

Міністерство освіти й науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
з предмета: ***Комп'ютерна лінгвістика***

Звіт
про виконання лабораторної роботи № 1
«Препроцесинг текстових документів»

Виконав:
Студент групи
Фес-32с
Бойко Кирило

Львів 2024

Завдання

використовуючи видані програми, здійснити попереднє опрацювання окремих текстів і текстових баз різними мовами; порівняти функціонал та інтерфейс різних програм, їхні переваги та недоліки; дослідити залежність часу опрацювання великих текстових баз від їхнього розміру.

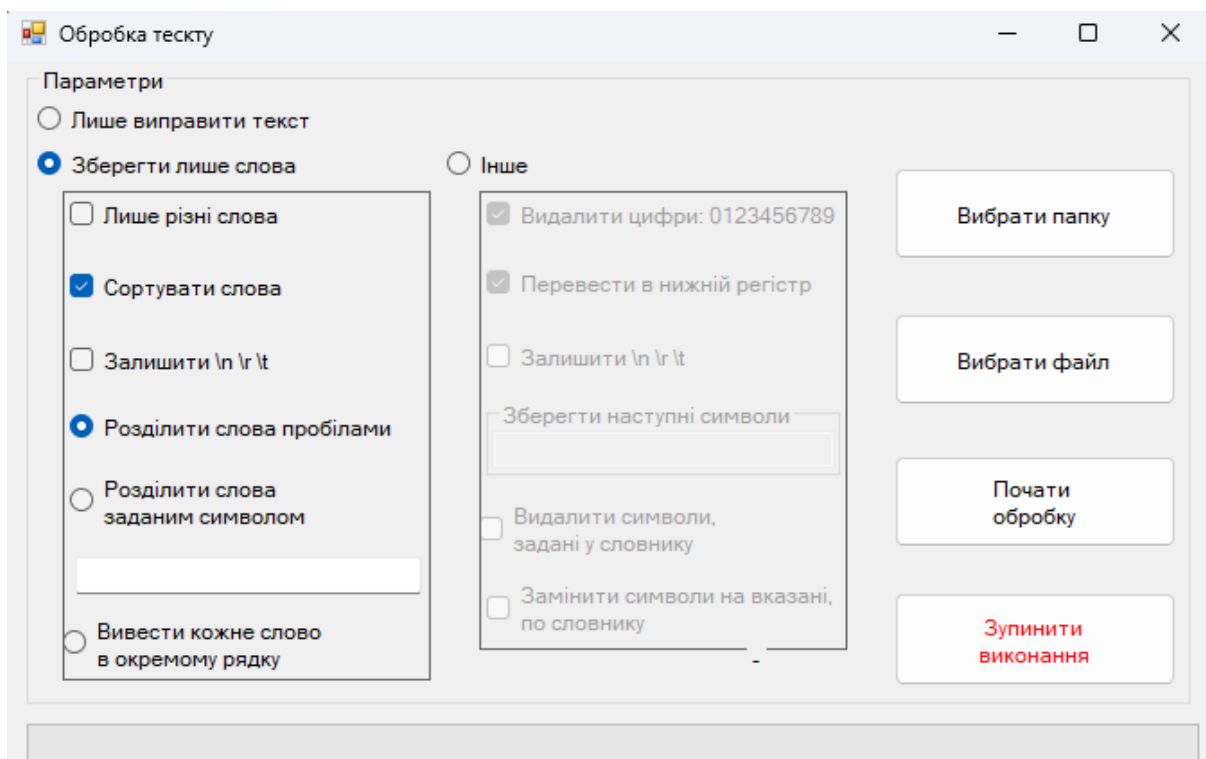
Хід роботи

Перед початком виконання лабораторної роботи, я обрав одну текстову базу, та один досліджуваний текст з архіву **main text corora2023-24.zip**

Clemencia Novela de costumbres by Fernán Caballero

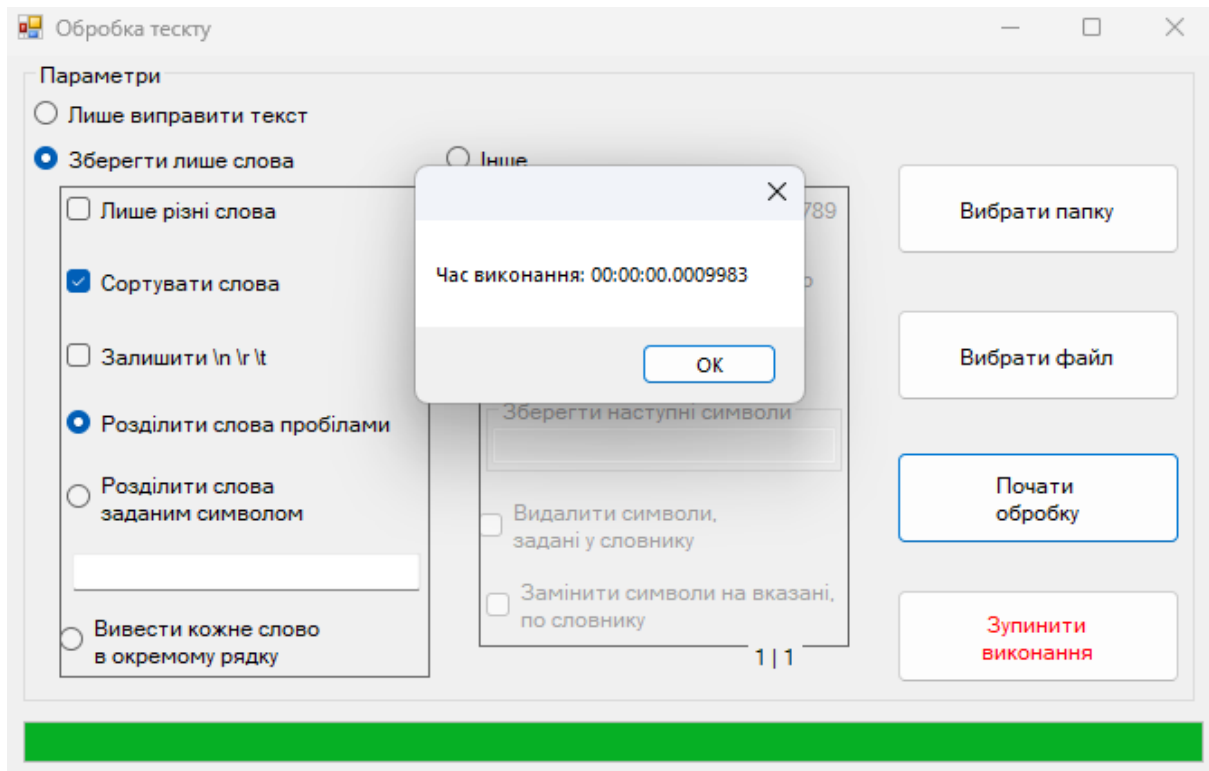
DONALD J. TRUMP January 20, 2017

Я ініціював програму **+Text cleaner&processor (main)** для обробки тексту, завантажив файл **Clemencia Novela de costumbres by Fernán Caballero.txt** і встановив такі параметри для обробки:

