

# Аналитический обзор проделанной работы

В данном обзоре будет рассмотрен кейс 1.

Условия: Дан одномерный массив  $A$  размерности  $N$ . Найти сумму отрицательных элементов, расположенных между максимальным и минимальным.

Функциональность: программа выполняет свои задачи в строгом соответствии с техническим заданием. Такие программы в основном необходимы программистам для решения алгоритмической задачи, поэтому в ней необходимо строго утвердить перечень входных данных, ограничившись целыми числами, к примеру. С учётом этих ограничений программа защищена от ошибок ввода и имеет универсальный вывод.

Производительность: Программа имеет высокую скорость по двум параметрам:

1. Выбран язык C++ и его стандартная библиотека, что обеспечивает достаточно низкий уровень абстракции, а т.к. программа спроектирована качественно - и лучшую скорость.
2. Асимптотический анализ программы обнаруживает максимальную сложность алгоритма в  $O(n)$ , что является отличным результатом.

При этом, программа не очень эффективна по памяти за счёт использования динамических массивов, занимающих довольно большое пространство ОП.

Удобство использования (юзабилити): Учитывая, что данная программа вероятнее всего будет использоваться в качестве функции, оценить её удобство можно достаточно высоко - она имеет два входных параметра и один выходной. С точки зрения прототипа функции - так же удобна, поскольку проста в отладке.

Безопасность: Программа не защищена ни от чего кроме алгоритмических крайних точек. При строгом условии ввода она будет достаточно защищена от поломки некорректными входными данными.

Масштабируемость: Программу можно с лёгкостью обернуть в функцию и использовать в любом проекте, где она может понадобиться. Масштабируемость удовлетворительна.

Сопровождаемость: Программа не прокомментирована ввиду отсутствия необходимости пояснения каких либо моментов для человека, квалификация которого позволяет писать самые простые программы. Сопровождаемость удовлетворительна.

Переносимость: За счёт использования только стандартной библиотеки C++ можно говорить о лёгкости портирования программы на любую платформу или даже трансляции

на другой ЯП.

Качество кода: Алгоритм, применённый в программе реализован максимально примитивно и является чистым и читаемым.

Тестирование: Программа была протестирована на алгоритмических крайних точках и защищена от случайных ошибок.

Выявленные проблемы:

- Сложность по памяти - решается за счёт использования статических массивов (по возможности)
- Возможность ошибки из-за некорректных входных данных - необходимо при встраивании программы учесть жёсткие требования к входным данным, или защитить её проверяя входные данные, если программа будет использоваться отдельно.