**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІ**

Кафедра **ПЗ**



### ЗВІТ

До лабораторної роботи № 3

**На тему:** *“*ЗМІСТ ТА РОЛЬ ТЕСТУВАННЯ У ЖИТТЄВОМУ ЦИКЛІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ*”*

**З дисципліни:** *“Вступ до інженерії програмного забезпечення”*

**Лектор:**

доцент каф. ПЗ

Левус Є. В.

**Виконав:**

ст. гр. ПЗ-15

Шпортко Т. О.

**Прийняв:**

асистент каф. ПЗ

Самбір А. А.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2023 р.

∑= \_\_\_\_ .

Львів – 2023

**Тема роботи:** Зміст та роль тестування у життєвому циклі розробки програмного забезпечення.

**Мета роботи:** Вивчити основні методи та принципи тестування на прикладі створення власної програми.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

**26)** Наведіть приклади нефункціонального тестування, що стосуються атрибутів якості ПЗ.

Нефункціональне тестування є процесом перевірки атрибутів якості програмного забезпечення (ПЗ), які не відносяться до його функціональної специфікації. Ось декілька прикладів нефункціонального тестування залежно від атрибутів якості, що перевіряються:

* Тестування продуктивності: це включає в себе перевірку швидкодії, відмовостійкості та масштабованості ПЗ. Прикладом може бути тестування швидкодії веб-додатку, під час якого перевіряється, як швидко відбувається завантаження сторінок під великим навантаженням користувачів.
* Тестування надійності: це включає в себе перевірку стійкості, надійності та безпеки ПЗ. Прикладом може бути тестування безпеки додатку, під час якого перевіряється, як добре він захищений від хакерських атак та вірусів.
* Тестування сумісності: це включає в себе перевірку сумісності ПЗ з іншими системами, апаратним та програмним забезпеченням. Прикладом може бути тестування сумісності гри на певній операційній системі та апаратурі.
* Тестування ефективності: це включає в себе перевірку того, наскільки ефективно ПЗ виконує свої функції та відповідає потребам користувача. Прикладом може бути тестування ефективності системи управління базами даних, під час якого перевіряється, наскільки швидко та точно відбувається пошук та збереження даних.
* Тестування доступності: це включає в себе перевірку того, наскільки доступне ПЗ для користувачів з обмеженими можливостями. Прикладом може бути тестування доступності веб-сайту, під час якого перевіряється, як добре він доступний для користувачів з візуальними чи слуховими порушеннями.
* Стрес-тестування: спрямований на визначення максимальної працездатності системи, при якій вона може працювати без збоїв та затримок в роботі. Це важливо для систем, які мають велику кількість користувачів або оброблюють великі обсяги даних, оскільки це дозволяє виявляти проблеми та усувати їх до того, як вони стануть критичними.
* UI-тестування (тестування користувацького інтерфейсу) - це процес перевірки коректності та зручності взаємодії користувача з програмним забезпеченням через його інтерфейс. Головна мета UI-тестування полягає в тому, щоб переконатися, що програмне забезпечення функціонує належним чином та задовольняє потреби користувачів.
* UX-тестування - це процес перевірки користувальницького досвіду при взаємодії з програмним забезпеченням, зокрема перевірка зручності, задоволеності та ефективності користування програмним забезпеченням.
* Тестування зручності супроводу - це процес перевірки, який виконується з метою оцінки зручності використання підтримки, такої як документація, пошукові системи, системи підтримки клієнтів тощо. Тестування зручності супроводу дозволяє виявити можливі проблеми, з якими користувачі можуть стикнутися під час спроб використовувати підтримку, що надається для продукту.
* Тестування безпеки - це процес перевірки програмного забезпечення на вразливості та можливість злому, з метою забезпечення високого рівня безпеки та захисту від різних загроз.

**41)** Чи є автоматичне тестування? Відповідь поясніть.

Ні, автоматичне тестування не існує.

Неіснування повного автоматичного тестування обумовлено тим, що тестування ПЗ - це процес, який включає не тільки запуск програми і отримання результатів, а й інші складові, які вимагають участі людини. Ось кілька причин, чому не можна повністю автоматизувати процес тестування:

* Необхідність зрозуміти вимоги користувача: Тестування повинно спрямовуватись на відповідність вимогам користувача, що вимагає розуміння специфіки діяльності клієнта. Тому, щоб створити ефективний набір тестів, необхідно співпрацювати з людьми, які розуміють бізнес-цілі ПЗ.
* Тестування на відсутність відхилень: Іноді тестування включає перевірку того, що програма не робить нічого неочікуваного, тобто не має відхилень. Неможливо автоматизувати перевірку відсутності відхилень, оскільки для цього необхідно аналізувати тестові результати та реагувати на них.
* Тестування на коректність: Інші тести вимагають перевірки правильності виконання програмою певних дій. Ці тести також потребують присутності людини для аналізу результатів тестування та визначення, чи відповідає програма очікуванням.
* Перевірка коректності відображення: Деякі тести пов'язані з перевіркою коректності відображення графічного інтерфейсу користувача (GUI). Автоматизація таких тестів може бути дуже складною, оскільки вони вимагають зображення та оцінки багатьох деталей візуального інтерфейсу.

**4)** Які є типи тестування в контексті умов функціонування програми?

У контексті умов функціонування програми існує три типи тестування:

1) тестування нормальних умов,

2) тестування екстремальних(граничних) умов,

3) тестування виняткових умов.

Перевірка в нормальних умовах передбачає тестування на основі даних, які характерні для реальних умов функціонування програми. Випадки, коли програма має працювати зі всіма можливими даними, надзвичайно рідкісні. Звичайно мають місце конкретні обмеження на область зміни даних, в якій програма має зберігати свою працездатність.

Перевірка в екстремальних умовах має слідувати відразу за перевіркою програми в нормальних умовах. Тестові дані цього етапу містять граничні значення області зміни вхідних даних, які мають сприйматися програмою як правильні дані. Для числових даних тестують програму при початковому та кінцевому значеннях допустимої області зміни даних, при зміні довжини відповідного поля від мінімальної до максимальної. Типовими прикладами таких екстремальних значень є дуже великі числа, дуже малі числа і відсутність інформації.

Перевірка у виняткових ситуаціях передбачає використання даних, значення яких знаходяться за межами допустимої області змін. Програма має відкидати будь-які дані, які вона не в стані опрацювати правильно.

**ПОСТАНОВКА ЗАВДАННЯ**

1. Провести функціональне тестування програми для роботи з динамічними списками (розробленої у ході виконання лабораторної роботи з дисципліни «Основи програмування»), використавши такі тести:

1. димовий тест,

2. тест нормальних умов,

3. тест екстремальних(граничних) умов,

4. тест виняткових умов.

1.1. Результати тестів 1, 2, 3 та 4 оформити у вигляді таблиці (за зразок взяти таблицю 2). У таблиці інформація заноситься в зрозумілій формі, достатньо повно, щоб не вимагалося додаткових (усних) пояснень. Для представлення *тестових даних* та *фактичних результатів* бажано використовувати посилання на зображення екранів чи відповідні файли. В *очікуваних результатах* зафіксувати, яка функція була протестована та відповідно, що є очікуваним результатом. Якщо помилки не знайдено у *Тип звіту* ставимо знак -. Примітки використовуємо для додаткових пояснень.

1.2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.

2. Окремо провести тестування логічної схеми програми (головна програма та дві функції-підпрограми). Для цього зобразити її у вигляді графів управління. Вершини графів пронумерувати. Записати усі отримані маршрути роботи програми, використовуючи номери вершин.

2.1. Для кожного маршруту провести тестування і записати результати тестування у звітній таблиці.

2.2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.