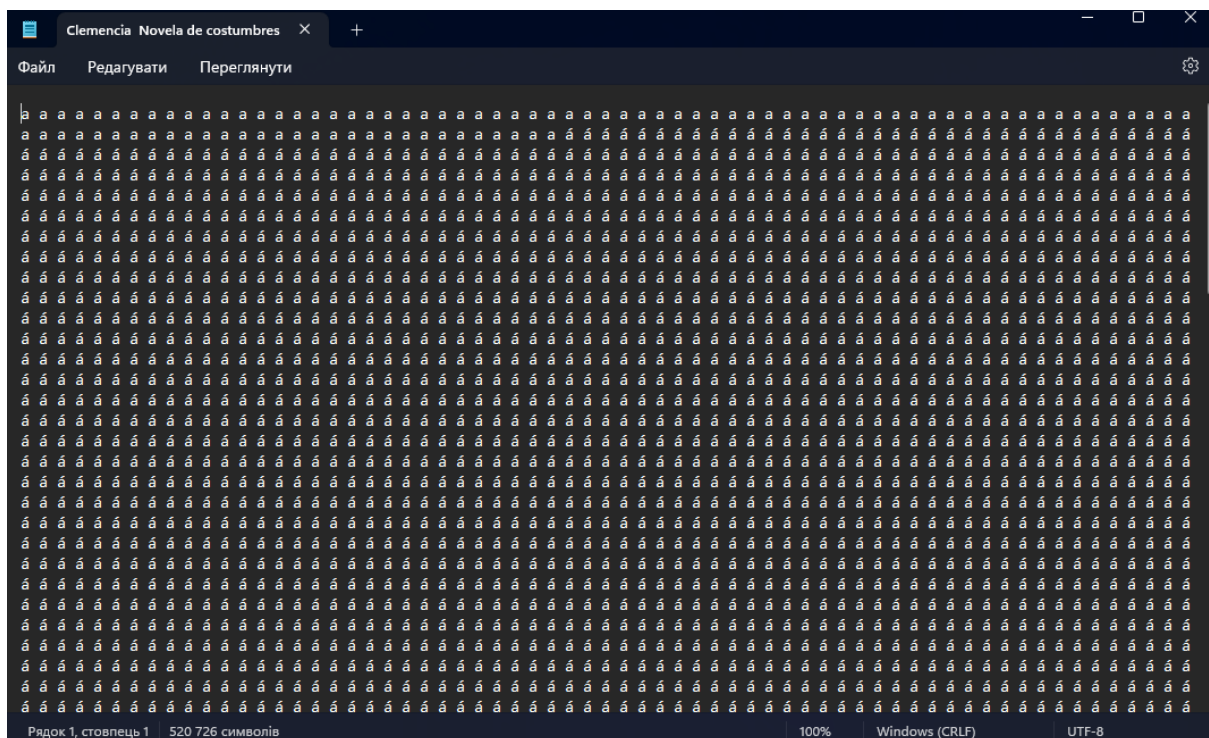


Інтерфейс утиліти

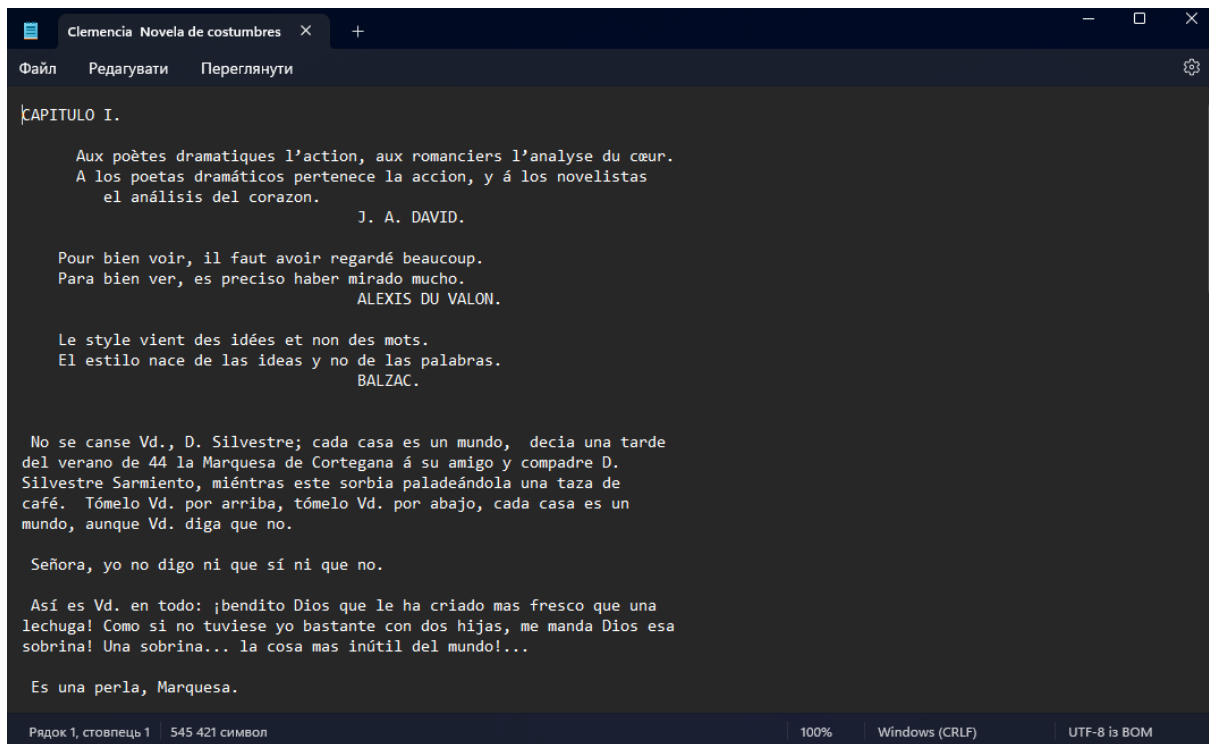
Результати



Інтерфейс утиліти після виконання роботи

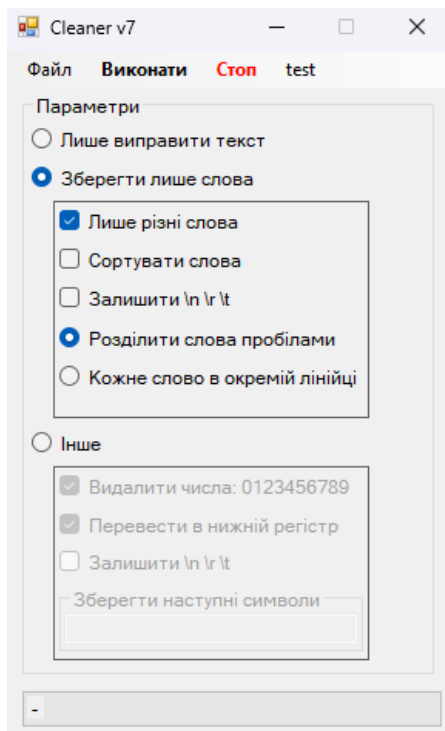


Оброблений файл



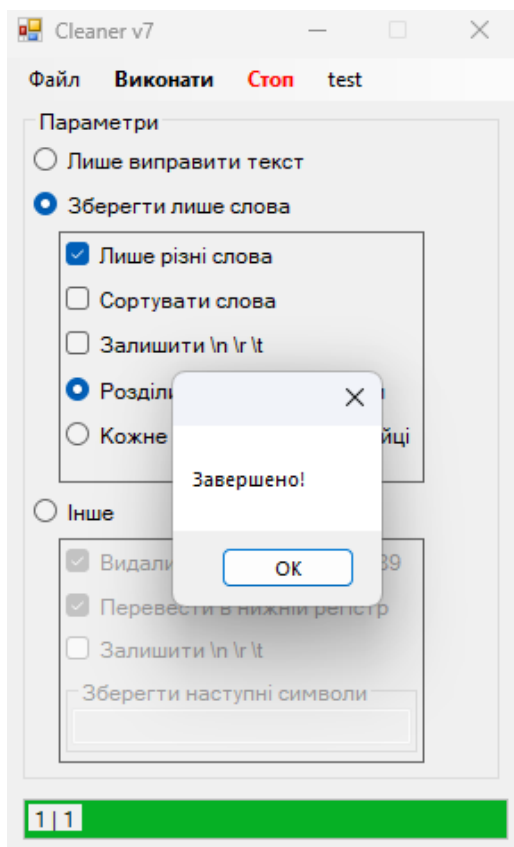
Оригінальний текстовий файл

Тепер виконую ті самі дії в програмі **+Text cleaner(for English only)** з файлом **DONALD J. TRUMP January 20, 2017.txt** та вибраними параметрами:

