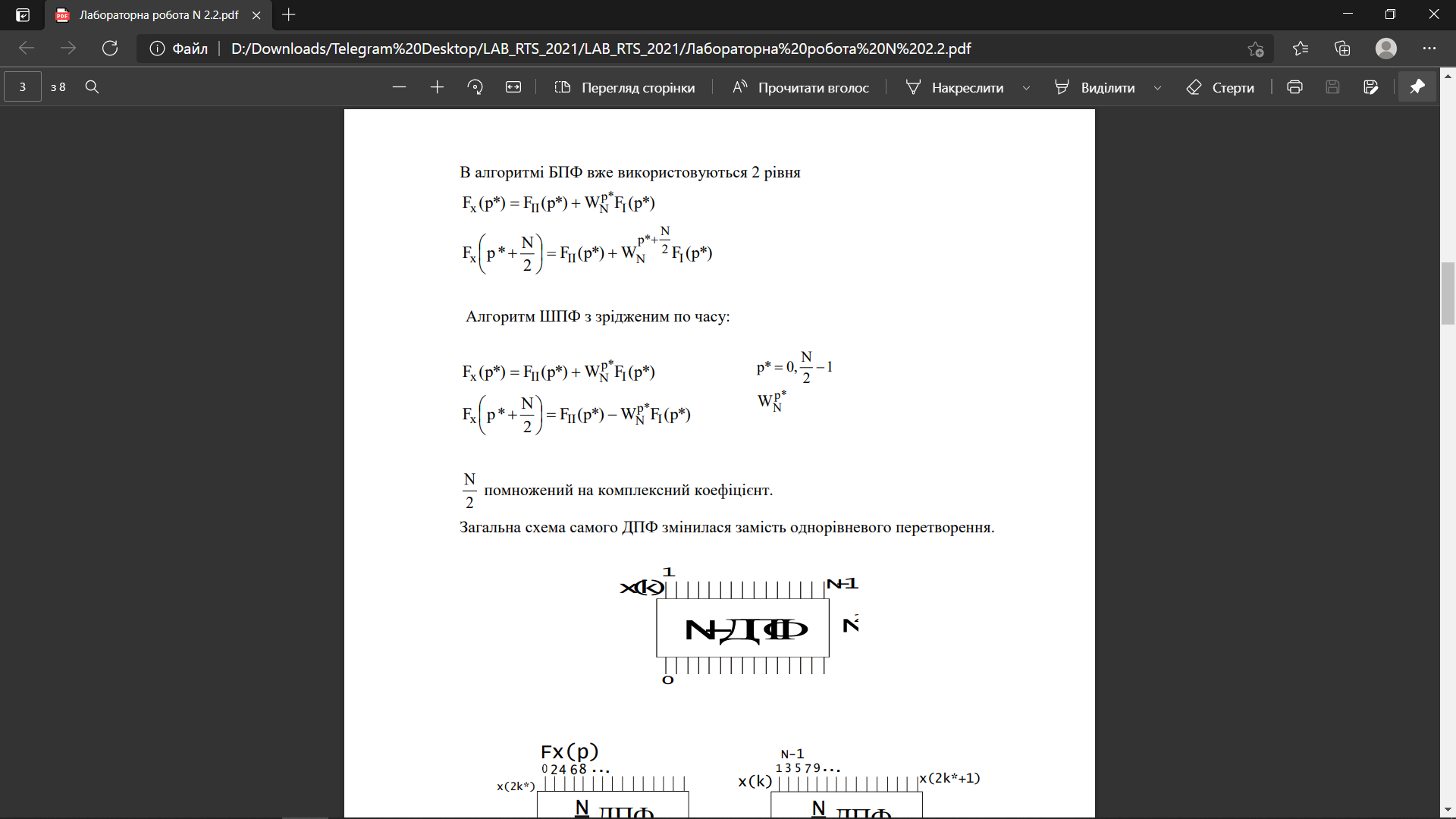
Київ 2020

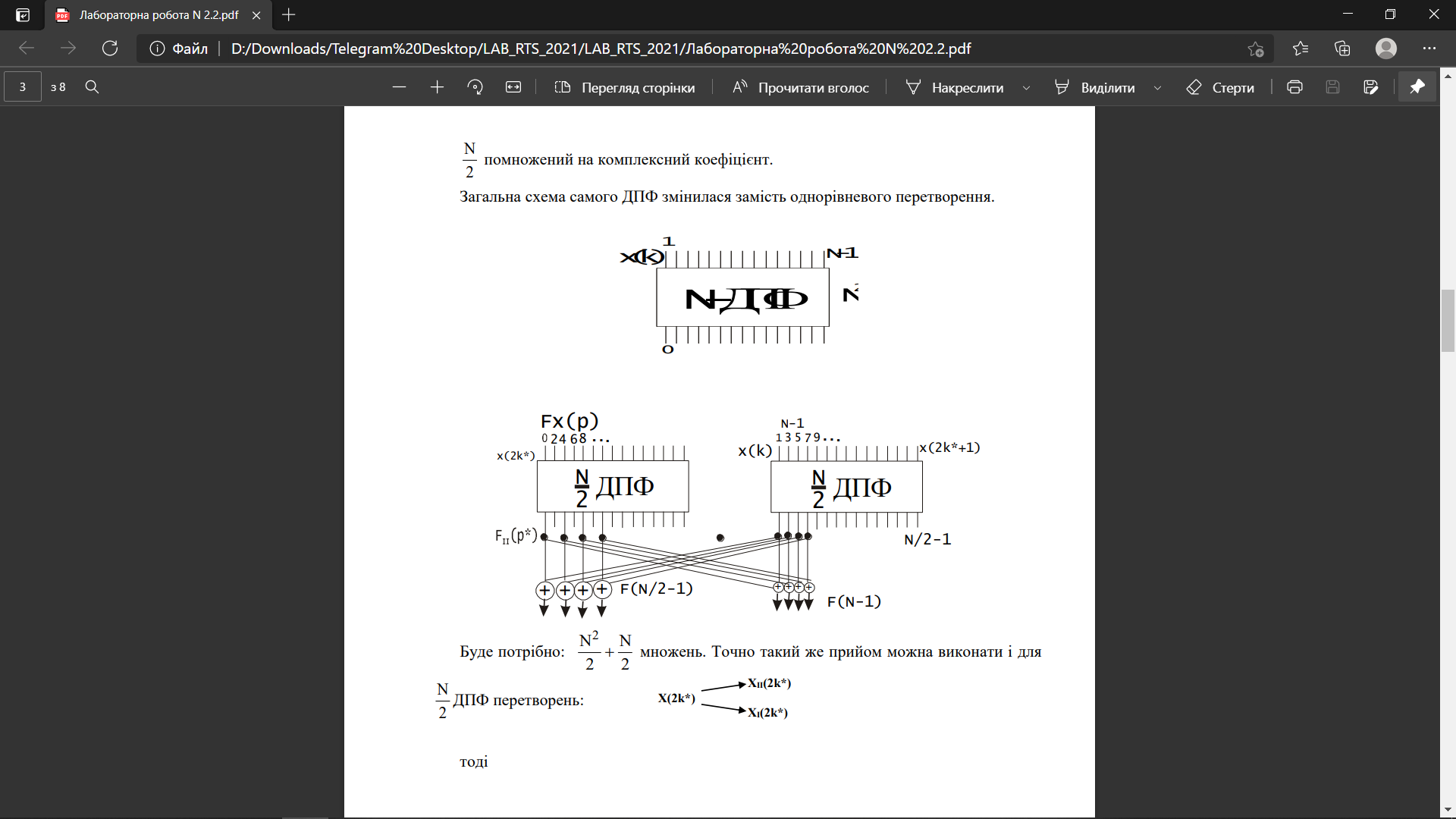
**Основні теоретичні відомості:**

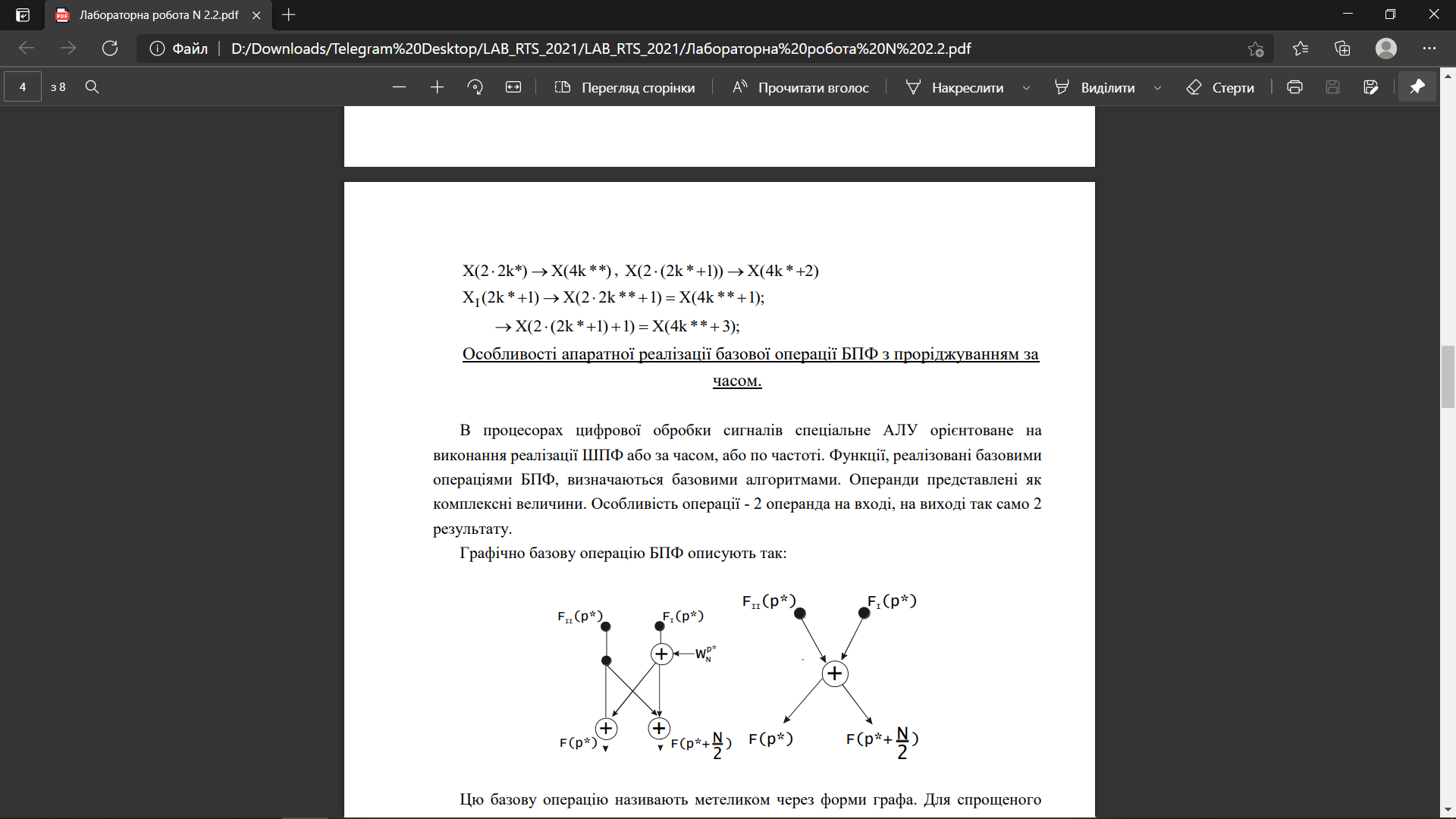












**Завдання за варіантом:**

Варіант 12

Число гармонік в сигналі n = 8

Гранична частота, = 1200

Кількість дискретних відліків, N = 1024

**Лістинг програми**:

**import** random

**import** math

**import** matplotlib**.**pyplot **as** plt

**import** time

**import** numpy **as** np

n **=** 8

tics **=** 1024

max\_amp **=** 1200

delta\_amp **=** max\_amp**/**n

freqs **=** **[]**

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

freq **=** delta\_amp **\*** i

freqs**.**append**(**freq**)**

**def** generate\_params**(**time**):**

amps **=** **[]**

