**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп’ютерних наук та кібернетики

**Звіт до лабораторної роботи №1**

**з дисципліни «Прикладні алгоритми»**

Виконав студент 4-го курсу

Групи МІ-41

Єгор ЄЛЕНІН

Київ – 2025

**ЗМІСТ**

**ПОСТАНОВКА ЗАДАЧІ 3**

**РОЗВ’ЯЗОК 4**

**ВИСНОВКИ 6**

**ПОСТАНОВКА ЗАДАЧІ**

**І. Спільна частина.**

Реалізувати структуру даних типу «множина рядків». Рядки – непусті послідовності довжиною до 15 символів з рядкових латинських літер. Структура даних повинна підтримувати операції додавання рядку до множини, вилучення та перевірки належності даного рядку множині. Максимальна кількість рядків у множині, що зберігається, не більше 106.

Вхідні дані. Кожен рядок вхідних даних задає одну операцію над множиною. Запис операції складається з типу операції та наступного за ним через пробіл ряду, над яким проводять операцію. Типи операції вказують символи: «+» – додавання рядку, «–» – вилучення, «?» – перевірка на належність. Загальні кількість операцій у вхідному файлі не більше 106. Список операцій завершується рядком із символом # (ознака кінця вхідних даних).

При додаванні елементу до множини не гарантується, що він відсутній у множині, а при вилученні елементу з множини не гарантується, що він є у множині.

Вихідні дані. Виводяться для кожної операції типу «?» рядок yes або no, в залежності від того, чи зустрічається дане слово у множині.

**ІІ. Частина за варіантами.**

Варіант 2. Знайти усі паліндроми в множині рядків. Оцінити час виконання.

**РОЗВ’ЯЗОК**

Код на гітхаб: <https://github.com/g-gi-g/PA_Lab1>

Для тестування використовувалися дані з рис. 1.

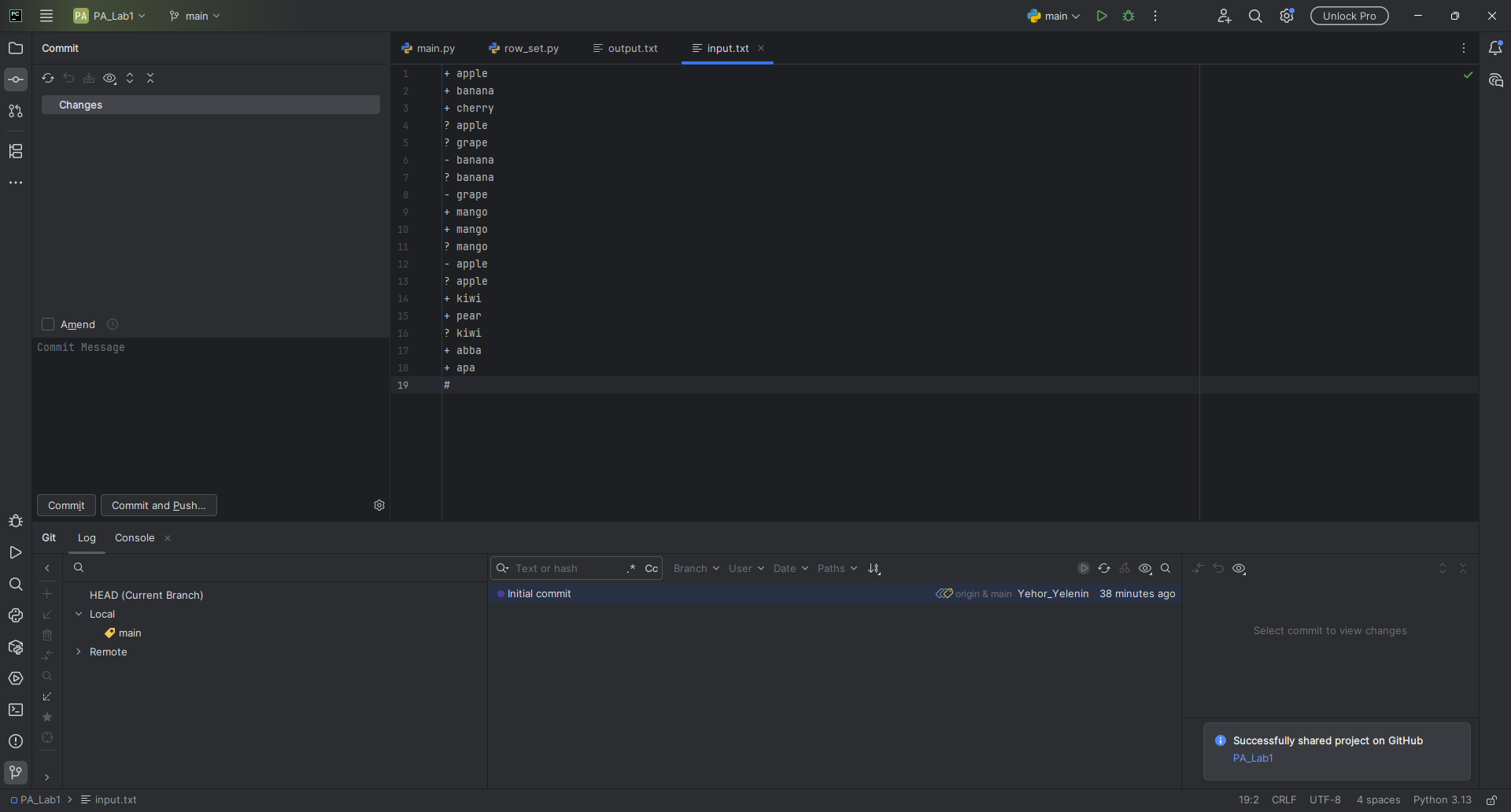


Рис. 1

Вихідний результат на рис. 2.