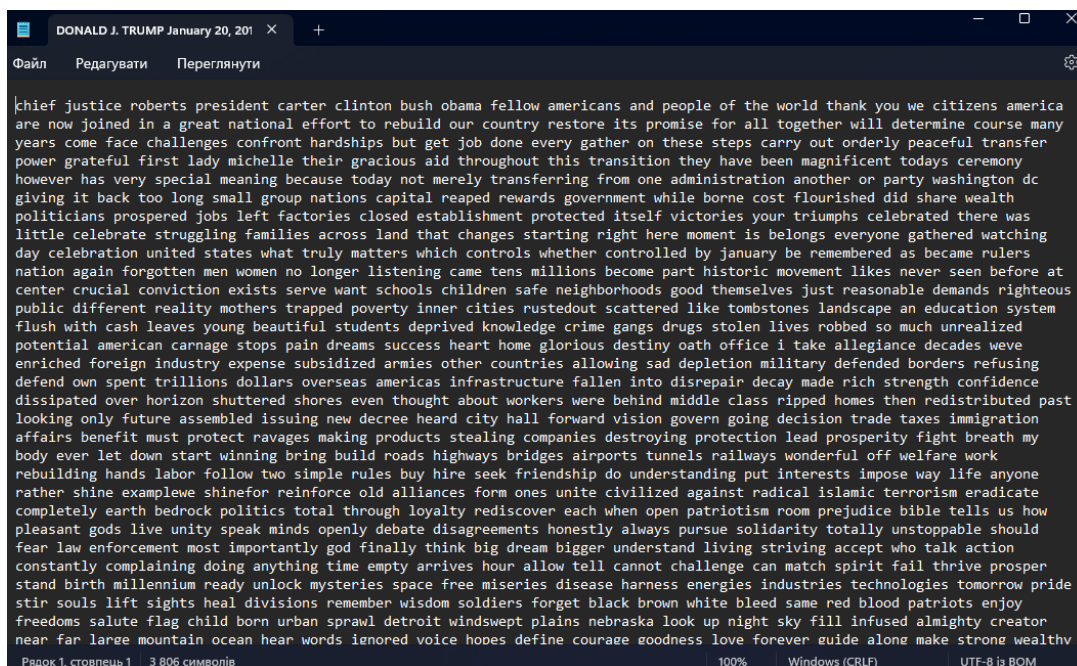


Інтерфейс програми та вибрані параметри

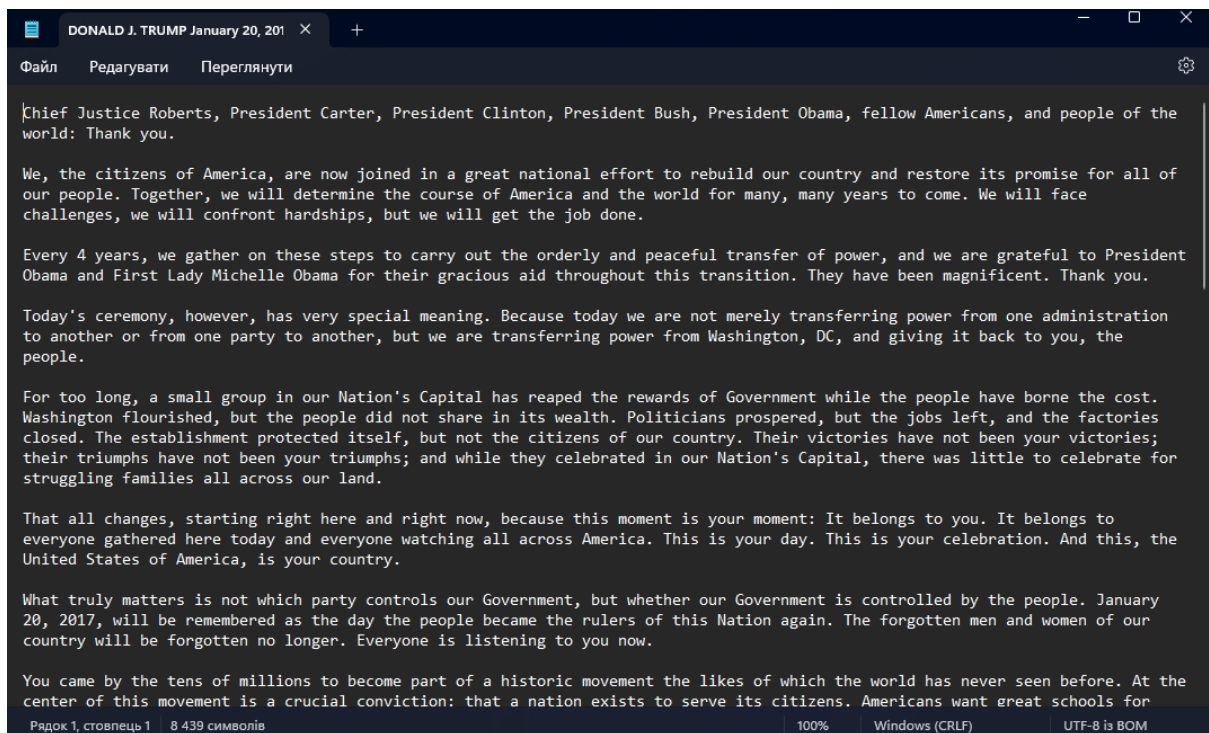
Результати



Результат виконання роботи



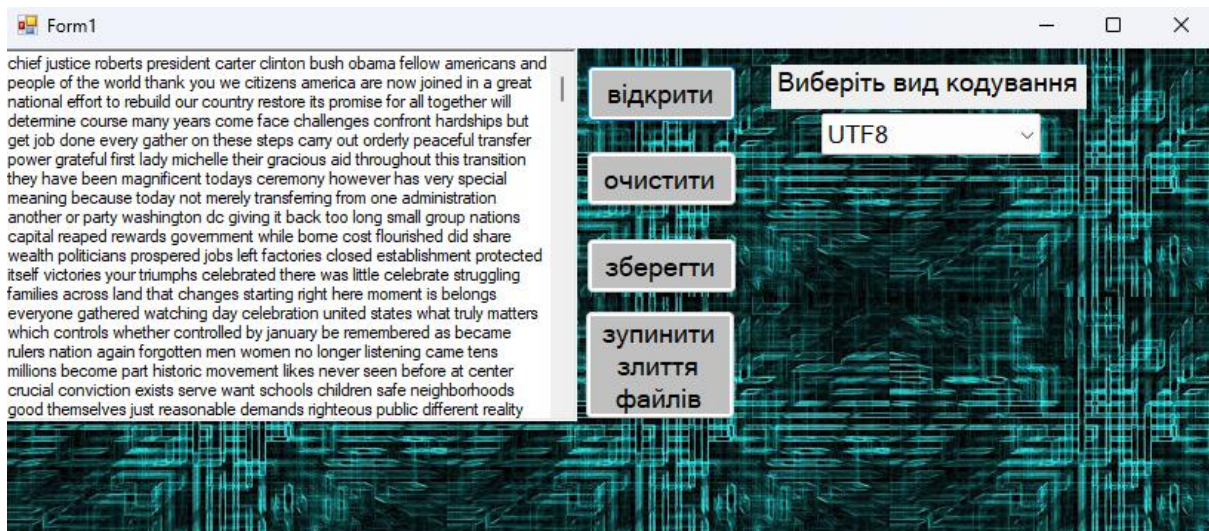
Текст після проведення чистки в програмі