**ХІД РОБОТИ**

**Результати функціонального тестування програми**

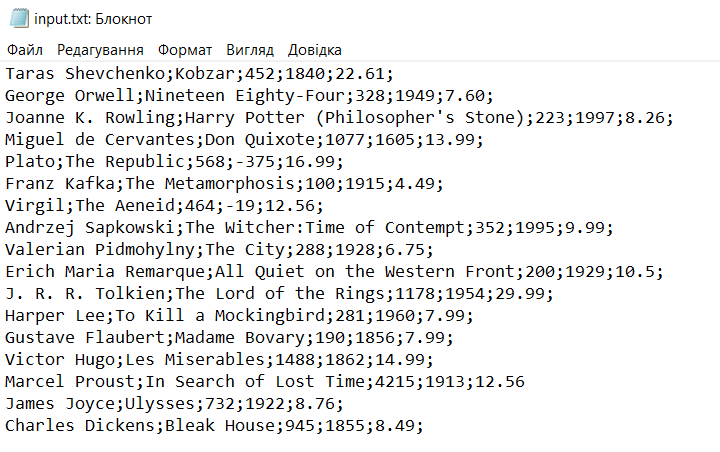
**BooksDB, версія 1.0**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  з/п | Тестові дані | Фактичні  результати | Очікувані  результати | Ступінь  критичності | Тип  звіту | Примітки |
| Димове тестування | | | | | | |
| 1 | Запуск програми | Програма успішно запустилася, на екран вивелося меню програми | Програма запустилася, на екран вивелося меню програми | Високий | - | Запуск програми  ([рис.10](#р10)) |
| Тест нормальних умов | | | | | | |
| 1 | Виведення списку у консоль | Програма успішно виводить список у консоль | Програма виводить список у консоль | Високий | - | Список  виведено  ([рис.11](#р11)) |
| 2 | Зчитування даних з вибраного файлу  ([рис.1](#р1)) | Програма  успішно  зчитує дані  з файлу у список | Програма  зчитує дані  з файлу у список, якщо введена назва файлу коректна | Високий | - | Зчитування  відбулося  ([рис.11](#р11)) |
| 3 | Запис даних у вибраний файл у текстовому режимі | Програма  успішно створює файл із введеною назвою та записує туди дані списку у текстовому режимі | Програма створює файл із введеною назвою та записує туди дані списку у текстовому режимі | Середній | - | Дані записано у файл  ([рис.12](#р12), [13](#р13)) |
| 4 | Запис даних у вибраний файл у бінарному режимі | Програма успішно створює файл із введеною назвою та записує туди дані списку у бінарному  режимі | Програма створює файл із введеною назвою та записує туди дані списку у бінарному режимі | Середній | - | Дані записано у файл  ([рис.14](#р14), [15](#р15)) |
| 5 | Додавання книг у список введенням з консолі | Програма успішно  додає книги у кінець списку із введеними з консолі даними | Програма додає книги  у кінець списку із введеними з консолі даними | Середній | - | Книга додалася  ([рис. 16](#р16)) |
| 6 | Видалення книг на певних позиціях  зі списку | Програма успішно видаляє книги зі списку за вказаними позиціями, виводить в консоль результуючий список | Програма видаляє книги зі списку за вказаними позиціями, виводить в консоль результуючий список | Середній | - | Книги зникли зі списку  ([рис. 17](#р17)) |
| 7 | Сортування списку за назвою | Програма успішно сортує список у алфавітному порядку за  назвою книги | Програма сортує список у алфавітному порядку за назвою книги | Середній | - | Посортований список  ([рис.18](#р18)) |
| 8 | Видалення зі списку книжок, чия назва починається з літер  P, K і L | Програма  успішно  видаляє  потрібні книги зі списку | Програма  видаляє  потрібні книги зі списку | Середній | - | Книги зникли зі списку  ([рис.19](#р19)) |
| 9 | Сортування в алфавітному порядку списку з тих книжок, чия ціна вища за середню ціну наявних | Програма створює другий список, що складається з потрібних книг, сортує його та виводить в консоль | Програма створює другий список, що складається з потрібних книг, сортує його та виводить в консоль | Середній | - | Посортований список  ([рис.20](#р20)) |
| 10 | Вихід з програми | Програма успішно завершує свою роботу | Програма завершує свою роботу | Високий | - | Повідомлення про  завершення роботи  ([рис.21](#р21)) |
| Текст екстремальних умов | | | | | | |
| 1 | Введення назви книги  та імені, прізвища довжиною 50 символів  ([рис.5](#р5)) | Програма успішно  обробляє такі дані, проте  табличний вивід ламається  ([рис.25](#р25)) | Програма успішно  обробляє такі дані | Середній | Пропозиція | Збільшити довжину рядків таблиці при виводі списку у консоль |
| 2 | Введення довгих назв книг та імен, прізвищ авторів | Програма успішно  обробляє такі дані | Програма успішно  обробляє такі дані | Високий | - | Введено  ([рис.27](#р27)) |
| 3 | Введення року створення >999 до н.е.  (необхідно 5 розрядів) | Програма успішно  обробляє такі дані | Програма успішно  обробляє такі дані | Середній | - | Введено  ([рис.28](#р28)) |
| 4 | Зчитування з пустого файлу  ([рис.3](#р3)) | Програма ламається | Програма виводить повідомлення про помилку | Середній | Помилка кодування | Помилка  ([рис.35](#р35)) |
| 5 | Зчитування з файлу з однією книгою  ([рис.2](#р2)) | Програма успішно  зчитує цю книгу | Програма зчитує цю книгу | Середній | - | Зчитано  ([рис. 22](#р22)) |
| 6 | Зчитування з файлу з  великою кількістю книг  ([рис.8](#р8)) | Програма успішно зчитує усі книги з файлу  незалежно від кількості  (обмеження лише у витратах пам’яті) | Програма успішно зчитує усі книги з файлу  незалежно від кількості | Високий | - | Зчитано |
| 7 | Зчитування рядка даних з файлу довжиною 100 символів  ([рис.6](#р6)) | Програма ламається | Програма успішно  обробляє такі дані | Середній | Помилка  кодування | Помилка  ([рис.26](#р26)) |
| 8 | Видалення книги з початку списку | Програма успішно  видаляє книгу з початку списку | Програма видаляє книгу з початку списку | Високий | - | Книга зникла  зі списку  ([рис.17](#р17)) |
| 9 | Видалення книги з кінця списку | Програма успішно  видаляє книгу з кінця списку | Програма видаляє книгу з кінця списку | Високий | - | Книга зникла  зі списку  ([рис.50](#р50)) |
| 10 | Сортування книг, вищих середнього за ціною, якщо у списку лише одна книга | Програма успішно створює  другий список, аналогічний першому та залишає його незмінним | Програма створює  другий список, аналогічний першому та залишає його незмінним | Середній | - | Одна книга, список без змін  ([рис.23](#р23)) |
| 11 | Сортування книг, вищих середнього за ціною, якщо список порожній | Програма успішно  виводить повідомлення про помилку | Програма виводить повідомлення про помилку | Низький | - | Повідомлення про помилку  ([рис.24](#р24)) |
| Тест виняткових умов | | | | | | |
| 1 | Вибір некоректної опції | При введенні чисел, що не є цілими чи будь-яких інших символів програма ламається  ([рис.31](#р31)) | Програма виводить повідомлення про помилку, дає можливість обрати коректну опцію | Високий | Помилка кодування | Помилка |
| 2 | Введення кириличних назв книг так імен, прізвищ авторів  ([рис.9](#р9)) | Програма реагує  не-  передбачувано  ([рис.29](#р29), [36](#р36)) | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Середній | Невідповідність  ТЗ  (пункт 3) | Передбачити  оброблення таких ситуацій  (пропускання книги, виведення повідомлення  про помилку, тощо) |
| 3 | Введення  від’ємної ціни книги | Програма успішно  виводить повідомлення про помилку | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Середній | - | Повідомлення про помилку  ([рис.51](#р51)) |
| 4 | Спроба маніпуляцій над пустим списком | Програма успішно  виводить повідомлення про помилку | Програма виводить повідомлення про помилку | Середній | - | Повідомлення про помилку  ([рис.24](#р24), [39](#р39)) |
| 5 | Додавання і видалення книги за некоректним індексом (поза межами списку) | Програма не приймає такі дані, проте не виводить повідомлення про помилку | Програма не приймає такі дані та виводить повідомлення про помилку | Середній | Пропозиція | Передбачити виведення повідомлення  про помилку  ([рис.32](#р32)) |
| 6 | Некоректна назва файлу | Програма аварійно завершує свою роботу | Програма виводить повідомлення про помилку | Середній | Помилка кодування | Помилка  ([рис.30](#р32)) |
| 7 | Зчитування рядка даних з файлу довжиною >100 символів  ([рис.4](#р4)) | Програма аварійно завершує свою роботу | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу | Середній | Помилка кодування | Помилка  ([рис.34](#р34)) |
| 8 | Введення назви книги, імені та прізвища автора довжиною >50 символів  ([рис.7](#р7)) | Програма аварійно завершує свою роботу | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Середній | Помилка  кодування | Помилка  ([рис.33](#р33)) |
| 9 | Введення року створення, що перевищує 4 розряди (>9999) | Програма не виводить повідомлення про помилку та працює так, ніби такі дані є коректні | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Низький | Невідповідність ТЗ  (пункт 3) | Передбачити  оброблення таких ситуацій  (пропускання книги, виведення повідомлення  про помилку, тощо) |
| 10 | Видалення книг з назвою, що починається з літер P, K і L, якщо таких немає в списку | Програма успішно обробляє дані, проте не виводить повідомлення-попередження  ([рис.37](#р37)) | Програма успішно обробляє дані та виводить повідомлення-попередження, що таких книг не було в списку | Середній | Пропозиція | Передбачити виведення повідомлення-попередження |

**Результати функціонального тестування програми**

**BooksDB, версія 2.0**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  з/п | Тестові дані | Фактичні  результати | Очікувані  результати | Ступінь  критичності | Тип  звіту | Примітки |
| Димове тестування | | | | | | |
| 1 | Запуск програми | Програма успішно запустилася, на екран вивелося меню програми | Програма запустилася, на екран вивелося меню програми | Високий | - | Запуск програми  ([рис.10](#р10)) |
| Тест нормальних умов | | | | | | |
| 1 | Виведення списку у консоль | Програма успішно виводить список у консоль | Програма виводить список у консоль | Високий | - | Список  виведено  ([рис.11](#р11)) |
| 2 | Зчитування даних з вибраного файлу  ([рис.1](#р1)) | Програма  успішно  зчитує дані  з файлу у список | Програма  зчитує дані  з файлу у список, якщо введена назва файлу коректна | Високий | - | Зчитування  відбулося  ([рис.11](#р11)) |
| 3 | Запис даних у вибраний файл у текстовому режимі | Програма  успішно створює файл із введеною назвою та записує туди дані списку у текстовому режимі | Програма створює файл із введеною назвою та записує туди дані списку у текстовому режимі | Середній | - | Дані записано у файл  ([рис.12](#р12), [13](#р13)) |
| 4 | Запис даних у вибраний файл у бінарному режимі | Програма успішно створює файл із введеною назвою та записує туди дані списку у бінарному  режимі | Програма створює файл із введеною назвою та записує туди дані списку у бінарному режимі | Середній | - | Дані записано у файл  ([рис.14](#р14), [15](#р15)) |
| 5 | Додавання книг у список введенням з консолі | Програма успішно  додає книги у кінець списку із введеними з консолі даними | Програма додає книги  у кінець списку із введеними з консолі даними | Середній | - | Книга додалася  ([рис. 16](#р16)) |
| 6 | Видалення книг на певних позиціях  зі списку | Програма успішно видаляє книги зі списку за вказаними позиціями, виводить в консоль результуючий список | Програма видаляє книги зі списку за вказаними позиціями, виводить в консоль результуючий список | Середній | - | Книги зникли зі списку  ([рис. 17](#р17)) |
| 7 | Сортування списку за назвою | Програма успішно сортує список у алфавітному порядку за  назвою книги | Програма сортує список у алфавітному порядку за назвою книги | Середній | - | Посортований список  ([рис.18](#р18)) |
| 8 | Видалення зі списку книжок, чия назва починається з літер  P, K і L | Програма  успішно  видаляє  потрібні книги зі списку | Програма  видаляє  потрібні книги зі списку | Середній | - | Книги зникли зі списку  ([рис.19](#р19)) |
| 9 | Сортування в алфавітному порядку списку з тих книжок, чия ціна вища за середню ціну наявних | Програма успішно створює другий список, що складається з потрібних книг, сортує його та виводить в консоль | Програма створює другий список, що складається з потрібних книг, сортує його та виводить в консоль | Середній | - | Посортований список  ([рис.20](#р20)) |
| 10 | Вихід з програми | Програма успішно завершує свою роботу | Програма завершує свою роботу | Високий | - | Повідомлення про  завершення роботи  ([рис.21](#р21)) |
| Текст екстремальних умов | | | | | | |
| 1 | Введення назви книги  та імені, прізвища довжиною 50 символів  ([рис.5](#р5)) | Програма успішно  обробляє такі дані | Програма успішно  обробляє такі дані | Середній | - | Введено |
| 2 | Введення довгих назв книг та імен, прізвищ авторів | Програма успішно  обробляє такі дані | Програма успішно  обробляє такі дані | Високий | - | Введено  ([рис.27](#р27)) |
| 3 | Введення року створення >999 до н.е.  (необхідно 5 розрядів) | Програма успішно  обробляє такі дані | Програма успішно  обробляє такі дані | Середній | - | Введено  ([рис.28](#р28)) |
| 4 | Зчитування з пустого файлу  ([рис.3](#р3)) | Програма успішно  виводить повідомлення про помилку | Програма виводить повідомлення про помилку | Середній | - | Повідомлення про помилку  ([рис.38](#р38)) |
| 5 | Зчитування з файлу з однією книгою  ([рис.2](#р2)) | Програма успішно  зчитує цю книгу | Програма зчитує цю книгу | Середній | - | Зчитано  ([рис. 22](#р22)) |
| 6 | Зчитування з файлу з  великою кількістю книг  ([рис.8](#р8)) | Програма успішно зчитує усі книги з файлу  незалежно від кількості  (обмеження лише у витратах пам’яті) | Програма успішно зчитує усі книги з файлу  незалежно від кількості | Високий | - | Зчитано |
| 7 | Зчитування рядка даних з файлу довжиною 100 символів  ([рис.6](#р6)) | Програма успішно  обробляє такі дані | Програма успішно  обробляє такі дані | Середній | - | Зчитано  ([рис.40](#р40)) |
| 8 | Видалення книги з початку списку | Програма успішно  видаляє книгу з початку списку | Програма видаляє книгу з початку списку | Високий | - | Книга зникла  зі списку  ([рис.17](#р17)) |
| 9 | Видалення книги з кінця списку | Програма успішно  видаляє книгу з кінця списку | Програма видаляє книгу з кінця списку | Високий | - | Книга зникла  зі списку  ([рис.50](#р50)) |
| 10 | Сортування книг, вищих середнього за ціною, якщо у списку лише одна книга | Програма успішно створює  другий список, аналогічний першому та залишає його незмінним | Програма створює  другий список, аналогічний першому та залишає його незмінним | Середній | - | Одна книга, список без змін  ([рис.23](#р23)) |
| 11 | Сортування книг, вищих середнього за ціною, якщо список порожній | Програма успішно  виводить повідомлення про помилку | Програма виводить повідомлення про помилку | Низький | - | Повідомлення про помилку  ([рис.24](#р24)) |
| Тест виняткових умов | | | | | | |
| 1 | Вибір некоректної опції | Програма успішно виводить повідомлення про помилку, дає можливість обрати коректну опцію | Програма виводить повідомлення про помилку, дає можливість обрати коректну опцію | Високий | - | Повідомлення про помилку  ([рис.43](#р43)) |
| 2 | Введення кириличних назв книг так імен, прізвищ авторів  ([рис.9](#р9)) |  | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Середній | - | Повідомлення про помилку  ([рис. 45](#р45), [47](#р47)) |
| 3 | Введення  від’ємної ціни книги | Програма успішно виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Середній | - | Повідомлення про помилку  ([рис.51](#р51)) |
| 4 | Спроба маніпуляцій над пустим списком | Програма успішно  виводить повідомлення про помилку | Програма виводить повідомлення про помилку | Середній | - | Повідомлення про помилку  ([рис.24](#р24), [39](#р39)) |
| 5 | Додавання і видалення книги за некоректним індексом (поза межами списку) | Програма не приймає такі дані та виводить повідомлення про помилку | Програма не приймає такі дані та виводить повідомлення про помилку | Середній | - | Повідомлення про помилку  ([рис.48](#р48)) |
| 6 | Некоректна назва файлу | Програма успішно виводить повідомлення про помилку | Програма виводить повідомлення про помилку | Середній | - | Повідомлення про помилку  ([рис.49](#р49)) |
| 7 | Зчитування рядка даних з файлу довжиною >100 символів  ([рис.4](#р4)) | Програма успішно  виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу | Середній | - | Повідомлення про помилку  ([рис.44](#р44)) |
| 8 | Введення назви книги, імені та прізвища автора довжиною >50 символів  ([рис.7](#р7)) | Програма успішно виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Середній | - | Повідомлення про помилку  ([рис.41](#р41)) |
| 9 | Введення року створення, що перевищує 4 розряди (>9999) | Програма успішно виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Програма виводить повідомлення про помилку, пропускає цю книгу при зчитуванні з файлу, або просить ввести коректні дані при зчитуванні з консолі | Низький | - | Повідомлення про помилку  ([рис.46](#р46)) |
| 10 | Видалення книг з назвою, що починається з літер P, K і L, якщо таких немає в списку | Програма успішно обробляє дані та виводить повідомлення-попередження | Програма успішно обробляє дані та виводить повідомлення-попередження, що таких книг не було в списку | Середній | - | Повідомлення-попередження  ([рис.42](#р42)) |