phases **=** **[]**

**for** i **in** **range(**n**):**

phase **=** **[]**

amp **=** **[]**

**for** j **in** **range(**time**):**

amp**.**append**(**random**.**random**())**

phase**.**append**(**random**.**random**())**

amps**.**append**(**amp**)**

phases**.**append**(**phase**)**

**return** **{**"amps"**:** amps**,** "phases"**:** phases**}**

**def** generate\_signal**(**time**):**

res **=** generate\_params**(**time**)**

amps **=** res**[**"amps"**]**

phases **=** res**[**"phases"**]**

signal\_sum **=** **[]**

**for** t **in** **range(**time**):**

signals **=** **[]**

**sum** **=** 0

**for** j **in** **range(**n**):**

signal **=** **(**amps**[**j**][**t**])\*** math**.**sin**(**freqs**[**j**]\***t**+** phases**[**j**][**t**])**

signals**.**append**(**signal**)**

**for** s **in** signals**:**

**sum** **=** **sum** **+** s

signal\_sum**.**append**(sum)**

**return** signal\_sum

signal **=** generate\_signal**(**tics**)**

#functions for plots

**def** plot\_f**(**x**):**

plt**.**xlabel**(**'частота'**)**

plt**.**title**(**'Графік перетвореного сигналу'**)**

plt**.**plot**(range(len(**x**)),** x**,** '#F25278'**)**

plt**.**show**()**

**def** plot\_N**(**x**,** times**):**

plt**.**plot**(**x**,** times**,** color **=** '#51074a'**)**

plt**.**xlabel**(**'час - кількість дискретних відліків'**)**

plt**.**ylabel**(**'час обчислення'**)**

plt**.**title**(**'Графік залежності складності обчислень від часу'**)**

plt**.**show**()**

#main part - function for calculation

**def** fast\_fourier\_transform**(**x**):**

N **=** **len(**x**)**

middle **=** **int(**N**/**2**)**

f **=** **[**0**]** **\*** N

**if** N **<=** 1**:**

**return** x

#using recursive calls

x\_even **=** fast\_fourier\_transform**(**x**[::**2**])**

x\_odd **=** fast\_fourier\_transform**(**x**[**1**::**2**])**

#combining

**for** p **in** **range(**middle**):**

w\_num **=** p**\***2**\***math**.**pi**/**N

w **=** math**.**cos**(**w\_num**)** **-** math**.**sin**(**w\_num**)\***1j

f**[**p**]** **=** x\_even**[**p**]** **+** w**\***x\_odd**[**p**]**

f**[**middle **+** p**]** **=** x\_even**[**p**]** **-** w**\***x\_odd**[**p**]**