Оригінальний текст

Щоб об'єднати два текстові файли в один, я запускаю програму **+Text merger2022** та вибираю попередньо оброблені тексти

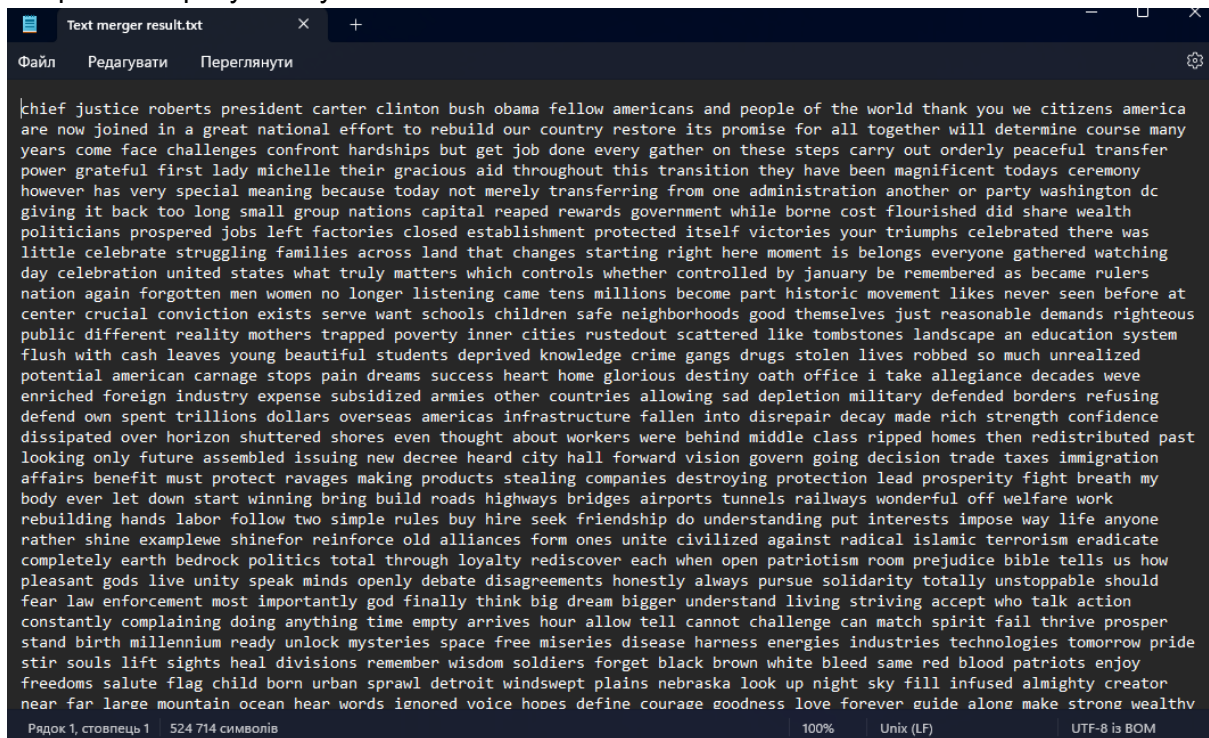
Результати



Інтерфейс програми



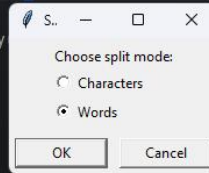
Збереження результату



Результат злиття двох текстів

Для поділу двох файлів я відкрив програму **+TextSplitter(2halves)**, вказав папку зі злитими файлами та вибрав місце для збереження результату.

```
+Splitter2parts_word&char6fin.py x
28
29 @'
30     def apply(self):
31         self.result = self.var.get()
32
33 1 usage
34 def get_control_chars():
35     all_chars = (chr(i) for i in range(sys.maxunicode))
36     control_chars = ''.join(
37         c for c in all_chars if unicodedata.category
38     )
39     return control_chars
40
41 1 usage
42 def remove_control_chars(s):
43     control_chars = get_control_chars()
44     control_char_re = re.compile('[%s]' % re.escape(control_chars))
45     return control_char_re.sub( repl: ' ', s)
46
47 1 usage
48 def split_by_words(text):
49     words = text.split()
50     half_words = len(words) // 2
51     return [' '.join(words[:half_words]), ' '.join(words[half_words:])]
52
53 1 usage
54 def split_by_chars(text):
```



Интерфейс програми

Результати

The image shows a Python script in a file named `+Splitter2parts_word&char6fin.py` and its output in a text merger window titled `1_text merger result.txt`.

The Python script defines a class with the following methods:

- `apply(self)`: Calls `self.result = self.var.get()`.
- `get_control_chars()`: Returns a string of control characters by iterating over `sys.maxunicode` and checking `unicodedata.category(c) == 'Cc'`.
- `remove_control_chars(s)`: Removes control characters from a string `s` using a regular expression `re.compile('[%s]' % control_chars)`.
- `split_by_words(text)`: Splits text into words and returns two halves joined by a space.
- `split_by_chars(text)`: (Partially visible)

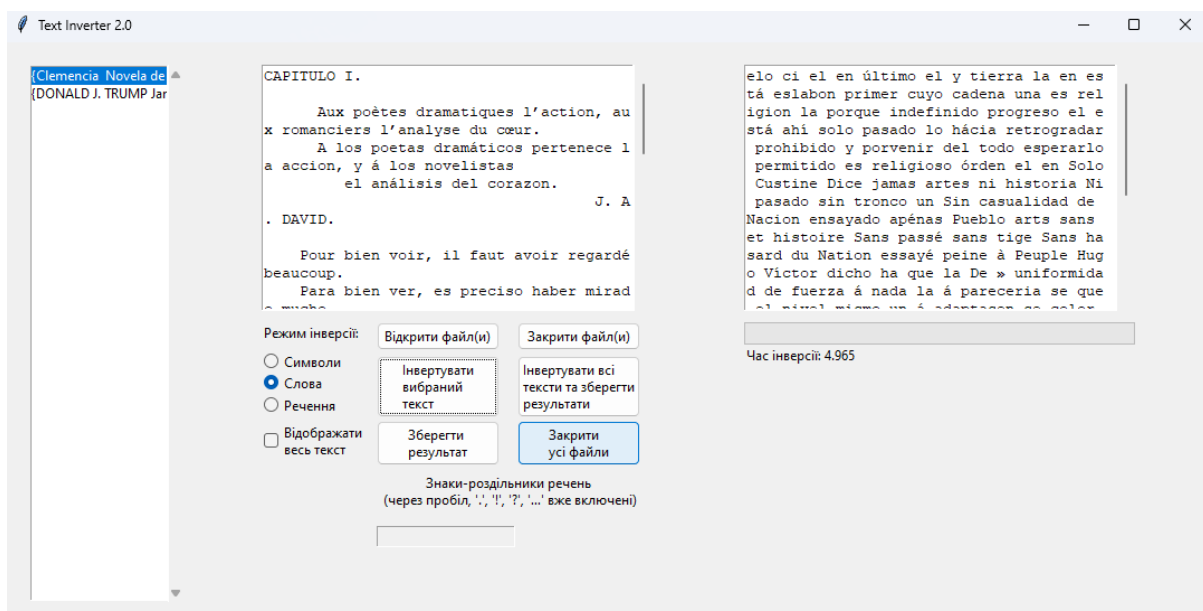
The text merger window displays the output of the script, which is a single line of text containing a large block of words and phrases, including:

chief justice roberts president carter clinton bush obama fellow americans and people of the world thank you we citizens america are now joined in a great national effort to rebuild our country restore its promise for all together will determine course many years come face challenges confront hardships but get job done every gather on these steps carry out orderly peaceful transfer power grateful first lady michelle their gracious aid throughout this transition they have been magnificent todays ceremony however has very special meaning because today not merely transferring from one administration another or party washington dc giving it back too long small group nations capital reaped rewards government while borne cost flourished did share wealth politicians prospered jobs left factories closed establishment protected itself victories your triumphs celebrated there was little celebrate struggling families across land that changes starting right here moment is belongs everyone gathered watching day celebration united states what truly matters which controls whether controlled by january be remembered as became rulers nation again forgotten men women no longer listening came tens millions become part historic movement likes never seen before at center crucial conviction exists serve want schools children safe neighborhoods good themselves just reasonable demands righteous public different reality mothers trapped poverty inner cities rustedout scattered like tombstones landscape an education system flush with cash leaves young beautiful students deprived knowledge crime gangs drugs stolen lives robbed so much unrealized potential american carnage stops pain dreams success heart home glorious destiny oath office i take allegiance decades weve enriched foreign industry expense subsidized armies other countries allowing sad depletion military defended borders refusing defend own spent trillions dollars overseas americas infrastructure fallen into disrepair decay made rich strength confidence dissipated over horizon shuttered shores even thought about workers were behind middle class ripped homes then redistributed past looking only future assembled issuing new decree heard city hall forward vision govern going decision trade taxes immigration affairs benefit must protect ravages making products stealing companies destroying protection lead prosperity fight breath my body ever let down start winning bring build roads highways bridges airports tunnels railways wonderful off welfare work rebuilding hands labor follow two simple rules buy hire seek friendship do understanding put interests impose way life anyone rather shine examplewe shinefor reinforce old alliances form ones unite civilized against radical islamic terrorism eradicate completely earth bedrock politics total through loyalty rediscover each when open patriotism room prejudice bible tells us how pleasant gods live unity speak minds openly debate disagreements honestly always pursue solidarity totally unstoppable should fear law enforcement most importantly god finally think big dream bigger understand living striving accept who talk action constantly complaining doing anything time empty arrives hour allow tell cannot challenge can match spirit fail thrive prosper stand birth millennium ready unlock mysteries space free miseries disease harness energies industries technologies tomorrow pride stir souls lift sights heal divisions remember wisdom soldiers forget black brown white bleed same red blood patriots enjoy freedoms salute flag child born urban sprawl detroit windswept plains nebraska look up night sky fill infused almighty creator near far large mountain ocean hear words ignored voice hopes define courage goodness love forever guide along make strong wealth

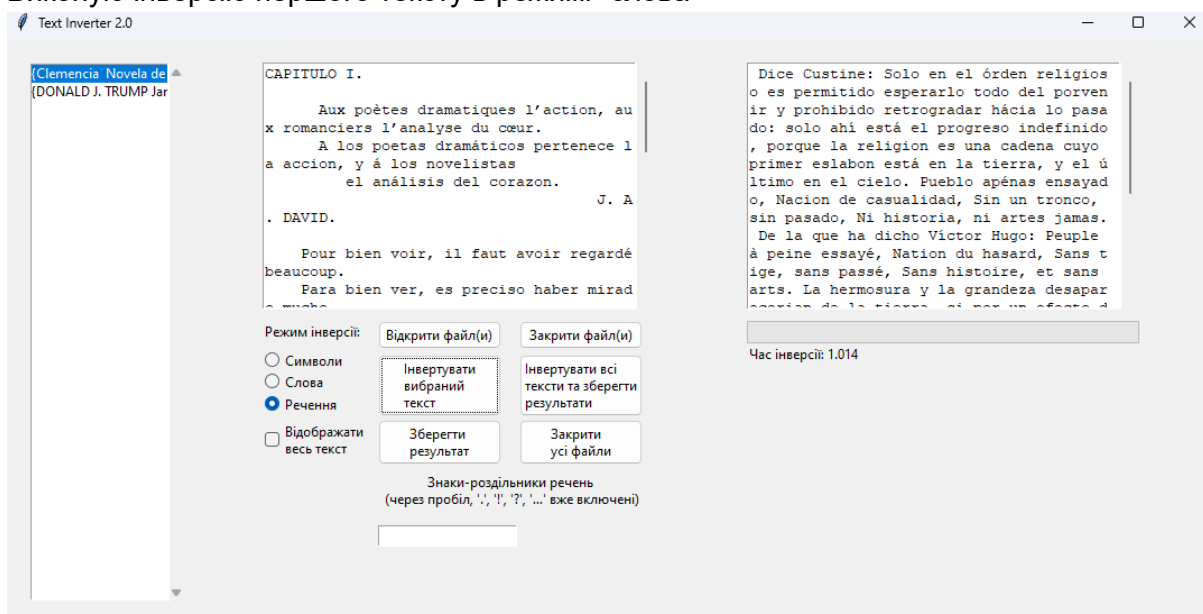
The text merger window also shows the status bar: `Рядок 1, столбець 1 272 643 символи 100% Windows (CRLF) UTF-8 is BOM`.

Результат розбиття 1 тексту

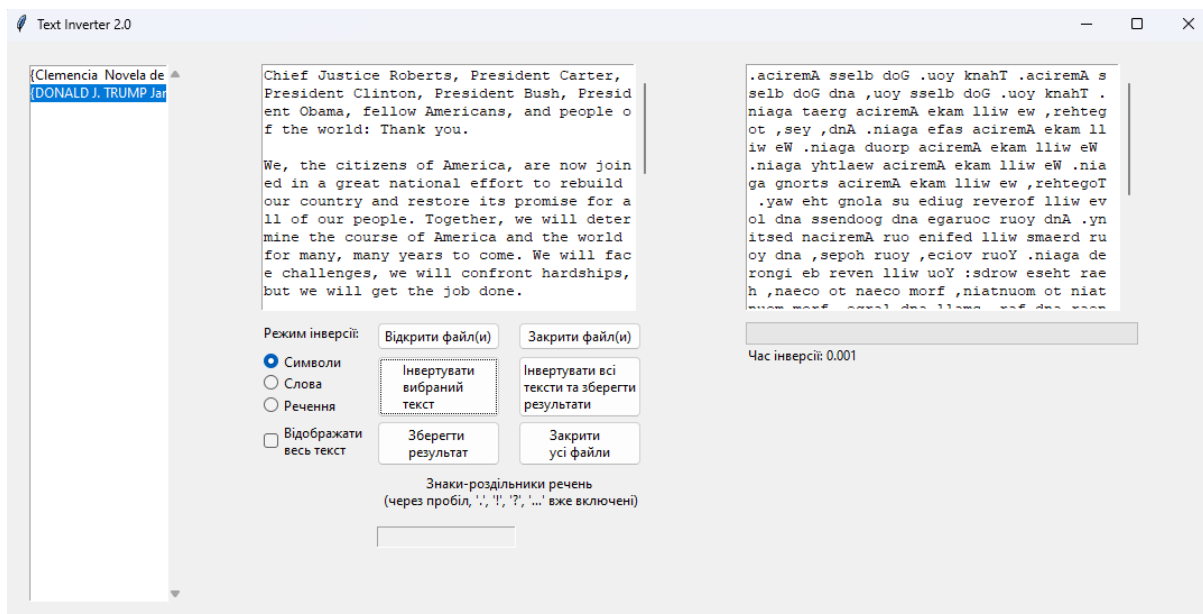
Виконую інверсію першого тексту в режимі "символи"