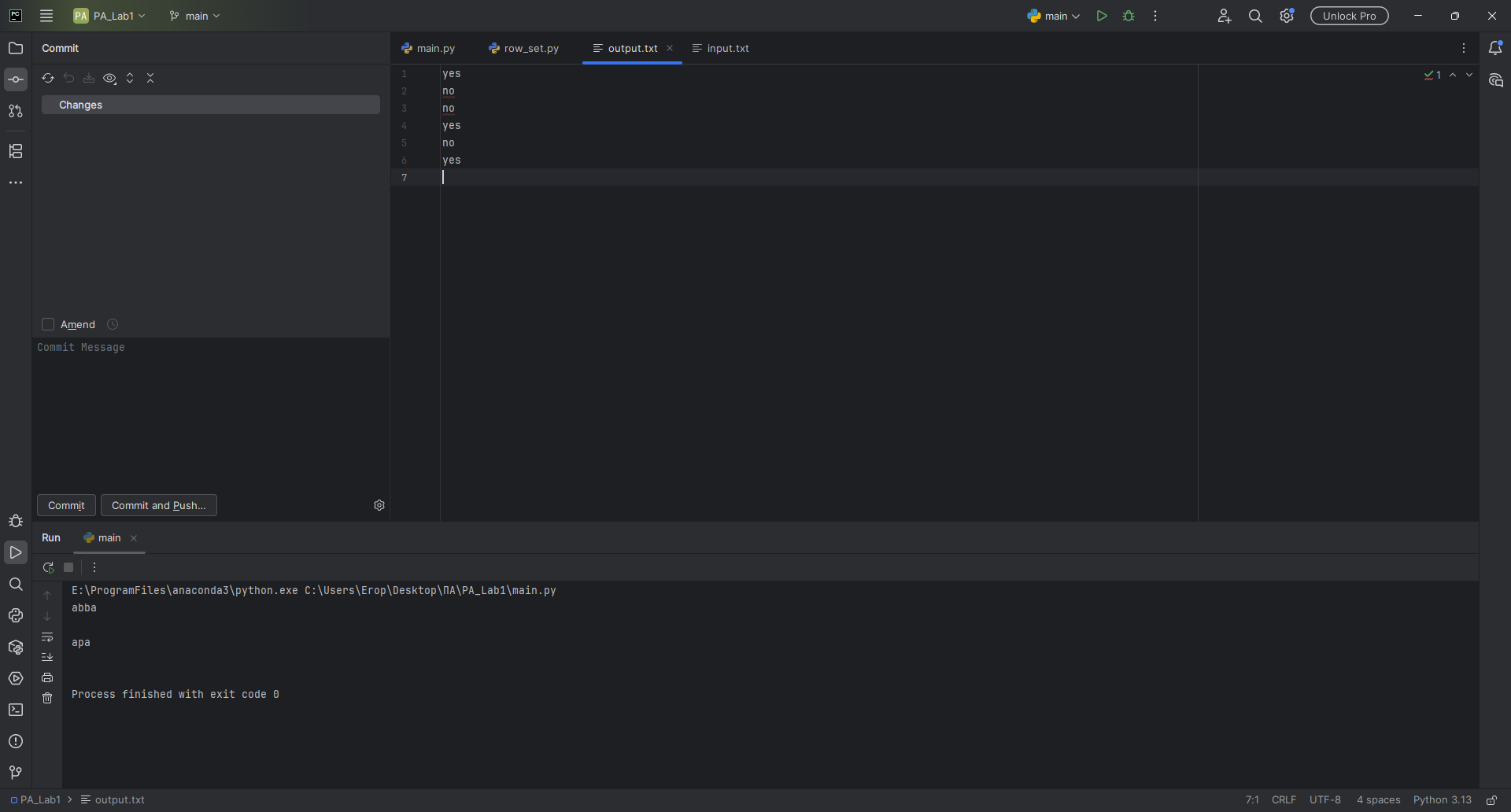


Рис. 2

Як видно з рис. 2, у терміналі вивелися всі слова-паліндроми, а в файл вивелися відповіді на запити по наявності слів у множині. Все відпрацювало коректно.

Оцінка часової складності функції пошуку паліндромів дуже наближено , де – кількість слів. Це точно краще, ніж прямий перебір.

**ВИСНОВКИ**

Було реалізовано структуру даних, яка підтримує вставку, вилучення непорожніх слів довжиною до 15 літер латинського алфавіту. Також є підтримка запитів до структури на предмет наявності того чи іншого слова. Також є функція пошуку всіх слів-паліндромів. Все було реалізовано з використанням поліноміальної хеш-функції, що пришвидшило обчислення у порівнянні з прямим перебором.