Вхідні дані

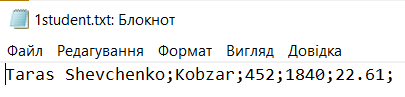
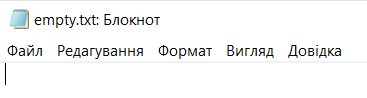


Рис. 1.

Рис. 3.

Рис. 2.

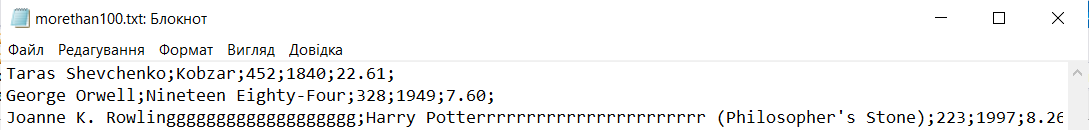


Рис. 4.

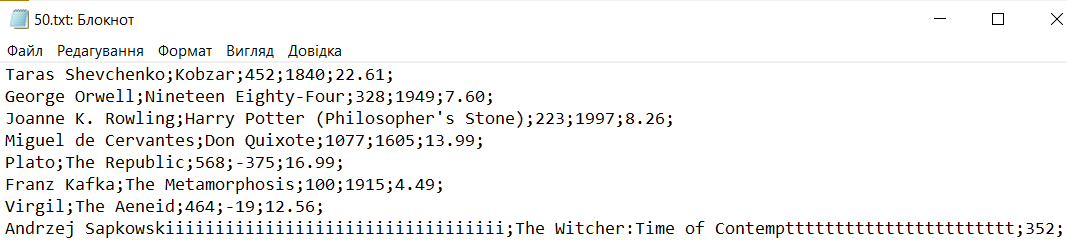


Рис. 5.

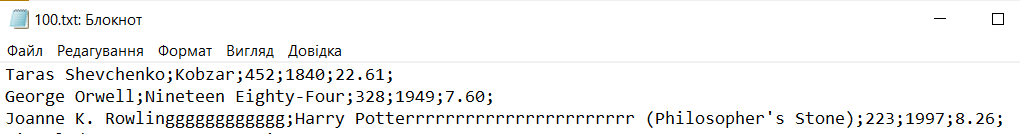


Рис. 6.

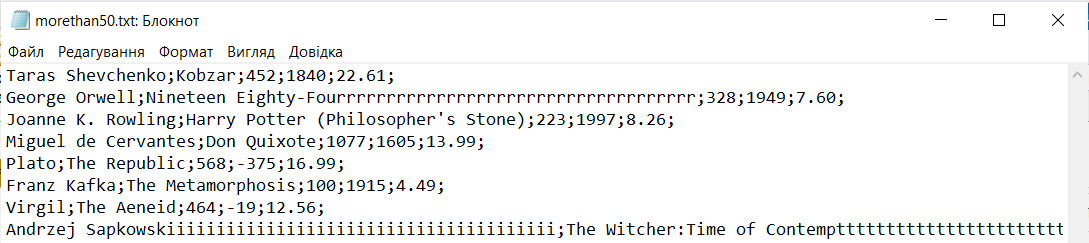


Рис. 7.

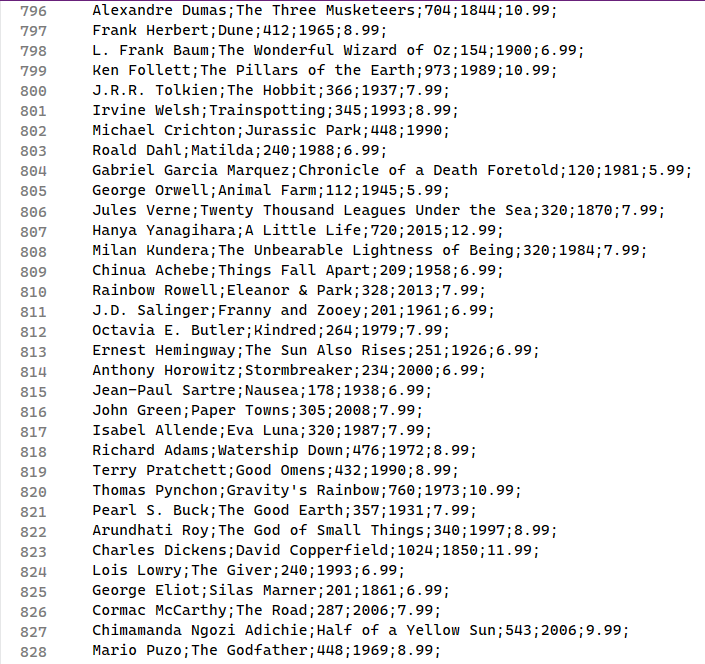


Рис. 8.

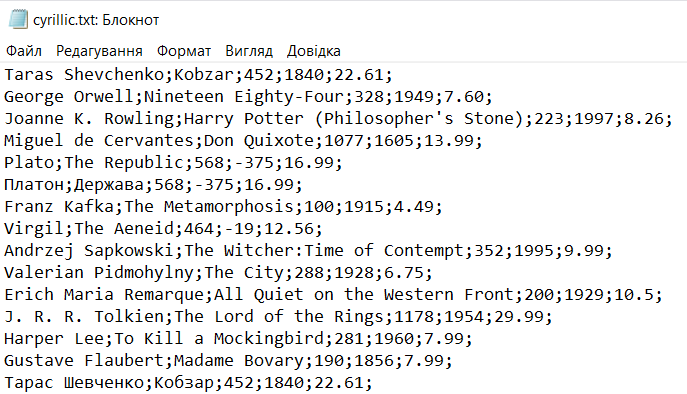


Рис. 9.

Результати тестів

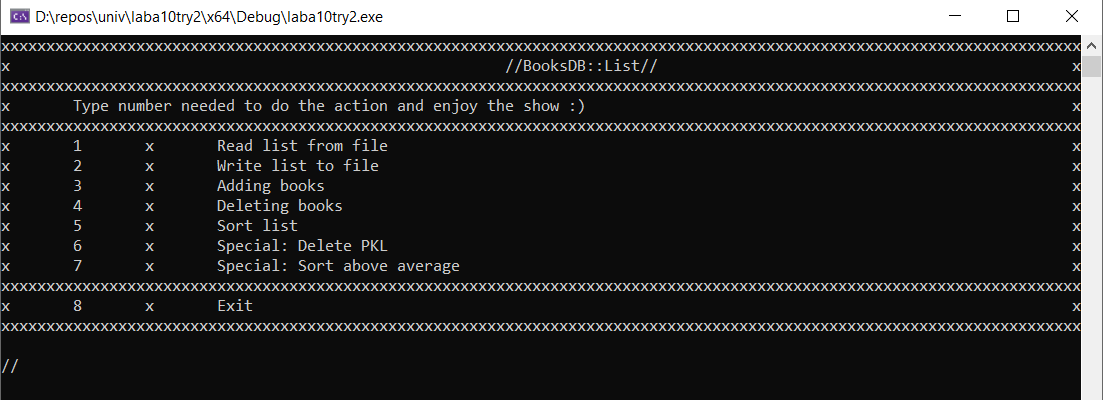


Рис. 10.