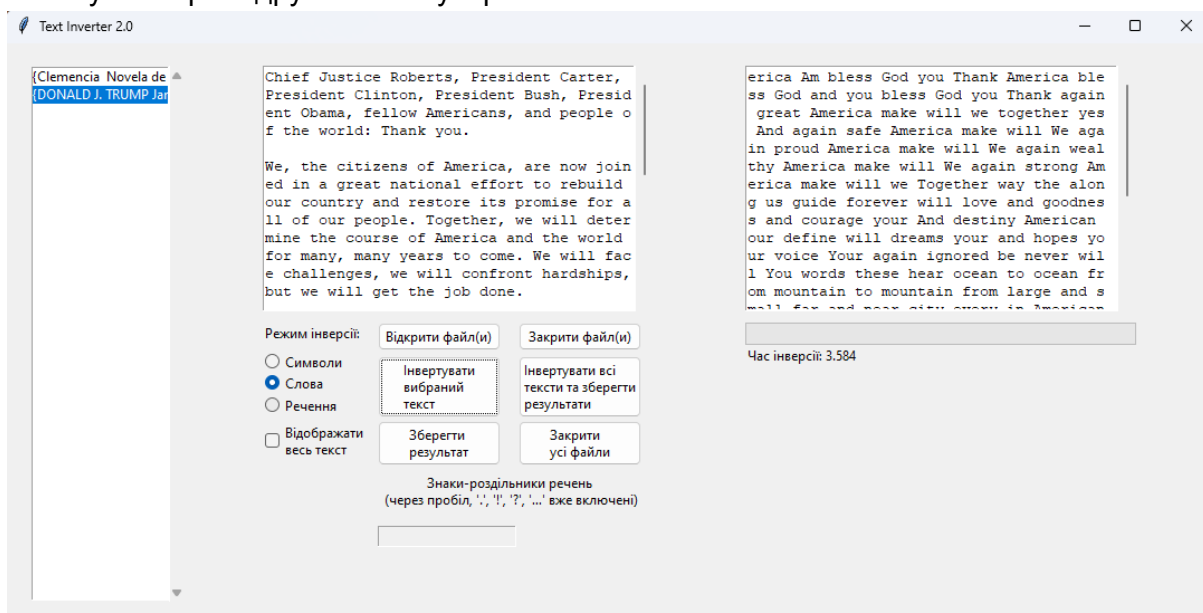
Виконую інверсію першого тексту в режимі "слова"



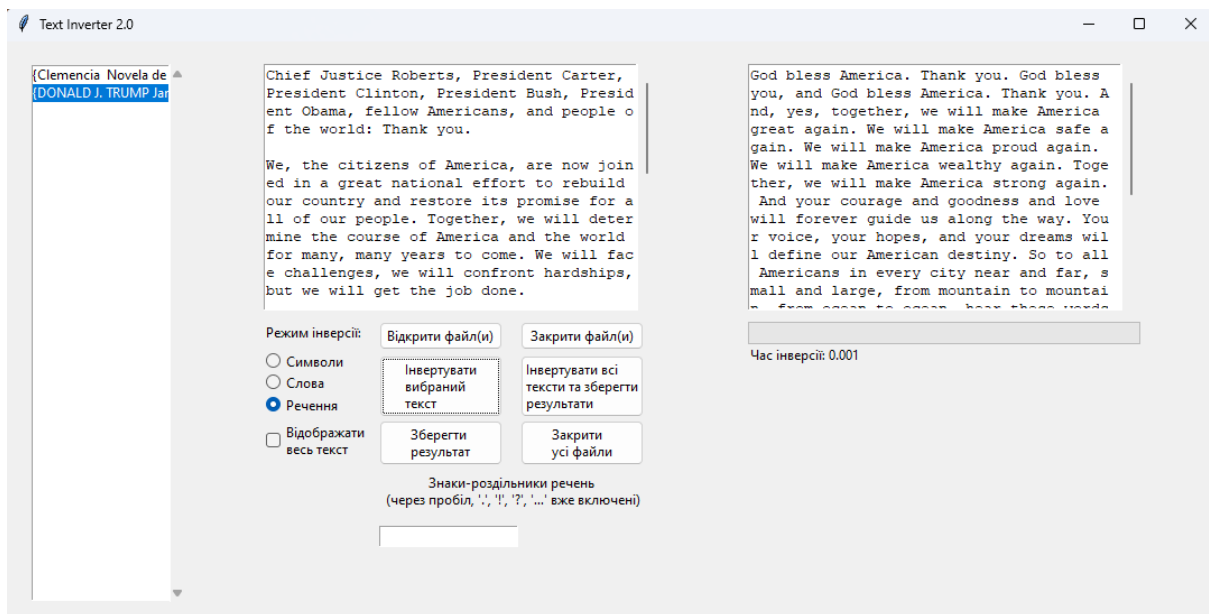
Виконую інверсію першого тексту в режимі "речення"



Виконую інверсію другого тексту в режимі "символи"

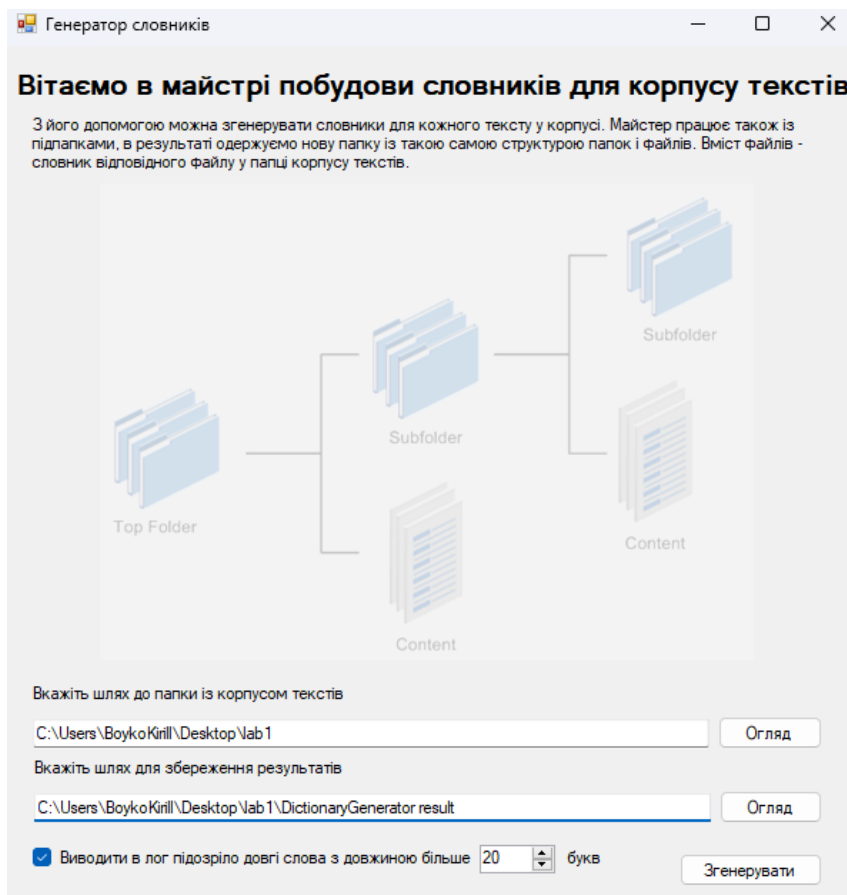


Виконую інверсію другого тексту в режимі "слова"

