1. Синтаксис
2. Ошибки, подсвечиваемые редактором кода (убрать точку с запятой, сделать опечатку, удалить двоеточие (python)/скобки (c++) и т.д.) (Сценарий: вставить правильный код, в котором программа сделает опечатки)
3. Добавление новых циклов (Объединение с d, сценарий: подаётся программа, содержащая цикл (циклы), и будут добавлены лишние циклы, которые необязательны и можно  сделать это в одном цикле)
4. Замена типа переменных/класса (Объединение с 4d, сценарий: вставить правильный код, в котором будут изменены простые типы переменных (например, double -> int), полей класса, а также могут быть изменены модификаторы доступа на неверные (нужен public, а программа изменит на private))
5. Изменение количества повторений в циклах
6. Перемена названий между двумя переменными в коде (Сценарий: случайным образом меняет название переменных из-за чего рушится программа или нарушается логика)
7. Логика
8. Нарушение логики условий (замена знаков (<, <=, >, >=), неверное использование логических операторов (and, or, not – python)/(&&, ||, ! – c++)) (Сценарий: в условии задачи должно быть ясно, какая операция нужна, а программа её поменяет и нарушится логика, также может привести к ошибки компиляции или времени выполнения)
9. Ошибки в функциях (в неправильном порядке переданы аргументы в функцию, возврат не того аргумента) (Сценарий: для c++, в функции убирается возвращаемое значение, меняется тип возвращаемого значения на неверный или при void функция начинает возвращать значение)
10. Создание ошибок с указателями (c++), питон как доп. задача (неверное выделение/освобождение динамической памяти) (Сценарий: подается программа с динамически выделяемой памятью, в ней происходит замена динамической памяти на статическую (с этим могут быть проблемы), заменено delete [] на просто delete и т.п).
11. Область видимости (переменная внутри блока не доступна за его пределами, проблемы с глобальными переменными, затенение переменных) (Сценарий: из программы, в которой нет нарушений области видимости меняется область видимости некоторых переменных)
12. Ошибка в обработке массивов или списков (неправильное обращение к элементам массивов или списка, выход за их границы, неправильное копирование элементов)
13. Замена одного цикла на другой (while на do while, for на while и т.д.) (Сценарий: меняется цикл do-while на while или подобным образом)
14. Алгоритмы
15. Выполнение арифметических операций с элементами массива (поиск суммы/разности элементов с помощью цикла/рекурсии) (Сценарий: на вход - программа, содержащая массив, на выходе – задание по данному массиву типа того, что описано выше)
16. Требование выполнить конкретный алгоритм (поиск элементов (обход дерева в глубину/обход в ширину/рекурсивный обход), нахождение кратчайшего пути в графе (алгоритм Дейкстры, алгоритм Флойда-Уоршелла) и т.д.) (Сценарий: на вход - программа, содержащая массив, на выходе – задание, описанное выше)
17. Требование: на входе сортировка. Замена одной сортировки на другую (пузырьком/быстрой сортировкой/слиянием/двоичная куча и др.)
18. Оптимизация
19. Память. Добавление новых тяжеловесных типов данных (замена всех интов на дабл и вставка в формулы этих дополнительных данных)
20. Затирание кода
21. Затирание части строк
22. Затирание несколько строчек кода
23. Генерация теор. вопросов. Требование: достаточно подробное описание кода (Сценарий: на входе – код и описание кода, на выходе – теоретический вопрос)
24. Принцип разработки
25. Применение принципов kiss, dry (реализация более простого решения, оптимизация дублирующегося кода) (Сценарий: подается программа, код в которой не оптимизирован, а на выходе – программа, которая практически лишена этих недостатков)
26. Обработка исключений
27. Требование: код с ошибкой (деление на ноль, неправильный тип аргумента для функции или операции, обращение по несуществующему индексу), возможно, в описании – подсказка по её устранению.
28. Написание тестов (Обработка всевозможных исключительных ситуаций)