**Аналитический обзор кейс-задачи № 1**

**1. Функциональность**

**Описание:** Программа должна корректно находить сумму отрицательных элементов, расположенных между максимальным и минимальным элементами массива.

**Анализ:**

* Программа успешно обрабатывает массивы различной длины и значений.
* Проверка на наличие отрицательных элементов между максимальным и минимальным элементами.
* Обработка случаев, когда максимальный и минимальный элементы находятся рядом или в одном и том же месте.

**2. Производительность**

**Описание:** Время выполнения программы должно быть приемлемым даже для больших массивов.

**Анализ:**

* Алгоритм имеет линейную сложность O(N), где N — размер массива. Это означает, что время выполнения увеличивается пропорционально размеру входных данных.
* Для массивов размером до 10^6 элементы программа работает быстро и эффективно.

**3. Удобство использования**

**Описание:** Программа должна иметь простой интерфейс и понятные сообщения об ошибках.

**Анализ:**

* Ввод данных осуществляется через консоль, что удобно для тестирования.
* Вывод результатов ясен и понятен.
* Можно добавить обработку ошибок, например, если массив пустой или не содержит отрицательных элементов.

**4. Безопасность**

**Описание:** Программа должна защищать данные от некорректного ввода.

**Анализ:**

* В текущей реализации нет защиты от некорректного ввода (например, ввод нечисловых значений).
* Рекомендуется добавить валидацию входных данных, чтобы избежать сбоев.

**5. Масштабируемость**

**Описание:** Программа должна быть легко модифицируема для обработки других типов данных или дополнительных функций.

**Анализ:**

* Код написан с использованием функций, что облегчает его модификацию.
* Можно добавить дополнительные функции, такие как нахождение суммы положительных элементов или элементов, находящихся между двумя заданными значениями.

**6. Сопровождаемость**

**Описание:** Код должен быть хорошо структурирован и документирован.

**Анализ:**

* Код написан с использованием функций, что улучшает его читаемость.
* Рекомендуется добавить комментарии к ключевым частям кода для лучшего понимания.

**7. Переносимость**

**Описание:** Программа должна работать на различных платформах.

**Анализ:**

* Код написан на Python, который является кроссплатформенным языком.
* Программа может быть запущена на любой системе с установленным интерпретатором Python.

**8. Качество кода**

**Описание:** Код должен быть чистым, структурированным и следовать стандартам.

**Анализ:**

* Код следует стандартам написания Python.
* Рекомендуется использовать PEP 8 для улучшения читаемости.

**9. Тестирование**

**Описание:** Программа должна быть протестирована на различных входных данных.

**Анализ:**

* Рекомендуется создать набор тестов для проверки различных сценариев, включая:
  + Пустой массив.
  + Массив без отрицательных элементов.
  + Массив с максимальным и минимальным элементами, находящимися рядом.