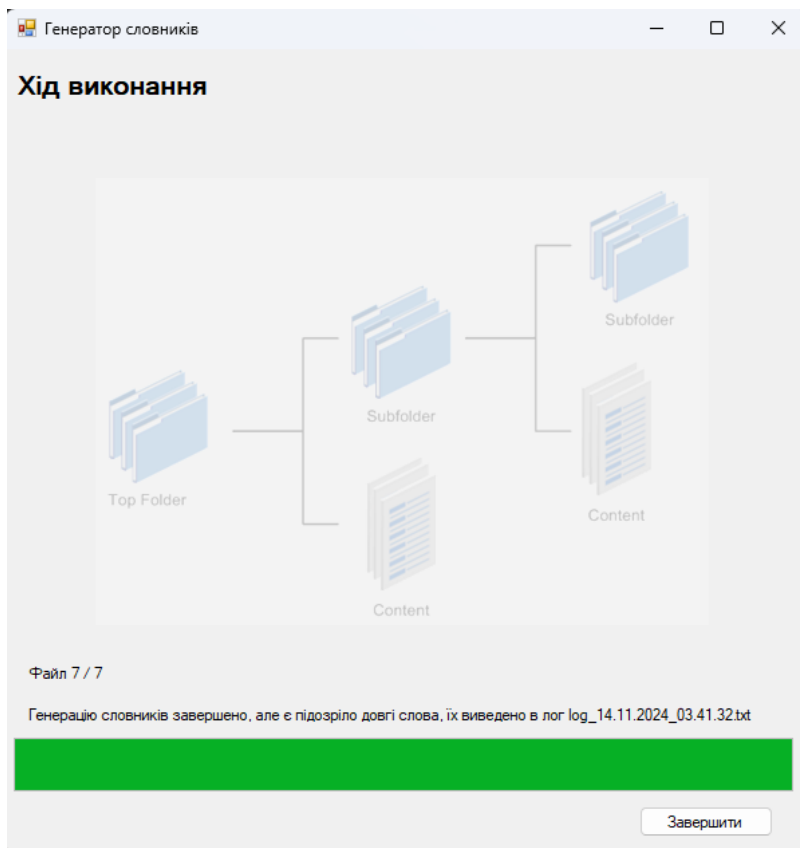


Виконую інверсію другого тексту в режимі "речення"