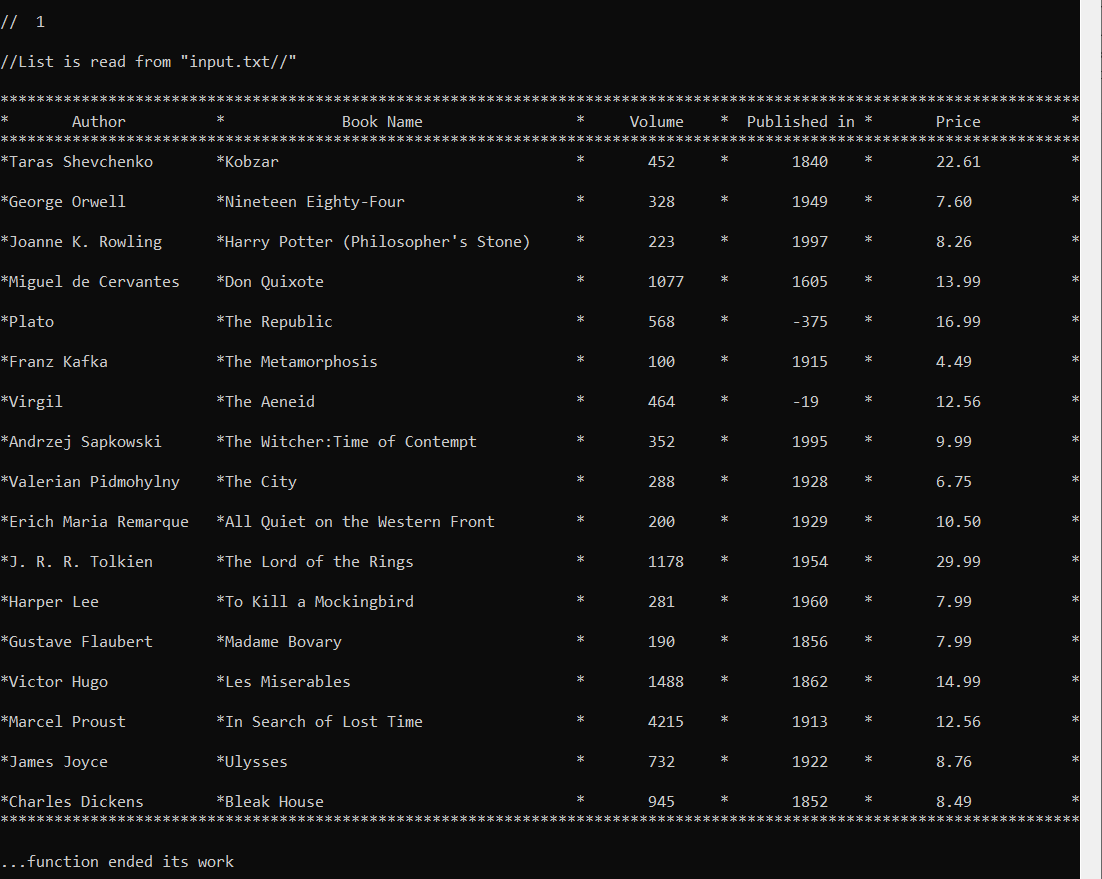


Рис. 11.

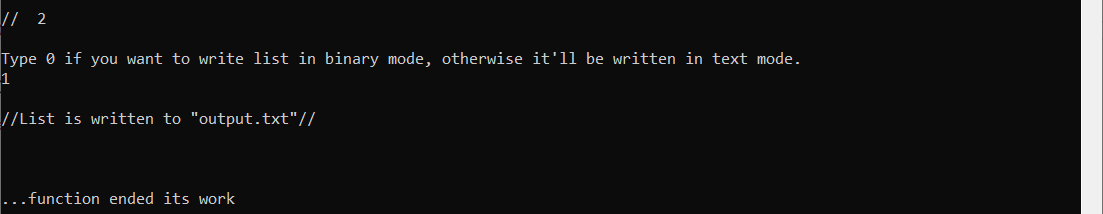


Рис. 12.

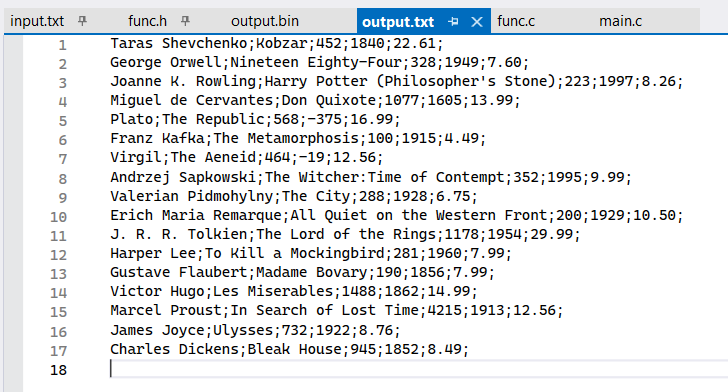


Рис. 13.

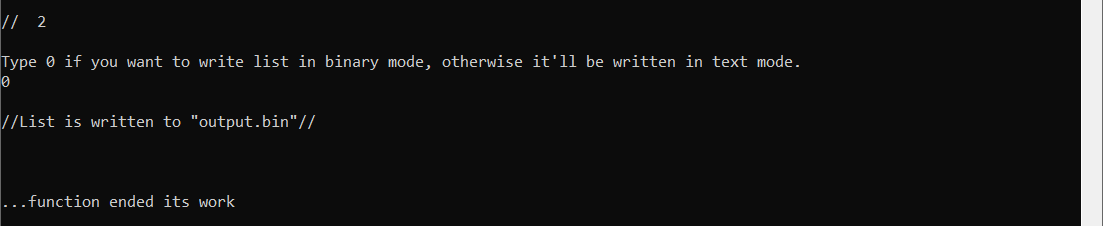


Рис. 14.

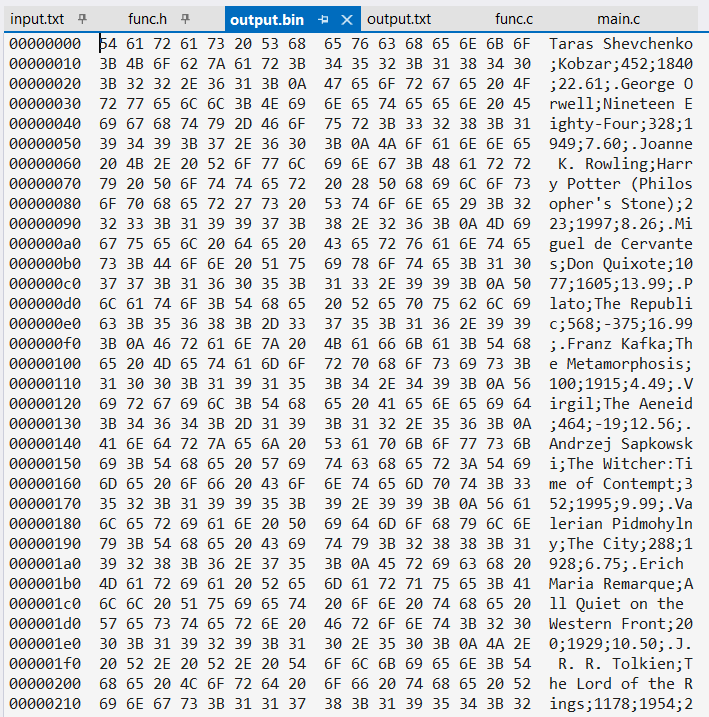


Рис. 15.

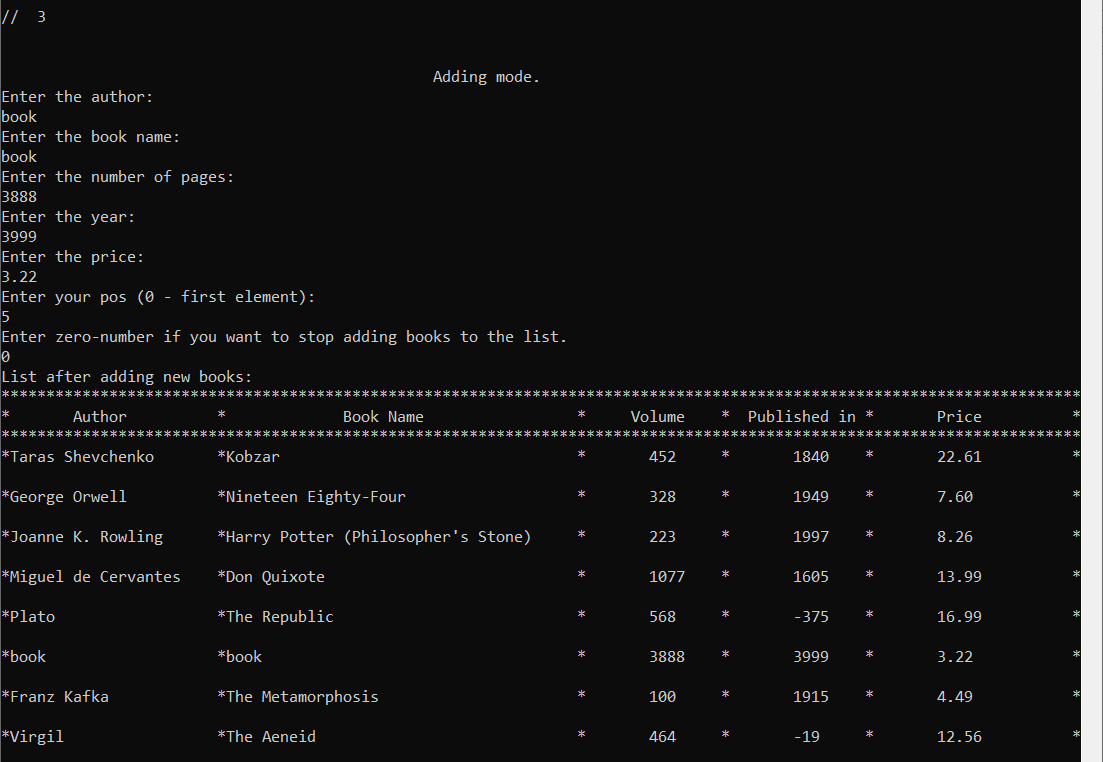


Рис. 16.