**return** f

res **=** fast\_fourier\_transform**(**signal**)**

**for** i **in** **range(len(**res**)):**

res**[**i**]** **=** **abs(**res**[**i**])**

plot\_f**(**res**)**

#building time complexity

min\_ticks **=** 100

max\_tick **=** 10000

times **=** **[]**

N **=** **[]**

**for** tic **in** **range(**min\_ticks**,** max\_tick**+**1**,** 100**):**

signal **=** generate\_signal**(**tic**)**

N**.**append**(**tic**)**

start\_time **=** time**.**perf\_counter**()**

fast\_fourier\_transform**(**signal**)**

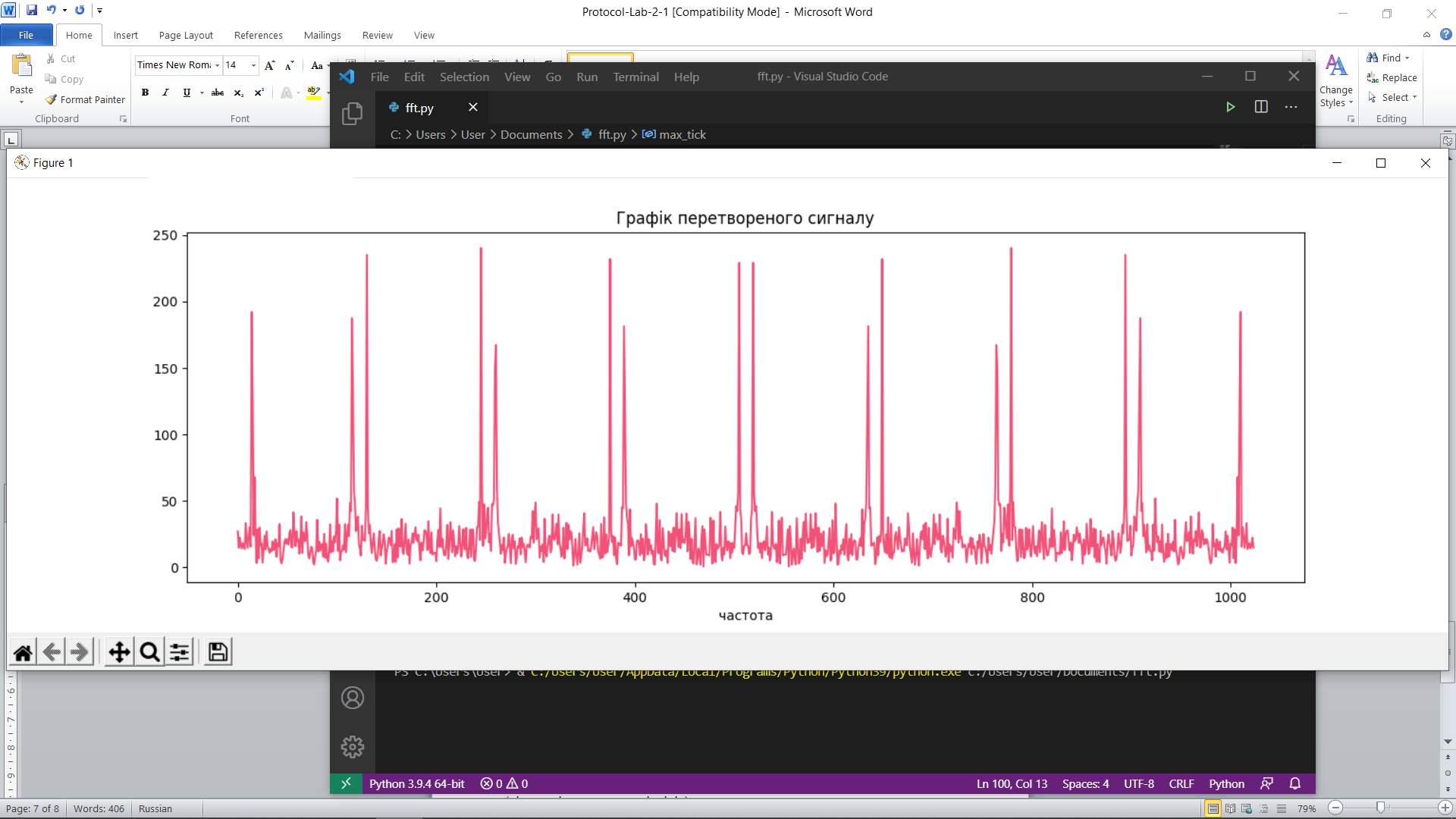
end\_time **=** time**.**perf\_counter**()**

