

Практична робота 05
Організація обміну інформації з програмою
Система оцінювання

№	Тема	К-ть балів
1.	Завдання 1	2
2.	Завдання 2	2
3.	Завдання 3	2
4.	Завдання 4	2
5.	Завдання 5	2
	Всього за практичну роботу	10
6.	ІНДЗ-1	3
7.	ІНДЗ-2	3
	Всього	16

* – діє бонусна система

1. Прості операції вводу-виводу

1. Напишіть програму, яка буде підраховувати у введеному з клавіатури тексті унікальні символи та їх кількість. У результаті потрібно вивести таблицю у вигляді «символ – кількість у тексті».
2. Напишіть програму, яка зчитуватиме вміст файлу та виводитиме його в консоль. Доповніть програмний код можливістю копіювати вміст цього файлу в інший файл.
3. (*Comma quibbling*) Нехай задано текст, наприклад, “АБВ ГДЄ Ж З”. Розставте коми між цими словами у такому стилі: “АБВ, ГДЄ, Ж та З”. Правила розстановки наступні:
 - 1) Якщо ввід порожній, то і вивід порожній.
 - 2) Якщо введено лише одне слово, воно і виводиться.
 - 3) Якщо введено два слова, вивід розділяється за допомогою “та”.
 - 4) Якщо введено більше 2 слів, вони розділяються комами, проте остання кома замінюється на «та».

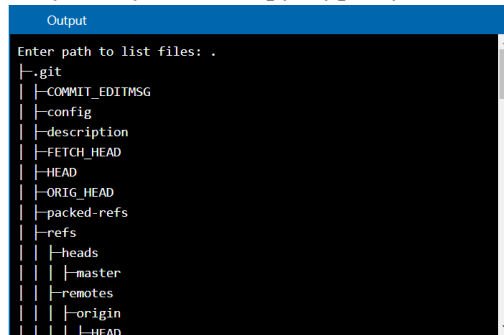
Словом вважається послідовність символів, що не містить пробіл або кому. Розгляньте функцію [strtok\(\)](#), яка допоможе відокремити слова. Напишіть функцію, яка виконуватиме дану процедуру.

4. Напишіть функцію, яка буде видаляти порожні рядки з тексту. Узагальніть код для випадку видалення конкретного слова в усьому файлі. Доповніть програму можливістю заміни одного слова на інше в усьому файлі.
5. Розгляньте [приклад реалізації стеку](#) мовою C. Реалізуйте на базі запропонованої структури даних перевірку дужкової послідовності в первинному коді з іншого файлу. Ідея перевірки наступна: при появі в тексті символу ‘{’, він додається в стек, а при появі символу ‘}’ – видаляється. Якщо стек у кінці перевірки залишається порожнім, значить програма має коректну дужкову послідовність, інакше – ні. Зверніть увагу на випадок, коли йде спустошення

стеку (закриваючих фігурних дужок більше, ніж відкриваючих – це негайний привід перервати перевірку, оскільки код некоректний).

ІНДЗ

1. Розгляньте приклад [коду](#), який демонструє можливість виводу всіх файлів і папок у поточній директорії. Детальніше дослідіть заголовковий файл `dirent.h` та спробуйте рекурсивно вивести файлову структуру на певному диску. Вивід структури буде подібним до такого:



```
Output
Enter path to list files: .
|--.git
|  |--COMMIT_EDITMSG
|  |--config
|  |--description
|  |--FETCH_HEAD
|  |--HEAD
|  |--ORIG_HEAD
|  |--packed-refs
|  |--refs
|     |--heads
|     |--master
|     |--remotes
|         |--origin
|         |--HEAD
```

2. **(Зв'язні списки)** Розгляньте [код реалізації двозв'язного списку](#) та змодельуйте роботу з плейлістом аудіоплеєра. Кожна пісня (вузол списку) має інформацію про свою назву, виконавця, альбом та рік випуску. Плейліст допускає вставку та видалення пісень, перехід до наступної або попередньої пісні, а також вивід переліку своїх пісень або поточної пісні, що відтворюється. Створіть плейліст на 6 пісень та продемонструйте дані операції.