|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_lab7 |
| Назва проекта / ПЗ  Name of Project / Software | Ковальчук-task\_7\_1.exe |
| Рівень тестування  Level of Testing | Системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Ковальчук Володимир Олегович |
| Виконавець  Implementer | Ковальчук Володимир Олегович |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TC-0 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення : Я хочу стати девелопером! | 1.Виведення : Так | Passed |
| TC-1 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення : Він працює як деволопер. | 1.Виведення : Так | Passed |
| TC-2 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення :Being a developer is fun | 1.Виведення : Некоректний формат речення. | Passed |
| TC-3 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення :This is a professional Developer? | 1.Виведення : Так | Passed |
| TC-4 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення : знаю одного девелопера ) | 1.Виведення : Так | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| TC-5 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення : Я люблю програмувати. | 1.Виведення : ні | Passed |
| TC-6 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення : девелопера багато працюють. | 1.Виведення : ні | Passed |
| TC-7 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення : Coding is interesting and fun | 1.Виведення : Некоректний формат речення. | Passed |
| TC-8 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення : Я хочу стати девелопером | 1.Виведення : Некоректний формат речення. | Passed |
| TC-9 | 1.Зайдіть в код блокс ;  2.Задачу 7.1 перетворити в код  3.Введіть речення : Developer is a cool job | 1.Виведення : Некоректний формат речення. | Passed |

**Висновки**

1.Ми з командою зробили алгоритм дії ,це нам допомогло зробити роботу швитще.

2.Ми з командою навчилися працювати з мовою С.

3.Також ми з командою виправили всі помилки які виникли.

4.Покрокове тестування та перевірки кожного етапу зменшували шанси на помилкую

5.Всі етапи документувалися для кращого розуміння.

6.Можливість до адаптації : Код був написаний так щоб можна було вести нові данні.

7.Ми розподілили завдання для більш єфективної праці.

8. Код працює як з короткими, так і з довгими реченнями.

9. Довжина рядка перевіряється перед доступом до останнього символу, що запобігає помилкам.

10. Код коректно працює з порожнім введенням, виводячи повідомлення про некоректний формат.

11. Підтримується робота як з кирилицею, так і з латиницею.

12. sizeof(input) гарантує, що буфер не вийде за межі допустимої пам’яті.

13. Використання int length = strlen(input); дозволяє зберегти довжину рядка для подальших перевірок.

14. Код не допускає аварійного завершення через неправильний ввід.

15.Використання return ch == '.' ch == '!' ch == '?' || ch == ')'; робить функцію компактною та ефективною.

16.Код не дає змоги вводити речення без розділового знака наприкінці.

17.Можливість використання дужок додає гнучкість (наприклад, перевірка цитат).

18.Код працює незалежно від довжини речення.

19.Враховується випадок, коли введено лише розділовий знак (що не вважається коректним реченням).

20.Функція не вимагає додаткових бібліотек або складних алгоритмів.

21.Використання окремої функції hasValidEnding() робить код зрозумілим і розширюваним.

22.Код коректно працює з випадком, коли після допустимого розділового знака є пробіли.

23.Реалізація дозволяє легко змінювати список допустимих символів у майбутньому.

24.convertToLower() гарантує коректну обробку великих літер.

25.tolower() працює як для латиниці, так і для кирилиці.

26.Використання (unsigned char)text[i] у tolower() запобігає помилкам із знаковими типами.

27.Цикл проходить лише один раз по рядку, що забезпечує високу продуктивність.

28.Використання окремої функції для зниження регістру робить код чистішим.

29.Код не змінює вихідне введення, що дозволяє його подальше використання.

30.Робота функції не залежить від розміру буфера.

31.Перетворення працює швидко навіть для довгих речень.

32.Змінюються лише літери, інші символи залишаються без змін.

33.Код підтримує мови з різними розкладками клавіатури

34.Використання strstr() дозволяє швидко знаходити підрядки без додаткових алгоритмів.

35.Код перевіряє не тільки "девелопер", а й "девелопером" та "developer".

36.Можливість додавання нових ключових слів без зміни основної логіки.

37.Працює як з одним словом у реченні, так і з кількома.

38.Чутливість до регістру усунена завдяки convertToLower().

39.strstr() не змінює вихідний рядок, що запобігає небажаним побічним ефектам.

40.Також ми покрашили здадність працювати в команді .

41.Код оптимізований та працює швидко навіть для довгих речень.

42.Не використовуються зайві змінні або складні структури даних.

43.Програма не використовує додаткову пам’ять, окрім основного буфера.

44.Код не має циклів з надлишковими перевірками.

45.Використання окремих функцій покращує читабельність.

46.Можливість легко адаптувати під інші мови програмування.

47.Код легко тестувати з різними вхідними даними.

48.Обробляє помилковий ввід без аварійного завершення програми.

49.Це завдання допомогло зрозуміти як краще адаптувати інші мови

50.Кожен елемент виконує свою задачуб що робить її дуже ефективною.