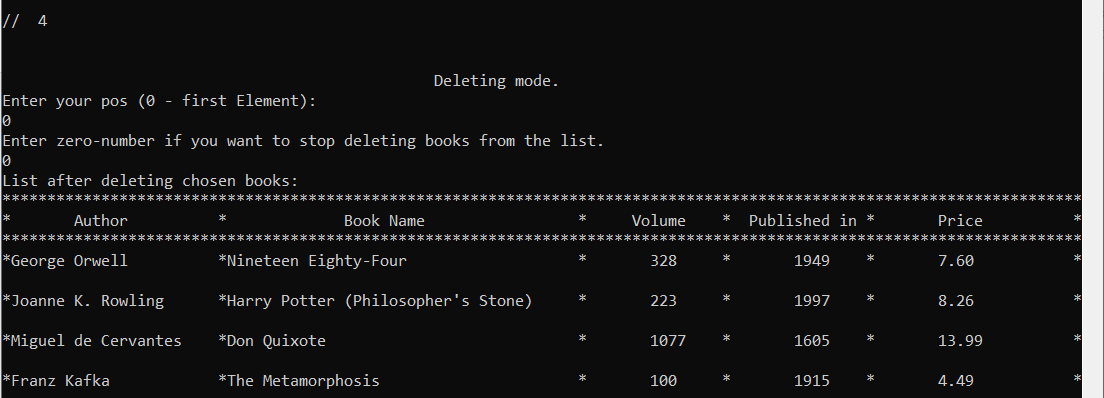


Рис. 17.

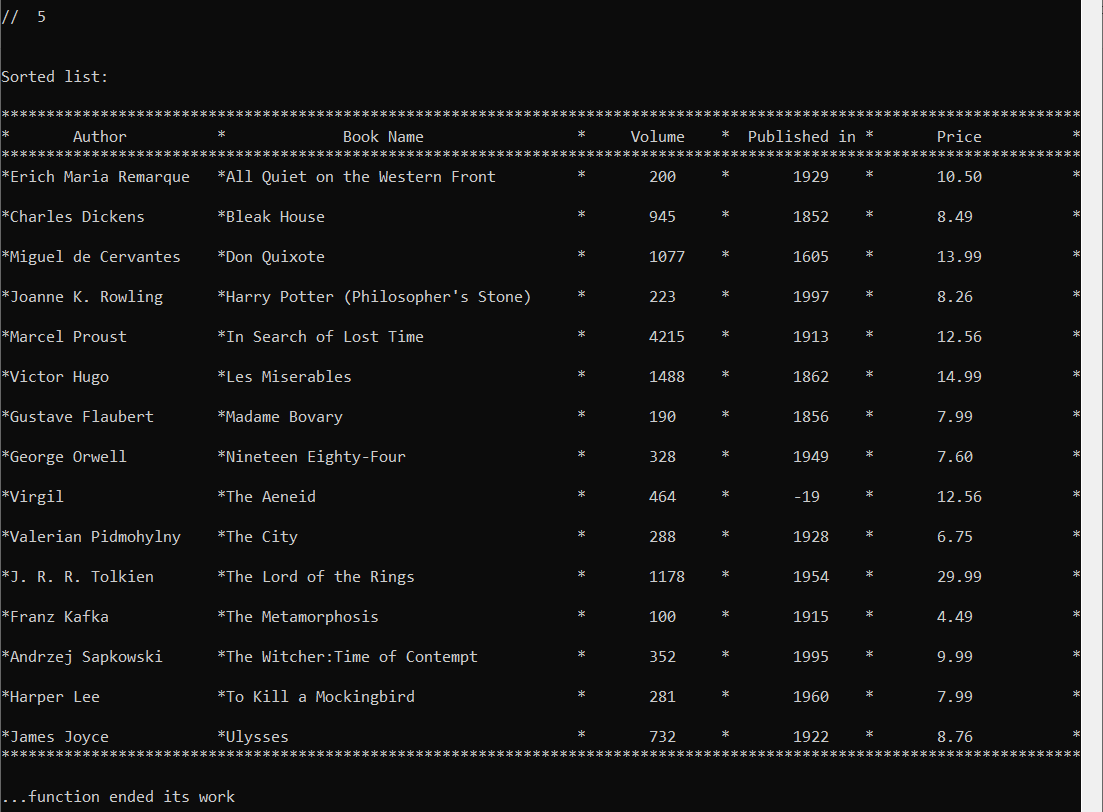


Рис. 18.

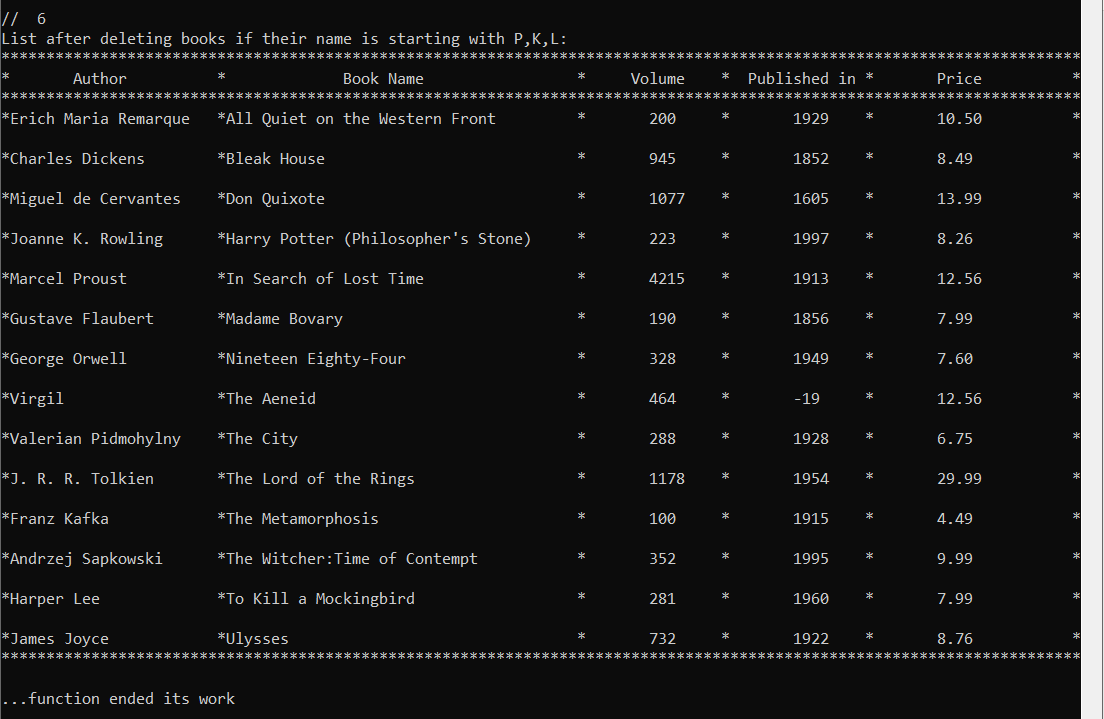


Рис. 19.

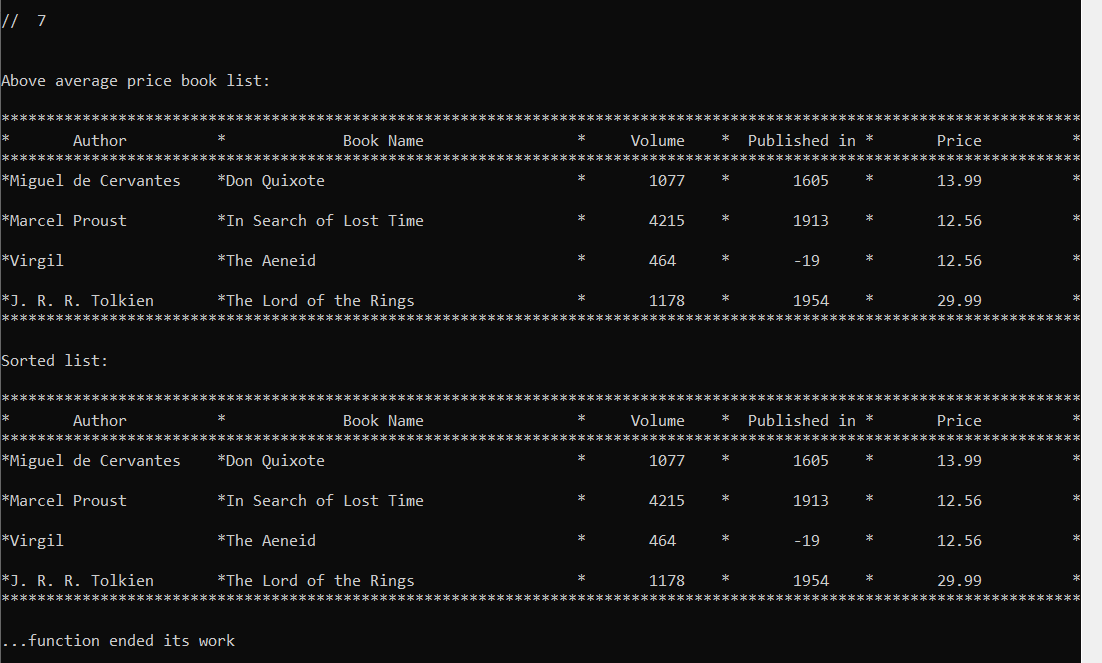


Рис. 20.

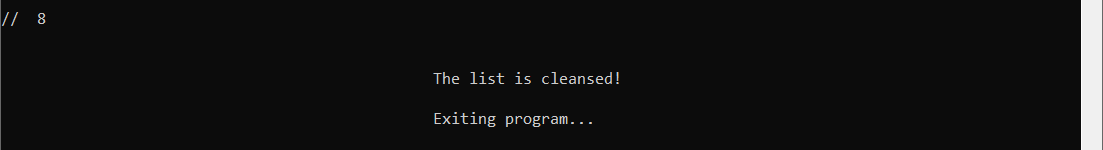


Рис. 21.

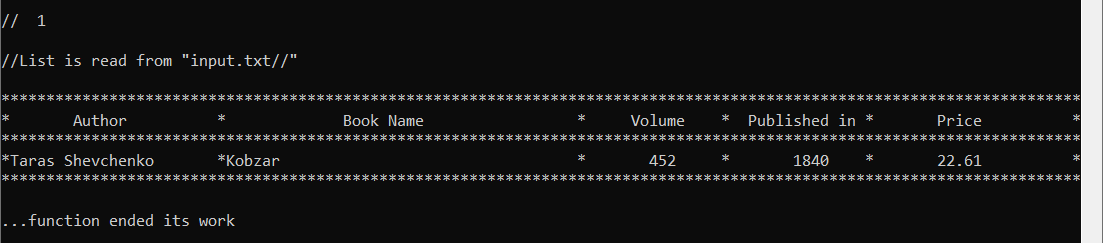


Рис. 22.