times**.**append**(**end\_time **-** start\_time**)**

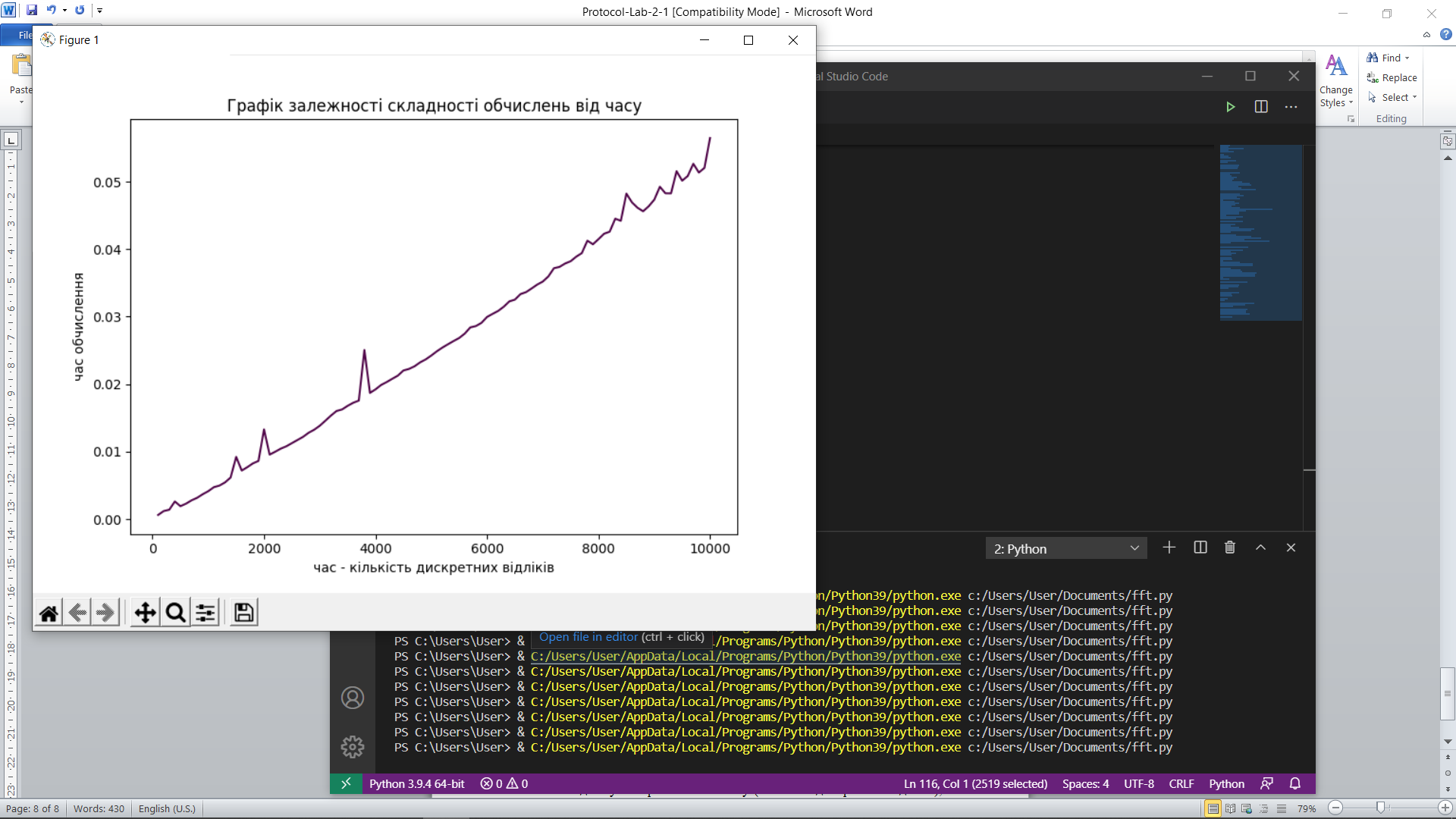
plot\_N**(**N**,** times**)**

**Приклад роботи програми**:

1) Згенерований сигнал після дискретного перетворення Фур’є:



2) Результат вимірювання складності обчислень в залежності від часу (кількості дискретних відліків)



**Висновки**:

Під час виконання лабораторної роботи було досліджено принципи реалізації спектрального аналізу на основі алгоритму швидкого перетворення Фур'є (Кулі-Тьюкі). Було реалізовано алгоритм та перетворено функцію сигналу на його спектр, використовуючи процедуру швидкого перетворення Фур'є з проріджуванням відліків сигналу за часом..

Також було проведено дослідження залежності складності обчислень від часу вимірювання сигналу (кількості дискретних відліків).