

# Завдання 1. Обробка вхідних аргументів програми.

Ціль цього завдання — написати просту програму для практики важливих базових технік доступних користувачу C++. Програма має робити просту річ: зчитувати список переданих їй аргументів, які в нашому випадку будуть назвами уявних (неіснуючих) файлів з їх розширенням (але без шляху чи директорії), і казати користувачу скільки файлів якого типа вона нарахувала.

Наприклад,

---

```
> ./build/task1 myfile.txt mydoc.doc yourdoc.doc otherfile.txt
You have provided 2 txt and 2 doc files.
```

---

Формат вихідного повідомлення залишається на вибір автора, але має бути зрозумілим, інформативним і граматично вірним.

## Додатковий бал

- щоб трохи розширити свою програму, збережіть назви файлів у `stringstream`, `std::vector`, або інше за вибором, і покажіть користувачу списки файлів для кожного розширення окремо,
- напишіть найпростіші автоматизовані тести для вашої програми.<sup>1</sup> Для цього можна зробити наступне
  1. Перенесіть основну логіку з `main()` в окрему функцію, щоб її можна було перевіряти.
  2. Створіть функцію-тестувальницю.
  3. Придумайте декілька різних типових (або навпаки неочікуваних!) варіантів вводу від користувача.

---

<sup>1</sup>Це незамінна навичка, якщо ви плануєте так чи інакше пов'язати ваше життя з розробкою чи тестуванням програм.

4. Всередині, оформіть їх в якусь форму, що буде зрозумілою для функції, що ви тестуєте.
5. Там же, викличте функцію-рахувальницю з різними варіантами списків файлів, перевірте коректність виводу.<sup>2</sup>
6. У `main()`, перед тим як перевіряти ввід користувача, викличте тести та виведіть одне з двох повідомлень: “All tests passed!”, або ж “Some tests failed!”.

---

<sup>2</sup>Якщо ви не хочете прив’язуватись до конкретного тексту повідомлення, перевіряйте тільки те, що має значення на вашу думку. Наприклад, у виводі програми, що наведено вище, достатньо пересвідчитись, що повідомлення має наступні підрядки “2 txt” та “2 doc”.