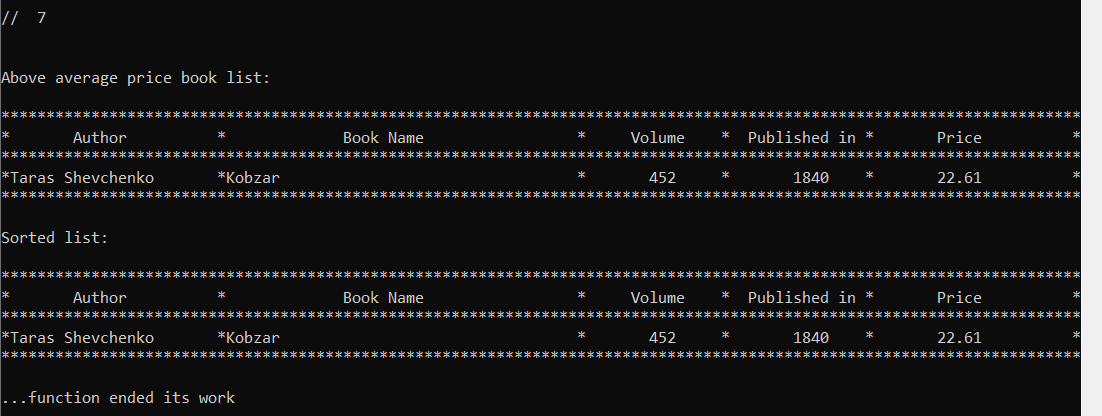
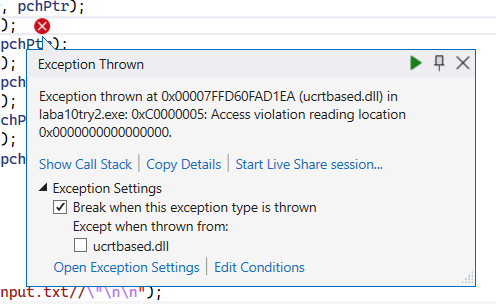


Рис. 23.



Рис. 24.



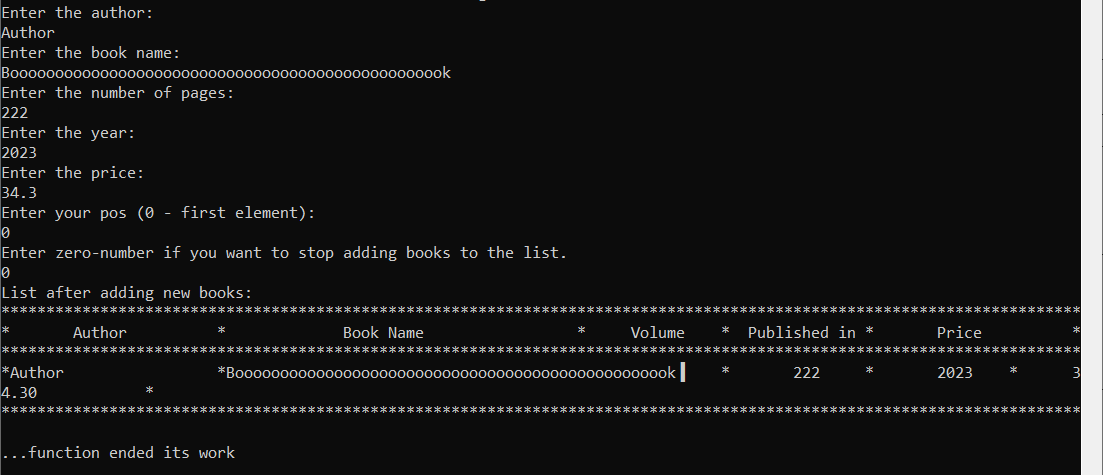


Рис. 26.

Рис. 25.

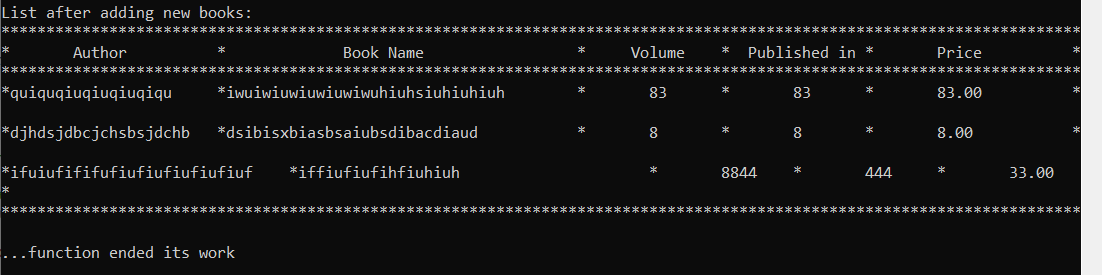


Рис. 27.

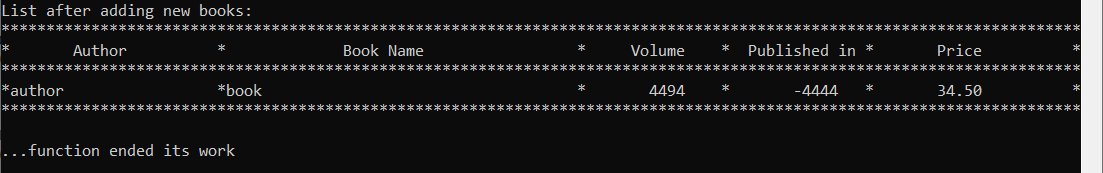


Рис. 28.

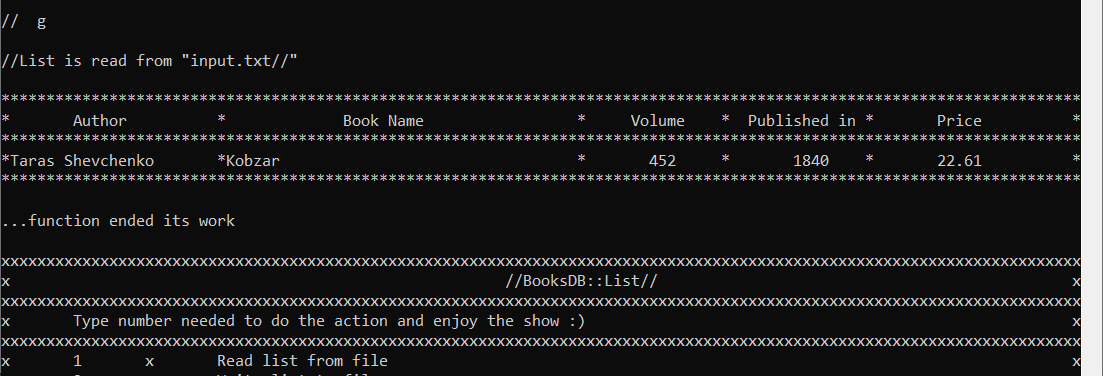
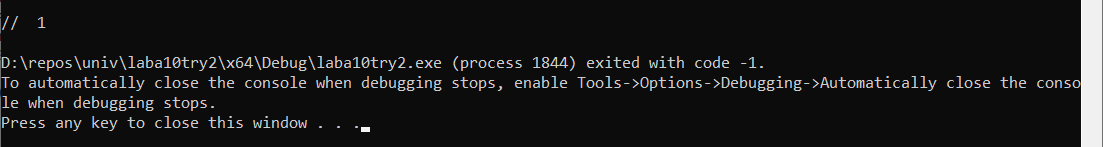
****

Рис. 29.

****

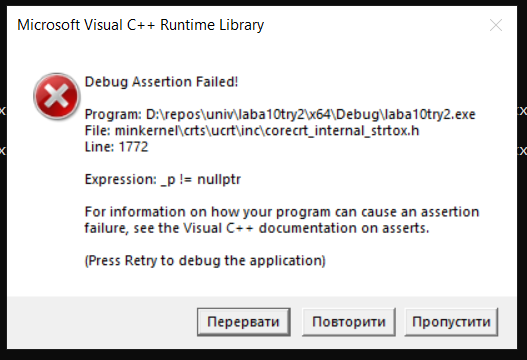
****

Рис. 30.

Рис. 31.

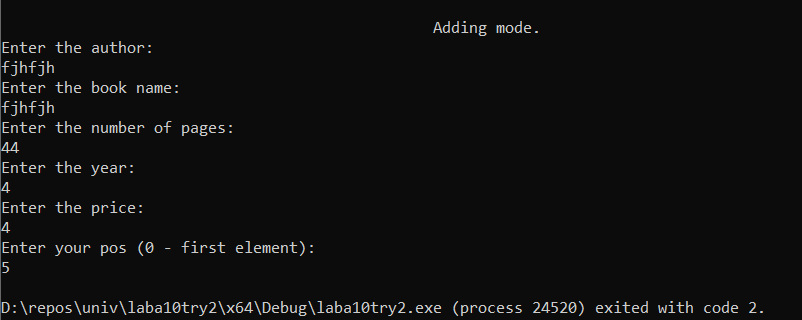
****

Рис. 32.

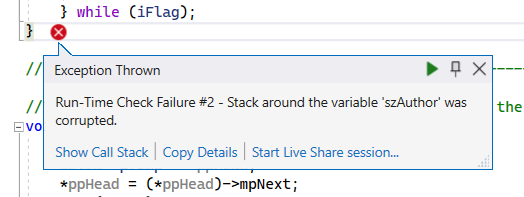
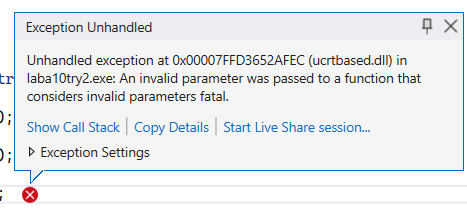
********

Рис. 34.

Рис. 33.

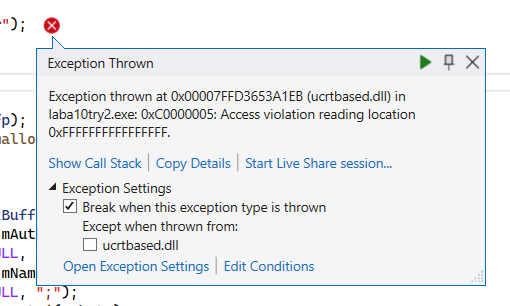
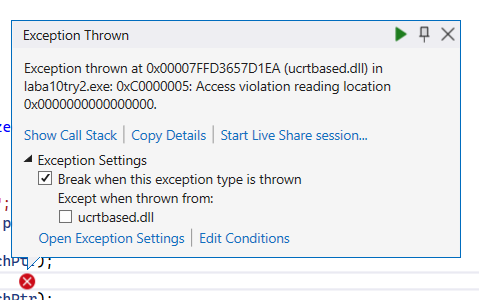
********

Рис. 35.

Рис. 36.

