

## Графічні інтерфейси

1. Створити GUI для компілятора за допомогою PyQt (Python).
  2. Використати Flutter для створення кросплатформного GUI (Dart).
  3. Реалізувати GUI на Electron.js із використанням Node.js і HTML/CSS.
  4. Зробити мобільний додаток на Swift для iOS, що інтерфейсується з компілятором.
  5. Створити мобільний додаток для Android на Kotlin для роботи з компілятором.
  6. Зробити GUI на JavaFX для редагування коду й виклику компілятора.
  7. Розробити GUI на WinForms (C#).
  8. Створити графічний інтерфейс на Avalonia (C#), орієнтуючись на кросплатформність.
  9. Використати GTK+ (C) для створення GUI.
  10. Створити десктопний додаток на Unreal Engine з вбудованим кодовим редактором.
- 

## Веб-орієнтовані рішення

11. Створити веб-інтерфейс компілятора на Flask (Python).
  12. Зробити аналогічний веб-інтерфейс на FastAPI.
  13. Використати React для розробки веб-редактора коду та роботи з компілятором.
  14. Створити веб-додаток на Angular.
  15. Реалізувати інтерфейс у Vue.js з інтеграцією вашого компілятора через API.
  16. Використати ASP.NET Core для створення веб-додатку.
  17. Реалізувати серверний REST API на Spring Boot, який взаємодіє з компілятором.
  18. Розробити веб-інтерфейс з використанням Svelte.
  19. Використати Deno для створення бекенду до веб-додатку.
  20. Створити веб-додаток з Next.js для роботи з компілятором.
- 

## Командні утиліти

21. Реалізувати утиліту на Go для спрощення запуску компілятора з аргументами.
  22. Створити консольний інтерфейс на Rust із зручним меню вибору параметрів.
  23. Написати консольний додаток на Haskell, який викликає компілятор.
  24. Реалізувати командний інтерфейс на Scala.
  25. Написати утиліту на F# для автоматизації запуску компілятора.
  26. Створити обгортку для компілятора на Julia.
  27. Реалізувати CLI на Elixir.
  28. Написати Bash-скрипт, який генерує параметри та запускає компілятор.
  29. Використати PowerShell для створення інтерактивного інтерфейсу запуску.
  30. Написати Ruby-скрипт для автоматизації тестування компілятора.
- 

## Редактори коду

31. Розширити VS Code плагіном для інтеграції з компілятором (JavaScript/TypeScript).
32. Створити плагін для Sublime Text на Python.
33. Реалізувати інтеграцію для JetBrains IDE з використанням Kotlin.
34. Створити розширення для Vim, що викликає компілятор.

35. Реалізувати плагін для Emacs на Lisp.
  36. Написати Atom-пакет для компіляції коду через компілятор.
  37. Використати Qt Creator для створення IDE з інтеграцією компілятора.
  38. Створити плагін для Eclipse.
  39. Додати функцію виклику компілятора до Notepad++ через Python Script.
  40. Інтегрувати компілятор у Geany за допомогою C.
- 

## **Інструменти аналізу**

41. Реалізувати лог-аналізатор на Python для журналів роботи компілятора.
  42. Використати R для візуалізації даних про час компіляції.
  43. Написати інструмент на Matlab для аналізу продуктивності.
  44. Реалізувати лог-фільтр на Perl.
  45. Написати аналізатор журналів на PHP із веб-інтерфейсом.
  46. Використати Lua для створення скрипту-обробника виводу компілятора.
  47. Написати інструмент для порівняння вихідних файлів компіляції на Swift.
  48. Зробити утиліту на Dart для перевірки залежностей.
  49. Написати Ruby-інструмент для підрахунку кількості помилок компіляції.
  50. Використати Nim для автоматизації аналізу компіляторних звітів.
- 

## **Автоматизація**

51. Створити Makefile для проєктів, що використовують компілятор.
  52. Використати CMake для генерації конфігурацій із параметрами компіляції.
  53. Реалізувати сценарій Gradle для автоматизації запуску компілятора.
  54. Написати скрипт на Groovy для побудови великих проєктів.
  55. Використати Ansible для автоматизації середовища компіляції.
  56. Реалізувати Jenkins-пайплайн для тестування компілятора.
  57. Написати автоматизатор тестів на JavaScript.
  58. Використати Terraform для підготовки середовищ компіляції.
  59. Реалізувати CI/CD-процес у GitLab з інтеграцією компілятора.
  60. Використати Docker для запуску ізольованого середовища з компілятором.
- 

## **Документація та допоміжні інструменти**

61. Реалізувати Markdown-конвертер для опису коду на Python.
62. Написати генератор документації на C++.
63. Використати Jekyll для створення сайту-довідки компілятора.
64. Реалізувати скрипт для створення PDF-звіту про помилки компіляції.
65. Написати JSON-інструмент для аналізу помилок.
66. Створити засіб перевірки синтаксису в коді на TypeScript.
67. Використати Pandoc для створення багатомовної документації.
68. Реалізувати засіб для автогенерації коду-шаблону.
69. Створити утиліту для перевірки сумісності версій компілятора.
70. Використати YAML для зберігання параметрів конфігурації.

---

## Тестування

71. Написати засіб для автоматичного створення тестових файлів.
72. Реалізувати фреймворк для юніт-тестування на Go.
73. Написати Python-скрипт для створення тестових наборів.
74. Реалізувати генератор випадкових тестів на Haskell.
75. Використати ScalaTest для автоматизації перевірок.
76. Реалізувати валідацію результатів роботи компілятора на Elixir.
77. Написати тестовий рушій для порівняння продуктивності.
78. Використати JavaScript для автоматизації валідації результатів.
79. Реалізувати R-скрипт для статистичного аналізу помилок.
80. Створити Python-інструмент для тестування різних параметрів компілятора.

---

## Різне

81. Створити гру, яка використовує мову програмування компілятора як сценарний рушій.
82. Написати навчальний посібник із прикладами використання компілятора.
83. Розробити гру-пазл для навчання синтаксису мови.
84. Використати Blender для створення 3D-редактора сценаріїв.
85. Створити Telegram-бота для компіляції коду.
86. Реалізувати Discord-бота для компілятора.
87. Написати Slack-інтеграцію для автоматизації роботи.
88. Зробити інтеграцію компілятора в Azure Functions.
89. Реалізувати інструмент для автоматичного оновлення версії компілятора.
90. Написати середовище інтерактивного навчання для початківців.