Для створення словника я використав утиліту **DictionaryGenerator**

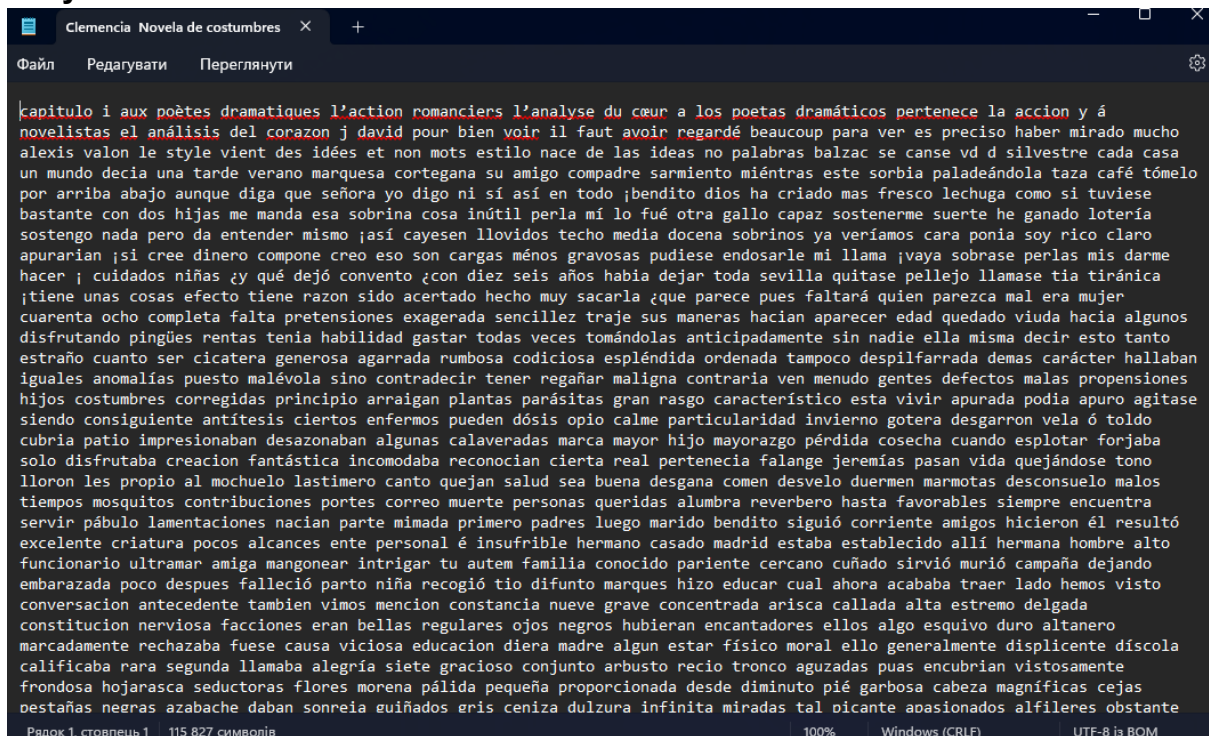


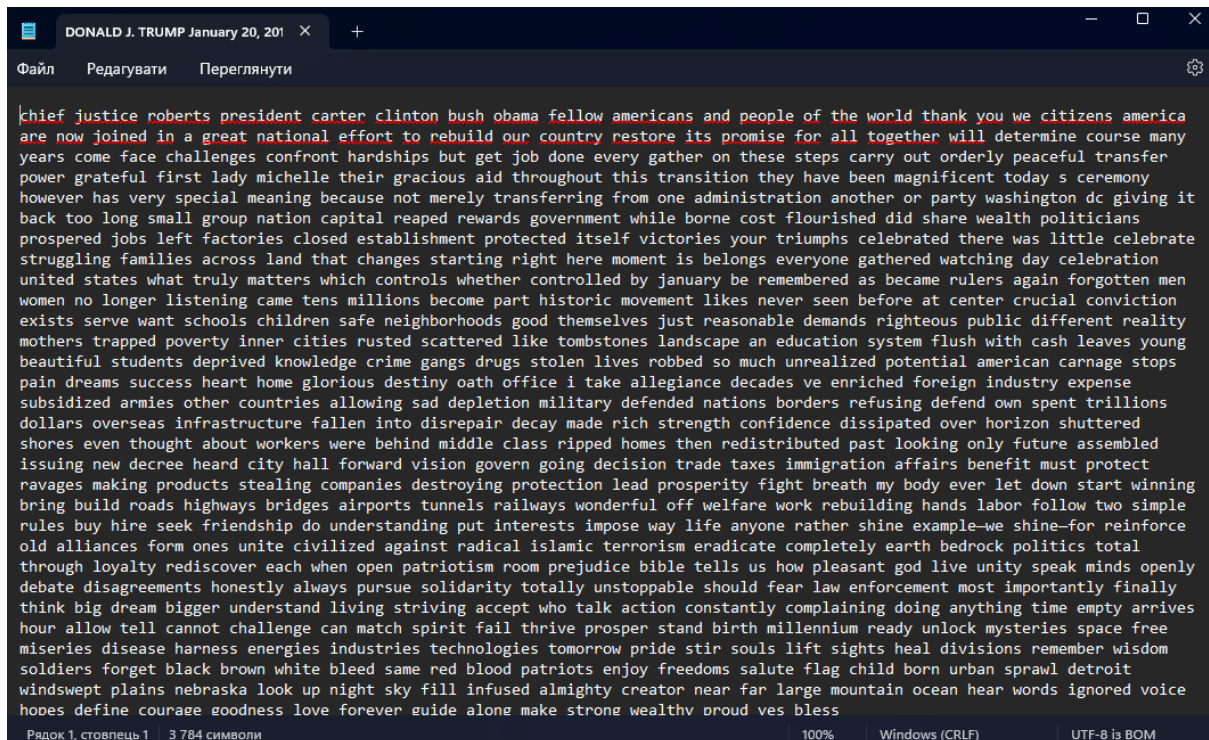
Інтерфейс програми



Інтерфейс програми вже з виконаною роботою

Результати





Зменшуючи розмір файлу поступово, я отримав відповідні результати та розробив програму для побудови графіків залежностей: $t(L)$, $\log t(L)$, $t(\log L)$ і $\log t(\log L)$ на основі зібраних даних

Розмір файлу (L, МБ)	Час обробки (t, секунди)
1.5	0.07
0.8	0.02
0.2	0.004

Код програми:

```
import matplotlib.pyplot as plt
import numpy as np

sizes = np.array([1.5, 0.8, 0.2])
times = np.array([0.07, 0.02, 0.004])

fig, axs = plt.subplots(2, 2, figsize=(12, 8))

axs[0, 0].plot(sizes, times, 'o-', color='purple', label='t(L)',
markersize=8)
axs[0, 0].set_xlabel("Розмір файлу (L, МБ)")
axs[0, 0].set_ylabel("Час обробки (t, с)")
axs[0, 0].set_title("Графік t(L)")
axs[0, 0].legend()

axs[0, 1].plot(sizes, np.log(times), 'o-', color='orange',
label='log t(L)', markersize=8)
```

```

axs[0, 1].set_xlabel("Розмір файлу (L, МБ)")
axs[0, 1].set_ylabel("log Часу обробки (log t)")
axs[0, 1].set_title("Графік log t(L)")
axs[0, 1].legend()

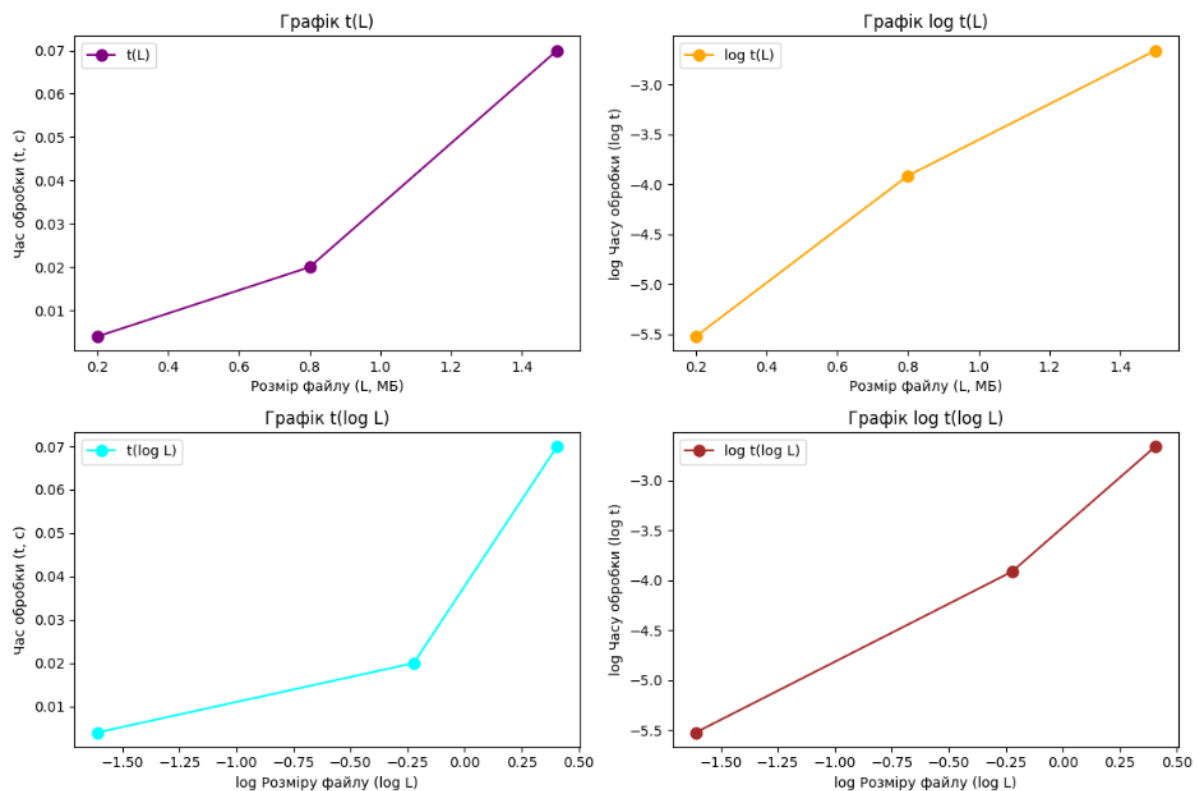
axs[1, 0].plot(np.log(sizes), times, 'o-', color='cyan',
label='t(log L)', markersize=8)
axs[1, 0].set_xlabel("log Розміру файлу (log L)")
axs[1, 0].set_ylabel("Час обробки (t, c)")
axs[1, 0].set_title("Графік t(log L)")
axs[1, 0].legend()

axs[1, 1].plot(np.log(sizes), np.log(times), 'o-', color='brown',
label='log t(log L)', markersize=8)
axs[1, 1].set_xlabel("log Розміру файлу (log L)")
axs[1, 1].set_ylabel("log Часу обробки (log t)")
axs[1, 1].set_title("Графік log t(log L)")
axs[1, 1].legend()

plt.tight_layout()
plt.show()

```

Результат



Висновок: У цій лабораторній роботі я дослідив, як час обробки текстових файлів змінюється залежно від їхнього розміру при використанні обраної програми для препроцесингу текстів. Аналіз графіків показав, що з ростом розміру файлу час обробки зменшується, що можна пояснити особливостями алгоритму, який більш ефективно працює з більшими обсягами даних.