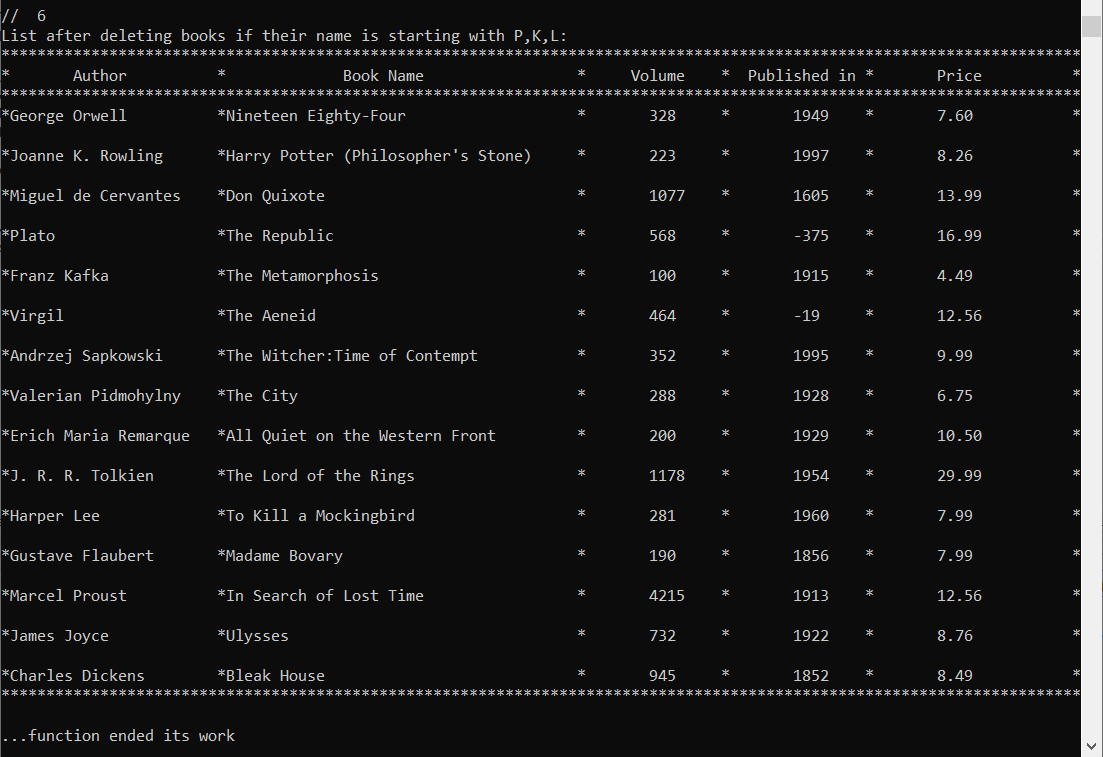
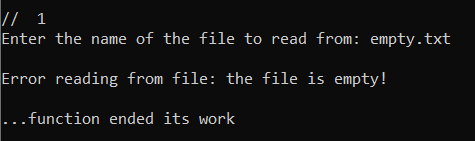
****

Рис. 37.

****

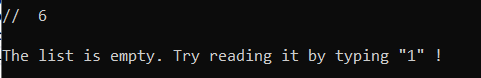
****

Рис. 38.

Рис. 39.

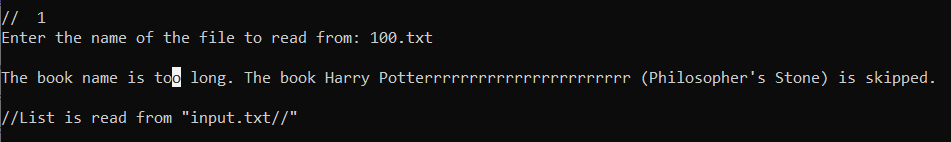
****

Рис. 40.

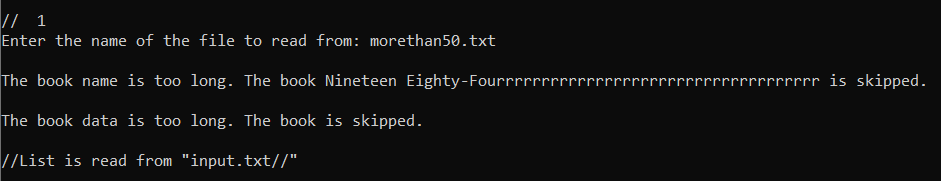
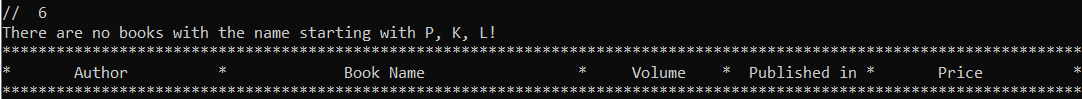
****

Рис. 41.

****

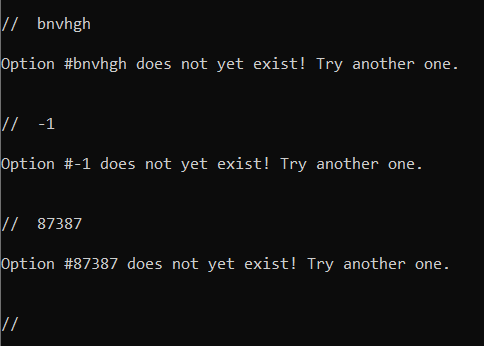
****

Рис. 42.

Рис. 43.

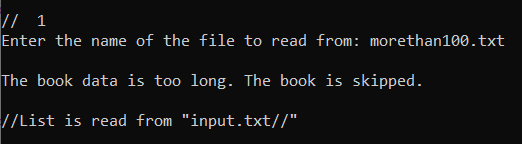
****

Рис. 44.

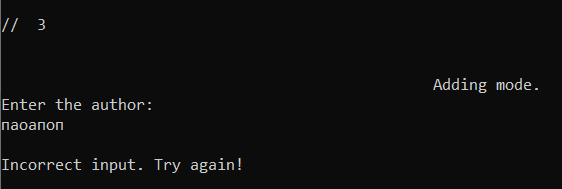
****

Рис. 45.

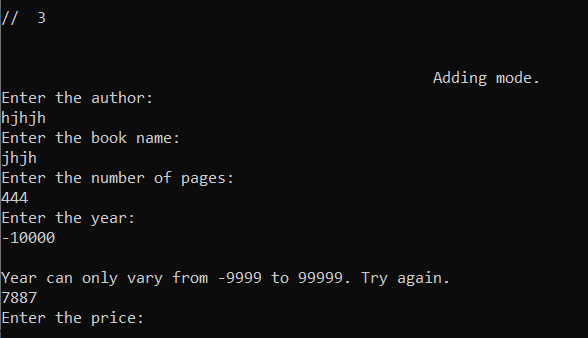
****

Рис. 46.

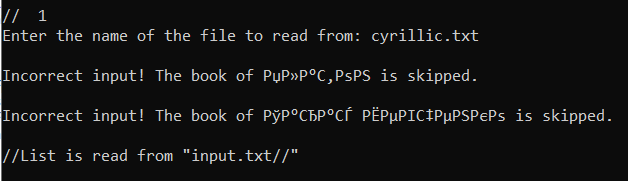
****

Рис. 47.

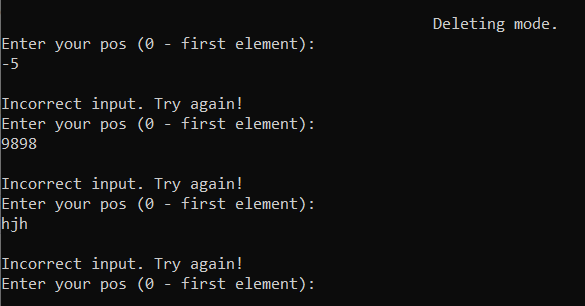
****

Рис. 48.

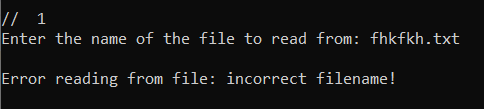
****

Рис. 49.

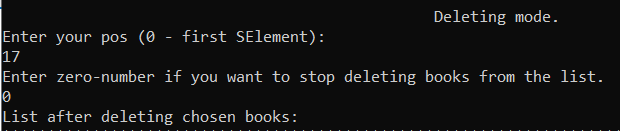
****

Рис. 50.

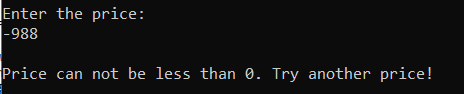
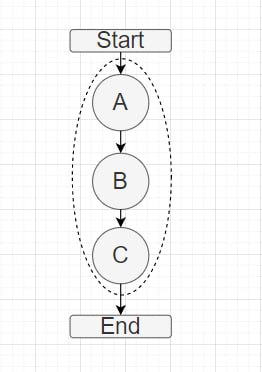
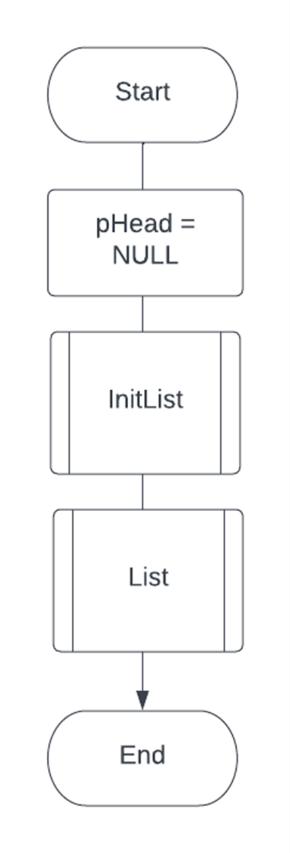
****

Рис. 51.

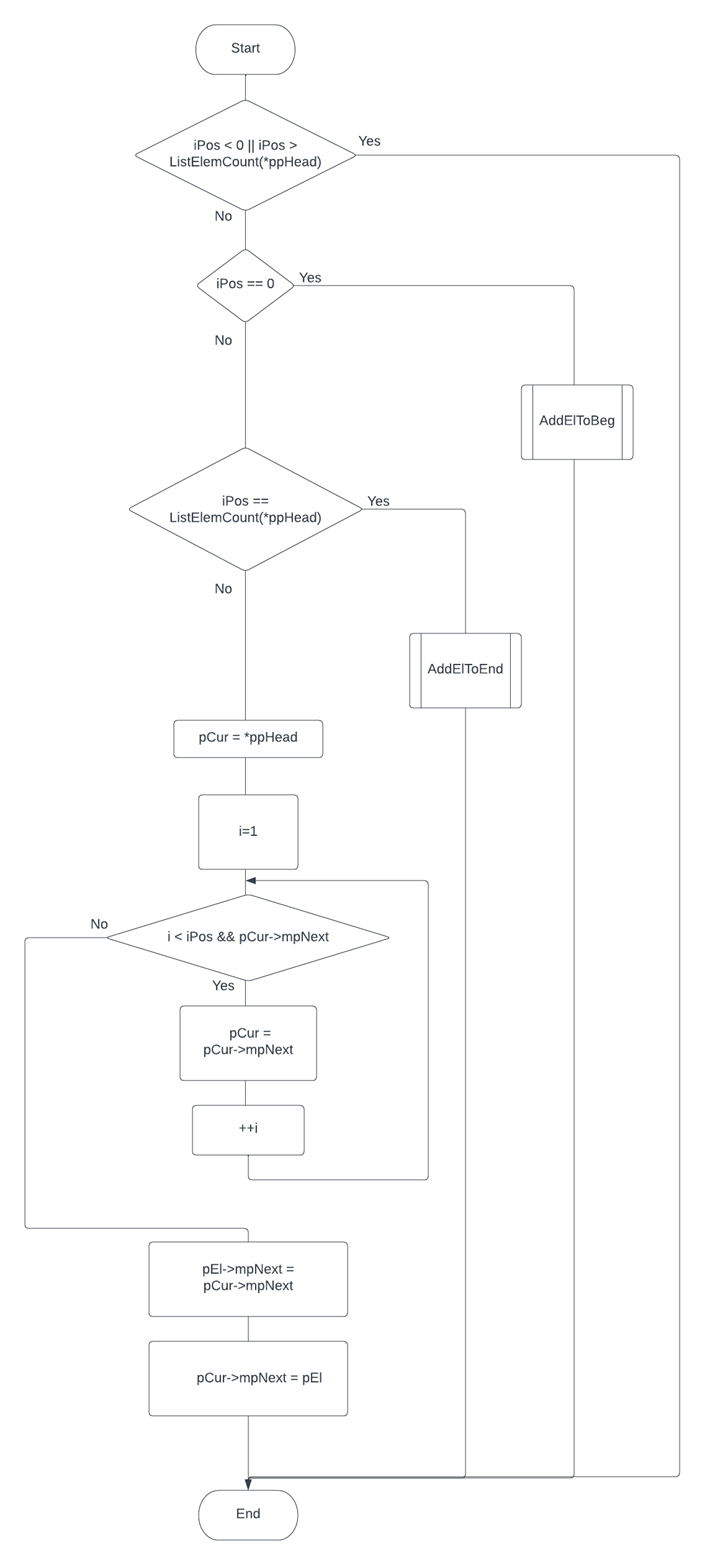
**Результати структурного тестування**

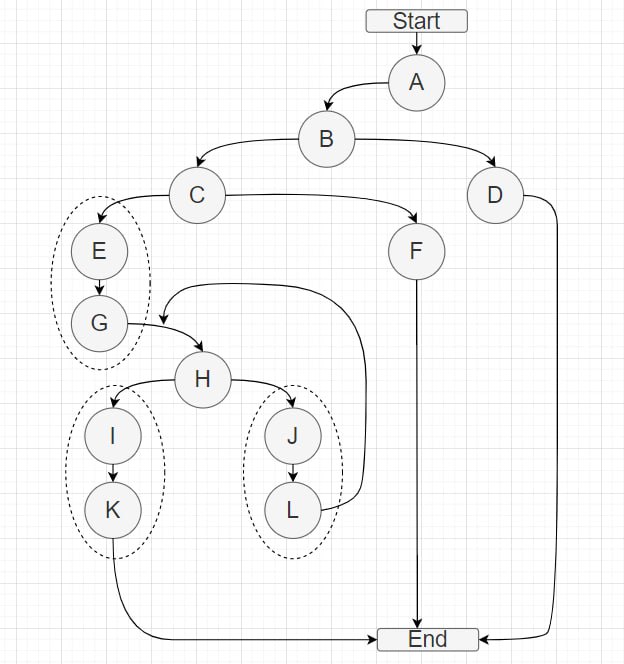
Граф управління функції main

****

|  |  |  |  |
| --- | --- | --- | --- |
| № | Маршрут | Результат | Оцінка роботи |
| 1 | [Start-A-B-C-End] | Програма ініціалізовує  список, запускає роботу меню | Тест успішний |

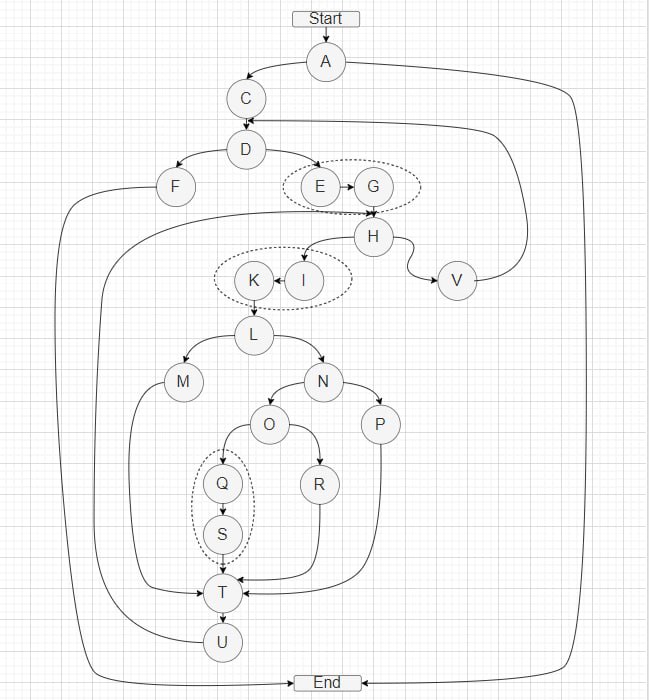
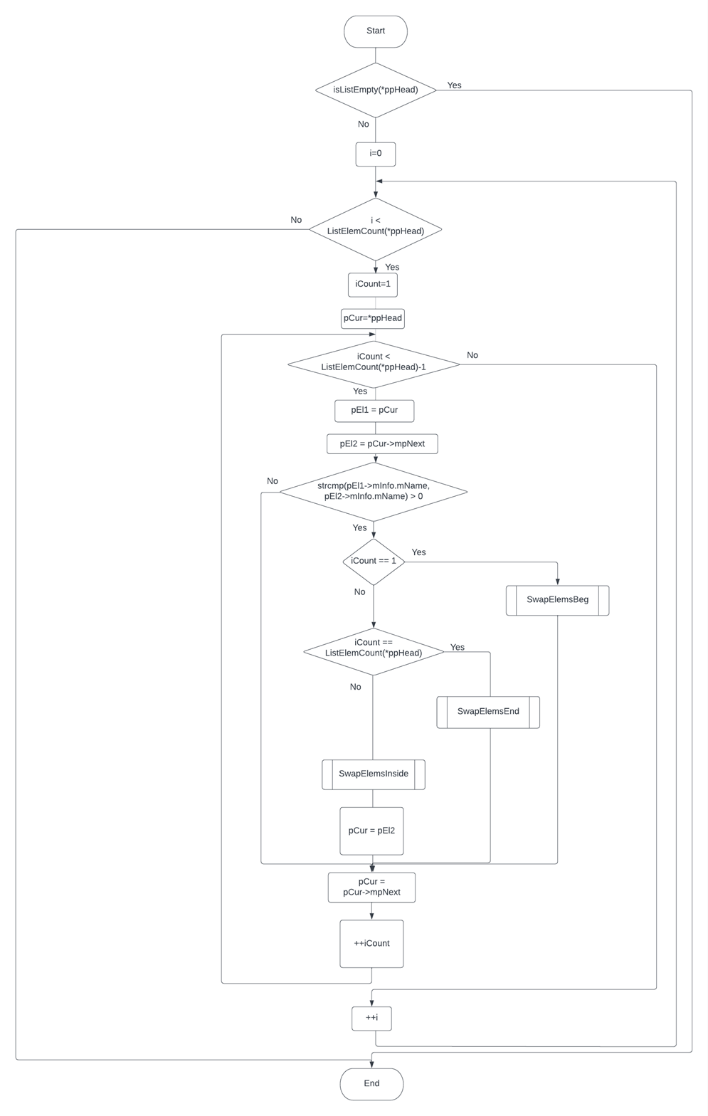
Граф управління функції AddElToPos

****

****

|  |  |  |  |
| --- | --- | --- | --- |
| № | Маршрут | Результат | Оцінка роботи |
| 1 | [Start-A-B-D -End] | Функція додає елемент в початок списку | Тест успішний |
| 2 | [Start-A-B-C-F-End] | Функція додає елемент в кінець списку | Тест успішний |
| 3 | [Start-A-B-C-E-G-H-I-K-End] | Функція додає елемент в середину списку | Тест успішний |
| 4 | [Start-A-B-C-E-G-H-J-L-H-I-K-End] | Функція додає елемент в середину списку, ітеруючи по списку | Тест успішний |

Граф управління функції ListSortByName

****

|  |  |  |
| --- | --- | --- |
| № | Маршрут | Оцінка роботи |
| 1 | [Start-A-End] | Тест успішний |
| 2 | [Start-A-C-D-F-End] | Тест успішний |
| 3 | [Start-A-C-D-E-G-H-V-D-F-End] | Тест успішний |
| 4 | [Start-A-C-D-E-G-H-V-D-E-G-H-I-K-L-M-T-U-H-V-D-F-End] | Тест успішний |
| 5 | [Start-A-C-D-E-G-H-V-D-E-G-H-I-K-L-N-P-T-U-H-V-D-F-End] | Тест успішний |
| 6 | [Start-A-C-D-E-G-H-V-D-E-G-H-I-K-L-N-O-R-T-U-H-V-D-F-End] | Тест успішний |
| 7 | [Start-A-C-D-E-G-H-V-D-E-G-H-I-K-L-N-O-Q-S-T-U-H-V-D-F-End] | Тест успішний |
| 8 | [Start-A-C-D-E-G-H-I-K-L-M-T-U-H-V-D-F-End] | Тест успішний |
| 9 | [Start-A-C-D-E-G-H-I-K-L-N-O-Q-S-T-U-H-V-D-F-End] | Тест успішний |
| 10 | [Start-A-C-D-E-G-H-I-K-L-N-O-R-T-U-H-V-D-F-End] | Тест успішний |
| 11 | [Start-A-C-D-E-G-H-I-K-L-N-P-T-U-H-V-D-F-End] | Тест успішний |

**ВИСНОВКИ**

Виконуючи цю лабораторну роботу, я познайомився із важливим етапом ЖЦ ПЗ – тестуванням. Вивчив основні методи та принципи тестування на прикладі роботи з власною програмою. Провів функціональне та структурне тестування своєї програми та задокументував його результати у формі таблиці, складених графів управління. Тестування є одним з найважливіших етапів ЖЦ ПЗ, оскільки воно дозволяє перевірити якість програми та виправити помилки якомога раніше, що, відповідно, зменшить витрати.